CHAPTER 3

Archives and information retrieval

Chapter contents

Learning goals

1. To understand the general types of data about the molecules and processes of life that are assembled in support of research and applications in biology, medicine, agriculture and technology.
2. To know the basic infrastructure of bioinformatics, in terms of the sites and responsibilities of the major archival projects.
3. To understand the basic concepts of information retrieval, including how to frame queries.
4. To have facility with general search engines on the Web, and specific sites for bioinformatics.
5. To know how to search for specific information about sequences, structures, metabolic pathways, relationships to disease, and how to launch analyses of the data recovered.

Introduction

This chapter introduces the information retrieval skills that will allow you to make effective use of the databanks. The goal is to give you familiarity with basic operations. It will then be easy to improve and develop your technique. Indeed, embedded in many databanks are tutorials which make it easy to explore their facilities.

Database indexing and specification of search terms

An index is a set of pointers to information in a database. In searching the entire World Wide Web, or a specialized database in molecular biology, you propose one or more search terms, and a program checks for them in its tables of indices. The model is that the entire database is composed of **entries**—discrete coherent parcels of information. The information retrieval software identifies entries with contents relevant to your interest. An example of the simplest paradigm is that you submit the term 'horse' and the program returns a list of entries that contain the term horse.

A full search of the Web would turn up information about many different aspects of horses—molecular biology, breeding, racing, poems about horses—most of which you don't want to see. For a successful search, it is not enough to mention what you *do* want—you must ensure that the desired responses don't get buried in extraneous rubbish. (Of course rubbish is merely whatever *other* people are interested in.)

To focus the results, information retrieval engines accept multiple query terms or keywords. A search for 'horse liver alcohol dehydrogenase' would produce responses specialized to this enzyme. The search would identify entries that contain all four keywords that you submitted: horse AND liver AND alcohol AND dehydrogenase. It would not return poems about horses among its top hits (except in the unlikely event that a poem contained all four keywords).

It is possible to ask for other logical combinations of indexing terms. For instance, if a search engine didn't know about transatlantic spelling differences, it would be useful to be able to search for 'hemoglobin OR haemoglobin'. (Note that a search for 'hemoglobin haemoglobin' would probably be interpreted as 'hemoglobin AND haemoglobin' which would pick up only documents written by international committees or orthographically-challenged expatriates.)

If you wanted to know about other dehydrogenases, you could ask for 'dehydrogenase NOT alcohol'. This would retrieve entries that contain the term dehydrogenase but did NOT contain the word alcohol. You would find entries about lactate dehydrogenase, malate dehydrogenase, etc. You would miss references to review articles that compared alcohol dehydrogenases to other dehydrogenases, or alignments of the sequences of many dehydrogenases including alcohol dehydrogenase. You might regret missing these.

Many database search engines will allow complex logical expressions such as '(haemoglobin OR hemoglobin) AND (dehydrogenase NOT alcohol)'. Construction of such expressions is an exercise in set theory, and it is helped by drawing Venn diagrams. Although the logic of a search is independent of the software used to query a database, different programs demand different syntax to express the same conditions. For example the query for dehydrogenase NOT alcohol might have to be entered as DEHYDROGENASE −ALCOHOL or DEHYDROGENASE !ALCOHOL.

Specialized databases, including those in molecular biology, impose a structure on the information, to separate different categories of information. This is essential. There are currently active biomedical scientists named E(lisabetta) Coli, (John D.) Yeast, (Patrice) Rat, and a large number of Rabbits, as well as several Crystals and Blots. If you wanted to find papers published by these investigators, it would be naive to perform a general search of a molecular biology database with any of their surnames. Many databases provide separate indexing and searching of different categories of information. They permit searching for papers of which E. Coli is an AUTHOR.

Some of the categories, such as taxonomy, have **controlled vocabularies**. Often these are presented to the user as pull-down menus. To do a search for 'globin NOT mammal', and pick out the relatively few entries about nonmammalian globins rather than the very many entries about globins, including human haemoglobins, that do not explicitly mention the term mammal, requires an information retrieval system that 'understands' the taxonomic hierarchy. Controlled

vocabularies—limited, explicit, and carefully defined sets of terms—are also important in distributing queries among several databases.

A technical problem that frequently creates difficulty is how to enter terms containing nonstandard characters such as accent marks or umlauts, Greek letters, and, as already mentioned, differences between US and British spelling. A specialized database such as NCBI's ENTREZ can handle the US–British spelling differences with a synonym dictionary. Programs that index the entire Web usually do not. Ignore the accent marks and hope for the best.

Follow-up questions

When searching in databases, it is rare that you will find exactly what you want on the first round of probing. Usually you have to modify the query, on the basis of the results initially returned. Most information retrieval software permits consecutive, cumulative searches, with altered sets of search terms and/or logical relationships. Conversely, once you find what you looked for, you will often want to extend your search to find related material. If you find a gene sequence, you might want to know about homologous genes in other organisms. Or whether a three-dimensional structure of the corresponding protein is available. Or you might want to know about papers published about the gene.

For these subsidiary queries you need links between entries in the same or different databases. This is a special example of the question of how one 'browses' in electronic libraries—a difficult problem, the subject of current research.

To find homologous genes you would like links to other items in the *same* database (a database of gene sequences). To find structures, or bibliographical references, related to a gene, you would like links *between* different databases (from the database of gene sequences to a database of three-dimensional structures, or to a bibliographical database). As the number of databases grows, intercommunication among them has become a high-priority goal. Indeed, the interactivity of the databases in molecular biology is growing more and more effective, so that these operations are fairly easy now—formerly one had to do separate searches on isolated databases. This is a generalization of the original model of a database as a closed set of independent entries that can be selected only by their indexed contents.

To some extent database activities in bioinformatics can be classified into *archiving*—with the major goals of conservation and curation—and *interpreting*— the compilation of biological information in a form most useful to support research. Different archives specialize in different kinds of data—nucleic acid sequences, protein sequences, structures—for reasons in part historical and in part because of the different specialized curatorial skills required. Interpretative databases are free to combine information from any available sources. In most cases, archival and interpretative projects are carried out at the same institution and even by the same people.

Two aspects of the development of bioinformatics databases are apparent. One is the very great growth of individual database projects that recombine the archived data in different ways. The other is the combination of many individual databases into 'umbrella' sites. There is really no paradox—both are going on. (Genes also both multiply and combine.)

Most database unifications are merely extensions, with greater appearance of intimacy, of the collaborations or competitive efforts that in most cases formerly existed. We shall see, for instance, that protein sequence databases are coordinating their activities as UniProt; and that the protein structure databases are coordinating as the Worldwide Protein Data Bank. Interpro is an umbrella database that integrates the contents, features, and annotation of individual databases of protein families, domains, and functional sites, and contains links to others, including the Gene Ontology Consortium™ functional classification. It currently subsumes the PROSITE, Pfam, PRINTS, SMART and ProDom databases, and intends to assimilate others. (Resistance is futile.)

Analysis of retrieved data

Sometimes as a result of a search you will want to launch a program using the results retrieved as input. For instance, if you identify a protein sequence of interest, you might want to perform a PSI-BLAST search. This is not strictly a database-lookup problem, and formerly you would have to run a separate job, and feed the retrieved sequence to the application program by hand. However, like searches in multiple databases, information retrieval systems in molecular biology often provide facilities for initiating such processes. This makes for very much improved fluency in your sessions at the computer.

The archives

Although our knowledge of biological sequence and structure data is very far from complete, it is of quite respectable size, and growing extremely rapidly. Many scientists are working to generate the data, or to carry out research projects analysing the results. Archiving and distribution are carried out by particular databanking organizations.

Archiving of bioinformatics data was originally carried out by individual research groups motivated by an interest in the associated science. As the requirements for equipment and personnel grew—and the nature of the skills required changed, to include much more emphasis on computing—they have been made the responsibility of special national and even international projects, on a very large scale indeed. Anyone who has followed the entire history of these projects cannot help being impressed by their growth from small, low-profile and ill-funded projects carried out by a few dedicated individuals, to a multinational

heavy industry subject to political takeovers and the scientific equivalent of leveraged buyouts.

Primary data collections related to biological macromolecules include:

- Nucleic acid sequences, including whole-genome projects
- Amino acid sequences of proteins
- Protein and nucleic acid structures
- Small-molecule crystal structures
- Protein functions
- Expression patterns of genes
- Metabolic pathways, and networks of interaction and control
- Publications

Nucleic acid sequence databases

The worldwide nucleic acid sequence archive is a triple partnership of The National Center for Biotechnology Information (USA), the EMBL Data Library (European Bioinformatics Institute, UK), and the DNA Data Bank of Japan (National Institute of Genetics, Japan). The groups exchange data daily. As a result the raw data are identical, although the format in which they are stored, and the nature of the annotation, vary among them. These databases curate, archive, and distribute DNA and RNA sequences collected from genome projects, scientific publications, and patent applications. To make sure that these fundamental data are freely available, scientific journals require deposition of new nucleotide sequences in a database as a condition for publication of an article. Similar conditions apply to amino acid sequences, and to nucleic acid and protein structures.

The nucleic acid sequence databases, as distributed, are collections of entries. Each entry has the form of a text file containing data and annotations for a single contiguous sequence. Many entries are assembled from several published papers reporting overlapping fragments of a complete sequence. Others are complete genomes.

Entries have a life cycle in the database. Because of the desire on the part of the user community for rapid access to data, new entries are made available before annotation is complete and checks are performed. Entries mature through the classes:

$$\text{Unannotated} \rightarrow \text{Preliminary} \rightarrow \text{Unreviewed} \rightarrow \text{Standard}$$

Rarely, an entry 'dies'—a few have been removed when they were determined to be erroneous.

A sample DNA sequence entry from the EMBL data library, including annotations as well as sequence data, is the gene for bovine pancreatic trypsin inhibitor (the Box shows part of this entry, omitting most of the sequence itself).

The EMBL Data Library entry for the bovine pancreatic trypsin inhibitor gene

```
ID   BTBPTIG    standard; DNA; MAM; 3998 BP.
XX
AC   X03365; K00966;
XX
DT   18-NOV-1986 (Rel. 10, Created)
DT   20-MAY-1992 (Rel. 31, Last updated, Version 3)
XX
DE   Bovine pancreatic trypsin inhibitor (BPTI) gene
XX
KW   Alu-like repetitive sequence; protease inhibitor;
KW   trypsin inhibitor.
XX
OS   Bos taurus (cattle)
OC   Eukaryota; Animalia; Metazoa; Chordata; Vertebrata; Mammalia;
OC   Theria; Eutheria; Artiodactyla; Ruminantia; Pecora; Bovidae.
XX
RN   [1]
RP   1-3998
RA   Kingston I.B., Anderson S.;
RT   "Sequences encoding two trypsin inhibitors occur in strikingly
RT   similar genomic environments";
RL   Biochem. J. 233:443-450(1986).
XX
RN   [2]
RA   Anderson S., Kingston I.B.;
RT   "Isolation of a genomic clone for bovine pancreatic trypsin
RT   inhibitor by using a unique-sequence synthetic dna probe";
RL   Proc. Natl. Acad. Sci. U.S.A. 80:6838-6842(1983).
XX
DR   SWISS-PROT; P00974; BPT1_BOVIN.
XX
CC   Data kindly reviewed (08-DEC-1987) by Kingston I.B.
XX
FH   Key             Location/Qualifiers
FH
FT   misc_feature    795..800
FT                   /note="pot. polyA signal"
FT   misc_feature    835..839
FT                   /note="pot. polyA signal"
FT   repeat_region   837..847
FT                   /note="direct repeat"
FT   misc_feature    930..945
FT                   /note="sequence homologous to Alu-like
FT                   consensus seq."
FT   repeat_region   1035..1045
FT                   /note="direct repeat"
FT   misc_feature    2456..2461
FT                   /note="pot. splice signal"
FT   CDS             2470..2736
FT                   /note="put. precursor"
FT   misc_feature    2488..2489
FT                   /note="pot. intron/exon splice junction"
FT   misc_feature    2506..2507
FT                   /note="pot. intron/exon splice junction"
FT   CDS             2512..2685
FT                   /note="trypsin inhibitor (aa 1-58)"
FT   misc_feature    2698..2699
FT                   /note="pot. exon/intron splice junction"
FT   misc_feature    3690..3695
FT                   /note="pot. polyA signal"
FT   misc_feature    3729..3733
FT                   /note="pot. polyA signal"
XX
SQ   Sequence 3998 BP; 1053 A; 902 C; 892 G; 1151 T; 0 other;
     aattctgata atgcagagaa ctggtaagga gttctgattg ttctgcttga ttaaatgggt
     tgtaacagga tagtgtcttg tcctgatcct agcattcata tggtgtgtgt tctggggcaa
     gtcatctgca gtttcttcac ctgaacaggg ggaccaggtt acatgagttt cttaaaagat
     taccagtcat gagtatgaag agtttacact ttcctgatca atgacgtcca tttcccatca
                                   3720 nucleotides deleted ...
     gccaggtcaa actttggggt gtgttatttc cctgaatt
//
```

A **feature table** (lines beginning FT) is a component of the annotation of an entry that reports properties of specific regions, for instance coding sequences (CDS). Because these are designed to be readable by computer programs—for example, to translate a coding region to an amino acid sequence—they have a more carefully

controlled format and a more restricted vocabulary. Development of controlled vocabularies and a shared dictionary and thesaurus for keywords and feature tables is also important in establishing links between different databases.

The feature table may indicate regions that:

- perform or affect function
- interact with other molecules
- affect replication
- are involved in recombination
- are a repeated unit
- have secondary or tertiary structure
- are revised or corrected

Genome databases

Although genome sequences form entries in the standard nucleic acid sequence archives, many species have special databases that bring together the genome sequence and its annotation with other data related to the species.

Protein sequence databases

In 2002, three protein sequence databases—The Protein Information Resource, at the National Biomedical Research Foundation of the Georgetown University Medical Center in Washington, DC, USA; and SWISS-PROT and TrEMBL, from the Swiss Institute of Bioinformatics in Geneva and the European Bioinformatics Institute in Hinxton, UK—coordinated their efforts, to form the *UniProt* consortium. The partners in this enterprise share the database but continue to offer separate information retrieval tools for access.

The PIR grew out of the very first sequence database, developed by Margaret O. Dayhoff—the pioneer of the field of bioinformatics. SWISS-PROT was developed at the Swiss Institute of Bioinformatics. TrEMBL contains the translations of genes identified within DNA sequences in the EMBL Data Library. TrEMBL entries are regarded as preliminary, and are converted—after curation and extended annotation—to full SWISS-PROT entries.

Today, almost all amino acid sequence information arises from translation of nucleic acid sequences. Information about ligands, disulphide bridges, subunit associations, post-translational modifications, glycosylation, effects of mRNA editing, etc., are not available from gene sequences. For instance, from genetic information alone one would not know that human insulin is a dimer linked by disulphide bridges. Protein sequence databanks collect this additional information from the literature and provide suitable annotations.

From UniProt, the entry for the amino acid sequence of the protein bovine pancreatic trypsin inhibitor, in SWISS-PROT format, is shown in the Box, pages 126–127. (The comparison of SWISS-PROT format with ENTREZ and PIR formats is the subject of Weblem 3.12.)

Databases associated with SWISS-PROT

Two related databases closely associated with SWISS-PROT are the ENZYME DB, and PROSITE, a set of motifs.

The ENZYME DB stores the following information about enzymes:

- EC Number: a numerical identifier assigned by the Enzyme Commission (authorized by the International Union of Biochemistry and Molecular Biology; see http://www.chem.qmw.ac.uk/iubmb/enzyme/)
- Recommended name
- Alternative names, if any
- Catalytic activity
- Cofactors, if any
- Pointers to SWISS-PROT and other data banks
- Pointers to disease associated with enzyme deficiency if any known

A sample entry in ENZYME DB

```
ID 1.14.17.3
DE PEPTIDYLGLYCINE MONOOXYGENASE.
AN PEPTIDYL ALPHA-AMIDATING ENZYME.
AN PEPTIDYLGLYCINE 2-HYDROXYLASE.
CA PEPTIDYLGLYCINE + ASCORBATE + O(2) = PEPTIDYL(2-HYDROXYGLYCINE) +
CA DEHYDROASCORBATE + H(2)O.
CF COPPER.
CC -!- PEPTIDYLGLYCINES WITH A NEUTRAL AMINO ACID RESIDUE IN THE
CC PENULTIMATE POSITION ARE THE BEST SUBSTRATES FOR THE ENZYME.
CC -!- THE ENZYME ALSO CATALYZES THE DISMUTATATION OF THE PRODUCT TO
CC GLYOXYLATE AND THE CORRESPONDING DESGLYCINE PEPTIDE AMIDE.
DR P10731, AMD_BOVIN ; P19021, AMD_HUMAN ; P14925, AMD_RAT ;
DR P08478, AMD1_XENLA; P12890, AMD2_XENLA;
```

The first two characters of each line identify the information that the line contains. For instance, ID = Identification, DE = Description = Official name, AN = Alternate name(s), CA = catalytic activity, CF = cofactor(s), CC = comments, DR = database reference (to SWISS-PROT).

PROSITE contains common patterns of residues of sets of proteins. Such a pattern (or motif, or signature, or fingerprint, or template) appears in a family of related proteins usually because of the requirements of binding sites that constrain the evolution of a protein family. Often they indicate distant relationships not otherwise detectable by comparing sequences. The consensus pattern for inorganic pyrophosphatase is: `D-[SGN]-D-[PE]-[LIVM]-D-[LIVMGC]`. The three conserved aspartates (D) bind divalent metal cations.

The PIR and associated databases

The PIR maintains several databases about proteins:

- PIR-PSD: the main protein sequence database
- iProClass: classification of proteins according to structure and function

Amino acid sequence entry for bovine pancreatic trypsin inhibitor

NiceProt View of Swiss-Prot: P00974

Entry information

Entry name	**BPT1_BOVIN**
Primary accession number	**P00974**
Secondary accession numbers	None
Entered in Swiss-Prot in	Release 01, July 1986
Sequence was last modified in	Release 10, March 1989
Annotations were last modified in	Release 44, June 2004

Name and origin of the protein

Protein name	**Pancreatic trypsin inhibitor [Precursor]**
Synonyms	**Basic protease inhibitor** **BPI** **BPTI** **Aprotinin**
Gene name	None
From	Bos taurus (Bovine) [TaxID: 9913]
Taxonomy	Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Cetartiodactyla; Ruminantia; Pecora; Bovidae; Bovinae; Bos.

References

[1] SEQUENCE FROM NUCLEIC ACID
MEDLINE=87283904; PubMed=2441071;
Creighton T.E., Charles I.G.;
"Sequences of the genes and polypeptide precursors for two bovine protease inhibitors";
J. Mol. Biol. 194:11-22(1987).
ADDITIONAL REFERENCES DELETED

Comments

- *FUNCTION*: Inhibits trypsin, kallikrein, chymotrypsin, and plasmin.
- *SUBCELLULAR LOCATION*: Secreted.
- *PHARMACEUTICAL*: Available under the name Trasylol (Mile). Used for inhibiting coagulation so as to reduce blood loss during bypass surgery.
- *SIMILARITY*: Contains 1 BPTI/Kunitz inhibitor domain.
- *DATABASE*: Name=Trasylol; Note=Clinical information on Trasylol; www="http://www.trasylol.com/".

ADDITIONAL COMMENTS DELETED

Copyright

Cross-references

EMBL	M20934; AAD13685.1;-.
	ADDITIONAL CROSS-REFERENCES TO EMBL DELETED
PIR	S00277; TIBO.
PDB	1K09; 10-JUL-02.
	ADDITIONAL CROSS-REFERENCES TO PDB DELETED
InterPro	IPR002223; Kunitz_BPTI.
Pfam	PF00014; Kunitz_BPTI; 1.
	Pfam graphical view of domain structure.
PRINTS	PR00759; BASICPTASE.
ProDom	PD000222; Kunitz_BPTI; 1.
	[Domain structure/List of seq. sharing at least 1 domain]
SMART	SM00131; KU; 1.
PROSITE	PS00280; BPTI_KUNITZ_1;1.
	PS50279; BPTI_KUNITZ_1; 2.
	PROSITE graphical view of domain structure.
Implicit links to	HOVERGEN; BLOCKS; ProtoNet; ProtoMap; PRESAGE; DIP; ModBase; SMR; SWISS-2DPAGE; UniRef.

Keywords

Serine protease inhibitor; Signal; Pharmaceutical; 3D-structure.

Features

Key	From	To	Length	Description
SIGNAL	1	21	21	Potential.
PROPEP	22	35	14	
CHAIN	36	93	58	Pancreatic trypsin inhibitor.
PROPEP	94	100	7	
DOMAIN	40	90	51	BPTI/Kunitz inhibitor.
SITE	50	51	2	Reactive bond for trypsin.
DISULFID	40	90		
DISULFID	49	73		
DISULFID	65	86		
HELIX	38	41	4	
STRAND	53	59	7	
TURN	60	63	4	
STRAND	64	70	7	
STRAND	80	80	1	
HELIX	83	90	8	

Sequence information

Length: **100 AA** [This is the length of the unprocessed precursor]
Molecular weight: **10903 Da** [This is the MW of the unprocessed precursor]
CRC64: **6A778A4AD763FB19** [This is a checksum on the sequence]

```
         10          20          30          40          50          60
          |           |           |           |           |           |
MKMSRLCLSV  ALLVLLGTLA  ASTPGCDTSN  QAKAQRPDFC  LEPPYTGPCK  ARIIRYFYNA

         70          80          90         100
          |           |           |           |
KAGLCQTFVY  GGCRAKRNNF  KSAEDCMRTC  GGAIGPWENL
```

- ASDB: annotation and similarity database; each entry is linked to a list of similar sequences
- NRL_3D: a database of sequences and annotations of proteins of known structure deposited in the Protein Data Bank
- ALN: a database of protein sequence alignments
- RESID: a database of covalent protein structure modifications (recall that important structural features of proteins such as disulphide bridges are not inferrable from gene sequences, and will not appear in protein sequence databases derived solely by translation of genomic data)

The PIR has also created IESA: The Integrated Environment for Sequence Analysis, a site for information retrieval and launching of calculations.

The web server of PIR shows some of the richness of information retrieval tools available. It includes:

- FETCH DATABASE ENTRY
- PAIRWISE SEQUENCE ALIGNMENT
- PROT-FAM: classification by protein family of over 7000 multiple sequence alignments of protein families
- ATLAS: search text fields of databases, or scan sequences for short peptides
- ALERT: receive information about new database entries of interest by e-mail, automatically
- GATEWAY: pattern recognition, homology identification

Databases of structures

Structure databases archive, annotate and distribute sets of atomic coordinates. The major database for biological macromolecular structures is the Protein Data Bank (PDB). It contains structures of proteins, nucleic acids, and a few carbohydrates. Started by the late Walter Hamilton at Brookhaven National Laboratories, Long Island, New York, USA in 1971, the PDB is now managed by the Research Collaboratory for Structural Bioinformatics (RCSB), a distributed organization based at Rutgers University, in New Jersey; the San Diego Supercomputer Center, in California; and the National Institute of Standards and Technology, in Maryland, all in the USA. The parent web site of the Protein Data Bank is at http://www.rcsb.org.

The home page of the PDB contains links to the data files themselves, to expository and tutorial material including short news items and the PDB Newsletter, to facilities for deposition of new entries, and to specialized search software for retrieving structures.

Recently, the RCSB, the Molecular Structure Database and the European Bioinformatics Institute, and the Protein Data Bank Japan have formed the Worldwide Protein Data Bank (wwPDB), with the goal of producing a unified archive.

The box shows part of a Protein Data Bank* entry for a structure of *E. coli* thioredoxin.† The information contained includes:

- What protein is the subject of the entry, and what species it came from
- Who solved the structure, and references to publications describing the structure determination
- Experimental details about the structure determination, including information related to the general quality of the result such as resolution of an X-ray structure determination and stereochemical statistics
- The amino acid sequence
- What additional molecules appear in the structure, including cofactors, inhibitors, and water molecules
- Assignments of secondary structure: helix, sheet
- Disulphide bridges
- The atomic coordinates

Protein Data Bank entry 2TRX, *E. Coli* thioredoxin

```
HEADER    ELECTRON TRANSPORT                      19-MAR-90   2TRX
COMPND    THIOREDOXIN
SOURCE    (ESCHERICHIA $COLI)
AUTHOR    S.K.KATTI,D.M.LE*MASTER,H.EKLUND
REVDAT   2   15-JAN-93 2TRXA   1       HEADER COMPND
REVDAT   1   15-OCT-91 2TRX    0
JRNL        AUTH   S.K.KATTI,D.M.LE*MASTER,H.EKLUND
JRNL        TITL   CRYSTAL STRUCTURE OF THIOREDOXIN FROM ESCHERICHIA
JRNL        TITL 2 $COLI AT 1.68 ANGSTROMS RESOLUTION
JRNL        REF    J.MOL.BIOL.                   V. 212   167 1990
JRNL        REFN   ASTM JMOBAK  UK ISSN 0022-2836                070
REMARK   1
REMARK   1 REFERENCE 1
REMARK   1  AUTH   A.HOLMGREN,B.-*O.SODERBERG,H.EKLUND,C.-*I.BRANDEN
REMARK   1  TITL   THREE-DIMENSIONAL STRUCTURE OF ESCHERICHIA COLI
REMARK   1  TITL 2 THIOREDOXIN-*S=2= TO 2.8 ANGSTROMS RESOLUTION
REMARK   1  REF    PROC.NAT.ACAD.SCI.USA         V.  72  2305 1975
REMARK   1  REFN   ASTM PNASA6  US ISSN 0027-8424                040
REMARK   1 REFERENCE 2
REMARK   1  AUTH   B.-*O.SODERBERG,A.HOLMGREN,C.-*I.BRANDEN
REMARK   1  TITL   STRUCTURE OF OXIDIZED THIOREDOXIN TO 4.5 ANGSTROMS
REMARK   1  TITL 2 RESOLUTION
REMARK   1  REF    J.MOL.BIOL.                   V.  90   143 1974
REMARK   1  REFN   ASTM JMOBAK  UK ISSN 0022-2836                070
REMARK   1 REFERENCE 3
REMARK   1  AUTH   A.HOLMGREN,B.-*O.SODERBERG
REMARK   1  TITL   CRYSTALLIZATION AND PRELIMINARY CRYSTALLOGRAPHIC
REMARK   1  TITL 2 DATA FOR THIOREDOXIN FROM ESCHERICHIA $COLI B
REMARK   1  REF    J.MOL.BIOL.                   V.  54   387 1970
REMARK   1  REFN   ASTM JMOBAK  UK ISSN 0022-2836                070
REMARK   2
REMARK   2 RESOLUTION. 1.68 ANGSTROMS.
REMARK   3
REMARK   3 REFINEMENT. BY THE RESTRAINED LEAST-SQUARES PROCEDURE OF J.
REMARK   3  KONNERT AND W. HENDRICKSON AS MODIFIED BY B. FINZEL
REMARK   3  (PROGRAM *PROFFT*).  THE R VALUE IS 0.165 FOR 25969
REMARK   3  REFLECTIONS IN THE RESOLUTION RANGE 8.0 TO 1.68 ANGSTROMS
REMARK   3  WITH FOBS .GT. 3.0*SIGMA(FOBS)
REMARK   3
```

→

* Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N.,Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000), The Protein Data Bank Nucleic Acids Research, **28**, 235–242.

† Katti, S. K., LeMaster, D. M. & Eklund, H. (1990), Crystal structure of thioredoxin from *Escherichia coli* at 1.68 Å resolution, *J. Mol. Biol.*, **212**, 167–184.

Protein Data Bank entry 2TRX, ***E. Coli*** thioredoxin *(continued)*

```
REMARK   3  RMS DEVIATIONS FROM IDEAL VALUES (THE VALUES OF
REMARK   3      SIGMA, IN PARENTHESES, ARE THE INPUT ESTIMATED
REMARK   3      STANDARD DEVIATIONS THAT DETERMINE THE RELATIVE
REMARK   3      WEIGHTS OF THE CORRESPONDING RESTRAINTS)
REMARK   3    DISTANCE RESTRAINTS (ANGSTROMS)
REMARK   3      BOND DISTANCE                                0.015(0.020)
REMARK   3      ANGLE DISTANCE                               0.035(0.030)
REMARK   3      PLANAR 1-4 DISTANCE                          0.055(0.050)
REMARK   3    PLANE RESTRAINT (ANGSTROMS)                    0.021(0.020)
REMARK   3    CHIRAL-CENTER RESTRAINT (ANGSTROMS**3)         0.131(0.150)
REMARK   3    NON-BONDED CONTACT RESTRAINTS (ANGSTROMS)
REMARK   3      SINGLE TORSION CONTACT                       0.165(0.500)
REMARK   3      MULTIPLE TORSION CONTACT                     0.174(0.500)
REMARK   3      POSSIBLE HYDROGEN BOND                       0.180(0.500)
REMARK   3    CONFORMATIONAL TORSION ANGLE RESTRAINT (DEGREES)
REMARK   3      PLANAR (OMEGA)                                4.0(3.0)
REMARK   3      STAGGERED                                    16.3(15.0)
REMARK   3      ORTHONORMAL                                  11.7(20.0)
REMARK   3    ISOTROPIC THERMAL FACTOR RESTRAINTS (ANGSTROMS**2)
REMARK   3      MAIN-CHAIN BOND                              1.38(1.000)
REMARK   3      MAIN-CHAIN ANGLE                             2.28(1.000)
REMARK   3      SIDE-CHAIN BOND                              1.97(1.000)
REMARK   3      SIDE-CHAIN ANGLE                             3.27(1.500)
REMARK   4
REMARK   4 THERE ARE TWO MOLECULES IN THE ASYMMETRIC UNIT.  THEY HAVE
REMARK   4 BEEN ASSIGNED CHAIN INDICATORS *A* AND *B*.  THEY HAVE BEEN
REMARK   4 REFINED INDEPENDENTLY WITHOUT IMPOSING NON-CRYSTALLOGRAPHIC
REMARK   4 SYMMETRY RESTRAINTS.
REMARK   5
REMARK   5 IN ADDITION TO THE METAL COORDINATION SPECIFIED ON CONECT
REMARK   5 RECORDS BELOW, THERE ARE BONDS TO OD1 AND OD2 OF ASP 10 IN
REMARK   5 A SYMMETRY-RELATED MOLECULE.  DUE TO SOME LIMITATIONS OF
REMARK   5 PROTEIN DATA BANK FORMAT, THESE BONDS CANNOT BE PRESENTED
REMARK   5 ON CONECT RECORDS.
REMARK   6
REMARK   6 CORRECTION. CORRECT CLASSIFICATION ON HEADER RECORD AND
REMARK   6  REMOVE E.C. CODE.  15-JAN-93.
SEQRES   1 A  108  SER ASP LYS ILE ILE HIS LEU THR ASP ASP SER PHE ASP
SEQRES   2 A  108  THR ASP VAL LEU LYS ALA ASP GLY ALA ILE LEU VAL ASP
SEQRES   3 A  108  PHE TRP ALA GLU TRP CYS GLY PRO CYS LYS MET ILE ALA
SEQRES   4 A  108  PRO ILE LEU ASP GLU ILE ALA ASP GLU TYR GLN GLY LYS
SEQRES   5 A  108  LEU THR VAL ALA LYS LEU ASN ILE ASP GLN ASN PRO GLY
SEQRES   6 A  108  THR ALA PRO LYS TYR GLY ILE ARG GLY ILE PRO THR LEU
SEQRES   7 A  108  LEU LEU PHE LYS ASN GLY GLU VAL ALA ALA THR LYS VAL
SEQRES   8 A  108  GLY ALA LEU SER LYS GLY GLN LEU LYS GLU PHE LEU ASP
SEQRES   9 A  108  ALA ASN LEU ALA
SEQRES   1 B  108  SER ASP LYS ILE ILE HIS LEU THR ASP ASP SER PHE ASP
SEQRES   2 B  108  THR ASP VAL LEU LYS ALA ASP GLY ALA ILE LEU VAL ASP
SEQRES   3 B  108  PHE TRP ALA GLU TRP CYS GLY PRO CYS LYS MET ILE ALA
SEQRES   4 B  108  PRO ILE LEU ASP GLU ILE ALA ASP GLU TYR GLN GLY LYS
SEQRES   5 B  108  LEU THR VAL ALA LYS LEU ASN ILE ASP GLN ASN PRO GLY
SEQRES   6 B  108  THR ALA PRO LYS TYR GLY ILE ARG GLY ILE PRO THR LEU
SEQRES   7 B  108  LEU LEU PHE LYS ASN GLY GLU VAL ALA ALA THR LYS VAL
SEQRES   8 B  108  GLY ALA LEU SER LYS GLY GLN LEU LYS GLU PHE LEU ASP
SEQRES   9 B  108  ALA ASN LEU ALA
FTNOTE   1
FTNOTE   1 RESIDUES PRO A 76 AND PRO B 76 ARE CIS PROLINES.
FTNOTE   2
FTNOTE   2 RESIDUES HIS A 6, LEU A 7, ILE A 23, ASP A 47, GLU A 48,
FTNOTE   2 LEU A 58, LEU A 80, HIS B 6, ASP B 47, LEU B 58, AND
FTNOTE   2 LEU B 80 HAVE BEEN MODELED AS TWO CONFORMERS.
FTNOTE   3
FTNOTE   3 RESIDUES 11 - 21 IN CHAIN B ARE DISORDERED.
HET     CU    109       1     COPPER ++ ION
HET     CU    109       1     COPPER ++ ION
HET    MPD    601       8     2-METHYL-2,4-PENTANEDIOL
HET    MPD    602       8     2-METHYL-2,4-PENTANEDIOL
HET    MPD    603       8     2-METHYL-2,4-PENTANEDIOL
HET    MPD    604       8     2-METHYL-2,4-PENTANEDIOL
HET    MPD    605       8     2-METHYL-2,4-PENTANEDIOL
HET    MPD    606       8     2-METHYL-2,4-PENTANEDIOL
HET    MPD    607       8     2-METHYL-2,4-PENTANEDIOL
HET    MPD    608       8     2-METHYL-2,4-PENTANEDIOL
FORMUL   3   CU    2(CU1 ++)
FORMUL   4  MPD    8(C6 H14 O2)
FORMUL   5  HOH   *140(H2 O1)
HELIX    1 A1A SER A   11  LEU A   17  1 DISORDERED IN MOLECULE B
HELIX    2 A2A CYS A   32  TYR A   49  1 BENT BY 30 DEGREES AT RES 39
HELIX    3 A3A ASN A   59  ASN A   63  1
HELIX    4 31A THR A   66  TYR A   70  5 DISTORTED H-BONDING C-TERMINS
HELIX    5 A4A SER A   95  LEU A  107  1
HELIX    6 A1B SER B   11  LEU B   17  1 DISORDERED IN MOLECULE B
HELIX    7 A2B CYS B   32  TYR B   49  1 BENT BY 30 DEGREES AT RES 39
HELIX    8 A3B ASN B   59  ASN B   63  1
HELIX    9 31B THR B   66  TYR B   70  5 DISTORTED H-BONDING C-TERMINS
HELIX   10 A4B SER B   95  LEU B  107  1
```

```
SHEET    1 B1A 5 LYS A   3  THR A   8  0
SHEET    2 B1A 5 LEU A  53  ASN A  59  1  O  VAL A  55   N  ILE A   5
SHEET    3 B1A 5 GLY A  21  TRP A  28  1  N  TRP A  28   O  LEU A  58
SHEET    4 B1A 5 PRO A  76  LYS A  82 -1  O  THR A  77   N  PHE A  27
SHEET    5 B1A 5 VAL A  86  GLY A  92 -1  N  GLY A  92   O  LYS A  82
SHEET    1 B1B 5 LYS B   3  THR B   8  0
SHEET    2 B1B 5 LEU B  53  ASN B  59  1  O  VAL B  55   N  ILE B   5
SHEET    3 B1B 5 GLY B  21  TRP B  28  1  N  TRP B  28   O  LEU B  58
SHEET    4 B1B 5 PRO B  76  LYS B  82 -1  O  THR B  77   N  PHE B  27
SHEET    5 B1B 5 VAL B  86  GLY B  92 -1  N  GLY B  92   O  LYS B  82
TURN     1 T1A THR A   8  SER A  11       III (TYPE I IN MOLECULE B)
TURN     2 T2A ALA A  29  CYS A  32       I
TURN     3 T3A TYR A  49  LYS A  52       II
TURN     4 T4A GLY A  74  THR A  77       VIB (INCLUDES CIS PRO 76)
TURN     5 T5A LYS A  82  GLU A  85       I'
TURN     6 T1B THR B   8  SER B  11       I (TYPE III IN MOLECULE A)
TURN     7 T2B ALA B  29  CYS B  32       I
TURN     8 T3B TYR B  49  LYS B  52       II
TURN     9 T4B GLY B  74  THR B  77       VIB (INCLUDES CIS PRO 76)
TURN    10 T5B LYS B  82  GLU B  85       I'
SSBOND   1 CYS A   32    CYS A   35
SSBOND   2 CYS B   32    CYS B   35
CRYST1   89.500   51.060   60.450  90.00 113.50  90.00 C 2           8
ORIGX1      1.000000  0.000000  0.000000        0.00000
ORIGX2      0.000000  1.000000  0.000000        0.00000
ORIGX3      0.000000  0.000000  1.000000        0.00000
SCALE1      0.011173  0.000000  0.004858        0.00000
SCALE2      0.000000  0.019585  0.000000        0.00000
SCALE3      0.000000  0.000000  0.018039        0.00000
ATOM      1  N   SER A   1      21.389  25.406  -4.628  1.00 23.22
ATOM      2  CA  SER A   1      21.628  26.691  -3.983  1.00 24.42
ATOM      3  C   SER A   1      20.937  26.944  -2.679  1.00 24.21
ATOM      4  O   SER A   1      21.072  28.079  -2.093  1.00 24.97
ATOM      5  CB  SER A   1      21.117  27.770  -5.002  1.00 28.27
ATOM      6  OG  SER A   1      22.276  27.925  -5.861  1.00 32.61
ATOM      7  N   ASP A   2      20.173  26.028  -2.163  1.00 21.39
ATOM      8  CA  ASP A   2      19.395  26.125  -0.949  1.00 21.57
ATOM      9  C   ASP A   2      20.264  26.214   0.297  1.00 20.89
ATOM     10  O   ASP A   2      19.760  26.575   1.371  1.00 21.49
ATOM     11  CB  ASP A   2      18.439  24.914  -0.856  1.00 22.14
ATOM     12  CG  ASP A   2      19.199  23.629  -0.576  1.00 23.23
ATOM     13  OD1 ASP A   2      20.107  23.371  -1.387  1.00 22.71
ATOM     14  OD2 ASP A   2      18.905  22.959   0.420  1.00 23.61

                        . . . protein atoms deleted

ATOM    844  N   ALA A 108      41.357  21.341   9.676  1.00 42.93
ATOM    845  CA  ALA A 108      42.151  20.619  10.674  1.00 46.31
ATOM    846  C   ALA A 108      42.632  19.312  10.013  1.00 48.21
ATOM    847  O   ALA A 108      41.703  18.483   9.767  1.00 49.54
ATOM    848  CB  ALA A 108      41.441  20.369  11.988  1.00 46.65
ATOM    849  OXT ALA A 108      43.857  19.249   9.766  1.00 49.19
TER     850      ALA A 108

  . . . second chain, and methane, pentane-diol molecules deleted

          HETATM 1749  O   HOH   401      30.339  33.478  16.727  1.00 17.61
          HETATM 1750  O   HOH   402      29.396  44.583   6.834  0.95 17.71

                  . . . 72 additional water molecules deleted
```

The PDB overlaps in scope with several other databases. The Cambridge Crystallographic Data Centre archives the structures of small molecules; oligonucleotides appear in both the CCDC and PDB. This information is extremely useful in studies of conformations of the component units of biological macromolecules, and for investigations of macromolecule-ligand interactions, including but not limited to applications to drug design. The Nucleic Acid Database (NDB) at Rutgers University, New Brunswick, New Jersey, USA complements the PDB. The BioMagResBank, at the Department of Biochemistry, University of Wisconsin, Madison, Wisconsin, USA, archives protein structures determined by Nuclear Magnetic Resonance.

The archives collect not only the results of structure determinations, but also the measurements on which they are based. The PDB keeps the new data from X-ray structure determinations, and the BioMagRes Bank those from NMR.

The PDB assigns a four-character identifier to each structure deposited. The first character is a number from 1–9. Do not expect mnemonic significance. In many cases several entries correspond to one protein—solved in different states of ligation, or in different crystal forms, or re-solved using better crystals or more accurate data collection techniques. For instance, there have been at least four generations of sperm whale myoglobin crystal structures.

It is easy to retrieve a structure if you know its identifier. From the RCSB home page, entering a PDB ID and selecting Explore gives a 1-page summary of the entry. Figure 3.1 shows the summary page for the thioredoxin structure, identifier 2TRX. Links from this page take you to:

- The publication in which the entry was described, via the bibliographic database PubMed

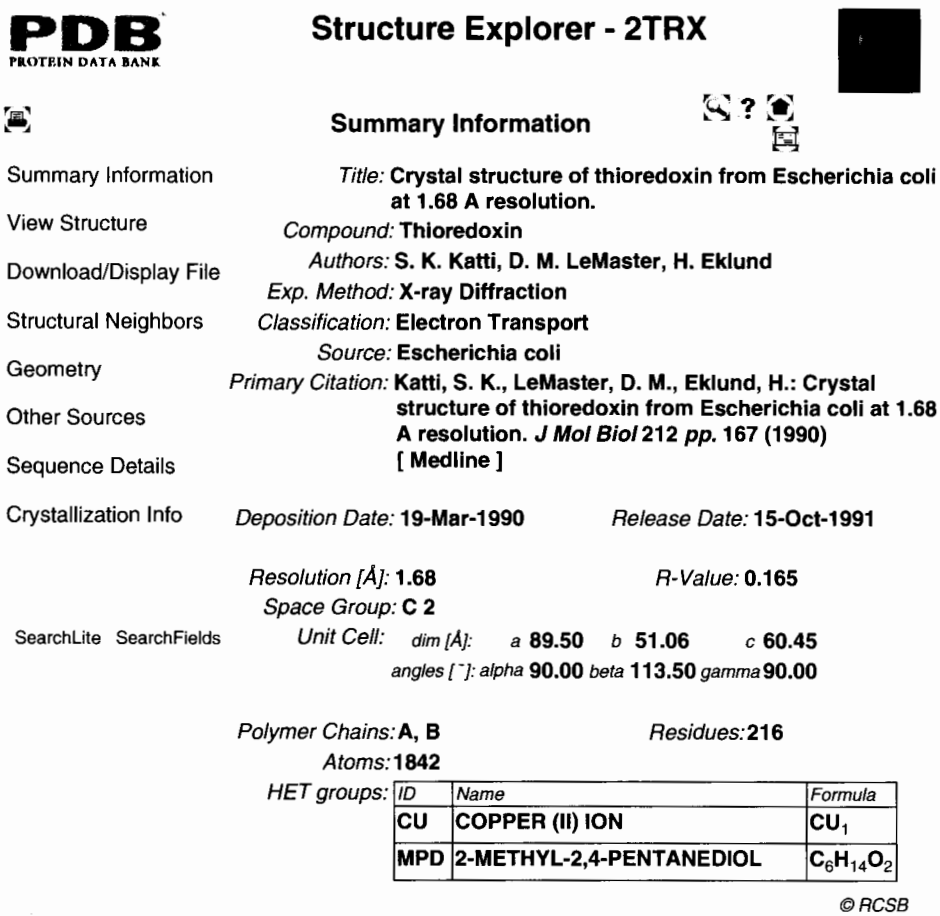


Fig. 3.1 The summary page for the PDB entry 2TRX, *E. coli* thioredoxin.

- Pictures of the structure (some of these may require that you install a viewing program on your computer)
- Access to the file containing the entry itself
- Lists of related structures, according to several different classifications of protein structures
- Stereochemical analysis—the distribution of bond lengths and angles, and conformational angles
- Sources of other information about this entry
- The sequence and secondary structure assignment
- Details about the crystal form and methods by which the crystals were produced

Fine, if you know the identifier. If not, how do you find it? A simple tool accessible from the PDB home page, called SearchLite, permits a search for keywords. Entering `coli and thioredoxin` returns 145 entries, including 2TRX and other crystal structures of the same molecule or mutants, but also several structures of Staphylococcal nuclease, because embedded in the nuclease structure entries is a reference to an article that contains the word thioredoxin in the title. The information returned would easily permit you to choose structures to look at or analyse, according to your particular interest in this family of molecules.

The PDB also offers more complex browsers. The Macromolecular Structure Database at the European Bioinformatics Institute (EBI) offers a useful list of facilities for searching and browsing the PDB including a search tool called OCA. OCA is a browser database for protein structure and function, integrating information from numerous databanks. Developed originally by J. Prilusky, OCA is supported by the EBI and is available there and at numerous mirror sites. (The name OCA, in addition to being the Spanish word for goose, has the same relationship to PDB as A. C. Clarke's computer HAL in the movie 2001 has to IBM.)

Another useful information source available at the EBI is the database of Probable Quaternary Structures (PQS) of the biologically active forms of proteins. Often the asymmetric unit of the crystal structure, as deposited in the PDB entry, contains only part of the active unit, or alternatively multiple copies of the active unit. In many cases it is not obvious how to go from the deposited entry to the active form, and this information is available in PQS.

Indicators of structure quality

X-ray crystal structure analysis produces estimates of the positions of the atoms in a molecule and of their effective sizes, known as **B-factors**. An important feature of the experimental data (the absolute values of the Fourier coefficients of the electron density) is that all atoms contribute to all observations. It is difficult to estimate errors in individual atomic positions.

Crystal structure determinations are at the mercy of the degree of order in different parts of the molecule. (Order is the extent to which different unit cells of the crystal are exact copies of one another.)

The degree of order governs the available **resolution** of the experimental data. Resolution is an index of potential quality of an X-ray structure determination. It measures the ratio of the number of parameters to be determined to the number of observations. In structure determinations of small organic molecules or of minerals, this ratio is usually generous: ~10. But for a typical protein crystal:

	Low resolution			...		**High**
Resolution in Å	4.0	3.5	3.0	2.5	2.0	1.5
Ratio of observations to parameters	0.3	0.4	0.6	1.1	2.2	3.8

(Resolution measures the fineness of the details that can be distinguished, hence the lower the number, the higher the resolution.)

In addition to disorder, errors in crystal structures reflect both errors in data and errors in solving the structure. A comparison of four independently-solved structures of interleukin-1β showed an average variation in atomic position of 0.84 Å, higher than the expected experimental error.

Many crystallographers deposit their experimental data along with the solved structures. This permits detailed checks on the results. But in many cases the experimental data are not available. How can one then assess the quality of a structure? B-factors provide important clues; high B-factors in an entire region suggest that the region has not been well-determined. This usually reflects imperfect order in the crystal. Programs can flag stereochemical outliers—exceptions to regularities common to well-determined protein structures. The entries corresponding to the PDB entries in `www.cmbi.kun.nl/gv/pdbreport` describe diagnostic analysis and identification of problems and outliers.

But although outliers are relatively easy to *detect*, it is difficult to decide whether they are correct but unusual features of the structure, or the result of errors in building the model, or the inevitable result of crystal disorder. Proper assessment requires access to the experimental data; and fixing real errors may well require the attention of an experienced crystallographer. The conclusion seems inescapable that structure factors should be archived and available.

Nuclear Magnetic Resonance (NMR)

NMR is the second major technique for determining macromolecular structure. It produces structures that are generally correct in topology but not as precise as a good X-ray structure determination and therefore less useful for the study of fine structural details. Crystallographers report a single structure, or only a small number. NMR spectroscopists usually produce a family of ~10–20 related structures or even more, calculated from the same experimental data. Comparison across such an ensemble indicates precision; regions in which the local variation in structure is small are well defined by the data. This is a rough equivalent of the crystallographer's B-factor.

Web resources: Protein and nucleic acid structures

Home page of Protein Data Bank:
http://www.rcsb.org

Home page of EBI Macromolecular Structure Database:
http://msd.ebi.ac.uk/

Home page of BioMagResBank:
http://www.bmrb.wisc.edu/

Searching the Protein Data Bank:

Home page of SCOP (Structural Classification of Proteins):
http://scop.mrc-lmb.cam.ac.uk/scop/

List of browsers: http://pdb-browsers.ebi.ac.uk/browse_it.shtml

OCA: http://oca.ebi.ac.uk/oca-bin/ocamain

Database of Protein Quaternary Structure:
http://pqs.ebi.ac.uk/

Reports of structure quality:
http://www.cmbi.kun.nl/gv/pdbreport

Classifications of protein structures

Several web sites offer hierarchical classifications of the entire Protein Data Bank according to the folding patterns of the proteins (see Chapter 5):

- SCOP: Structural Classification of Proteins
- CATH: Class/Architecture/Topology/Homologous superfamily?
- DALI: based on extraction of similar structures from distance matrices
- CE: a database of structural alignments

These sites are useful general entry points to protein structural data. For instance, SCOP offers facilities for searching on keywords to identify structures, navigation up and down the hierarchy, generation of pictures, access to the annotation records in the PDB entries, and links to related databases.

Specialized, or 'boutique' databases

Many individuals or groups select, annotate, and recombine data focused on particular topics, and include links affording streamlined access to information about subjects of interest.

For instance, the protein kinase resource is a specialized compilation that includes sequences, structures, functional information, laboratory procedures, lists of interested scientists, tools for analysis, a bulletin board, and links.

The HIV protease database archives structures of Human Immunodeficiency Virus 1 proteinases, Human Immunodeficiency Virus 2 proteinases, and Simian Immunodeficiency Virus Proteinases, and their complexes; and provides tools for their analysis and links to other sites with AIDS-related information. This database contains some crystal structures not deposited in the PDB.

In the field of immunology:

- IMGT, the international ImMunoGeneTics database, is a high-quality integrated database specializing in Immunoglobulins, T-cell receptors and Major Histocompatibility Complex (MHC) molecules of all vertebrate species. The IMGT server provides a common access to all Immunogenetics data. At present, it includes two databases: IMGT/LIGM-DB, a comprehensive database of immunoglobulin and T-cell receptor gene sequences from human and other vertebrates, with translation for fully annotated sequences; and IMGT/HLA-DB, a database of the human MHC referred to as HLA (Human Leucocyte Antigens).
- KABAT—Database of Sequences of Proteins of Immunological Interest—North-Western University (USA)
- MHCPEP—Major Histocompatibility Complex Binding Peptides Database—WEHI (Melbourne, Australia)

Web resources: Databases for specific protein families

Protein kinases:
http://www.sdsc.edu/kinases/

HIV proteases:
http://www-fbsc.ncifcrf.gov/HIVdb/

Immunology:
IGMT: http://imgt.cines.fr
KABAT: http://immuno.bme.nwu.edu/
MHCPEP: http://wehih.wehi.edu.au/mhcpep/

Expression and proteomics databases

Recall the central dogma: DNA makes RNA makes protein. Genomic databases contain DNA sequences. Expression databases record measurements of *mRNA* levels, usually via **ESTs** (expressed sequence tags–short terminal sequences of cDNA synthesized from mRNA) describing patterns of gene transcription. Proteomics databases record measurements on *proteins*, describing patterns of gene translation.

Comparisons of expression patterns give clues to: (1) the function and mechanism of action of gene products, (2) how organisms coordinate their control over metabolic processes in different conditions—for instance yeast under aerobic or anaerobic conditions, (3) the variations in mobilization of genes at different stages of the cell cycle, or of the development of an organism, (4) mechanisms of antibiotic resistance in bacteria, and consequent suggestion of targets for drug development (5) the response to challenge by a parasite, (6) the response to medications of different types and dosages, to guide effective therapy.

There are many databases of ESTs. In most, the entries contain fields indicating tissue of origin and/or subcellular location, state of development, conditions of

growth, and quantitation of expression level. Within GenBank the dbEST collection currently contains almost 23 million entries, from 719 species, led by:

Species with largest number of entries in dbEST

Species	Number of entries
Homo sapiens (human)	5 654 825
Mus musculus + *domesticus* (mouse)	4 235 142
Ciona intestinalis (primitive chordate)	684 280
Rattus sp. (rat)	636 658
Triticum aestivum (wheat)	559 149
Danio rerio (zebrafish)	532 545
Gallus gallus (chicken)	494 605
Bos *taurus* (cattle)	465 743
Zea mays (maize)	415 211
Xenopus tropicalis	392 901
Xenopus laevis (African clawed frog)	385 714
Drosophila melanogaster (fruit fly)	382 439
Hordeum vulgare + subsp. *vulgare* (barley)	356 856
Glycine max (soybean)	334 668
Sus scrofa (pig)	328 573
Arabidopsis thaliana (thale cress)	322 641
Caenorhabditis elegans (nematode)	298 805
Oryza sativa (rice)	284 007

Some EST collections are specialized to particular tissues (e.g. muscle, teeth) or to species. In many cases there is an effort to link expression patterns to other knowledge of the organism. For instance, the Jackson Lab Gene Expression Information Resource Project for Mouse Development coordinates data on gene expression and developmental anatomy.

Many databases provide connections between ESTs in different species, for instance, linking human and mouse homologues, or relationships between human disease genes and yeast proteins. Other EST collections are specialized to a type of protein, for instance, cytokines. A large effort is focussed on cancer: integrating information on mutations, chromosomal rearrangements, and changes in expression patterns, to identify genetic changes during tumour formation and progression.

Although of course there is a close relationship between patterns of transcription and patterns of translation, direct measurements of protein contents of cells and tissues—proteomics—provides additional valuable information. Because of differential rates of translation of different mRNAs, measurements of proteins directly give a more accurate description of patterns of gene expression than measurements of transcription. Post-translational modifications can be detected *only* by examining the proteins.

Proteome analysis involves separation, identification, and determination of the quantitative amounts of proteins in a sample (see Chapter 6). Proteome databases store images of gels, and their interpretation in terms of protein patterns. For each protein, an entry typically records (see Weblem 3.21):

- identification of protein
- relative amount
- function
- mechanism of action
- expression pattern
- subcellular localization
- related proteins
- post-translational modifications
- interactions with other proteins
- links to other databases

Bioinformatics is contributing to the development of these databases, and also to the development of algorithms for comparing and analysing the patterns they contain.

Databases of metabolic pathways

The Kyoto Encyclopedia of Genes and Genomes (KEGG) collects individual genomes, gene products and their functions, but its special strengths lie in its integration of biochemical and genetic information. KEGG focuses on interactions: molecular assemblies, and metabolic and regulatory networks. It has been developed under the direction of M. Kanehisa.

KEGG organizes five types of data into a comprehensive system:

1. Catalogues of chemical compounds in living cells
2. Gene catalogues
3. Genome maps
4. Pathway maps
5. Orthologue tables

The catalogues of chemical compounds and genes—items 1 and 2—contain information about particular molecules or sequences. Item 3, genome maps, integrates the genes themselves according to their appearance on chromosomes. In some cases knowing that a gene appears in an operon can provide clues to its function.

Item 4, the pathway maps, describe potential networks of molecular activities, both metabolic and regulatory. A metabolic pathway in KEGG is an idealization corresponding to a large number of possible metabolic cascades. It can generate a real metabolic pathway of a particular organism, by matching the proteins of that organism to enzymes within the reference pathways.

One enzyme in one organism would be referred to in KEGG in its orthologue tables, item 5, which link the enzyme to related ones in other organisms. This permits analysis of relationships between the metabolic pathways of different organisms.

KEGG derives its power from the very dense network of links among these categories of information, and additional links to many other databases to which the system maintains access. Two examples of the kinds of questions that can be treated by KEGG are:

- It has been suggested that simple metabolic pathways evolve into more complex ones by gene duplication and subsequent divergence. Searching the pathway catalogue for sets of enzymes that share a folding pattern will reveal clusters of paralogues.
- KEGG can take the set of enzymes from some organism and check whether they can be integrated into known metabolic pathways. A gap in a pathway suggests a missing enzyme or an unexpected alternative pathway.

Bibliographic databases

MEDLINE (based at the US National Library of Medicine) integrates the biomedical literature, including very many papers dealing with subjects in molecular biology not overtly clinical in content. It is included in PubMed, a bibliographical database offering abstracts of scientific articles, integrated with other information retrieval tools of the National Center for Biotechnology Information (NCBI) within the National Library of Medicine (http://www.ncbi.nlm.nih.gov/PubMed/).

One very effective feature of PubMed is the option to retrieve *related articles*. This is a very quick way to 'get into' the literature of a topic. Combined with the use of a general search engine for web sites that do not correspond to articles published in journals, fairly comprehensive information is readily available about most subjects. Here's a tip: if you are trying to start to learn about an unfamiliar subject, try adding the keyword *tutorial* to your search in a general search engine, or the keyword *review* to your search in PubMed.

Almost all scientific journals now place their tables of contents, and in many cases their entire issues, on web sites. The US National Institutes of Health have established a centralized web-based library of scientific articles, called PubMed Central (http://www.pubmedcentral.nih.gov/). In collaboration with scientific journals, the NCBI is organizing the electronic distribution of the full texts of published articles.

A new organization, the Public Library of Science, has the goal of making the scientific (including medical) literature publicly and freely accessible. A non-profit organization, the Public Library of Science has received support from foundations for its efforts in distributing literature published by others, and to start its own publications, which will permit exploration of different relationships—including but not limited to economic ones—between authors, publishers and readers.

Surveys of molecular biology databases and servers

It is difficult to explore any topic in molecular biology on the web without quickly bumping into a list of this nature. Lists of web resources in molecular biology are very common. They contain, to a large extent, the same information,

but vary widely in their 'look and feel' aspects. The real problem is that unless they are curated they tend to degenerate into lists of dead links. (A draft of this section contained a reference to a web site that contained a reasonable survey. Returning to it two months later, the name of the site had changed, and over half of the sites listed had disappeared.)

This book does not contain a long annotated list of relevant and recommended sites, for the following reasons: (1) You don't want a long list, you need a short one. (2) The Web is too volatile for such a list to stay useful for very long. *It is much more effective to use a general search engine to find what you want at the moment you want it.* Each year the January issue of the journal Nucleic Acids Research contains a set of articles on databases in molecular biology. This is an invaluable reference.

Moreover, the content of the databases is expanding all the time. If you try the searches described in examples in this chapter you will obtain more 'hits' than the results printed here. (Indeed, I have not hesitated to use older sets of results if, because they contain more variety than the latest results, they seem more informative. The problem of suppressing extensive redundancy in responses to websearches is a challenge for research in the field of information retrieval.)

My advice is: spend some time browsing; it won't take you long to find a site that appears reasonably stable and has a style compatible with your methods of work. Alternatively, here's a site that is comprehensive and shows signs of a commitment to keeping it up to date: http://www.expasy.org/alinks.html. It is a suitable site for starting a browsing session.

Gateways to archives

Databases of nucleic acid and protein sequences maintain facilities for a very wide variety of information retrieval and analysis operations. Categories of these operations include:

1. **Retrieval of sequences from the database** Sequences can be 'called up' either on the basis of features of the annotations, or by patterns found within the sequences themselves.
2. **Sequence comparison** This is not a facility, this is a heavy industry! It was introduced in Chapter 1 and will be discussed in detail in Chapter 4. It includes the very important searches for relatives.
3. **Translation of DNA sequences to protein sequences**
4. **Simple types of structure analysis and prediction** For example, statistical methods for predicting the secondary structure of proteins from sequences alone, including hydrophobicity profiles—from which the transmembrane proteins can generally be identified (see page 193).
5. **Pattern recognition** It is possible to search for all sequences containing a pattern or combination of patterns, expressed as probabilities for finding certain sets of residues at consecutive positions. In DNA sequences, these may be

recognition sites for enzymes such as those responsible for splicing interrupted genes. In proteins, short and localized patterns sometimes identify molecules that share a common function even if there is no obvious overall relationship between their sequences. PROSITE is a collection of these protein 'signature' patterns.

6. **Molecular graphics** is necessary to provide intelligible depictions of very complicated systems. Typical applications of molecular graphics include:
   - Mapping residues believed to be involved in function, onto the three-dimensional framework of a protein. Often this will isolate an active site.
   - Classifying and comparing the folding patterns of proteins.
   - Analysing changes between closely-related structures, or between two conformational states of a single molecule,
   - Studying the interaction of a small molecule with a protein, in order to attempt to assign function, or for drug development,
   - Interactive fitting of a model to the noisy and fuzzy image of the molecule that arises initially from the measurements in solving protein structures by X-ray crystallography.
   - Design and modelling of new structures.

Access to databases in molecular biology

How to learn web skills

It would be difficult to learn to ride a bicycle by reading a book describing the sets of movements required, much less one about the theory of the gyroscope. Similarly, the place to learn web skills is at a terminal, running a browser. True enough, but there is always a certain initial period of difficulty and imbalance. Here the goal is only to provide some temporary assistance to get you started. Then, off you go!

This section contains introductions to some of the major databanks and information retrieval systems in molecular biology. In each case we show relatively simple searches and applications. When appropriate, unique features of each system will be emphasized.

ENTREZ

The National Center for Biotechnology Information, a component of the United States National Library of Medicine, maintains databases and avenues of access to them. ENTREZ offers access via the following database divisions:

- Protein
- Peptide
- Nucleotide
- Structure
- Genome

- Popset—information about populations
- OMIM—Online Mendelian Inheritance in Man

Links between various databases are a strong point of NCBI's system. The starting point for retrieval of sequences and structures is called ENTREZ: http://www.ncbi.nlm.nih.gov/Entrez/.

Let us pick a molecule—human neutrophil elastase—and search for relevant entries in the different sections of ENTREZ.

Search in ENTREZ protein database

Go to http://www.ncbi.nlm.nih.gov/Entrez/. Select `Protein: sequence database`, enter the search terms HUMAN ELASTASE and click on GO.

The Box shows fifteen answers returned by the program. (In a browser, you will also find links to the sequence databank entries.) The top hit is ELASTASE 1 PRECURSOR [HOMO SAPIENS]; other responses include elastases from other species, inhibitors from human and from leech, and tyrosyl-tRNA synthetase. (Why should a leech protein and tRNA synthetase show up in a search for human elastase? See Weblem 3.9.) Later we shall see how to tune the query to eliminate these extraneous responses.

ENTREZ responses to *human elastase* in PROTEIN database

1. elastase 1 precursor [Homo sapiens]
   gi—4731318—gb—AAD28441.1—AF120493_1[4731318]
2. ALPHA-1-ANTITRYPSIN PRECURSOR (ALPHA-1 PROTEASE INHIBITOR) (ALPHA-1-ANTIPROTEINASE)
   gi—1703025—sp—P01009—A1AT_HUMAN[1703025]
3. elastase [Mus musculus]
   gi—7657060—ref—NP_056594.1—[7657060]
4. proteinase 3 [Mus musculus]
   gi—6755184—ref—NP_035308.1—[6755184]
5. ANTIMICROBIAL PEPTIDE ENAP-2
   gi—7674025—sp—P56928—ENA2_HORSE[7674025]
6. AMBP PROTEIN PRECURSOR [CONTAINS: ALPHA-1-MICROGLOBULIN (PROTEIN HC)
   (COMPLEX-FORMING GLYCOPROTEIN HETEROGENEOUS IN CHARGE); INTER-ALPHA-TRYPSIN INHIBITOR LIGHT CHAIN (ITI-LC) (BIKUNIN) (HI-30)]
   gi—122801—sp—P02760—AMBP_HUMAN[122801]
7. ELAFIN PRECURSOR (ELASTASE-SPECIFIC INHIBITOR) (ESI) (SKIN-DERIVED ANTILEUKOPROTEINASE) (SKALP)
   gi—119262—sp—P19957—ELAF_HUMAN[119262]
8. ANTILEUKOPROTEINASE
   gi—113637—sp—P22298—ALK1_PIG[113637]

→

→

9. ANTILEUKOPROTEINASE 1 PRECURSOR (ALP) (HUSI-1) (SEMINAL PROTEINASE
INHIBITOR) (SECRETORY LEUKOCYTE PROTEASE INHIBITOR) (BLPI) (MUCUS PROTEINASE INHIBITOR) (MPI)
gi—113636—sp—P03973—ALK1_HUMAN[113636]
10. ALPHA-2-MACROGLOBULIN PRECURSOR (ALPHA-2-M)
gi—112911—sp—P01023—A2MG_HUMAN[112911]
11. tyrosyl-tRNA synthetase [Homo sapiens]
gi—4507947—ref—NP_003671.1—[4507947]
12. pancreatic elastase IIB [Homo sapiens]
gi—7705648—ref—NP_056933.1—[7705648]
13. protease inhibitor 3, skin-derived (SKALP) [Homo sapiens]
gi—4505787—ref—NP_002629.1—[4505787]
14. pancreatic elastase I (allele HEL1-36)—human (fragment)
gi—7513237—pir—S70441[7513237]
15. guamerin—Korean leech

The format of the responses is as follows. In each case, the first line gives the name and synonyms of the molecule, and the species of origin. Note that Greek letters are spelt out. The last line gives references to the source databanks: gi = GenInfo Identifier, (see page 25), gb = GenBank accession number, sp = Swiss-Prot, pir = Protein Identification Resource, ref = the Reference Sequence project of NCBI. The entries retrieved include elastases from human and other species, and also inhibitors of elastase.

Opening the entry corresponding to the first hit retrieves the file shown in the next Box. The first lines are mostly database housekeeping—accession numbers, molecule name, date of deposition, etc. Then descriptive material such as the source, this case human, with the full taxonomic classification, credit to the scientists who deposited the entry, and literature references. Finally the particular scientific information: the location of the gene, and its product (CDS = coding sequence), and the sequence itself (see Exercise 3.2).

Searches in ENTREZ nucleotide database

We next look again for HUMAN ELASTASE, this time in the Nucleotide database. Let us try to tune the search, to eliminate the responses that refer to elastase inhibitors.

1. Select NUCLEOTIDE at the ENTREZ site.
2. Click on LIMITS, select ORGANISM from the pulldown menu, type HOMO SAPIENS in the search box.
3. Next select SUBSTANCE NAME from the pulldown menu, and then type AND ELASTASE in the search box.

Top result of search for human elastase in ENTREZ Protein database

```
LOCUS       AF120493_1    258 aa                    PRI       03-AUG-2000
DEFINITION  elastase 1 precursor [Homo sapiens].
ACCESSION   AAD28441
PID         g4731318
VERSION     AAD28441.1  GI:4731318
DBSOURCE    locus AF120493 accession AF120493.1
KEYWORDS    .
SOURCE      human.
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1  (residues 1 to 258)
  AUTHORS   Talas,U., Dunlop,J., Khalaf,S., Leigh,I.M. and Kelsell,D.P.
  TITLE     Human elastase 1: evidence for expression in the skin and the
            identification of a frequent frameshift polymorphism
  JOURNAL   J. Invest. Dermatol. 114 (1), 165-170 (2000)
  MEDLINE   20087075
   PUBMED   10620133
REFERENCE   2  (residues 1 to 258)
  AUTHORS   Talas,U., Dunlop,J., Leigh,I.M. and Kelsell,D.P.
  TITLE     Direct Submission
  JOURNAL   Submitted (15-JAN-1999) Centre for Cutaneous Research, Queen Mary
            and Westfield College, 2 Newark Street, London E1 2AT, UK
COMMENT     Method: conceptual translation supplied by author.
FEATURES             Location/Qualifiers
     source          1..258
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
                     /chromosome="12"
                     /map="12q13"
                     /cell_type="keratinocyte"
     Protein         1..258
                     /product="elastase 1 precursor"
     CDS             1..258
                     /gene="ELA1"
                     /coded_by="AF120493.1:42..818"
ORIGIN
        1 mlvlyghstq dlpetnarvv ggteagrnsw psqislqyrs ggsryhtcgg tlirqnwvmt
       61 aahcvdyqkt frvvagdhnl sqndgteqyv svqkivvhpy wnsdnvaagy diallrlaqs
      121 vtlnsyvqlg vlpqegaila nnspcyitgw gktktngqla qtlqqaylps vdyaicssss
      181 ywgstvkntm vcaggdgvrs gcqgdsggpl hclvngkysl hgvtsfvssr gcnvsrkptv
      241 ftqvsayisw innviasn
//
```

4. Finally select TEXT WORD from the pulldown menu, and then type NOT INHIBITOR in the search box. Now click on GO.

If you click on Details, you will find:

HOMO SAPIENS[ORGANISM] AND ELASTASE[SUBSTANCE NAME] NOT INHIBITOR [TEXT WORD]

The search returns over 400 hits, including many individual clones. The top hit (see Box) is: HOMO SAPIENS ELASTASE 1 PRECURSOR (ELA1) MRNA, COMPLETE CDS. The term 'complete cds' means complete coding sequence.

Compare this file with the result of searching in the Protein database (see Exercise 3.5).

Top result of search for human elastase in ENTREZ Nucleotide database

```
LOCUS       AF120493      952 bp    mRNA            PRI       03-AUG-2000
DEFINITION  Homo sapiens elastase 1 precursor (ELA1) mRNA, complete cds.
ACCESSION   AF120493
VERSION     AF120493.1  GI:4731317
KEYWORDS    .
SOURCE      human.
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 952)
  AUTHORS   Talas,U., Dunlop,J., Khalaf,S., Leigh,I.M. and Kelsell,D.P.
  TITLE     Human elastase 1: evidence for expression in the skin and the
            identification of a frequent frameshift polymorphism
  JOURNAL   J. Invest. Dermatol. 114 (1), 165-170 (2000)
  MEDLINE   20087075
   PUBMED   10620133
REFERENCE   2  (bases 1 to 952)
  AUTHORS   Talas,U., Dunlop,J., Leigh,I.M. and Kelsell,D.P.
  TITLE     Direct Submission
  JOURNAL   Submitted (15-JAN-1999) Centre for Cutaneous Research, Queen Mary
            and Westfield College, 2 Newark Street, London E1 2AT, UK
FEATURES             Location/Qualifiers
     source          1..952
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
                     /chromosome="12"
                     /map="12q13"
                     /cell_type="keratinocyte"
     gene            1..952
                     /gene="ELA1"
     CDS             42..818
                     /gene="ELA1"
                     /codon_start=1
                     /product="elastase 1 precursor"
                     /protein_id="AAD28441.1"
                     /db_xref="GI:4731318"
                     /translation="MLVLYGHSTQDLPETNARVVGGTEAGRNSWPSQISLQYRSGGSR
                     YHTCGGTLIRQNWVMTAAHCVDYQKTFRVVAGDHNLSQNDGTEQYVSVQKIVVHPYWN
                     SDNVAAGYDIALLRLAQSVTLNSYVQLGVLPQEGAILANNSPCYITGWGKTKTNGQLA
                     QTLQQAYLPSVDYAICSSSSYWGSTVKNTMVCAGGDGVRSGCQGDSGGPLHCLVNGKY
                     SLHGVTSFVSSRGCNVSRKPTVFTQVSAYISWINNVIASN"
BASE COUNT      226 a    261 c    250 g    215 t
ORIGIN
        1 ttggtccaag caagaaggca gtggtctact ccatcggcaa catgctggtc ctttatggac
       61 acagcaccca ggaccttccg gaaaccaatg cccgcgtagt cggagggact gaggccggga
      121 ggaattcctg gccctctcag atttccctcc agtaccggtc tggaggttcc cggtatcaca
      181 cctgtggagg gacccttatc agacagaact gggtgatgac agctgctcac tgcgtggatt
      241 accagaagac tttccgcgtg gtggctggag accataacct gagccagaat gatggcactg
      301 agcagtacgt gagtgtgcag aagatcgtgg tgcatccata ctggaacagc gataacgtgg
      361 ctgccggcta tgacatcgcc ctgctgcgcc tggcccagag cgttaccctc aatagctatg
      421 tccagctggg tgttctgccc caggagggag ccatcctggc taacaacagt ccctgctaca
      481 tcacaggctg gggcaagacc aagaccaatg ggcagctggc ccagaccctg cagcaggctt
      541 acctgccctc tgtggactat gccatctgct ccagctcctc ctactggggc tccactgtga
      601 agaacaccat ggtgtgtgct ggtggagatg gagttcgctc tggatgccag ggtgactctg
      661 ggggcccccc ccattgcttg gtgaatggca agtattctct ccatggagtg accagctttg
      721 tgtccagccg gggctgtaat gtctccagga agcctacagt cttcacccag gtctctgctt
      781 acatctcctg gataaataat gtcatcgcct ccaactgaac attttcctga gtccaacgac
      841 cttcccaaaa tggttcttag atctgcaata ggacttgcga tcaaaaagta aaacacattc
      901 tgaaagacta ttgagccatt gatagaaaag caaataaaac tagatataca tt
//
```

Searches in ENTREZ genome database

A search for HUMAN ELASTASE returns:

1. NC_000967 CAENORHABDITIS ELEGANS CHROMOSOME III[64] LCL—WORM_CHR_III
2. NC_001099 HOMO SAPIENS CHROMOSOME 19[19] REF—NC_001099—HSAP-19
3. NC_001065 HOMO SAPIENS CHROMOSOME 14[14] REF—NC_001065—HSAP-14
4. NC_001044 HOMO SAPIENS CHROMOSOME 11[11] REF—NC_001044—HSAP-11
5. NC_001008 HOMO SAPIENS CHROMOSOME 6[6] REF—NC_001008—HSAP-6

Why should a *C. elegans* protein appear in a search for human elastase? The entry NC_000967 is chromosome III of *C. elegans* in its entirety. Comments on one of the genes detected include:

```
gene="T07A5.1" /note="weak similarity with elastase (PIR accesssion number A406659)"
```

Many other genes in *C. elegans* are annotated with similarities to human proteins. However, although *C. elegans* does contain an elastase, this is *not* flagged as similar to human elastase, although it is a homologue.

Searches in ENTREZ structure database

Is the three-dimensional structure of human elastase known? Select the STRUCTURE database, from the choices to the left of the query box, and rerun the search. The program returns at least five answers:

1JK3	CRYSTAL STRUCTURE OF HUMAN MMP-12 (MACROPHAGE ELASTASE) AT TRUE ATOMIC RESOLUTION
1HAZ	SNAPSHOTS OF SERINE PROTEASE CATALYSIS: (C) ACYL-ENZYME INTERMEDIATE BETWEEN PORCINE PANCREATIC ELASTASE AND HUMAN BETA-CASOMORPHIN-7 JUMPED TO PH 9 FOR 1 MINUTE
1HAX	SNAPSHOTS OF SERINE PROTEASE CATALYSIS: (A) ACYL-ENZYME INTERMEDIATE BETWEEN PORCINE PANCREATIC ELASTASE AND HUMAN BETA-CASOMORPHIN-7 AT PH 5
1B0F	CRYSTAL STRUCTURE OF HUMAN NEUTROPHIL ELASTASE WITH MDL 101, 146
1QIX	PORCINE PANCREATIC ELASTASE COMPLEXED WITH HUMAN BETA- CASOMORPHIN-7

The designations 1JK3, 1HAZ, 1HAZ, 1BOF, and 1QIX are entry codes from the Protein Data Bank.

OOPS!—we may not realize it, but we have missed many useful entries. There are many elastase structures solved in complex with inhibitors, which we have asked the system to reject. Deleting NOT INHIBITORS and rerunning the query returns several more structures.

Searches in the bibliographic database PubMed

Perhaps it is time to look at what people have had to say about our molecule. Of course the literature on elastase is huge. A search in PubMed for HUMAN ELASTASE returns over 7500 entries. To prune the results, let us try to find citations to articles describing the role of elastase in disease. A search for HUMAN ELASTASE DISEASE returns over 1600 entries. What about specific elastase **mutants** related to human disease? A search for HUMAN ELASTASE DISEASE MUTATION returns more than 40 articles, in reverse chronological order. Here are 10 of them:

Hermans MH, Touw IP. Significance of neutrophil elastase mutations versus G-CSF receptor mutations for leukemic progression of congenital neutropenia. Blood. 2001 Apr 1;97(7):2185–6. No abstract available.

Li FQ, Horwitz M. Characterization of mutant neutrophil elastase in severe congenital neutropenia. J Biol Chem. 2001 Apr 27;276(17):14230–41.

Ye S. Polymorphism in matrix metalloproteinase gene promoters: implication in regulation of gene expression and susceptibility of various diseases. Matrix Biol. 2000 Dec;19(7):623–9. Review.

Dale DC, Person RE, Bolyard AA, Aprikyan AG, Bos C, Bonilla MA, Boxer LA, Kannourakis G, Zeidler C, Welte K, Benson KF, Horwitz M. Mutations in the gene encoding neutrophil elastase in congenital and cyclic neutropenia. Blood. 2000 Oct 1;96(7):2317–22.

McGettrick AJ, Knott V, Willis A, Handford PA. Molecular effects of calcium binding mutations in Marfan syndrome depend on domain context. Hum Mol Genet. 2000 Aug 12;9(13):1987–94.

Rashid MH, Rumbaugh K, Free in PMC, Passador L, Davies DG, Hamood AN, Iglewski BH, Kornberg A. Polyphosphate kinase is essential for biofilm development, quorum sensing, and virulence of Pseudomonas aeruginosa. Proc Natl Acad Sci U S A. 2000 Aug 15;97(17):9636–41.

Jormsjo S, Ye S, Moritz J, Walter DH, Dimmeler S, Zeiher AM, Henney A, Hamsten A, Eriksson P. Allele-specific regulation of matrix metalloproteinase-12 gene activity is associated with coronary artery luminal dimensions in diabetic patients with manifest coronary artery disease. Circ Res. 2000 May 12;86(9):998–1003.

Talas U, Dunlop J, Khalaf S, Leigh IM, Kelsell DP. Human elastase 1: evidence for expression in the skin and the identification of a frequent frameshift polymorphism. J Invest Dermatol. 2000 Jan;114(1):165–70.

Horwitz M, Benson KF, Person RE, Aprikyan AG, Dale DC. Mutations in ELA2, encoding neutrophil elastase, define a 21-day biological clock in cyclic haematopoiesis. Nat Genet. 1999 Dec;23(4):433–6.

Griffin MD, Torres VE, Grande JP, Kumar R. Vascular expression of polycystin. J Am Soc Nephrol. 1997 Apr;8(4):616–26.

There are references to a relation between mutations in neutrophil elastase and neutropenia—a low level of a type of white blood cells called neutrophils. To pursue this, we can look for elastase in the database of human genetic disease:

Online Mendelian Inheritance in Man (OMIM™)

OMIM is a database of human genes and genetic disorders. It was originally compiled by V. A. McKusick, M. Smith and colleagues and published on paper. The National Center for Biotechnology Information (NCBI) of the US National Library of Medicine has developed it into a database accessible from the Web, and introduced links to other archives of related information, including sequence databanks and the medical literature. OMIM is now well integrated with the NCBI information retrieval system ENTREZ. A related database, the OMIM Morbid Map, treats genetic diseases and their chromosomal locations.

The response to ELASTASE in a search of OMIM describes the results linking mutations in the gene to cyclic neutropenia.

The collection of results on elastase that we have assembled would support research on the system; for instance, we could map elastase mutants onto the structure of the molecule to see whether we could derive clues to the cause of cyclic neutropenia.

The Sequence Retrieval System (SRS)

SRS, originally developed by T. Etzold, is an integrated system for information retrieval from many different sequence databases, and for feeding the sequences retrieved into analytic tools such as sequence comparison and alignment programs.

SRS can search a total of 141 databases of protein and nucleotide sequences, metabolic pathways, 3D structures and functions, genomes, and disease and phenotype information (see Box). These include many small databases such as the Prosite and Blocks databases of protein structural motifs, transcription factor databases, and databases specialized to certain pathogens.

Some categories of databases searchable from SRS

Nucleotide Sequence	Literature
Uniprot	Mapping
Protein Function	Protein Structure
Enzymes	Metabolic Pathways
Mutation, SNP	Gene Ontology

In addition to the number and variety of databases to which it offers access, SRS offers tight links among the databases, and fluency in launching applications. A search in a single database component can be extended to a search in the complete network; that is, entries in all databases pertaining to a given protein can be found easily. Similarity searches and alignments can be launched directly, without saving the responses in an intermediate file.

In an SRS session, you begin by selecting one or more of the databases in which to search. The databases are grouped by category: nucleotide sequence-related, protein-related, etc. Then you can enter a set of query terms. As with ENTREZ, you may search for them either in all fields, or assign terms to categories. The program will respond with a set of entries containing your terms. As follow-on queries one might:

1. Examine one of the sequences identified by linking to the file retrieved.
2. Select one or more of the sequences identified and search other databases for related entries.
3. Launch an application, such as a secondary structure prediction or a multiple sequence alignment.

Other options on the search results page allow you to create and download reports on the selected matches. This might be simply a listing of the sequences, or the result of a more complex analysis of the results. Applying the multiple sequence alignment program CLUSTAL-W to the results produces an alignment such as appears in Plate III.

The Protein Identification Resource (PIR)

The PIR is an effective combination of a carefully curated database, information retrieval access software, and a workbench for investigations of sequences. The PIR also produces the Integrated Environment for Sequence Analysis (IESA). Think of this as an analysis package sitting on top of a retrieval system. Its functionality includes browsing, searching and similarity analysis, and links to other databases. Users may:

- Browse by annotations.
- Search selected text fields for different annotations, such as Superfamily, Family, Title, Species, Taxonomy group, Keywords and Domains.
- Analyse sequences using BLAST or FASTA Searches, Pattern Match, Multiple alignment.
- Global and Domain Search, and Annotation-sorted Search.
- View Statistics for Superfamily, Family, Title, Species, Taxonomy group, Keywords, Domains, Features.
- View Links to other databases, including PDB, COG, KEGG, WIT, and BRENDA.
- Select Specialized Sequence Groups such as Human, Mouse, Yeast and *E. coli* genomes.

The URLs for search of PIR by Text terms are:
In the US: `http://www-nbrf.georgetown.edu/pirwww/search/textpsd.html`
In Europe: `http://www.mips.gsf.de`

One feature of the PIR International system is the search for a specific peptide. Looking at the alignment of mammalian elastases in Plate III, we note at positions 220–228 a conserved motif: most of the sequences contain CNGDSGGPLN.

In the PIR, we can select PATTERN/PEPTIDE MATCH and search for exact matches for the subsequence CNGDSGGPLN giving 63 results.

Returning to the alignment table (Plate III), variations in the pattern appear in some molecules. The more general search for C[RNQF]GDSG[GS]PL[HNV], in which [XYZ] means a position containing either X or Y or Z, would pull out all the mammalian elastases in the alignment, plus a total of 82 sequences in all. Even these are not all the elastase homologues in the databank, as one could find by running a PSI-BLAST search for any of the sequences, or, remaining strictly within PIR, by looking up elastase in the PROT-FAM database. The pattern matches 20 families, all serine proteinases.

We are well on the way to generating a complete list of homologues.

ExPASy—Expert Protein Analysis System

ExPASy is the information retrieval and analysis system of the Swiss Institute of Bioinformatics, which (in collaboration with the European Institute of Bioinformatics) also produces the protein sequence databases SWISS-PROT and TrEMBL. TrEMBL contains translations of nucleotide sequences from the EMBL Data Library not yet fully integrated into SWISS-PROT.

Opening the main web page of ExPASy (http://www.expasy.org) and selecting SWISS-PROT and TrEMBL gives access to a set of information retrieval tools, including a link to SRS. There is also the option of searching SWISS-PROT directly. If we select FULL TEXT SEARCH and probe SWISS-PROT with the single term ELASTASE, we find ELNE_HUMAN, the real goal of our search, and around 150 other hits, including many inhibitors. One elastase homologue found is from the blood fluke: CERC_SCHMA. Both sequences are precursors; in the following alignment of these two sequences, upper case letters indicate the mature enzyme:

```
CERC_SCHMA  --msnrwrfvvvvtlftycltfervstwlIRSGEPVQHPAEFPFIAFLTTER-TMCTGSL 57
ELNE_HUMAN  mtlgrrlaclflacvlpalllggtalaseIVGGR-RARPHAWPFMVSLQLRGGHFCGATL 59
              :..*    :.:. ::.  *    . : * .*.    :*  :**:. *  .  :* .:*

CERC_SCHMA  VSTRAVLTAGHCVCSPLPVIRVSFLTLRNGDQQGIHHQPSGVKVAPGYMPSCMSARQRRP 117
ELNE_HUMAN  IAPNFVMSAAHCVAN--VNVRAVRVVLGAHNLSRREP--TRQVFAVQRIFENGYDP      111
            ::.. *::*.***..  :.*  : : * ::  :::*   .  : . : .  *

CERC_SCHMA  IAQTLSGFDIAIVMLAQMVNLQSGIRVISLPQPSDIPPPGTGVFIVGYGRDDNDRDPSRK 177
ELNE_HUMAN  VNLLN--DIVILQLNGSATINANVQVAQLPAQGRRLGNGVQCLAMGWGLLGRNRG--     164
            :      **.*: *   ..:::.::*  .**  .    *. : :*:*  ..:*.

CERC_SCHMA  NGGILKKGRATIMECRHATNGNPICVKAGQNFGQLPAPGDSGGPLLPS-LQGPVLGVVSH 236
ELNE_HUMAN  IASVLQELNVTVVTS-LCRRSNVCTLVRGRQAG-VCFGDSGSPLVCNGLIHGIASFVRG  221
            ..:*::  ..*:: .   . ..*  :  *:: * .  ****.**: . *    : ..*

CERC_SCHMA  GVTLPNLPDIIVEYASVARMLDFVRSNI----------------   264
ELNE_HUMAN  GCASGLYPDAFAPVAQFVNWIDSIIQRSEDNPCPHPRDPDPASRTH 267
            * :   ** :. *.... :* : ..
```

The structure of human neutrophil elastase is known from X-ray crystallography, but that of the blood fluke elastase is not.

One of the unique facilities of the ExPASy server is the link to SWISS-MODEL, an automatic web server for building homology models. Opening SWISS-MODEL and choosing FIRST APPROACH MODE (the simplest), we can simply enter the

SWISS-PROT code CERC_SCHMA, and launch the application. Model building is not a trivial operation, so the job is done off-line and the results sent by e-mail.

We shall discuss SWISS-MODEL further in Chapter 5.

Ensembl

Ensembl (http://www.ensembl.org) is intended to be the universal information source for the human genome. The goals are to collect and annotate all available information about human DNA sequences, link it to the master genome sequence, and make it accessible to the many scientists who will approach the data with many different points of view and different requirements. To this end, in addition to collecting and organizing the information, very serious effort has gone into developing computational infrastructure. Suitable conventions of nomenclature are established: it is not trivial to devise a scheme for maintaining stable identifiers in the face of data that will be undergoing not only growth but revision. The most visible result of these efforts is the web site, very rich in facilities both for browsing and for focussing in on details.

Ensembl is a joint project of the European Bioinformatics Institute and The Sanger Centre; participants include E. Birney, M. Clamp, T. Cox and T. J. P. Hubbard. However, Ensembl is organized as an open project, encouraging outside contributions. All but the most naive of readers must recognize the great demands this will place on quality control procedures.

Data collected in Ensembl includes genes, SNPs, repeats, and homologies. Genes may either be known experimentally, or deduced from the sequence. Because the experimental support for annotation of the human genome is so variable, Ensembl presents the supporting evidence for identification of every gene. Very extensive linking to other databases containing related information, such as Online Mendelian Inheritance in Man (OMIM), or expression databases, extend the accessible information.

Ensembl is structured around the human genome sequence. Users may identify regions via several types of lookups or searches:

- BLAST searches on a sequence or fragment
- Browsing—starting at the chromosome level then zooming in
- Gene name
- Relation to diseases, via OMIM
- ENSEMBL ID if the user knows it
- General text search

A text search in Ensembl for BRCA1 produced the page displayed, showing the region around the BRCA1 locus. The upper frame shows a megabase, mapped to the q21.2 and q21.31 bands of chromosome 17. It reports markers, and assigned genes. The bottom frame shows a more detailed view. Note the control panels between the two frames that permit navigation and 'zooming'. The bottom frame shows a 0.1 megabase region, reporting many more details, including the detailed structure of the BRCA1 gene, and the SNPs observed.

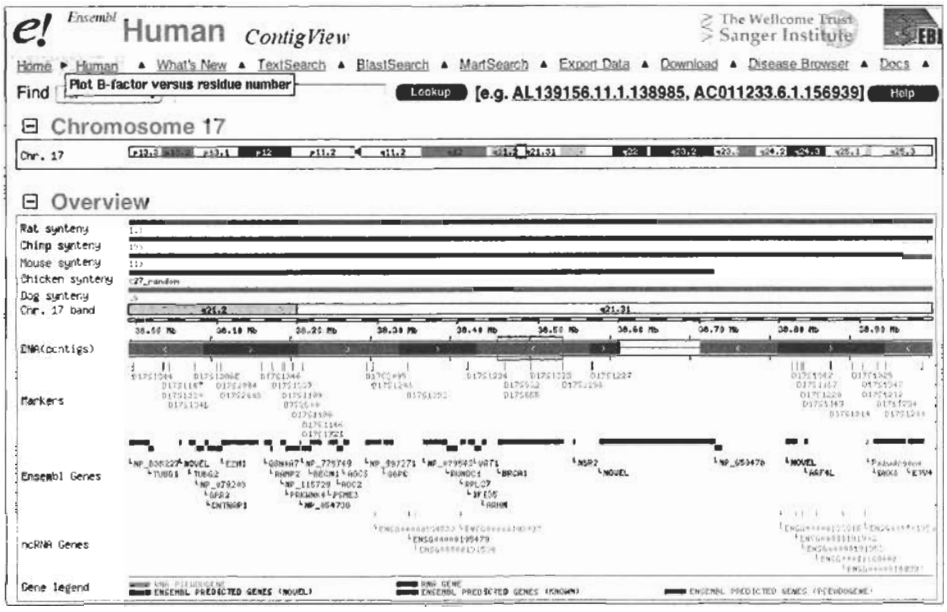


Fig. 3.4 A portion of the ENSEMBL page showing the region of the human genome surrounding BRAC1.

Where do we go from here?

We have visited only a few of the many databanks in molecular biology accessible on the Web. In the short term, readers will explore these sites and others, and become familiar with not only with the contents of the Web but its dynamics—the appearance and disappearance of sites and links. There are various biological metaphors for the Web—as an ecosystem that is evolving, or that is growing polluted by dead sites and links to dead sites. Unfortunately, there is no effective mechanism for decay and recycling as in the organic world!

Databanks are developing more effective avenues of intercommunication, to the point where ever more intimate links shade into apparent coalescence. The time is not far off when there will be one molecular biology databank, with many avenues of access. Scientists will be able to configure their own access to selected slices of the information, creating 'virtual databases' tailored to their own needs.

Recommended Reading

Each year the January issues of the journal *Nucleic Acid Research* contains a set of articles on databases in molecular biology. This is an invaluable reference.

Bishop, M. J., *Genetics databases* (London: Academic, 1999). [A compendium of databases, access and analysis.]

Zdobnov, E. M., Lopez, R., Apwelier, R. & Etzold, T. (2002). The EBI SRS server—new features, *Bioinformatics*, **18**, 1149–1150.

Exercises, Problems, and Weblems

Exercises

3.1 A database of vehicles has entries for the following: bicycle, tricycle, motorcycle, car. It stores only the following information about each entry: (1) how many wheels (a number), and (2) source of propulsion = human or engine. For every possible pair of vehicles, devise a logical combination of query terms referring to either the exact value or the range in the number of wheels, and to the source of propulsion, that will return the two selected vehicles and no others.

3.2 The Box on page 144 showed the NCBI protein entry for human elastase 1 precursor. On a photocopy of this page, indicate which items are (a) purely database housekeeping, (b) peripheral data such as literature references, (c) the results of experimental measurements, (d) information inferred from experimental measurements.

3.3 Write a PERL script to extract the amino acid sequence from an entry in the ENTREZ protein sequence database as shown in the Box, page 144, and convert it to FASTA format.

3.4 Compare the file retrieved by a search in NCBI for human elastase under Protein (page 144) and Nucleotide (page 145). On photocopies of these two pages, mark with a highlighter all items that the two files have in common.

Weblems

3.1 Retrieve the complete SWISS-PROT entry for bovine pancreatic trypsin inhibitor (*not* pancreatic secretory trypsin inhibitor) and the complete PIR entry for this protein. What information does each have that the other does not?

3.2 Find a list of official and unofficial mirror sites of the Protein Data Bank. Which is closest to you?

3.3 Find all structures of sperm whale myoglobin in the Protein Data Bank and draw a histogram of their dates of deposition.

3.4 Find protein structures determined by Peter Hudson, alone or with colleagues.

3.5 Design a search string for use with the Protein Data Bank tool SearchLite that would return *E. coli* thioredoxin structures but *not* Staphylococcal nuclease structures.

3.6 For what fraction of structures determined by X-ray crystallography deposited in the Protein Data Bank have structure factor files also been deposited?

3.7 Protein Data Bank entry 8XIA contains the structure of one monomer of D-Xylose isomerase from *Streptomyces rubiginosus*. What is the probable quaternary structure? How was the geometry of the assembly corresponding to the probable quaternary structure derived from the coordinates in the entry?

3.8 Find structural neighbours of Protein Data Bank entry 2TRX (*E. coli* thioredoxin), according to SCOP, CATH, FSSP, and CE. Which, if any, structures do *all* these classifications consider structural neighbours of 2TRX? Which structures are considered structural neighbours in some but not all classifications?

3.9 Why did an ENTREZ search in the protein category for HUMAN ELASTASE return a tRNA synthetase?

3.10 The Box on page 154 contains the amino acid sequence of human elastase 1 precursor. What sequence differences are there between this and the mature protein?

3.11 What is the relation between the elastase sequences recovered from searching the NCBI and the PIR?

3.12 Using SWISS-PROT directly, or SRS, recover the SWISS-PROT entry for human elastase. What information does this file contain that does not appear in (a) the corresponding entry in ENTREZ (protein) and (b) the corresponding entry in PIR?

3.13 What homologues of human neutrophil elastase can be identified by PSI-BLAST?

3.14 Search for structures of elastases using the Protein Data Bank search facilities. Compare the results with those from ENTREZ, described in the text.

3.15 Which gene in *C. elegans* encodes a protein similar in sequence to human elastase?

3.16 What is the chromosomal location of the human gene for glucose-6-phosphate dehydrogenase?

3.17 Pseudogenes in eukaryotes can be classified into those that arose by gene duplication and divergence, and those reinserted into the genome from mRNA by a retrovirus, called *processed* pseudogenes. Processed pseudogenes can be identified by the absence of introns. Which if any of the pseudogenes in the human globin gene clusters are processed pseudogenes?

3.18 Preliminary genetic analysis on the way to isolating the gene associated with cystic fibrosis bracketed it between the MET oncogene and RFLP D7S8. It was then estimated that this region contained 1–2 million bp, and might contain 100–200 genes. (a) How many base pairs long did this region actually turn out to be? (b) How many expressed genes is this region now believed to contain?

3.19 The gene for Berardinelli-Seip syndrome was initially localized between two markers on chromosome band 11q13—D11S4191 and D11S987. How many base pairs are there in the interval between these two markers?

3.20 Is there a database available on the Web that specifically collects structural and thermodynamic information on protein-nucleic acid interactions?

3.21 The Yeast proteome database contains an entry for cdc6, the protein that regulates initiation of DNA replication. (a) On what chromosome is the gene for

yeast cdc6? (b) What post-translational modification does this protein undergo to reach its mature active state? (c) What are the closest known relatives of this protein in other species? (d) With what other proteins is yeast cdc6 known to interact? (e) What is the effect of distamycin A on the activity of yeast cdc6? (f) What is the effect of actinomycin D on the the activity of yeast cdc6?

CHAPTER 4

Alignments and phylogenetic trees

Chapter contents

Learning goals

1. To understand the concept of sequence alignment: the assignment of residue-residue correspondences.
2. To know how to construct and interpret dotplots, and the relationship between dotplots and alignments.
3. To be able to define and distinguish the Hamming distance and Levenshtein distance as measures of dissimilarity of character strings.
4. To understand the basis of scoring schemes for string alignment, including substitution matrices and gap penalties.
5. To appreciate the difference between global alignments and local alignments, and to understand the use of approximate methods for quick screening of databases.
6. To understand the significance of *Z*-scores, and to know how to interpret *P*-values and *E*-values returned by database searches.
7. To be able to interpret multiple alignments of amino acid sequences, and to make inferences from multiple sequence alignments about protein structures.
8. To be able to define and distinguish the concepts of homology, similarity, clustering, and phylogeny.
9. To become expert in the use of PSI-BLAST and related programs.
10. To appreciate the use of profile methods and Hidden Markov Models in database searching.
11. To understand the contents and significance of phylogenetic trees, and the methods available for deriving them, including maximum parsimony and maximum likelihood; to know the role and use of an outgroup in derivation of a phylogenetic tree.

Introduction to sequence alignment

Given two or more sequences, we initially wish to:

- measure their similarity
- determine the residue-residue correspondences
- observe patterns of conservation and variability
- infer evolutionary relationships

If we can do this, we will be in a good position to go fishing in databanks for related sequences. A major application is to the annotation of genomes, involving assignment of structure and function to as many genes as possible.

How can we define a quantitative measure of sequence similarity? To compare the nucleotides or amino acids that appear at corresponding positions in two or more sequences, we must first assign those correspondences. *Sequence alignment is the identification of residue-residue correspondences.* It is *the* basic tool of bioinformatics.

Any assignment of correspondences that preserves the order of the residues within the sequences is an alignment. Gaps may be introduced.

Given two text strings: first string = a b c d e

second string = a c d e f

a reasonable alignment would be:

```
a b c d e -
a - c d e f
```

We must define criteria so that an algorithm can choose the *best* alignment. For the sequences `gctgaacg` and `ctataatc`:

An uninformative alignment:

```
- - - - - - - g c t g a a c g
c t a t a a t c - - - - - - - -
```

An alignment without gaps:

```
g c t g a a c g
c t a t a a t c
```

An alignment with gaps:

```
g c t g a - a - - c g
- - c t - a t a a t c
```

And another:

```
g c t g - a a - c g
- c t a t a a t c -
```

Most readers would consider the last of these alignments the best of the four. To decide whether it is the best of *all* possibilities, we need a way to examine all possible alignments systematically. Then we need to compute a score reflecting the quality of each possible alignment, and to identify the alignment with the optimal score. The optimal alignment may not be unique: several different alignments may give the same best score. Moreover, even minor variations in the scoring scheme may change the ranking of alignments, causing a different one to emerge as the best.

These examples illustrate **pairwise sequence alignments.** However, usually we can find large families of similar sequences, by identifying homologues in different species. A mutual alignment of more than two sequences is called a **multiple sequence alignment.** Multiple sequence alignments are much more informative than pairwise sequence alignments, in terms of revealing patterns of conservation.

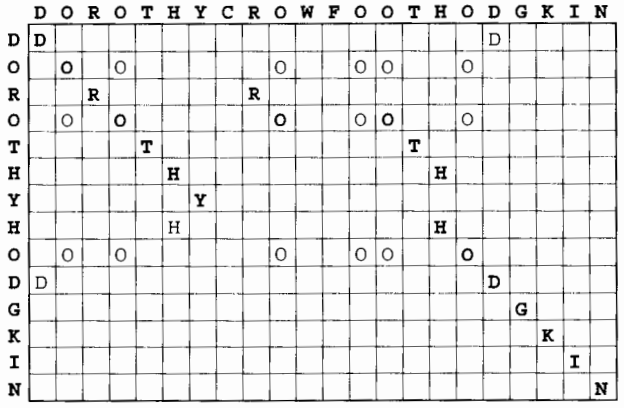
The dotplot

The dotplot is a simple picture that gives an overview of pairwise sequence similarity. Less obvious is its close relationship to alignments.

The dotplot is a table or matrix. The rows correspond to the residues of one sequence and the columns to the residues of the other sequence. In its simplest form, the positions in the dotplot are left blank if the residues are different, and filled if they match. Stretches of similar residues show up as diagonals in the upper left–lower right (Northwest–Southeast) direction.

Example 4.1

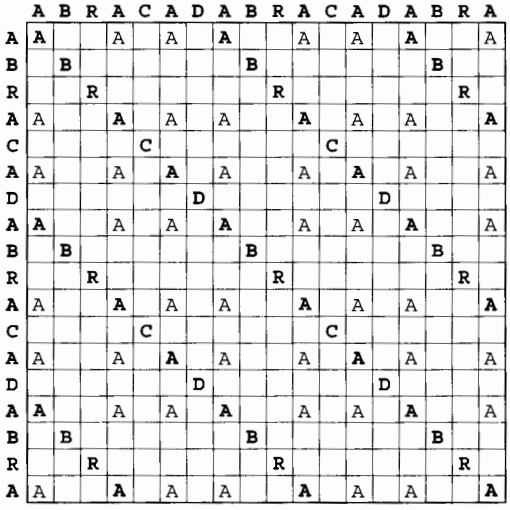
Dotplot showing identities between short name (DOROTHYHODGKIN) and full name (DOROTHYCROWFOOTHODGKIN) of a famous protein crystallographer.



Letters corresponding to *isolated* matches are shown in non-bold type. The longest matching regions, shown in red, are the first and last names DOROTHY and HODGKIN. Shorter matching regions, such as the OTH of dorOTHy and crowfoOTHodgkin, or the RO of doROthy and cROwfoot, are noise.

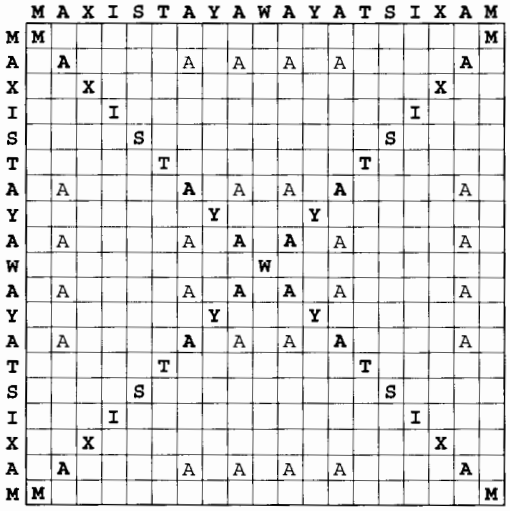
Example 4.2

Dotplot showing identities between a repetitive sequence (ABRACADABRA-CADABRA) and itself. The repeats appear on several subsidiary diagonals parallel to the main diagonal.



Example 4.3

Dotplot showing identities between the palindromic sequence MAX I STAY AWAY AT SIX AM and itself. The palindrome reveals itself as a stretch of matches *perpendicular* to the main diagonal.



This is not just word play—regions in DNA recognized by transcription regulators or restriction enzymes have sequences related to palindromes, crossing from one strand to the other:

→

→

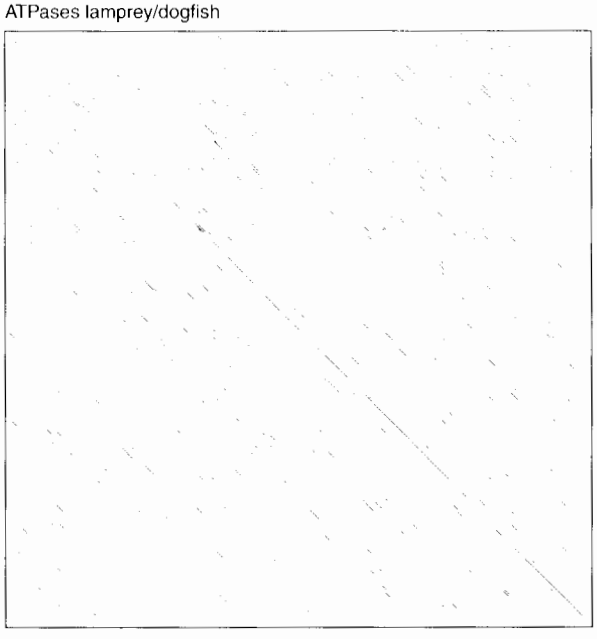
Example 4.3 (*continued*)

EcoRI recognition site: GAATTC
CTTAAG

Within each strand a region is followed by its reverse complement (see Exercise 4.9 and Problem 4.8). Longer regions of DNA or RNA containing inverted repeats of this form can form stem-loop structures. In addition, some transposable elements in plants contain true (approximate) palindromic sequences—inverted repeats of noncomplemented sequences, on the same strand. The following example appears in the Wheat dwarf virus genome: `ttttcgtgagtgcgcggaggctttt`.

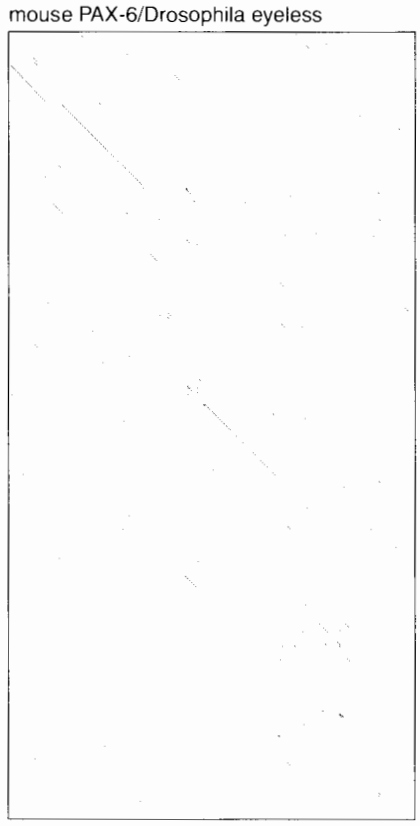
The dotplot gives a quick pictorial statement of the relationship between two sequences. Obvious features of similarity stand out. For example, a dotplot relating the mitochondrial ATPase-6 genes from a lamprey (*Petromyzon marinus*) and dogfish shark (*Scyliorhinus canicula*) shows that the similarity of the sequences is weakest near the beginning. This gene codes for a subunit of the ATPase complex. In the human, mutations in this gene cause Leigh syndrome, a neurological disorder of infants produced by the effects of impaired oxidative metabolism on the brain during development.

A disadvantage of the dotplot is that its 'reach' into the realm of distantly-related sequences is poor. In analysing sequences, one should always look at a dotplot to be sure of not missing anything obvious, but be prepared to apply more subtle tools.



Often regions of similarity are displaced, to appear on parallel but not collinear diagonals. This indicates that insertions or deletions have occurred in the segments between the similar regions. A dotplot relating the PAX-6 protein of mouse and

the eyeless protein of *Drosophila melanogaster* shows three extended regions of similarity with different lengths of sequence between them. Two of the regions are near the beginning of the sequences and one is near the middle. The section between the second and third regions of similarity is longer in the mouse sequence than in the *Drosophila* sequence.



Filtering the results can reduce the noise in a dotplot. In the comparison of the ATPase sequences, dots were not shown unless they were at the centre of a consecutive region or 15 residues containing at least 6 matches. The PERL program for dotplots (see Box) allows the user to set values for a **window** (length of region of consecutive residues) and a **threshold** (number of matches required within the window).

Web resources: Dotplots

E. L. Sondhammer's program Dotter computes and displays dotplots. It allows the user to control the calculation and alter the appearance of the display by adjusting parameters interactively.
http://www.cgb.ki.se/cgb/groups/sonnhammer/Dotter.html
To use the full set of features of Dotter it is necessary to install it locally.

A web site that offers interactive dotplotting is:
http://www.isrec.isb-sib.ch/java/dotlet/Dotlet.html

PERL Example 4.1 A program to draw dotplots

The program shown reads:

1. A general title for the job, printed at the top of the output drawing (first line of input).
2. Parameters specifying the filtering parameters *window* and *threshold* (second line of input). A dot will appear in the dotplot if it is in the centre of a stretch of residues of length *window* such that the number of matches is $\geq$ *threshold*.
3. The two sequences, each beginning with a title line and ending with a *.

The program draws a dotplot similar to those shown in the text. The output is in a graphical language called PostScript™, which can be displayed or printed on many devices.

```
#!/usr/bin/perl
#dotplot.pl -- reads two sequences and prints dotplot

# read input

$/ = "";
$_ = <DATA>; $_ =~ s/#(.*)\n/\n/g;
$_ =~ /^(.*)\n\s*(\d+)\s+(\d+)\s*\n(.*)\n([A-Za-z\n]*)\*\s*\n(.*)\n([A-Za-z\n]*)\*/
$title = $1; $nwind = $2; $thresh = $3;
$seqt1 = $4; $seq1 = $5; $seqt2 = $6; $seq2 = $7;
$seq1 =~ s/\n//g; $seq2 =~ s/\n//g; $n = length($seq1); $m = length($seq2);

# postscript header

print <<EOF;
%!PS-Adobe-
/s /stroke load def /l /lineto load def /m /moveto load def /r /rlineto load def
/n /newpath load def /c /closepath load def /f /fill load def
1.75 setlinewidth 30 30 translate /Helvetica findfont 20 scalefont setfont
EOF

#print matrix

$dx = 500.0/$n;  $mdx = -$dx; $dy = 500.0/$m;
if ($dy < $dx) {$dx = $dy;} $dy = $dx; $xmx = $n*$dx; $ymx = $m*$dx;
print "0 510 m ($title NWIND = $nwind) show\n";
printf "0 0 m 0 %9.2f l %9.2f %9.2f l %9.2f 0 l c s\n", $ymx,$xmx,$ymx,$xmx;

for ($k = $nwind - $m + 1; $k < $n - $nwind; $k++) {
    $i = $k; $j = 1; if ($k < 1) {$i = 1; $j = 2 - $k;}
    while ($i <= $n - $nwind && $j <= $m - $nwind) {
        $_ = (substr($seq1,$i -1,$nwind) ^ substr($seq2,$j -1,$nwind));
        $mismatch = ($_ =~ s/[^\x0]//g);
        if ($mismatch < $thresh) {
            $xl = ($i - 1)*$dx; $yb = ($m - $j)*$dy;
            printf "n %9.2f %9.2f m %9.2f 0 r 0 %9.2f r %9.2f 0 r c f\n",
                    $xl,$yb,$dx,$dy,$mdx;
        }
        $i++; $j++;
    }
}
print "showpage\n";

__END__
ATPases lamprey / dogfish                    #TITLE
15 6                                         #WINDOW, THRESHOLD
Petromyzon marinus mitochondrion             #SEQUENCE 1
atgacactagatatctttgaccaatttacctccccaaca
atatttgggcttccactagcctgattagctatactagcccctagctta
```

```
atattagtttcacaaacaccaaaatttatcaaatctcgttatcacacacta
cttacacccatcttaacatctattgccaaacaactctttcttccaataaac
caacaagggcataaatgagccttaatttgtatagcctctataatatttatc
ttaataattaatcttttaggattattaccatatacttatacaccaactacc
caattatcaataaacataggattagcagtgccactatgactagctactgtc
ctcattgggttacaaaaaaaaaccaacagaagccctagcccacttattacca
gaaggtaccccagcagcactcattcccatattaattatcattgaaactatt
agtctttttatccgacctatcgccctaggagtccgactaaccgctaattta
acagctggtcacttactttatacaactagtttctataacaacctttgtaata
attcctgtcatttcaatttcaattattacctcactacttcttctatta
ctaacaattctggagttagctgttgctgtaatccaggcatatgtatttatt
ctacttttaactctttatctgcaagaaaacgttt*
Scyliorhinus canicula mitochondrion          #SEQUENCE 2
atgattataagctttttgatcaattcctaagtccctcctttctagga
atcccactaattgccctagctatttcaattccatgattaatatttccaacaccaacc
aatcgttgacttaataatcgattattaactcttcaagcatgatttattaaccgatttatt
tatcaactaatacaacccataaatttaggaggacataaatgagctatcttatttacagcc
ctaatattatttttaattaccatcaatcttctaggtctccttccatatacttttacgcct
acaactcaactttctcttaatatagcctttgccctgcccttatggcttacaactgtatta
attggtatatttaatcaaccaaccattgccctagggcacttattacctgaaggtacccca
acccctttagtaccagtactaatcattatcgaaaccatcagtttatttattcgaccatta
gccttaggagtccgattaacagccaacttaacagctggacatctccttatacaattaatc
gcaactgcggcctttgtccttttaactataataccaaccgtggccttactaacctcccta
gtcctgttcctattgactattttagaagtggctgtagctataattcaagcatacgtattt
gtccttctttaagcttatatctacaagaaaacgtataa*
```

Dotplots and sequence alignments

The dotplot captures in a single picture not only the overall similarity of two sequences, but also the complete set and relative quality of different possible alignments. Any path through the dotplot from upper left to lower right, moving at each point only East, South or Southeast, corresponds to a possible alignment. If two sequences are closely related, the alignment can be read directly off the dotplot.

Figure 4.1 shows an example based on the Dorothy Hodgkin dotplot. If the direction of the 'move' between successive cells is diagonal, two pairs of

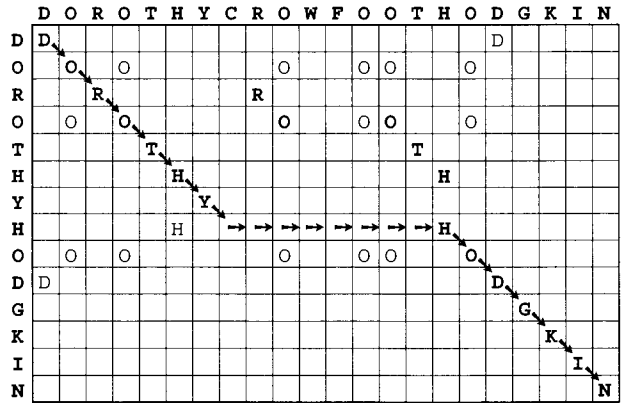


Fig. 4.1 Any path through the dotplot from upper left to lower right passes through a succession of cells, *each of which picks out a pair of positions, one from the row and one from the column, that correspond in the alignment; or that indicates a gap in one of the sequences.* The path need not pass through filled-in points only. However, the more filled-in points on the path, the more matching residues in the alignment.

successive residues appear in the alignment without an insertion between them. If the direction of the move is horizontal, a gap is introduced in the sequence indexing the rows. If the direction of the move is vertical (none in this example), a gap would be introduced in the sequence indexing the columns. Note that no moves can be directed up or to the left, as this would correspond to aligning several residues of one sequence with only one residue of the other. The path indicated by the arrows corresponds to the obvious alignment:

```
DOROTHY--------HODGKIN
DOROTHYCROWFOOTHODGKIN
```

Another way to think of a path through the dotplot is as an *edit script*; that is, the prescription of a series of operations that transforms the sequence that indexes the columns—the 'horizontal' sequence—into the sequence that indexes the rows—the 'vertical' sequence. Each move tells us to perform an operation—a substitution, an insertion, or a deletion. When the end of the path is reached, the effect will be to change one sequence into the other. In general, several different sequences of edit operations may convert one string into the other in the same number of steps, but they may induce different alignments.

It should be emphasized that although a sequence of edit operations derived from an optimal alignment *may* correspond to an actual evolutionary pathway, it is impossible to *prove* that it does. The larger the edit distance, the larger the number of reasonable evolutionary pathways between two sequences.

Example 4.4 Dotplots and alignments

Let us compare the appearance of dotplots between pairs of proteins with increasingly more distant relationships. Figure 4.2 shows the dotplot comparisons of the sulphydryl proteinase papain from papaya, with four homologues—the close relative, kiwi fruit actinidin, and more distant relatives, human procathepsin L, human cathepsin B, and *Staphyloccus aureus* Staphopain. The sequence alignments are also shown. As the sequences progressively diverge, it becomes more and more difficult to spot the correct alignment in the dotplot. The alignments shown were derived from comparisons of the structures.

→

```
ALIGNMENT OF 9pap      and 2act
SCORE = 5324  NPOS = 219 NIDENT = 102  %IDENT = 46.58

IPEYVDWRQKGAVTPVKNQGSCGSCWAFSAVVTIEGIIKIRTGNLNQYSEQELLDCDR--
 | |||||  |||   | || || ||||||  | ||| ||  | |   ||||| || |
LPSYVDWRSAGAVVDIKSQGECGGCWAFSAIATVEGINKITSGSLISLSEQELIDCGRTQ

RSYGCNGGYPWSALQ-LVAQYGIHYRNTYPYEGVQRYCRSREKGPYAAKTDGVRQVQPYN
   || |||     |       ||     |||       |              |    |   |
NTRGCDGGYITDGFQFIINDGGINTEENYPYTAQDGDCDVALQDQKYVTIDTYENVPYNN

QGALLYSIANQPVSVVLQAAGKDFQLYRGGIFVGPCGNKVDHAVAAVGYGP----NYILI
  ||      ||||| | |||  |  |  ||| ||||  ||||   ||||      |
EWALQTAVTYQPVSVALDAAGDAFKQYASGIFTGPCGTAVDHAIVIVGYGTEGGVDYWIV

KNSWGTGWGENGYIRIKRGTGNSYGVCGLYTSSFYPVKN
|||| | ||| || || |  |   | ||  |   ||||
KNSWDTTWGEEGYMRILRNVGGA-GTCGIATMPSYPVKY
```

PAPA_CARPA/ACTN_ACTCH

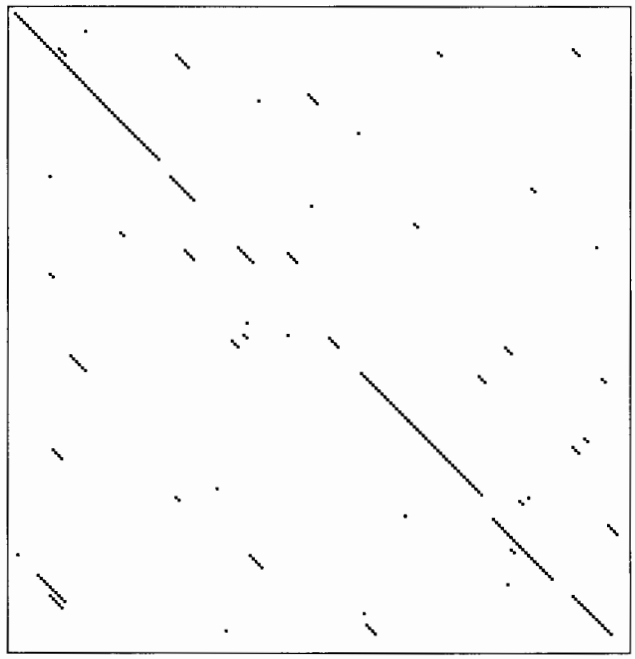


Fig. 4.2a Alignment of papaya papain and kiwi fruit actinidin, with the corresponding dotplot.

Example 4.4 (*continued*)

```
ALIGNMENT OF 9pap     and 1cjl

SCORE = 3214 NPOS = 220 NIDENT  = 81  %IDENT = 36.82

IPEYVDWRQKGAVTPVKNQGSCGSCWAFSAVVTIEGIIKIRTGNLNQYSEQELLDCD--R
     ||| || |||||||| ||| |||||    ||      || |   ||| | ||
V----DWREKGYVTPVKNQGQCGSSWAFSATGALEGQMFRKTGRLISLSEQNLVDCSGPE

RSYGCNGGYPWSALQLVAQY-GIHYRNTYPYEGVQRYCRSREKGPYAAKTDGVRQVQPYN
   |||||    | | |    |       ||||      |    |    |   |
GNEGCNGGLMDYAFQYVQDNGGLDSEESYPYEATEESCKYNPKYS-VANDAGFVDIPKQE

QGALLYSIANQPVSVVLQAAGKDFQLYRGGIFVGP--CGNKVDHAVAAVGYG---PNYIL
          | ||   |    |  |  ||   |        || |  ||||     | |
KALMKAVATVGPISVAIDAGHESFLFYKEGIYFEPDCSSEDMDHGVLVVGYGFESNKYWL

IKNSWGTGWGENGYIRIKRGTGNSYGVCGLYTSSFYPVKN
 |||||  ||  ||        |    ||       ||
VKNSWGEEWGMGGYVKMAKDRRN-H--CGIASAASYPTV-
```

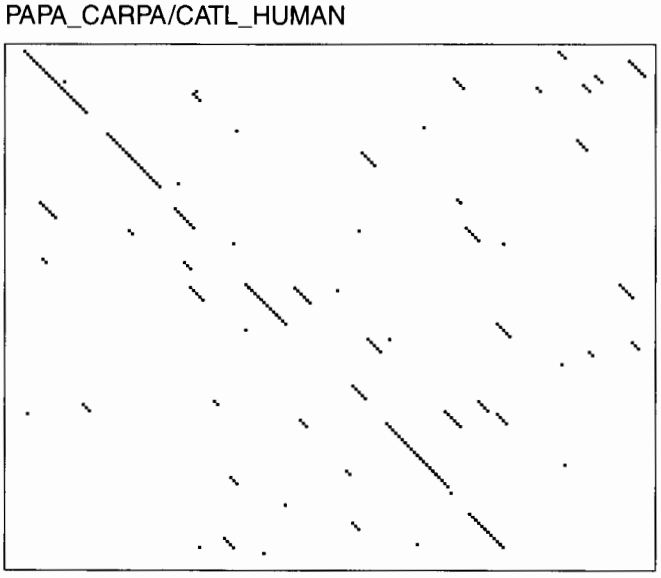


Fig. 4.2b Alignment of papaya papain and human procathepsin L, with the corresponding dotplot. This dotplot shows that there are several similar regions, but it would be difficult to generate a complete sequence alignment from the dotplot.

```
ALIGNMENT OF 9pap      and 1huc

SCORE   =  2073  NPOS = 251  NIDENT  = 66  %IDENT = 26.29

IPEYVD-WRQKGAVTPVKNQGSCGSCWAFSAVVTIEGIIKIRTGNLNQYSEQELLD-C-D
        | |          |||||||||| ||  |   | | |       |   ||  |
--DAREQWPQCPTIKEIRDQGSCGSCWAFGAVEAISDRICIHTNVSVEVSAEDLLTCCGS

RRSYGCNGGYP------WSALQLVAQYGI--HYRN-TY-----P--YEGVQRYCRSREKG
    |||||||       |    ||        |      |          |    |
MCGDGCNGGYPAEAWNFWTRKGLVSGGLYESHVGCRPYSIPPCEHHVNGSRPPCTGEGDT

PYAAK------TDGVRQVQPYNQGALLYSIANQPVSV-V-----LQ---AAGKDFQLYRG
|   |            |   |        |                          || ||
PKCSKICEPGYSPTYKQDKHYGYNSYSVSNSEKDIMAEIYKNGPVEGAFSVYSDFLLYKS

GIFVGPCGNKV-DHAVAAV--GY--GPNYILIKNSWGTGWGENGYIRIKRGTGNSYGVCG
|       |     ||       |   |  | |  ||| | || ||   | ||     |
GVYQHVTGEMMGGHAIRILGWGVENGTPYWLVANSWNTDWGDNGFFKILRGQ-DHCGIES

LYTSSFYPVKN
        |
EVVAGI-PRTD
```

PAPA_CARPA/CATB_HUMAN

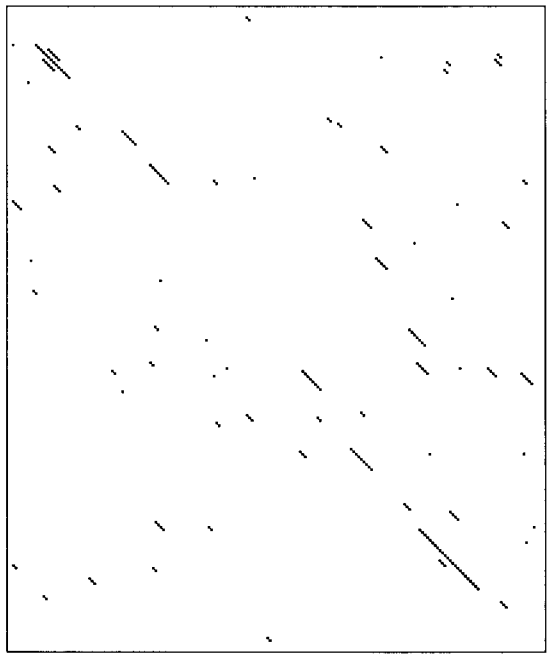


Fig. 4.2c Alignment of papaya papain and human liver cathepsin B, with the corresponding dotplot. Note, in *both* the sequence alignment and the dotplot, the higher similarity at the beginning and end of the sequences than in the middle region.

Example 4.4 (*continued*)

```
ALIGNMENT OF 9pap      and 1cv8

SCORE = -290  NPOS = 219  NIDENT  = 25 %IDENT = 11.42

IPEYVDWRQKGAVTPVKNQGSCGSCWAFSAVVTIEGIIKIRTGNLNQYSEQELLDCDRRS
                                                |       |
-------------------------------------------------EQYVNKLENFKIRE

YGCNGGYPWSALQLVAQYGIHYRNTYPYEGVQRYCRSREKG-PYAAKTDGVRQVQPY---
   | |                    | |   | | |        |           |
TQGNNGWCAGYTMSALLNATYNTNKYHAEAVMRFLHPNLQGQQFQFTGLTPREMIYFGQT

--NQGALLYSIANQPVSVVLQAAGKDFQLYRGGIFVGPCGNKVDHAVAAVGYGPNYILIK
      ||            |     |       |       |     || | ||
QGRSPQLLNRMTTYNEVDNLTKNNKGIAIL-GSRVESRNGMHAGHAMAVVGNAKLNNGQE

NSWGTGWGENGYIRIKRGTGNSYGVCGLYTSSFYPVKN-
         ||              |
VIIIWNPWDNGFMTQDAKNNVIPVSNGDHYQWYSSIYGY
```

PAPA_CARPA/STPA_STAAU

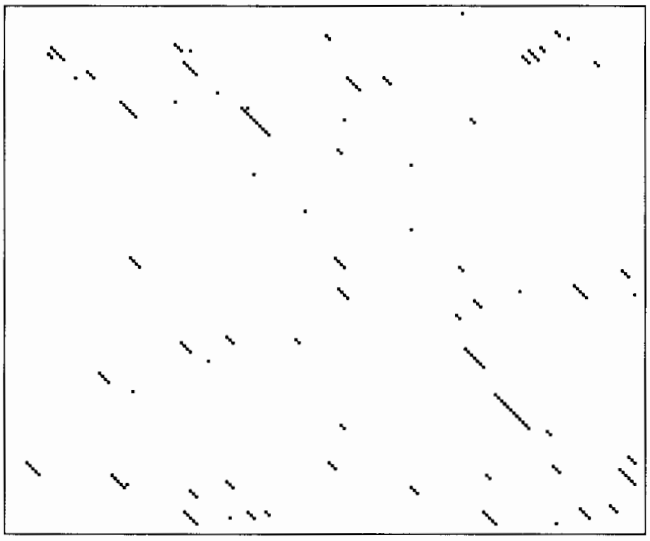


Fig. 4.2d Alignment of papaya papain and *S. aureus* staphopain, with the corresponding dotplot. The alignment of these two sequences is not derivable from this dotplot.

Measures of sequence similarity

To go beyond 'alignment by eyeball' via dotplots, we must define quantitative measures of sequence similarity and difference.

Given two character strings, two measures of the distance between them are:

(1) The **Hamming distance**, defined between two strings of equal length, is the number of positions with mismatching characters.

(2) The **Levenshtein**, or **edit distance**, between two strings of not necessarily equal length, is the minimal number of 'edit operations' required to change one string into the other, where an edit operation is a deletion, insertion or alteration of a single character in either sequence. A given sequence of edit operations induces a unique alignment, but not vice versa.

For example:

```
agtc        Hamming distance = 2
cgta

ag-tcc      Levenshtein distance = 3
cgctca
```

For applications to molecular biology, recognize that certain changes are more likely to occur naturally than others. For example, amino acid substitutions tend to be conservative: the replacement of one amino acid by another with similar size or physicochemical properties is more likely to have occurred than its replacement by another amino acid with dissimilar properties. Or, the deletion of a succession of contiguous bases or amino acids is a more probable event than the independent deletion of the same number of bases or amino acids at noncontiguous positions in the sequences. Therefore we wish to assign variable weights to different edit operations. A computer program can then determine not just minimal edit distances but optimal alignments. It can score each path through the dotplot, by adding up the scores of the individual steps. For each substitution, it adds the score of the mutation, depending on the pair of residues involved. For horizontal and vertical moves, it adds a suitable gap penalty.

Scoring schemes

A scoring system must account for residue substitutions, and insertions or deletions. (An insertion, from one sequence's point of view, is a deletion as seen by the other!) Deletions, or gaps in a sequence, will have scores that depend on their lengths.

Hamming and Levenshtein distances measure the *dissimilarity* of two sequences: similar sequences give small distances and dissimilar sequences give large distances. It is common in molecular biology to define scores as measures of sequence *similarity*. Then similar sequences give high scores and dissimilar sequences give low scores. These are equivalent formulations. Algorithms for optimal alignment can seek either to minimize a dissimilarity measure, or to maximize a scoring function.

Example 4.5

Transition mutations (*purine*↔*purine* and *pyrimidine* ↔ *pyrimidine;* that is, a↔g and t↔c) are more common than *transversions* (*purine* ↔ *pyrimidine;* that is, (a or g) ↔ (t or c)). Suggest a substitution matrix that reflects this.

One possibility is:

	a	t	g	c
a	20	10	5	5
t	10	20	5	5
g	5	5	20	10
c	5	5	10	20

For nucleic acid sequences, it is common to use a simple scheme for substitutions: +1 for a match, −1 for a mismatch, or a more complicated scheme based on the higher frequency of transition mutations than transversion mutations.

For proteins, a variety of scoring schemes have been proposed. We might group the amino acids into classes of similar physicochemical type, and score +1 for a match within residue class, and −1 for residues in different classes. We might try to devise a more precise substitution score from a combination of properties of the amino acids. Alternatively, we might try to let the proteins teach us an appropriate scoring scheme. M. O. Dayhoff did this first, by collecting statistics on substitution frequencies in the protein sequences then known. Her results were used for many years to score alignments. They have been superseded by newer matrices based on the very much larger set of sequences that has subsequently become available.

Derivation of substitution matrices

As sequences diverge, mutations accumulate. To measure the relative probability of any particular substitution, for instance Serine → Threonine, we can count the number of Serine → Threonine changes in pairs of aligned homologous sequences. We could use the relative frequencies of such changes to form a scoring matrix for substitutions. A likely change will score higher than a rare one. But, what if there have been multiple substitutions at certain sites? This will bias the statistics. We can avoid this problem by restricting our samples to sequences that are sufficiently similar that we can assume that no position has changed more than once.

A measure of sequence divergence is the **PAM**: 1 PAM = 1 Percent Accepted Mutation. Thus, two sequences 1 PAM apart have 99% identical residues. For pairs of sequences within the 1 PAM level of divergence, it is likely that there has been no more than one change at any position. Collecting statistics from pairs of sequences as closely related as this, and correcting for different amino acid abundances, produces the **1 PAM substitution matrix**. To produce a matrix appropriate to more widely divergent sequences, we can take powers of this matrix. The PAM250 level, corresponding to ~20% overall sequence identity, is the lowest sequence similarity for which we can generally hope to produce a correct

alignment by sequence analysis alone. It is therefore the appropriate level to choose for practical work (see Box, Substitution matrices for scoring amino acid sequence similarity). (Several authors have derived substitution matrices appropriate in different ranges of overall sequence similarity.)

The occurrence of reversions, either directly or via one or more other changes, produces an apparent slowdown in mutation rates as sequences progressively diverge. The relation between PAM score and % sequence identity is:

PAM	0	30	80	110	200	250
% identity	100	75	50	60	25	20

The PAM250 matrix of M. O. Dayhoff is shown in the Box. It expresses scores as **log-odds** values:

Score of mutation $i \leftrightarrow j$

$$= log_{10} \frac{\text{observed i} \leftrightarrow \text{j mutation rate}}{\text{mutation rate expected from amino acid frequencies}}$$

The numbers are multiplied by 10, simply to avoid decimal points. The matrix entries reflect the probabilities of mutational events. A value of +2—for instance, C ↔ S—implies that in related sequences the mutation would be expected to occur 1.6 times more frequently than random. The calculation is as follows: The matrix entry 2 corresponds to the actual value 0.2 because of the scaling. The value 0.2 is $\log_{10}$ of the relative expectation value of the mutation. Because $\log_{10}$ (1.6) = 0.2, the expectation value is 1.6.

The probability of two independent mutational events is the product of their probabilities. By using logs, we have scores that we can add up rather than multiply, a computational convenience.

The BLOSUM matrices

S. Henikoff and J. G. Henikoff developed the family of BLOSUM matrices for scoring substitutions in amino acid sequence comparisons. Their goal was to replace the Dayhoff matrix with one that would perform best in identifying distant relationships, making use of the much larger amount of data that had become available since Dayhoff's work.

The BLOSUM matrices are based on the BLOCKS database of aligned protein sequences; hence the name BLOcks SUbstitution Matrix. From regions of closely-related proteins alignable without gaps, Henikoff and Henikoff calculated the ratio of the number of observed pairs of amino acids at any position, to the number of pairs expected from the overall amino acid frequencies. Like the Dayhoff matrix, the results are expressed as log-odds. In order to avoid overweighting closely-related sequences, the Henikoffs replaced groups of proteins that have sequence identities higher than a threshold by either a single representative or a weighted average. The threshold 62% produces the commonly-used BLOSUM62 substitution matrix (see Box). This is offered by all programs as an option and as the default by most. BLOSUM matrices, and other recently-derived substitution parameters, have superseded the Dayhoff matrix in applications.

Substitution matrices used for scoring amino acid sequence similarity

Rows and columns are in alphabetical order of the THREE-letter amino acid names. Only the lower triangles of the matrices are shown, as the substitution probabilities are taken as symmetric. (Not because we are sure that the rate of any substitution is the same as the rate of its reverse, but because we cannot determine the differences between the two rates.)

The Dayhoff PAM250 matrix:

Ala (A)	2																		
Arg (R)	−2	6																	
Asn (N)	0	0	2																
Asp (D)	0	−1	2	4															
Cys (C)	−2	−4	−4	−5	12														
Gln (Q)	0	1	1	2	−5	4													
Glu (E)	0	−1	1	3	−5	2	4												
Gly (G)	1	−3	0	1	−3	−1	0	5											
His (H)	−1	2	2	1	−3	3	1	−2	6										
Ile (I)	−1	−2	−2	−2	−2	−2	−2	−3	−2	−5									
Leu (L)	−2	−3	−3	−4	−6	−2	−3	−4	−2	2	6								
Lys (K)	−1	3	1	0	−5	1	0	−2	0	−2	−3	5							
Met (M)	−1	0	−2	−3	−5	−1	−2	−3	−2	2	4	0	6						
Phe (F)	−3	−4	−3	−6	−4	−5	−5	−5	−2	1	2	−5	0	9					
Pro (P)	1	0	0	−1	−3	0	−1	0	0	−2	−3	−1	−2	−5	6				
Ser (S)	1	0	1	0	0	−1	0	1	−1	−1	−3	0	−2	−3	1	2			
Thr (T)	1	−2	0	0	−2	−1	0	0	−1	0	−2	0	−1	−3	0	1	3		
Trp (W)	−6	2	−4	−7	−8	−5	−7	−7	−3	−5	−2	−3	−4	0	−6	−2	−5	17	
Tyr (Y)	−3	−4	−2	−4	0	−4	−4	−5	0	−1	−1	−4	−2	7	−5	−3	−3	0	10	
Val (V)	0	−2	−2	−2	−2	−2	−2	−1	−2	4	2	−2	2	−1	−1	−1	0	−6	−2	4
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V

The BLOSUM62 matrix:

Ala (A)	4																		
Arg (R)	−1	5																	
Asn (N)	−2	0	6																
Asp (D)	−2	−2	1	6															
Cys (C)	0	−3	−3	−3	9														
Gln (Q)	−1	1	0	0	−3	5													
Glu (E)	−1	0	0	2	−4	2	5												
Gly (G)	0	−2	0	−1	−3	−2	−2	6											
His (H)	−2	0	1	−1	−3	0	0	−2	8										
Ile (I)	−1	−3	−3	−3	−1	−3	−3	−4	−3	4									
Leu (L)	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4								
Lys (K)	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5							
Met (M)	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5						
Phe (F)	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6					
Pro (P)	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7				
Ser (S)	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4			
Thr (T)	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5		
Trp (W)	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11	
Tyr (Y)	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
Val (V)	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V

Scoring insertions/deletions, or 'gap weighting'

To form a complete scoring scheme for alignments, we need, in addition to the substitution matrix, a way of scoring gaps. How important are insertions and deletions, relative to substitutions? Distinguish gap initiation:

```
aaagaaa
aaa-aaa
```

from gap extension:

```
aaaggggaaa
aaa----aaa
```

For aligning DNA sequences, the alignment program CLUSTAL-W recommends use of the identity matrix for substitution (+1 for a match, 0 for a mismatch) and gap penalties 10 for gap initiation and 0.1 for gap extension by one residue. For aligning protein sequences, the recommendations are to use the BLOSUM62 matrix for substitutions, and gap penalties 11 for gap initiation and 1 for gap extension by one residue.

Computing the alignment of two sequences

Now that we have a scoring scheme, we can apply it to finding optimal alignments—we seek the alignment that maximizes the score. A famous algorithm to determine the global optimal alignments of two sequences is based on a mathematical technique called dynamic programming. (Details are described at the end of this section.) This algorithm has been extremely important in molecular biology. Two of its noteworthy features are:

- The good news is that the method is guaranteed to give a *global* optimum. It will find the *best* alignment score, given the choice of parameters—substitution matrix (M) and gap penalty—with no approximation.
- The bad news is that many alignments may give the same optimal score. And none of these need correspond to the biologically correct alignment. For instance, in comparing the α- and β-chains of chicken haemoglobin, W. Fitch and T. Smith found 17 alignments all of which give the same optimal score, one of which is correct (on the basis of the structures, the court of last resort). There are 1317 alignments with scores within 5% of the optimum.

Another item of bad news is technical: The time required to align two sequences of lengths n and m is proportional to $n \times m$, because this is the size of the edit matrix that must be filled in. This means that the dynamic-programming method is not convenient to use for searching in an entire sequence database for a match to a probe sequence, and even less convenient for 'all-against-all' alignments. The database search problem is in effect the problem of matching a probe sequence to a region of a very long sequence, the length of the entire database.

Variations and generalizations

Variations of the dynamical-programming method apply to two related alignment questions (see also Box, page 26):

- ***global alignment:*** find the best alignment of one entire sequence with another entire sequence.
- ***local alignment:*** find the best alignment of some segment of one sequence against some segment of another sequence. (This includes probing a database with a single sequence, regarding a database as a single very long sequence.)

The global alignment algorithm was first applied to biological sequence alignment by S. B. Needleman and C. D. Wunsch. T. Smith and M. Waterman modified it to identify local matches.

Approximate methods for quick screening of databases

It is routine to screen genes from a new genome against the databases, for similarity to other sequences. Approximate methods can detect close relationships well and quickly but are inferior to the exact ones in picking up very distant relationships. In practice, they give satisfactory performance in the many cases in which the probe sequence is fairly similar to one or more sequences in the databank, and they are therefore worth trying first.

A typical approximation approach would take a small integer k, and determine all instances of each k-tuple of residues in the probe sequence that occur in any sequence in the database. A candidate sequence is a sequence in the databank containing a large number of matching k-tuples, with equivalent spacing in probe and candidate sequences. For a selected set of candidate sequences, approximate optimal alignment calculations are then carried out, with the time- and space-saving restriction that the paths through the matrix considered are restricted to bands around the diagonals containing the many matching k-tuples. There are several variations on this theme.

The dynamic programming algorithm for optimal pairwise sequence alignment*

A chart implicitly containing all possible alignments can be constructed as a matrix similar to that used in drawing the dotplot. The residues of one sequence index the rows, the residues of the other sequence index the columns. Any path through the matrix from upper left to lower right corresponds to an alignment (see Fig. 4.1). The task is to find the path that has the lowest cost, and the difficulty is that there are a very large number of paths to consider.

As an illustration, suppose you wanted to drive from Malmö in Southern Sweden to Tromsø in Northern Norway (see Fig. 4.3). Your route will consist of a number of segments, taking you through a succession of intermediate cities. There are many choices of different combinations of segments to produce a complete, continuous path.

The computational approach to finding the optimal path begins by assigning a numerical measure of the 'cost' to each of the possible individual segments of the journey. This 'cost' is not simply the financial outlay, but a more general estimate

* Optional section. Readers in doubt may consider the remarks in Lesk, A. M. (1988), TATA for now . . .. *Trends Biochem. Sci.*, **13**, 410.

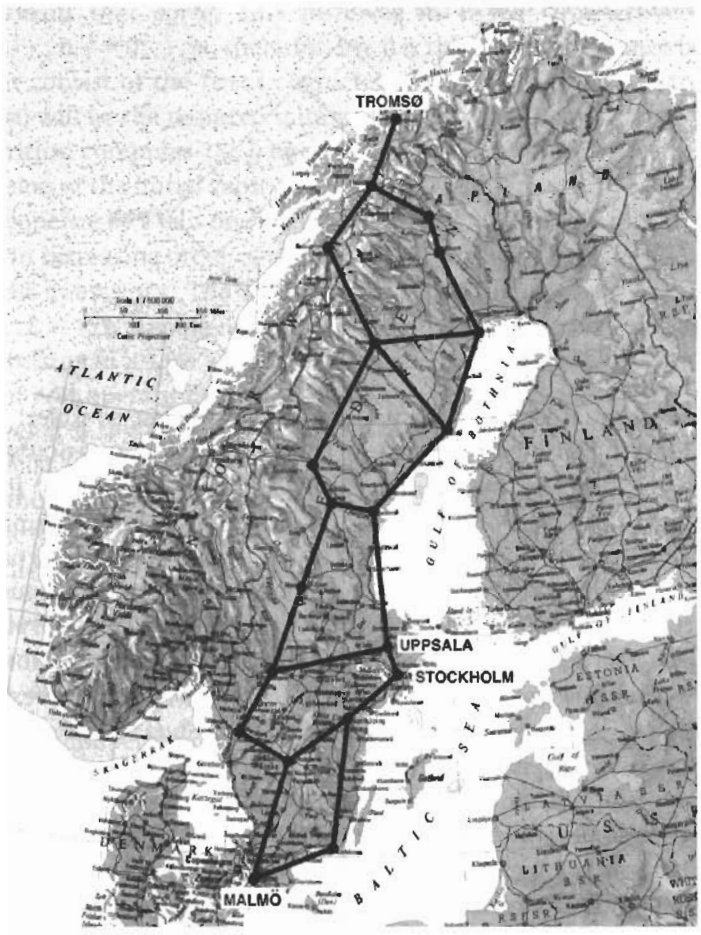
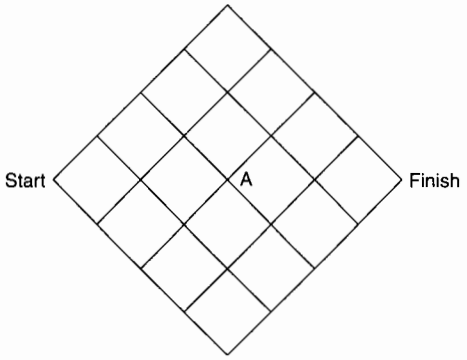


Fig. 4.3 Possible routes from Malmö to Tromsø. How can you determine an optimal route? (© Collins Bartholomew 1980. Reproduced by permission of HarperCollins Publishers.)

of your relative preferences for different portions of the route. The distance travelled will clearly be an important component of the cost, but other factors such as the quality of the roads and the opportunities for sightseeing also contribute. For any route selected, the overall cost of the trip is the sum of the costs of the individual segments. Clearly it is inefficient to repeat any leg of the journey, or to visit any city twice, so you will agree that every intermediate stop will be north of the previous one. This formalism is expressed in terms of minimizing a cost rather than maximizing a score; for our purposes the two approaches are equivalent. An algorithm can explore the possible combinations to determine an optimal overall route.

Here is an abstract version of the problem:



To grasp the essential idea of dynamic programming, first consider: How many paths from Start to Finish pass through A? There are 6 paths from Start to A. (Write them all down.) Therefore, by symmetry there are 6 paths from A to finish, and a total of 36 paths from Start to Finish passing through A. (Why?) Assuming that we have assigned costs to the individual steps, do we have to check all 36 paths to find the path of minimum cost that goes from Start to Finish, passing through A? No—here is the crucial observation: *The choice of the best path from A to Finish is independent of the choice of path from the Start to A.* If we determine the best of the 6 paths from Start to A, and we determine the best of the 6 paths from A to Finish, the best path from Start to Finish passing through A is: the best path from Start to A *followed by* the best path from A to finish. No more than 12 of the paths through A need be considered.

Even greater simplification is possible by systematically resubdividing the problem. The dynamic programming method for finding the optimal path through the matrix is based on this idea.

A statement of the optimal alignment problem and the dynamical programming solution is as follows: Given two character strings, possibly of unequal length: $A = a_1a_2 \ldots a_n$ and $B = b_1b_2 \ldots b_m$, where each a_i and b_j is a member of an alphabet set $\mathcal{A}$, consider sequences of edit operations that convert A and B to a common sequence. Individual edit operations include:

Substitution of b_j for a_i — represented (a_i, b_j).
Deletion of a_i from sequence A — represented (a_i, ϕ).
Deletion of b_j from sequence B — represented as (ϕ, b_j).

If we extend the alphabet set to include the null character ϕ: $\mathcal{A}^+ = \mathcal{A} \cup \{\phi\}$, a sequence of edit operations is a set of ordered pairs (x, y), with $x, y \in \mathcal{A}^+$.

A *cost function* d is defined on edit operations:

$d(a_i, b_j)$ = cost of a mutation in an alignment in which position i of sequence A corresponds to position j of sequence B, and the mutation substitutes $a_i \leftrightarrow b_j$.

$d(a_i, \phi)$ or $d(\phi, b_j)$ = cost of a deletion or insertion.

Define the minimum weighted distance between sequences A and B as

$$D(A, B) = \min_{A \to B} \Sigma d(x, y)$$

where $x, y \in A^+$ and the minimum is taken over all sequences of edit operations that convert A and B into a common sequence.

The problem is to find $D(A, B)$ and one or more of the alignments that correspond to it.

An algorithm that solves this problem in $\mathcal{O}(mn)$ time creates a matrix $\mathcal{D}(i, j)$, $i = 0, \ldots n; j = 0, \ldots m$, such that $\mathcal{D}(i, j)$ is the minimal distance between the strings that consist of the first i characters of A and the first j characters of B. Then $\mathcal{D}(n, m)$ will be the required minimal distance $D(A, B)$.

The algorithm computes $\mathcal{D}(i, j)$ by recursion. The value of $\mathcal{D}(i, j)$ corresponds to the conversion of the initial subsequences $A_i = a_1 a_2 \ldots a_i$ and $B_j = b_1 b_2 \ldots b_j$ into a common sequence by L edit operations S_k, $k = 1, \ldots L$, which can be considered to be applied in increasing order of position in the strings. Consider *undoing* the last of these edit operations. The resulting truncated sequence of edit operations, S_k, $k = 1, \ldots L - 1$, is a sequence of edit operations for converting a substring of A_i and a substring of B_j into a common result. What is more, it must be an *optimal* sequence of edit operations for these substrings, for if some other sequence S'_k were a lower-cost sequence of operations for these substrings, then S'_k followed by S_L would be a lower-cost sequence of operations than S_k for converting A_i to B_j. Therefore there should be a recursive method for calculating the $\mathcal{D}(i, j)$.

Recognize the correspondence between individual edit operations and steps between adjacent squares in the matrix (see Fig. 4.1):

$(i-1, j-1) \rightarrow (i, j)$	corresponds to the substitution $a_i \rightarrow b_j$.
$(i-1, j) \rightarrow (i, j)$	corresponds to the deletion of a_i from A.
$(i, j-1) \rightarrow (i, j)$	corresponds to the insertion of b_j into A at position i.

Sequences of edit operations correspond to stepwise paths through the matrix

$$(i_0, j_0) = (0, 0) \rightarrow (i_1, j_1) \rightarrow \cdots (n, m)$$

where $0 \leq i_{k+1} - i_k \leq 1$, (for $0 \leq k \leq n - 1$), $0 \leq j_{k+1} - j_k \leq 1$ (for $0 \leq k \leq m - 1$). Considering the possible sequences of edit operations and the corresponding paths through the matrix, the predecessor of an optimal string of edit operations leading from (0,0) to (i, j), where $i, j > 0$, must be an optimal sequence of edit operations leading to one of the cells $(i - 1, j)$, $(i - 1, j - 1)$, or $(i, j - 1)$; and, correspondingly, $\mathcal{D}(i, j)$ must depend only on the values of $\mathcal{D}(i - 1, j)$, $\mathcal{D}(i - 1, j - 1)$, and $\mathcal{D}(i, j - 1)$, (together of course with the parameterization specified by the cost function d).

The algorithm is then as follows:

Compute the $(m + 1) \times (n + 1)$ matrix $\mathcal{D}$ by applying:

(1) the initialization conditions on the top row and left column:

$$D(i, 0) = \sum_{k=0}^{i} d(a_k, \phi)$$

$$D(0, j) = \sum_{k=0}^{j} d(\phi, b_k)$$

These values impose the gap penalty on unmatched residues at the beginning of either sequence.

And then

(2) the recurrence relations:

$$D(i, j) = min\{D(i-1, j) + d(a_i, \phi, D(i-1, j-1) + d(a_i, b_j), D(i, j-1) + d(\phi, b_j)\}$$

for $i = 1, \ldots n$; $j = 1, \ldots m$. This means: consider all three possible steps to $\mathcal{D}(i, j)$:

Operation	Cumulative cost
insert a gap in sequence A	$\mathcal{D}(i-1,j) + d(a_i, \phi)$
substitute $a_i \leftrightarrow b_j$	$\mathcal{D}(i-1,j-1) + d(a_i, b_j)$
insert a gap in sequence B	$\mathcal{D}(i,j-1) + d(\phi, b_j)$

From these, choose the minimal value. For each cell record not only the value $\mathcal{D}(i, j)$ but a pointer back to (one or more of) the cell(s) $(i-1, j)$, $(i-1, j-1)$ or $(i, j-1)$ selected by the minimization operation. Note that more than one predecessor may give the same value.

When the calculations are complete, $\mathcal{D}(n, m)$ is the optimal distance $D(A, B)$. An alignment corresponding to the sequence of edit operations recorded by the pointers can be recovered by tracing a path back through the matrix from (n, m) to $(0, 0)$. This alignment corresponding to the minimal distance $D(A, B) = \mathcal{D}(n, m)$ may well not be unique.

Example 4.6 Pairwise Sequence Alignment

Align the strings A = ggaatgg and B = atg, according to the simple scoring scheme: match = 0, mismatch = 20, insertion or deletion = 25.

Here is the state of play after the top row and leftmost column have been initialized (italic), and the element in the second row and second column has been entered as **20** (boldface):

	ϕ	a	t	g
ϕ	*0*	*25*	*50*	*75*
g	*25*	**20**		
g	*50*			
a	*75*			
a	*100*			
t	*125*			
g	*150*			
g	*175*			

The value of **20** was chosen as the minimum of 25 + 25 (horizontal move, or insert gap into string atg), 0 + 20 (substitution a↔g), and 25 + 25 (vertical move, or insert gap into string ggaatgg). Because the substitution (the diagonal move) provided the minimal value, the cell containing 0 in the upper left hand corner of the matrix is the predecessor of the cell in which we have just entered the 20. (If two or even three of the possible moves produce the same value, the resulting cell has multiple predecessors.)

→

Here is the matrix after completion of the calculation:

	ϕ		a		t		g
ϕ	0	←	25	←	50	←	75
	↑	↖		↖		↖	
g	25		20	←	45		50
	↑	↖	↑	↖		↖	
g	50		45		40		45
	↑	↖		↖	↑	↖	
a	75		50		65		60
	↑	↖	↑	↖		↖	↑
a	100		75		70		85
	↑		↑	↖		↖	
t	125		100		75		90
	↑		↑		↑	↖	
g	150		125		100		75
	↑		↑		↑	↖	↑
g	175		150		125		100

It includes the traceback information in the form of arrows pointing from each cell to its predecessor(s). For some applications we may need only the value of $D(A, B)$ but not an alignment; if so, it is unnecessary to save the pointers. (This has implications for the space required to perform the calculation.) Boldface arrows delineate the paths of optimal alignment retracing a trail of predecessors from lower right, back to upper left. In some cases, one cell may show two predecessors. These correspond to alternative alignments with the same score.

There are two cells at which the traceback path branches. This gives a total of four optimal alignments with equal score:

```
ggaatgg    ggaatgg    ggaatgg    ggaatgg
---atg-    ---at-g    --a-tg-    --a-t-g
```

With a gap-weighting scheme assigning a smaller penalty to gap extension than to gap initiation the first two of these would score better than the others. However, more sophisticated gap-weighting schemes require more complicated recurrence formulas for filling the matrix.

This algorithm determines the optimal *global* alignment of two sequences. It is inappropriate for detection of local regions of high similarity within two sequences, or for probing a long sequence with a short fragment, because it imposes gap penalties *outside* the similar regions. The method of T. Smith and M. Waterman solves this problem. Their modifications of the basic dynamic programming algorithm finds optimal local alignments; that is, it selects the substrings from both sequences that are most similar to each other. Their changes affect:

(1) Initialization of the matrix—setting the values of the top row and left column. In the Smith–Waterman method the top row and left column are

Example 4.6 *(continued)*

set to 0. As a result, either sequence can slide along the other before alignment starts, without incurring any gap penalty against the residues it leaves behind.

(2) **Filling in the matrix** In the example of global alignment, at each step a choice is forced among match, insertion or deletion, even if none of these choices is attractive and even if a succession of unattractive choices degrades the score along a path containing a well-fitting local region. The Smith–Waterman method adds the fourth option: end the region being aligned.

(3) **Scoring and traceback** The score of a global alignment is the number in the matrix element at the lower right. In the Smith–Waterman method it is the optimal value encountered, wherever in the matrix it appears. For global alignment, traceback to determine the actual alignment starts at the lower-right cell. In the Smith–Waterman method it starts at the cell containing the optimal value and continues back only as far as the region of local similarity continues.

The Smith–Waterman method would report a unique local optimum for our example:

```
ggaatgg
  atg
```

Note that no gaps appear outside the region matched.
(Example adapted from: Tyler, E. C., Horton, M. R. & Krause, P. R. (1991), A review of algorithms for molecular sequence comparison, *Comp. Biomed. Res*, **24**, 72–96.)

Significance of alignments

Suppose alignment reveals an intriguing similarity between two sequences. Is the similarity significant or could it have arisen by chance? (We raised this question in Chapter 1.) For some simple phenomena—tossing a coin or rolling dice—it is possible to calculate exactly the expected distribution of results, and the likelihood of any particular result. For sequences it is not trivial to define the population from which the alignment is selected. For instance, to take random strings of nucleotides or amino acids as controls ignores the bias arising from nonrandom composition.

A practical approach to the problem is as follows: If the score of the alignment observed is no better than might be expected from a *random permutation* of the sequence, then it is likely to have arisen by chance. We may randomize one of the sequences, many times, realign each result to the second sequence (held fixed), and collect the distribution of resulting scores. Figure 4.4 shows a typical

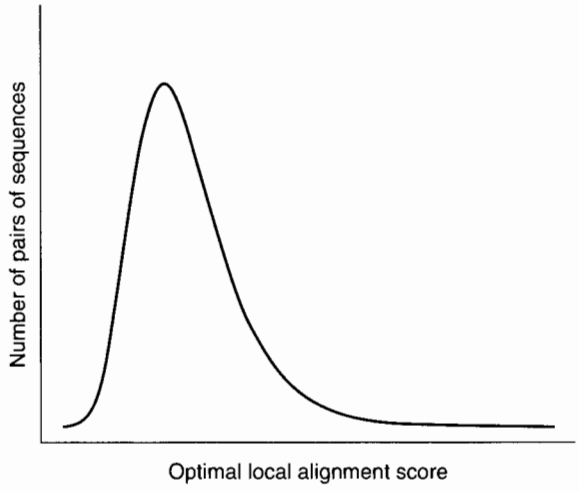


Fig. 4.4 Optimal local alignment scores for pairs of random amino acid sequences of the same length follow an extreme-value distribution. Note the long tail at the right. This means that a score several standard deviations above the mean has a higher probability of arising by chance (that is, it is *less* significant) than if the scores followed a normal distribution. This graph shows the probability distribution function. The formula for the corresponding cumulative distribution function, that is, for any score x, the probability of observing a score $\geq x$ is:

$$P(\text{Score} \geq x) = 1 - \exp(-Ke^{-\lambda x}),$$

where K and λ are parameters related to the position of the maximum and the width of the distribution.

result. For database searches, use the population of results returned from the entire database as the population with which to measure the statistics.

Clearly if the randomized sequences score as well as the original one, the alignment is unlikely to be significant. We can measure the mean and standard deviation of the scores of the alignments of randomized sequences, and ask whether the score of the original sequence is unusually high. The **Z-score** reflects the extent to which the original result is an outlier from the population:

$$\text{Z-score} = \frac{\text{score} - \text{mean}}{\text{standard deviation}}$$

A Z-score of 0 means that the observed similarity is no better than the average of random permutations of the sequence, and might well have arisen by chance. Other values used as measures of significance are P = the probability that the observed match could have happened by chance, and, for database searching, E = the number of matches as good as the observed one that would be expected to appear by chance in a database of the size probed (see Box, page 184).

Many 'rules of thumb' are expressed in terms of per cent identical residues in the optimal alignment. If two proteins have over 45% identical residues in their optimal alignment, the proteins will have very similar structures, and are very likely to have a common or at least a related function. If they have over 25% identical residues, they are likely to have a similar general folding pattern. On the other hand, observations of a lower degree of sequence similarity cannot rule out

How to play with matches but not get burned

Pairwise alignments and database searches often show tenuous but tantalizing sequence similarities. How can we decide whether we are seeing a true relationship? Statistics cannot answer biological questions directly, but can tell us the likelihood that a similarity as good as the one observed would appear, just by chance, among unrelated sequences. To do this we want to compare our result with alignments of the same sequences to a large population. This 'control' population should be similar in general features to our aligned sequences, but should contain few sequences related to them. Only if the observed match stands out from the population can we regard it as significant.

To what population of sequences should we compare our alignment? For pairwise alignments, we can pick one of the two sequences, make many scrambled copies of it using a random-number generator, and align each permuted copy to the second sequence. For probing a database, the entire database provides a comparison population.

Alignments of our sequence to each member of the control population generates a large set of scores. How does the score of our original alignment rate? Several statistical parameters have been used to evaluate the significance of alignments:

- The *Z*-score is a measure of how unusual our original match is, in terms of the mean and standard deviation of the population scores. If the original alignment has score S,

$$Z\text{-score of S} = \frac{\text{S} - \text{mean}}{\text{standard deviation}}$$

  A *Z*-score of 0 means that the observed similarity is no better than the average of the control population, and might well have arisen by chance. The higher the *Z*-score, the greater the probability that the observed alignment has not arisen simply by chance. Experience suggests that *Z*-scores ≥ 5 are significant.

- Many programs report P = the probability that the alignment is better than random. The relationship between Z and P depends on the distribution of the scores from the control population, which do *not* follow the normal distribution.

  A rough guide to interpreting P values:

$P \leq 10^{-100}$	exact match
P in range 10^{-100}–10^{-50}	sequences very nearly identical, e.g. alleles or SNPs
P in range 10^{-50}–10^{-10}	closely-related sequences, homology certain
P in range 10^{-5}–10^{-1}	distant relatives, usually
$P > 10^{-1}$	match probably insignificant

→

→

- For database searches, some programs (including PSI-BLAST) report *E*-values. The *E*-value of an alignment is the expected number of sequences that give the same *Z*-score or better if the database is probed with a random sequence. *E* is found by multiplying the value of *P* by the size of the database probed. Note that *E* but not *P* depends on the size of the database. Values of *P* are between 0 and 1.0. Values of *E* are between 0 and the number of sequences in the database searched.

  A rough guide to interpreting *E* values:

$E \leq 0.02$	sequences probably homologous
E between 0.02 and 1	homology can't be ruled out
$E > 1$	you'd have to expect this good a match just by chance

Statistics are a useful guide, but not a substitute for thinking carefully about the results, and further analysis of ones that look promising!

homology. R. F. Doolittle defined the region of 18%–25% sequence identity as the 'twilight zone' in which the suggestion of homology is tantalizing but dangerous. Below the twilight zone is a region where pairwise sequence alignments tell very little. Lack of significant sequence similarity does not preclude similarity of structure.

Although the twilight zone is a treacherous region, we are not entirely helpless. In deciding whether there is a genuine relationship, the 'texture' of the alignment is important—are the similar residues isolated and scattered throughout the sequence; or are there 'icebergs'—local regions of high similarity (another term of Doolittle's), which may correspond to a shared active site? We may need to rely on other information, about shared ligands or function. Of course if the structures are known, we could examine them directly.

Some illustrative examples:

- Sperm whale myoglobin and lupin leghaemoglobin have 15% identical residues in optimal alignment. This is even below Doolittle's definition of the twilight zone. But we also know that both molecules have similar three-dimensional structures, both contain a haem group, and both bind oxygen. They are indeed distantly-related homologues.
- The sequences of the N- and C-terminal halves of rhodanese have 11% identical residues in optimal alignment. If these appeared in independent proteins, one could not conclude from the sequences alone that they were related. However, their appearance in the same protein suggests that they arose via gene duplication and divergence. The striking similarity of their structures confirms their relationship.

- As a cautionary note, consider the proteinases chymotrypsin and subtilisin. They have 12% identical residues in optimal alignment. These enzymes have a common function, and a common Ser — His — Asp catalytic triad. However, they have dissimilar folding patterns, and are not related. Their common function and mechanism is an example of convergent evolution. This case serves as a warning against special pleading for relationships between proteins with dissimilar sequences on the basis of similarities of function and mechanism!

Multiple sequence alignment

'One amino acid sequence plays coy; a pair of homologous sequences whisper; many aligned sequences shout out loud.' In nature, even a single sequence contains all the information necessary to dictate the fold of the protein. How does a multiple sequence alignment make the information more intelligible? Alignment tables expose patterns of amino acid conservation, from which distant relationships may be more reliably detected. Structure prediction tools also give more reliable results when based on multiple sequence alignments than on single sequences.

Visual examination of multiple sequence alignment tables is one of the most profitable activities that a molecular biologist can undertake away from the lab bench. Don't even THINK about not displaying them with different colours for amino acids of different physicochemical type. A reasonable colour scheme (not the only one) is:

Colour	Residue Type	Amino Acids
Yellow	small nonpolar	Gly, Ala, Ser, Thr
Green	hydrophobic	Cys, Val, Ile, Leu, Pro, Phe, Tyr, Met, Trp
Magenta	polar	Asn, Gln, His
Red	negatively charged	Asp, Glu
Blue	postively charged	Lys, Arg

To be informative a multiple alignment should contain a distribution of closely- and distantly-related sequences. If all the sequences are very closely related, the information they contain is largely redundant, and few inferences can be drawn. If all the sequences are very distantly related, it will be difficult to construct an accurate alignment (unless all the structures are available), and in such cases the quality of the results, and the inferences they might suggest, are questionable. Ideally, one has a complete range of similarities, including distant relatives linked through chains of close relationships.

Case Study 4.1: Structural inferences from multiple sequence alignment of thioredoxins

Thioredoxins are enzymes found in all cells. They participate in a broad range of biological processes, including cell proliferation, blood clotting, seed germination, insulin degradation, repair of oxidative damage, and enzyme regulation. The common mechanism of these activities is the reduction of protein disulphide bonds.

Plate IV shows a multiple sequence alignment of 16 thioredoxins. The structure of *E. coli* thioredoxin contains a central five-stranded β-sheet flanked on either side by α-helices; these helices and strands are indicated by the symbols α and β. Other thioredoxins are expected to share most but not all of the secondary structure of the *E. coli* enzyme. The plate also shows a summary of the alignment as a *sequence logo*, in which letters of different sizes indicate different proportions of amino acids. T. Schneider and M. Stephens designed sequence logos; this example was produced using the web server http://weblogo.berkeley.edu.

Structural and functional features of thioredoxins that we might hope to identify from the multiple sequence alignment include (see Fig. 4.5 and Plate IV):

- **The most highly conserved regions probably correspond to the active site.** The disulphide bridge between residues 32 and 35 in *E. coli* thioredoxin is part of a WCGPC[K or R] motif conserved in the family. Other regions conserved in the sequences, including the PT at residues 75–77 and the GA at residues 92–93, are involved in substrate binding.
- **Regions rich in insertions and deletions probably correspond to surface loops. A position containing a conserved Gly or Pro probably corresponds to a turn.** Turns correlated with insertions and deletions occur at positions 9, 20, 60 and 95. The conserved glycine at position 92 in

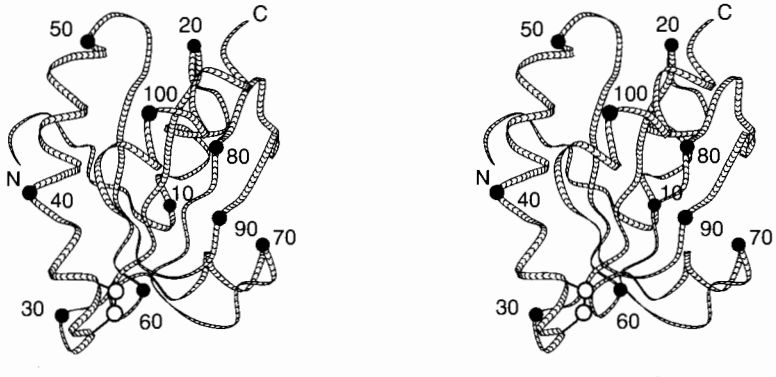


Fig. 4.5 The structure of *E. coli* thioredoxin [2TRX] (see also Plate IV). Residue numbers correspond to those in the multiple sequence alignment table. The N- and C-termini are also marked. Spheres indicate positions of the Cα atoms of every tenth residue. The reactive disulphide bridge between Cys32 and Cys35 appears between the numerals 30 and 60.

→

Case Study 4.1 *(continued)*

E. coli thioredoxin is indeed part of a turn. It is in an unusual mainchain conformation, one that is easily accessible only to glycine (see Chapter 5). The conserved proline at position 76 in *E. coli* thioredoxin is also associated with a turn. It is in another unusual mainchain conformation, this one easily accessible only to proline.

- **A conserved pattern of hydrophobicity with spacing 2 (that is, every other residue)—with the intervening residues more variable and including hydrophilic residues—suggests a β-strand on the surface.** This pattern is observable in the β-strand between residues 50 and 60.
- **A conserved pattern of hydrophobicity with spacing ~4 suggests a helix.** This pattern is observable in the region of helix between residues 40 and 49.

Thioredoxins are members of a superfamily including many more-distantly-related homologues. These include glutaredoxin (hydrogen donor for ribonucleotide reduction in DNA synthesis), protein disulphide isomerase (which catalyses exchange of mismatched disulphide bridges in protein folding), phosducin (a regulator of G-protein signalling pathways), and glutathione S-transferases (chemical defence proteins). Implicit in the multiple sequence alignment table of the thioredoxins themselves are patterns that should be applicable to identifying these more distant relatives.

Applications of multiple sequence alignments to database searching

Searching in databases for homologues of known proteins is a central theme of bioinformatics. Indeed it brooked no delay; we introduced it in Chapter 1 with the application of PSI-BLAST. We reconsider database searching here, with the goal of trying to understand how we can best use available information to build effective procedures. The goals are high **sensitivity**—picking up even very distant relationships—and high **selectivity**—minimizing the number of sequences reported that are not true homologues. Here we discuss how to apply multiple sequences. In Chapter 5 we shall discuss how to apply structural information in addition.

We recognize a familiar face by reacting to its integral appearance rather than to individual features. Similarly, multiple sequence alignments contain subtle patterns that characterize families of proteins.

During the last decade, great progress has been made in devising methods for applying multiple sequence alignments of known proteins to identify related sequences in database searches. The results are central to contemporary applications of bioinformatics, including the interpretion of genomes. Three important methods are: Profiles, PSI-BLAST and Hidden Markov Models.

Profiles

Profiles express the patterns inherent in a multiple sequence alignment of a set of homologous sequences. They have several applications:

- They permit greater accuracy in alignments of distantly-related sequences.
- Sets of residues that are highly conserved are likely to be part of the active site, and give clues to function.
- The conservation patterns facilitate identification of other homologous sequences.
- Patterns from the sequences are useful in classifying subfamilies within a set of homologues.
- Sets of residues that show little conservation, and are subject to insertion and deletion, are likely to be in surface loops. This information has been applied to vaccine design, because such regions are likely to elicit antibodies that will cross-react well with the native structure.
- Most structure-prediction methods are more reliable if based on a multiple sequence alignment than on a single sequence. Homology modelling, for instance, depends crucially on correct sequence alignments.

To use profile patterns to identify homologues, the basic idea is to match the query sequences from the database against the sequences in the alignment table, giving higher weight to positions that are conserved than to those that are variable. If a region is absolutely conserved, such as the WCGPC motif in thioredoxins, the procedure should all but insist on finding it. But being too compulsive risks missing interesting distant relatives; some leeway should be allowed.

What is needed is a quantitative measure of conservation. For each position in the table of aligned sequences, take an inventory of the distribution of amino acids. For instance, for positions 25–30 of the thioredoxin alignment:

Residue number	Number of each amino acid																		
	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
25	1									2								13		
26			16																	
27					16															
28																7	1		5	3
29	16																			
30			1	4									2			1	7	1		

Given a query sequence representing a potential thioredoxin homologue, we want to evaluate its similarity to the query sequence, in such a way that agreement with the known sequences at the absolutely conserved positions—for instance 26, 27 and 29—contributes a very high score, and disagreement at these positions contributes a very low score. For moderately conserved positions, such

as 28, we want a modest positive contribution to the score if the query sequence has an S or a W at this position, and a smaller contribution if it has T or Y. The general idea is to score each residue from the query sequence based on the amino acid distribution at that position in the multiple sequence alignment table.

A tempting but overly simple approach would be to use the inventories as scores directly. For example, if the residues in a query sequence that correspond to positions 25–30 in thioredoxin contains the sequence VDFSAE, this fragment would score 13 + 16 + 16 + 7 + 16 + 4 = 72. This is almost the greatest value possible. The alternative query sequence ACGWAP would score 1 + 0 + 0 + 5 + 16 + 2 = 24, a much lower value. Of course for each query sequence we have to test all possible alignments with the multiple-alignment table, and take the largest total score. The highest-scoring sequences best fit the patterns implicit in the table.

This simple approach would work if our table contained a large and unbiased sample of thioredoxin sequences. But only in this case would the simple inventory give a correct picture of the *potential* distributions of residues at each position. If our sample were small, the pattern derived would be unlikely to reflect the complete repertoire. Or, if the sample contained a large subset of similar sequences, these would be over-represented in the inventories. For instance, we can see in Plate IV that vertebrate thioredoxins form a very closely-related set. If we included twenty more vertebrate thioredoxins in the alignment, the profile would recognize only vertebrate thioredoxins effectively.

Substitution matrices suggest how to make the inventory 'fuzzy' and thereby more general.

The observed amino acid distribution at any residue position is a 20-membered array: $(a_1, a_2, a_3, \ldots a_{20})$, where a_i is the number of amino acids of type i observed at that position. (For position 25 of the thioredoxins, $a_1 = 1$ because 1 alanine is observed, and $a_{18} = 13$, representing the valines.) Then in the simplest scheme, the score of an alanine at position 25 is just 1; the score of a Val is 13; in general, the score of an amino acid of type i is a_i. In this scheme the rows of the inventory itself provide the arrays a needed for scoring each position.

A better scoring scheme would evaluate any amino acid according to its chance of being substituted for one of the observed amino acids. If $D(i, j)$ is the amino acid substitution matrix—BLOSUM62, perhaps—then amino acid i could score $a_1D(i, 1) + a_2D(i, 2) \ldots a_{20}D(i, 20)$. This scheme distributes the score among observed amino acids, weighted according to the substitution probability. An amino acid in the query sequence could score high *either* if it appears frequently in the inventory at this position, or if it has a high probability of arising by mutation from residue types that are common at this position. This approach is more effective in detection of distant relatives from a limited set of known sequences. In this case, the scoring vector for amino acids is the product of the substitution matrix and the rows of the inventory array. An even better approach is to use as the amino acid distribution a combination of the observed inventory and a general background level of amino acid composition.

The result is a set of probability scores for each amino acid (or gap) at each position of the alignment, called a **position-specific scoring matrix.** An alternative

method of deriving a position-specific scoring matrix, based on three-dimensional structures, is described in Chapter 5.

Given a query sequence, and the position-specific scoring matrices derived from a profile, the calculations required to find the optimal score over all alignments of the query sequence with the profile are extensions of the dynamic programming methods of aligning two sequences.

A weakness of simple profiles is that the multiple sequence alignment must be provided in advance, and is taken as fixed. PSI-BLAST and Hidden Markov Models gain power by integrating the alignment step with the collection of statistics.

PSI-BLAST

PSI-BLAST is a program that searches a databank for sequences similar to a query sequence. It is a development of the earlier program BLAST = Basic Local Sequence Alignment Tool. The BLAST program and its variants (see Box) check each entry in the databank *independently* against a query sequence. PSI-BLAST begins with such a one-at-a-time search. It then derives pattern information from a multiple sequence alignment of the initial hits, and reprobes the database using the pattern. Then it repeats the process, fine-tuning the pattern in successive cycles.

BLAST programs come in several flavours

Program	Type of query sequence	Search in database of
BLASTP	amino acid sequence	protein sequences
BLASTX	translated nucleotide sequence	protein sequences
TBLASTN	amino acid sequence	translated nucleotide sequences
TBLASTX	translated nucleotide sequence	translated nucleotide sequences
PSI-BLAST	amino acid sequence	protein sequence database

These programs compare amino acid sequences with amino acid sequences, using by default BLOSUM62 matrix. Searches involving nucleotide sequences, either as query sequence or in the database searched, are carried out by translating nucleotide sequences to amino acid sequence in all six possible reading frames. Another program in this family, BLASTN, compares nucleic acid query sequences with nucleic acid databanks directly.

The problem that BLAST was originally designed to solve is that full-blown dynamic programming methods are rather slow for complete searches in a large databank. Often the databank contains close matches to the query sequence. Less sensitive but faster programs are quite capable of identifying the close matches, and if that is what you want, fine. For example, if you want to search for homologues of a mouse protein in the human genome, the similarity is likely to be high and an approximate method likely to find it. But if you want to search for

homologues of a human protein in *C. elegans* or yeast, the relationship may be more tenuous; and more sophisticated, slower methods may be required. (It may come as a surprise, but computer time requirements are still a consideration. For although computing is becoming less expensive, the sizes of the databanks and the number of searches desired, on a worldwide basis, are growing. The net effect is that the pressure on computing resources is increasing.)

The method used by BLAST goes back, in a sense, to the dotplot approach, checking for well-matching local regions. For each entry in the database, it checks for short contiguous regions that match a short contiguous region in the query sequence, using a substitution scoring matrix but allowing no gaps. An approach in which candidate regions of *fixed length* are identified initially can be made very fast by the use of lookup tables.

Once BLAST identifies a well-fitting region, it tries to extend it. In some versions gaps are allowed. The output of BLAST is the set of local segment matches. In an example from Chapter 1:

```
My.care.is.loss.of.care,.by.old.care.done,
   |||||||||     |||||||||||||    ||||||  ||
Your.care.is.gain.of.care,.by.new.care.won
```

even a very simple algorithm could pick up all matching regions of four contiguous residues and then combine and extend them (see Problem 4.5).

PSI-BLAST, using iterated pattern search (see Box), is much more powerful than simple pairwise BLAST in picking up distant relationships. PSI-BLAST correctly identifies three times as many homologues as BLAST in the region below 30% sequence identity. It is therefore a very useful method for analysing whole genomes.

The only methods based entirely on sequence analysis that do better than PSI-BLAST are Hidden Markov Models. These are described in the next section. To achieve significantly better performance it is necessary to make explicit use of structural information. This is discussed in the next chapter.

A flowchart for PSI-BLAST

1. Probe each sequence in the chosen database independently for local regions of similarity to the query sequence, using a BLAST-type search but allowing gaps.
2. Collect significant hits. Construct a multiple sequence alignment table between the query sequence and the significant local matches.
3. Form a profile from the multiple sequence alignment.
4. Reprobe the database with the profile, still looking only for local matches.
5. Decide which hits are statistically significant and retain these only.
6. Go back to step 2, until a cycle produces no change. This accounts for the 'Iterated' in the program title (Position Sensitive Iterated, PSI).

Hidden Markov Models

A Hidden Markov Model (HMM for short) is a computational structure for describing the subtle patterns that define families of homologous sequences. HMMs are powerful tools for detecting distant relatives, and for prediction of protein folding patterns. They are the only method based entirely on sequences—that is, without explicitly using structural information—competitive with PSI-BLAST for identifying distant homologues. They also perform well at fold recognition, as assessed in CASP programmes.

Within an HMM is a multiple sequence alignment. However, HMMs are usually presented as *procedures for generating sequences*. A conventional multiple sequence alignment table could also be used to generate sequences, by selecting amino acids at successive positions, each amino acid chosen from a position-specific probability distribution derived from the profile. But HMMs are more general than profiles:

1. They include the possibility of introducing gaps into the generated sequence, with position-dependent gap penalties.
2. Application of profiles requires that the multiple sequence alignment be specified up front; the pattern statistics are then derived from the alignment. HMMs carry out the alignment and the assignment of probabilities together.

The internal structure of an HMM shows the mechanism for generating sequences (Fig. 4.6). Begin at Start, and follow some chain of arrows until arriving at End. Each arrow takes you to a state of the system. At each state (1) you take some action—emit a residue, perhaps—and (2) choose an arrow to take you to the next state. The action and the choice of successor state are governed by sets of probabilities. Associated with each state that emits a residue are: one probability distribution for the twenty amino acids, and a second probability distribution for the choice of successor state. Both of these probability distributions are calibrated to encode information about a particular sequence family. In this way, the same general structure can be specialized to many different sequence families.

The dynamics of the system are such that only the current state influences the choice of its successor—the system has no 'memory' of its history. This is characteristic of First-order Markov processes, studied by the nineteenth century Russian mathematician A. A. Markov. Distinguish the succession of states from the succession of amino acids emitted to form the output sequence. Several paths through the system can generate the same sequence. Only the succession of characters emitted is visible; the state sequence that generated the characters remains internal to the system, i.e. hidden. By the probability distributions associated with the individual states, the system captures—or models—the patterns inherent in a family of sequences. Hence the name, Hidden Markov Model.

Software for applying HMMs to biological sequence analysis can achieve:

1. **Training** Given a set of unaligned homologous sequences, it can align them and adjust the transition and residue output probabilities to define an HMM capturing the patterns inherent in the sequences submitted.

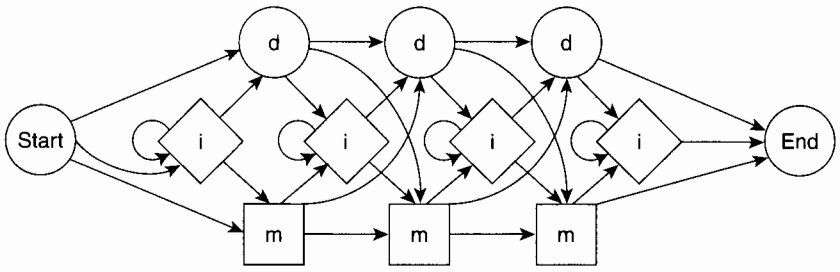


Fig. 4.6 The structure of a Hidden Markov Model (HMM). Corresponding to each residue position in a multiple sequence alignment, the HMM contains a match state (m), and a delete state (d). Insert states (i) appear between residue positions, and at the beginning and end.

- Match states emit a residue. Here the term match means only that there is *some* amino acid both in the model underlying the HMM and in the sequence emitted, not that these are necessarily the *same* amino acid. The probability of emitting each of the twenty amino acids in each of the match states is a property of the model. As with profiles, the probabilities are position dependent.
- Delete states skip a column in the multiple sequence alignment. Arriving at a delete state from a match or insert state corresponds to gap opening, and the probabilities of these transitions reflect a position-specific gap-opening penalty. Arriving at a delete state from a previous delete state corresponds to gap extension.
- Insert states appear between two successive positions in the alignment. If the system enters an insert state, a new residue that does not correspond to a position in the alignment table appears in the emitted sequence. An insert state can be followed by itself, to insert more than one residue. The succession of residues emitted from match and insert states generates the output sequence.

After taking the action appropriate to the type of state (m, d, or i), another probability distribution governs the choice of the next state. In every possible succession of states, every column of the embedded alignment must be visited, and either matched or deleted—there is no way to traverse the network without passing through either an m state or a d state at each position.

2. **Detection of distant homologues** Given an HMM and a test sequence, calculate the probability that the HMM would generate the test sequence. If an HMM trained on a known family of sequences would generate the test sequence with relatively high probability, the test sequence is likely to belong to the family.
3. **Align additional sequences** The probability of any sequence of states can be computed from the individual state-to-state transition probabilities. Finding the most likely sequence of states that the HMM would use to generate one or more test sequences reveals their optimal alignment to the family.

Case Study 4.2 *(continued)*

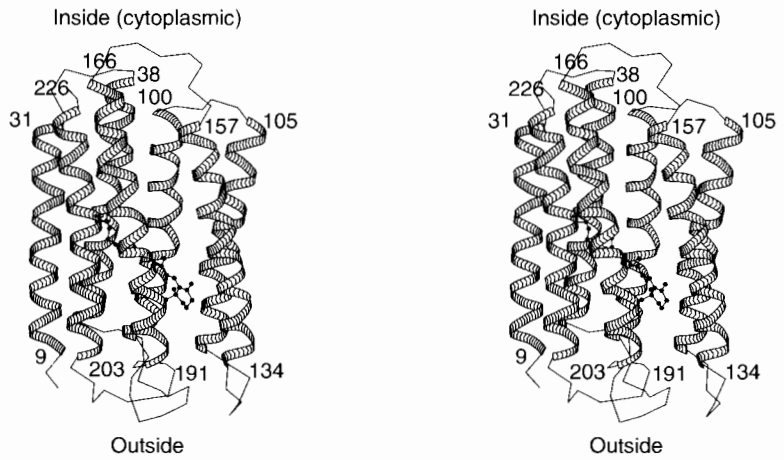


Fig. 4.7 Bacteriorhodopsin from the bacterium *Halobacterium salinarum*, (formerly *Halobacterium halobium*), [2brd] viewed in the plane of the membrane. The ligand shown in red is the chromophore, retinal.

Käll, Krogh and Sonnhammer trained Hidden Markov Models to test simultaneously for transmembrane helices and signal peptides.* The goals are to find both at the same time, to discriminate between them in the results, and to predict not only the positions of the transmembrane helices but the locations—cytoplasmic or interior—of the loops. The method, called **Phobius**, is available at http://phobius.cgb.ki.se/.

Developed from previous work, the HMM for the combined prediction contains separate HMMs for transmembrane helix prediction, and signal peptide prediction, hooked up in parallel. A connection links the models: a transition is possible from the last state of the signal peptide model to the cytoplasmic-side loop state of the helical transmembrane protein model. This makes it possible to treat sequences of proteins that contain a signal peptide sequence *followed* by a transmembrane helical structure. Because a signal peptide must *precede* a transmembrane helical structure in the sequence, no link is necessary *from* the transmembrane helical structure model *to* the signal peptide model.

Phobius was trained on a chosen set of several hundred sequences, retaining only one exemplar of sets of very close relatives. The set included membrane proteins, and control sequences containing neither signal peptides nor transmembrane helices. Annotation of the sequences of the membrane proteins in the training set classified each residue into:

- cytoplasmic loop
- transmembrane helix

* Käll, L., Krogh, A. & Sonnhammer, E. L. L. (2004), A combined transmembrane topology and signal peptide prediction method, *J. Mol. Biol.*, **338**, 1027–1036.

→

- non-cytoplasmic loop
- non-cytoplasmic long loop
- signal peptide cleavage site
- n-region of signal peptide
- h-region of signal peptide
- c-region of signal peptide

The available sequences were split into ten groups. In ten separate calculations, the system was trained on the union of nine of the ten groups with the tenth reserved for testing. After comparison and tuning, a final stage of refinement of the model used all the sequences together.

Application of Phobius to the sequence of bacteriorhodopsin, the protein shown in Fig. 4.7, gives the following results:

```
ID   2brd
FT   DOMAIN       7   11      NON CYTOPLASMIC.
FT   TRANSMEM    12   29
FT   DOMAIN      30   40      CYTOPLASMIC.
FT   TRANSMEM    41   63
FT   DOMAIN      64   82      NON CYTOPLASMIC.
FT   TRANSMEM    83  101
FT   DOMAIN     102  107      CYTOPLASMIC.
FT   TRANSMEM   108  129
FT   DOMAIN     130  134      NON CYTOPLASMIC.
FT   TRANSMEM   135  156
FT   DOMAIN     157  176      CYTOPLASMIC.
FT   TRANSMEM   177  199
FT   DOMAIN     200  204      NON CYTOPLASMIC.
FT   TRANSMEM   205  224
FT   DOMAIN     225  227      CYTOPLASMIC.
```

Readers can compare this prediction with the experimental structure (see Problem 4.9).

Phobius is the most successful algorithm currently available for recognizing signal peptides and helical transmembrane proteins, and for predicting the orientation of the transmembrane segments. Phobius is capable of distinguishing h-domains of signal peptides from transmembrane helices. The number of false classifications of signal peptides was 3.9%, and the number of false classifications of transmembrane helices was 7.7%. These results represent a great improvement over previous methods. It is interesting that addressing the two problems at once proved to be more successful than treating them separately.

Web resources: Hidden Markov Models

Two research groups specializing in biological applications of Hidden Markov Models run web servers and distribute their programs:

R. Hughey, K. Karplus and D. Haussler (University of California at Santa Cruz.):
http://cse.ucsc.edu/research/compbic/sam.html
http://cse.ucsc.edu/research/compbio/HMM-apps/HMM-applications.html

S. Eddy (Washington University, St. Louis, MO, USA)
http://hmmer.wustl.edu/

Results of analysis of known sequences and structures are also available on the Web:

Pfam is a database of multiple sequence alignments and HMMs for many protein domains, developed by A. Bateman, E. Birney, R. Durbin, S. R. Eddy, K. L. Howe and E. L. Sonnhammer:
http://www.sanger.ac.uk/Software/Pfam

J. Gough, K. Karplus, R. Hughey, and C. Chothia have generated HMMs for all PDB superfamilies:
http://stash. mrc-lmb.cam.ac.uk/SUPERFAMILY/

Phylogeny

We have now seen several examples of evolution, in proteins and in genomes. These represent the extension to the molecular level of concerns that have occupied biologists since Darwin and even before. The basic principle is that *the origin of similarity is common ancestry.* Although there are many exceptions, arising from convergent evolution or horizontal gene transfer, the importance of this principle both for rationalizing contemporary observations and giving a window into the history of life cannot be underestimated.

The field of phylogeny has the goals of working out the relationships among species, populations, individuals, or genes. (The general term is 'taxa'.) The *observable* taxa—for instance the extant species for which we wish to work out the pattern of ancestry, are called the 'operational taxonomic units', abbreviated to OTUs.) Relationship is taken in the literal sense of kinship or genealogy, that is. assignment of a scheme of descendants of a common ancestor (see Box). Evolutionary relationships give us a glimpse at the historical development of life (see Box: Time scale of Earth history). Although molecules themselves cannot be dated, the evolutionary events as observed on the molecular level can be calibrated with the fossil record.

Concepts related to biological classification and phylogeny (see page 29)

Homology means, specifically, descent from a common ancestor.
Similarity is the measurement of resemblance or difference, independent of the source of the resemblance. Similarity is observable in data collectable *now*, and involves no historical hypotheses. In contrast, assertions of homology require inferences about historical events which are almost always unobservable.

→

Clustering is bringing together similar items, distinguishing classes of objects that are more similar to one another than they are to other objects outside the classes. Most people would agree about degrees of similarity, but clustering is more subjective. When classifying objects, some people prefer larger classes, tolerating wider variation; others prefer smaller, tighter, classes. They are called *groupers* or *splitters*.

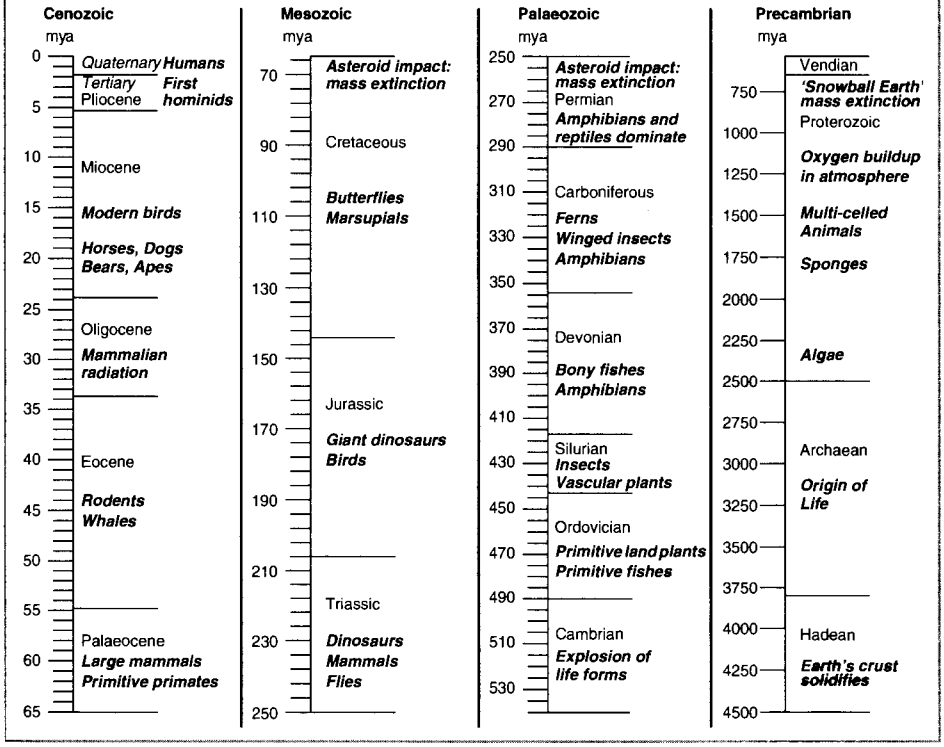
Hierarchial clustering is the formation of clusters of clusters of...

Phylogeny is the description of biological relationships, usually expressed as a tree. A statement of phylogeny among objects *assumes* homology and *depends* on classification. Phylogeny states a topology of the relationships based on classification according to similarity of one or more sets of characters, or on a model of evolutionary processes. In many cases, phylogenetic relationships based on different characters are consistent, and support one another. If different characters induce inconsistent phylogenetic relationships, then all the putative relationships are dubious. Conversely, note that the same similarity data may be consistent with different possible topologies or trees.

Time scale of Earth history

Geological ages (e.g. Cenozoic), epochs (e.g. Quaternary), and cataclysmic events (e.g. Asteroid impact: mass extinction) in black. First appearance of, or prevalence of, different life forms in red.

mya = millions of years ago.



The results of phylogenetic analyses are usually presented in the form of an evolutionary tree. The taxonomy of the ratites—large flightless birds—is a typical example (Fig. 4.8a). The ancestor of the ratites is believed to be a bird that could fly, probably related to the extant tinamous.

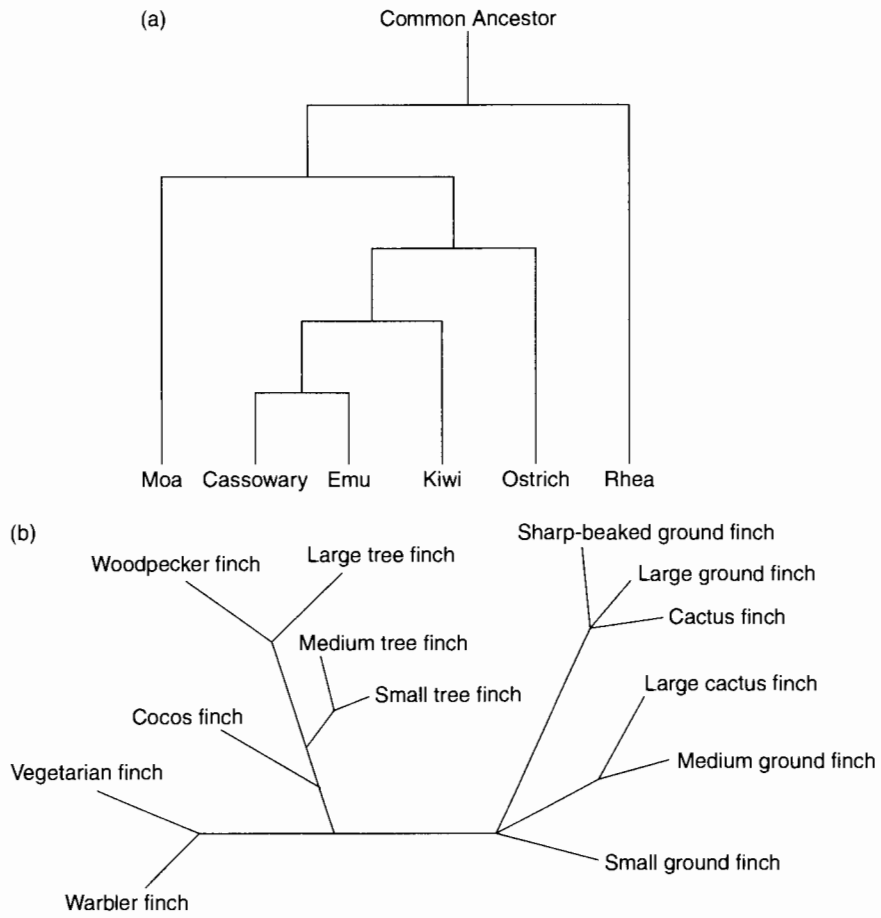


Fig. 4.8 (a) Phylogenetic tree of ratites (large flightless birds) based on mitochondrial DNA sequences. The common ancestor is at the *root* of this tree. A surprising implication of these DNA sequences is that the moa and kiwi are not closest relatives, and therefore that New Zealand must have been colonized twice by ratites or their ancestors. (b) *Unrooted* tree of relationships among finches from the Galapagos and Cocos Islands. Darwin studied the Galapagos finches in 1835, noting the differences in the shapes of their beaks and the correlation of beak shape with diet. Finches that ate fruits had beaks like those of parrots, and finches that ate insects had narrow, prying beaks. These observations were seminal to the development of Darwin's ideas. As early as 1839 he wrote, in *The Voyage of the Beagle*, 'Seeing this gradation and diversity of structure in one small, intimately related group of birds, one might really fancy that from an original paucity of birds in this archipelago, one species had been taken and modified for different ends.'

Such a tree, showing all descendants of a single original ancestral species, is said to be **rooted**. (The root of the tree typically appears at the top or the side; botanists will have to get used to this.) Alternatively, we may be able to specify relationships but not order them according to a history. The relationships among the finches of the Galapagos Islands, studied by Darwin, plus a related species from the nearby Cocos Island, are shown in an **unrooted tree** (Fig. 4.8b). Addition of data from a species on the South American mainland ancestral to the island finches would allow us to **root the tree.**

Statement of a tree of relationships may reveal only the connectivity or topology of the tree, in which case the lengths of the branches contain no information. A more ambitious goal is to show the distances between taxa quantitatively, for instance to label the branches with the time since divergence from common ancestor.

Given a set of data that characterizes different groups of organisms—for example, DNA or protein sequences, or protein structures, or shapes of teeth from different species of animals—how can we derive information about the relationships among the organisms in which they were observed? It is rare for species relationships and ancestry to be directly observable. Evolutionary trees determined from genetic data are often based on inferences from the patterns of similarity, which are all that is observable among species living now. We generally assume that the more similar the characters the more closely related the species, although this is a dangerous assumption. Nevertheless, from the relationships among the characters we wish to infer patterns of ancestry: the *topology* of the phylogenetic relationships (informally, the 'family tree').

To what extent do the topologies of the relationships depend on the choice of character? In particular, are there *systematic* discrepancies between the implications of molecular and palaeontological analysis?

Molecular approaches to phylogeny developed against a background of traditional taxonomy, based on a variety of morphological characters, embryology, and, for fossils, information about the geological context (stratigraphy). The classical methods have some advantages. Traditional taxonomists have much less restricted access to extinct organisms, via the fossil record. They can *date* appearances and extinctions of species by geological methods. Molecular biologists, in contrast, have very limited access to extinct species. Some subfossil remains of species which became extinct as recently as the last century or two have legible DNA, including specimens of the quagga (a relative of the zebra) and the thylacine (Tasmanian 'wolf', a marsupial), and some New Zealand birds (including moas). We have seen, in Chapter 1, an example of a sequence from the mammoth. Some DNA sequences from Neanderthal man have been recovered from an individual who died approximately 30 000 years ago. But *Jurassic Park* remains fiction!

A crucial event in the acceptance of molecular methods occurred in 1967 when V. M. Sarich and A. C. Wilson dated the time of divergence of humans from chimpanzees at 5 million years ago, based on immunological data. At that time

palaeontologists dated this split at 15 million years ago, and were reluctant to accept the molecular approach. Reinterpretation of the fossil record led to acceptance of a more recent split, and broke the barrier to general acceptance of molecular methods. (It is now generally accepted that human and chimpanzee lineages diverged between ~6–8 million years ago.)

Indeed, many molecular properties have been used for phylogenetic studies, some surprisingly long ago. Serological cross-reactivity was used from the beginning of the last century until superseded by direct use of sequences. In one of the most premature scientific studies I know of, E. T. Reichert and A. P. Brown published, almost a century ago (in 1909), a phylogenetic analysis of fishes based on haemoglobin crystals. Their work was based on Stenö's law (1669), that although different crystals of the same substance have different dimensions—some are big, some small—they have the same interfacial angles, reflecting the similarity in microscopic arrangement and packing of the atomic or molecular units within the crystals. Reichert and Brown showed that the interfacial angles of crystals of haemoglobins isolated from different species showed patterns of similarity and divergence parallel to the species' taxonomic relationships.

Reichert and Brown's results are replete with significant implications. They show that proteins have definite, fixed shapes, an idea by no means recognized at the time. They imply that as species progressively diverge, the structures of their haemoglobins progressively diverge also. In 1909, no one had a clue about nucleic acid or protein sequences. In principle, therefore, the recognition of evolution of protein structures preceded, by several decades, the idea of evolution of sequences.

Today, DNA sequences provide the best measures of similarities among species for phylogenetic analysis. The data are digital. It is even possible to distinguish selective from non-selective genetic change, using the third position in codons, or untranslated regions such as pseudogenes, or the ratio of synonymous to nonsynonymous codon substitutions. Many genes are available for comparison. This is fortunate, because given a set of species to be studied, it is necessary to find genes that vary at an appropriate rate. Genes that remain almost constant among the species of interest provide no discrimination of degrees of similarity. Genes that vary too much cannot be aligned. There is an analogous situation in radioactive dating requiring choice of an isotope with a half-life of the same general magnitude as the time interval to be determined.

Fortunately genes vary widely in their rates of change. The mammalian mitochondrial genome, a circular double-stranded DNA molecule approximately 16 000 bp long, provides a useful fast-changing set of sequences for the study of evolution among closely-related species. In contrast, ribosomal RNA sequences were used by C. Woese to identify the three major kingdoms: Archaea, Bacteria and Eukarya (see Fig. 1.2).

Conversely, different rates of change of sequences of different genes can lead to different and even contradictory results in phylogenetic studies. This is especially

true if what we want is not just the topology of the relationships but the branch lengths. In addition, horizontal gene transfer, and convergent evolution, are competing phenomena—that is, competing with descent—that interfere with the deduction of phylogenetic relationships. In attempts to push the determination of evolutionary relationships farther and farther back in time, it has been found that horizontal gene transfer appears to have been more important early in life history.

Phylogenetic trees

We describe phylogenetic relationships as trees. In computer science, a tree is a particular kind of graph. A graph is a structure containing nodes (abstract points) connected by edges (represented as lines between the points). (See Box; we shall develop the ideas of graphs in connection with the discussion of networks in Chapter 6.) A **path** from one node to another is a consecutive set of edges beginning at one point and ending at the other, like our trip from Malmö to Tromsø. A **connected graph** is a graph containing at least one path between any two nodes. From these we can define a **tree:** a connected graph in which there is *exactly* one path between every two points. A particular node may be selected as a **root**; but this is not necessary—abstract trees may be rooted or unrooted (see Fig. 4.8). Unrooted trees show the topology of relationship but not the pattern of descent. A rooted tree in which every node has two descendants is called a **binary tree** (see PERL program, page 204).

Another special kind of graph is a **directed graph** in which each edge is a one-way street (see page 315). Examples include the Hidden Markov Model diagram shown in Fig. 4.6, and the neural networks illustrated in Chapter 5. Rooted phylogenetic trees are, implicitly, directed graphs, the ancestor-descendent relationship implying the direction of each edge.

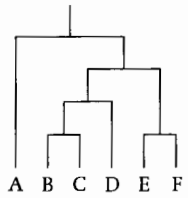
It may be possible to assign numbers to the edges of a graph to signify, in some sense, a 'distance' between the nodes connected by the edges. The graph may then be drawn to scale, with the sizes of the edges proportional to the assigned lengths. The length of a path through the graph is the sum of the edge lengths.

In phylogenetic trees, edge lengths signify either some measure of the dissimilarity between two species, or the length of time since their separation. The assumption that differences between properties of living species reflects their divergence times will be true only if the rates of divergence are the same in all branches of the tree. Many exceptions are known. For instance, among mammals many proteins from rodents show relatively fast evolutionary rates (see Weblem 4.8).

PERL Example 4.2 A program to draw binary trees

```
#!/usr/bin/perl
#drawtree.prl -- draws binary trees (root at top)
#usage:  echo '(A((BC)D)(EF))' | drawtree.prl > output.ps

print <<EOF;
%!PS-Adobe-\n%%BoundingBox: atend
/n /newpath load def /m /moveto load def /l /lineto load def
/rm /rmoveto load def /rl /rlineto load def /s /stroke load def
1.0 setlinewidth 50 100 translate 2 2 scale
/Helvetica findfont 10 scalefont setfont
EOF

$tree = <>; chop($tree); $_ = reverse($tree); s/[()]//g;

$x = 0; $y = 0;
while ($nd = chop()) {
   print "$x $y m ($nd) stringwidth pop -0.5 mul 0 rm ($nd) show\n";
   $xx{$nd} = $x; $x+=20; $yy{$nd} = 10;
}

while ($tree =~ s/\(?([A-Z])([A-Z])\)?/\1/) {
    print "n $xx{$1} $yy{$1} m\n";
    ($yy{$1} > $yy{$2}) || ($yy{$1} = $yy{$2}); $yy{$1} += 20;
    print "$xx{$1} $yy{$1} l $xx{$2} $yy{$1} l $xx{$2} $yy{$2} l s\n";

    $xx{$1} = 0.5*($xx{$1} + $xx{$2});
}
print "n $xx{$tree} $yy{$tree} m 0 20 rl s showpage\n";

$rx = 2*$x + 30; $yt = 2*$yy{$tree} + 146;
print "%%BoundingBox: 40 95 $rx $yt\n";
```

The input: (A((BC)D)(EF)) produces the following output, as a PostScript file, which can be printed on most printers and displayed on most terminals.



Glossary of terms related to graphs

Graph an abstract structure containing **nodes** (points) and **edges** (lines connecting points).
Path a consecutive set of edges.
Connected graph a graph in which there is at least one path between every two nodes.
Tree a connected graph with exactly one path between every two nodes.
Edge length a number assigned to each edge signifying in some sense the distance between the nodes connected by the edge.
Path length the sum of the lengths of the edges that comprise the path.

Broadly, there are two approaches to deriving phylogenetic trees. One approach makes no reference to any historical model of the relationships. Proceed by measuring a set of distances between species, and generate the tree by a hierarchical clustering procedure. This is called the **phenetic** approach. The alternative, the

cladistic approach, is to consider possible pathways of evolution, infer the features of the ancestor at each node, and choose an optimal tree according to some model of evolutionary change. Phenetics is based on similarity; cladistics is based on genealogy.

Clustering methods

Phenetic, or clustering, approaches to determination of phylogenetic relationships are explicitly non-historical. Indeed, hierarchical clustering is perfectly capable of producing a tree even in the absence of evolutionary relationships. A departmental store has goods clustered into sections according to the type of product—for instance, clothing or furniture—and subclustered into more closely-related subdepartments, such as men's and women's shoes. Men's and women's shoes have a common ancestor, but there is no implication that shoes and furniture do.

A simple clustering procedure works as follows: Given a set of species, determine for all pairs a measure of the similarity or difference between them. This could depend on a physical body trait such as the difference between the average adult height of members of two species. Or one could use the number of different bases in alignments of mitochondrial DNA. To create a tree from the set of dissimilarities, first choose the two most closely-related species and insert a node to represent their common ancestor. Then replace the two selected species by a set containing both, and replace the distances from the pair to the others by the average of the distances of the two selected species to the others. Now we have a set of pairwise dissimilarities, not between individual species, but between sets of species. (Regard each remaining individual species as a set containing only one element.) Then repeat the process, as in the following example.

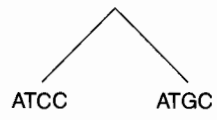
Example 4.7

Consider four species characterized by homologous sequences ATCC, ATGC, TTCG and TCGG. Taking the number of differences as the measure of dissimilarity between each pair of species, use a simple clustering procedure to derive a phylogenetic tree.

The distance matrix is:

	ATCC	ATGC	TTCG	TCGG
ATCC	0	**1**	2	4
ATGC		0	3	3
TTCG			0	2
TCGG				0

Because the matrix is symmetric, we need fill in only the upper half. The smallest nonzero distance is **1** (in boldface), between ATCC and ATGC. Therefore our first cluster is {ATCC, ATGC}. The tree will contain the fragment:



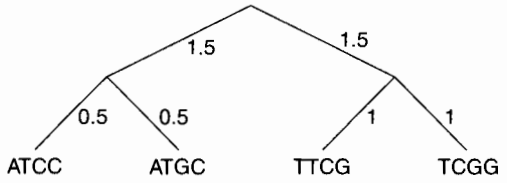
→

Example 4.7 *(continued)*

The reduced distance matrix is:

	{ATCC, ATGC}	TTCG	TCGG
{ATCC, ATGC}	0	$\frac{1}{2}(2+3) = 2.5$	$\frac{1}{2}(4+3) = 3.5$
TTCG		0	2
TCGG			0

The next cluster is {TTCG, TCGG}, distance 2. Finally, linking the clusters {ATCC, ATGC} and {TTCG, TCGG} gives the tree:



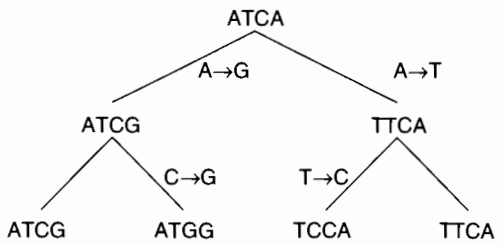
Branch lengths have been assigned according to the rule:
branch length of edge between nodes X and Y = $\frac{1}{2}$ distance between X and Y
Whether the branch lengths are truly proportional to the divergence times of the taxa represented by the nodes must be determined from external evidence.

This process of tree building is called the UPGMA method (Unweighted Pair Group Method with Arithmetic mean). A modification of the UPGMA method by N. Saitou and M. Nei, called Neighbour Joining, is designed to correct for unequal rates of evolution in different branches of the tree.

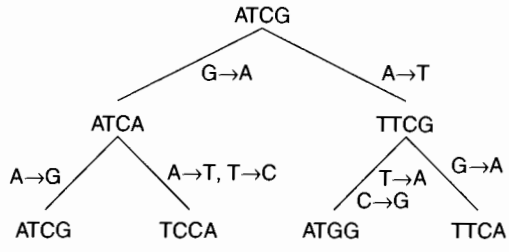
Cladistic methods

Cladistic methods deal explicitly with the patterns of ancestry implied by the possible trees relating a set of taxa. Their aim is to select the correct tree by utilizing an explicit model of the evolutionary process. The most popular cladistic methods in molecular phylogeny are the **maximum parsimony** and **maximum likelihood** approaches. They are specialized to sequence data, starting from a multiple sequence alignment. Neither maximum parsimony nor maximum likelihood could be applied to anatomic characters such as average adult height.

The *maximum parsimony* method of W. Fitch defines an optimal tree as the one that postulates the fewest mutations. For instance, given species characterized by homologous sequences ATCG, ATGG, TCCA and TTCA, the tree:



postulates four mutations. An alternative tree:



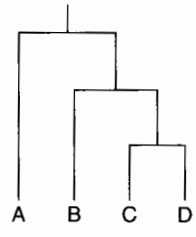
postulates seven mutations. Note that the second tree implies that the G → A mutation in the fourth position occurred twice independently. The former tree is optimal according to the maximum parsimony method, because no other tree involves fewer mutations. In many cases, several trees may postulate the same number of mutations, fewer than any other tree. For such cases the maximum parsimony approach does not give a unique answer.

The *maximum likelihood* method assigns quantitative probabilities to mutational events, rather than merely counting them. Like maximum parsimony, maximum likelihood reconstructs ancestors at all nodes of each tree considered; but it also assigns branch lengths based on the probabilities of the mutational events postulated. For each possible tree topology, the assumed substitution rates are varied to find the parameters that give the highest likelihood of producing the observed sequences. The optimal tree is the one with the highest likelihood of generating the observed data.

Both maximum parsimony and maximum likelihood methods are superior to clustering techniques. This has been demonstrated with cases where independent evidence—for instance, from palaeontology—provides a correct answer, and also with simulated data—computed generation of evolving sequences.

The problem of varying rates of evolution

Suppose that the four species A, B, C and D have the phylogenetic tree:



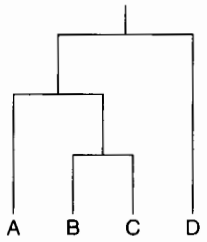
This tree is consistent with the dissimilarity matrix:

	A	B	C	D
A	0	3	3	3
B		0	2	2
C			0	1
D				0

Suppose, however, that taxon D is changing very fast, although the phylogeny is unaltered. The dissimilarity matrix might then be observed to be:

	A	B	C	D
A	0	3	3	20
B		0	2	20
C			0	20
D				0

from which we would derive the incorrect phylogenetic tree:



All the methods discussed here are subject to errors of this kind if the rates of evolutionary change vary along different branches of the tree. To test for varying rates, compare the species under consideration with an **outgroup**—a species more distantly related to all the species in question than any pair of them is to each other. For instance, if we are studying species of primates, a non-primate mammal such as the cow would be a suitable outgroup. If the rates of evolution among the primate species were constant, we should expect to observe approximately equal dissimilarity measures between all primate species and the cow. If this is not observed, the suggestion is that evolutionary rates have varied among the primates, and the character being used may well not provide the correct phylogenetic tree.

Computational considerations

Cladistic methods—maximum parsimony and maximum likelihood—are more accurate than simpler clustering methods such as UPGMA, but require large amounts of computer time if the number of species is appreciable. The total number of possible trees, which cladistic methods are committed to considering if they could, increases very rapidly with the number of species. As a result, in many cases of interest these methods can give only approximate answers, even with respect to their intrinsic assumptions.

Because calculated phylogenies are often approximations, it is important to try to test them. Methods include:

1. Comparison of phylogenies obtained from different characters describing the same set of taxa—are they consistent? If trees produced from different characters share a subtree, perhaps that portion of the phylogeny has been determined reliably and other portions have not.

2. Analysis of subsets of taxa should give the same answer—with respect to the subset—as appears within the full tree.
3. Formal statistical tests, involving rerunning the calculation on subsets of the original data, are known as **jackknifing** and **bootstrapping:**
   - **Jackknifing** is calculation with data sets sampled randomly from the original data. For phylogeny calculations from multiple sequence alignments, select different subsets of the positions in the alignment, and rerun the calculation. Finding that each subset gives the same phylogenetic tree lends it credibility. If each subset gives a different tree, none of them is trustworthy.
   - **Bootstrapping** is similar to jackknifing except that the positions chosen at random may include multiple copies of the same position, to form data sets of the same size as the original, to preserve statistical properties of the sampling.
4. If there are very long edges, consider seriously the possibility of unequal variation in evolutionary rate that may have perturbed the calculation. Introduce outgroup taxa to check.

Web resources: Phylogenetic trees

The taxonomic community has expended great effort to produce mature software. The PHYLIP package (PHYLogeny Inference Package) of J. Felsenstein is an integrated collection of many different techniques. The programs work on many different types of computers, and are freely distributed and easily obtained.

Summaries of tools for phylogenetics; includes useful list of web sites, and general listing of phylogeny software:

http://evolution.genetics.washington.edu/phylip/software.html

and Whelan, S., Liò, P. & Goldman, N. (2001), Molecular phylogenetics: State-of-the-art methods for looking into the past, *Trends in Genetics*, 17, 262–272.

Some multiple sequence alignment packages, such as CLUSTAL-W, provide facilities to launch a phylogenetic tree calculation from the alignments they produce.

Recommended reading

Altschul, S. F. & Koonin, E. V. (1998), Iterated profile searches with PSI-BLAST—a tool for discovery in protein databases, *Trends Biochem. Sciences*, **23**, 444–447. [Description of one of the most important tools for database searching for sequence similarity.]

Altschul, S. F., Boguski, M. S., Gish, W. & Wootton, J. C. (1994), Issues in searching molecular sequence databases, *Nature Genetics*, **6**, 119–129. [General background to challenges in designing information retrieval methods and interpreting the results.]

Eddy, S. (1996), Hidden Markov Models, *Current Opinion in Struct. Biol.*, **6**, 361–365. [Readable introduction to an important mathematical technique providing powerful tools for detection of distantly-related sequences, and protein fold recognition.]

Efron, B. & Gong, G. (1983), A leisurely look at the bootstrap, the jackknife, and cross-validation, *The American Statistician*, **37**, 36–48. [Classic paper on statistical methods for calibrating pattern recognition procedures.]

Gura, T. (2000), Bones, molecules . . . or both? *Nature*, **406**, 230–233. [Discussion on the congruences and conflicts between evolutionary trees based on molecules and morphology.]

Penny, D., Hendy, M. D., Zimmer, E. A & Hamby, R. K. (1990), Trees from sequences: Panacea or Pandora's box? *Aust. Syst. Bot.*, **3**, 21–38. [Cautionary notes about determination of phylogenetic trees.]

Whelan, S., Liò, P. & Goldman, N. (2001), Molecular phylogenetics: State-of-the-art methods for looking into the past, *Trends in Genetics*, **17**, 262–272. [Review, with links to software.]

Exercises, Problems, and Weblems

Exercises

4.1 What is the Hamming distance between the words DECLENSION and RECREATION?

4.2 What is the Levenshtein distance between the words BIOINFORMATICS and CONFORMATION?

4.3 The Levenshtein distance between the strings `agtcc` and `cgctca` is 3, consistent with the following alignment:

```
ag-tcc
cgctca
```

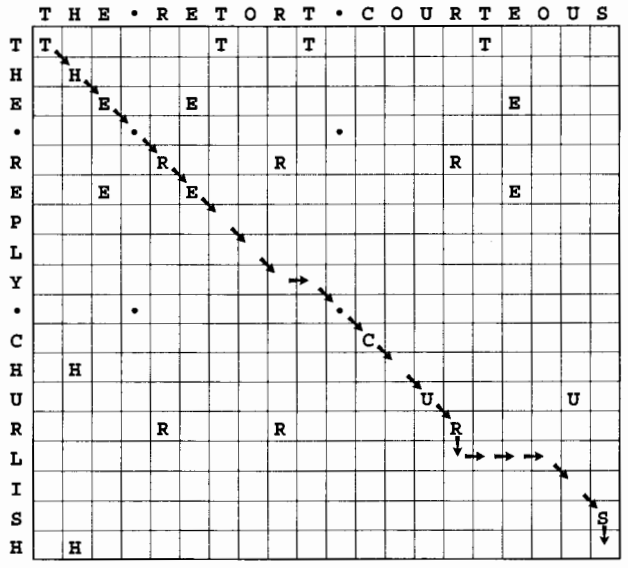
Provide a sequence of three edit operations that convert `agtcc` to `cgctca`.

4.4 'I wasted time and now doth time waste me.' (a) First sketch the expected appearance of a dotplot of this character string against itself. (b) Then calculate the dotplot exactly, recording only character identities as dots in the matrix, and compare with (a).

4.5 What values of window and threshold (see program, page 164) would you use to eliminate the singletons in the DOROTHYHODGKIN dotplot, but retain the other matches shown?

4.6 For each of the matrices (a) PAM250 and (b) BLOSUM62, which substitution is more probable, W↔F or H↔R?

4.7 To what alignment does the path through the following dotplot correspond?



4.8 In planning your trip from Malmö to Tromsø (see page 177), suppose that for personal reasons you wanted to include a visit to Uppsala. How could you

adjust the costs of the segments to ensure that the minimal-cost route passes through Uppsala?

4.9 How would you use a dotplot to pick up palindromic DNA sequences of the type that appear partly on each strand, as in the specificity sites of restriction endonucleases?

4.10 Modify the PERL program on page 164 that draws dotplots to accept sequences in FASTA format.

4.11 To what value of P would a Z-score of 1 correspond in a normal distribution?

4.12 For each of the alignments in Fig. 4.2, state whether it is in the twilight zone, more similar than the twilight zone or less similar than the twilight zone.

4.13 Figure 4.2a shows the sequence alignment of papaya papain and kiwi fruit actinidin, and the corresponding dotplot. The sequence alignment shows two places at which one or more residues are deleted from the papain sequence, and one place at which a residue is deleted from the actinidin sequence. On a photocopy of Fig. 4.2a, indicate in the dotplot the positions of these insertions and deletions.

4.14 Suppose it were argued that randomizing a sequence is not an appropriate way to generate a control population for analysis of the statistical significance of pairwise sequence alignments, because natural sequences have nonrandom dipeptide or tripeptide frequencies. What improved way to generate a control population would you suggest?

4.15 Comparisons of DNA sequences of homologous chromosomes in different people show that, on average, 1 of every 700 bp of noncoding DNA is different. 95% of the human genome is noncoding. Estimate the number of polymorphisms in the human genome, to give some idea of the number of potential DNA markers.

4.16 Show the calculations that lead to the entry with value 65 in Example 4.6. What is the significance of the observation that there two arrows coming from it?

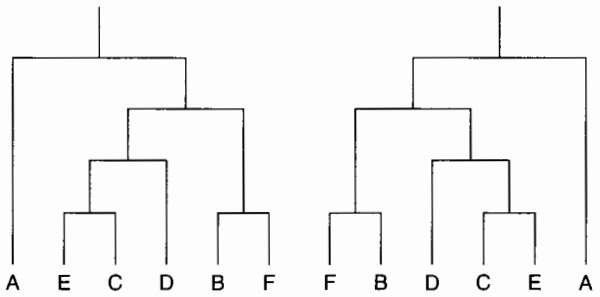
4.17 The α-helix formed by residues 32–49 in *E. coli* thioredoxin is interrupted. On a photocopy of Fig. 4.5 indicate where this interruption appears. At what residue is this distortion likely to occur?

4.18 At what positions in the *E. coli* thioredoxin sequence do turns occur in the structure that are *not* associated with insertions or deletions in the sequence alignment?

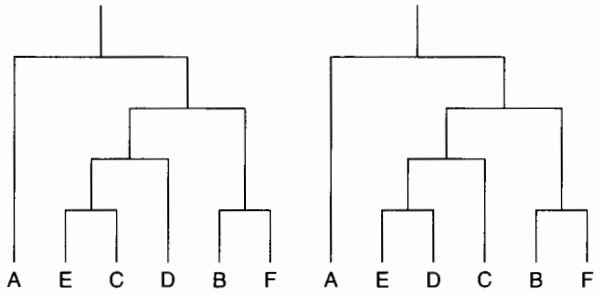
4.19 (a) Using simple 'inventory' scoring, what hexapeptide gives the greatest possible value for a match to positions 25–30 in thioredoxin scoring table (page 189). (b) Using a scoring scheme distributed among all 20 amino acids according to the BLOSUM62 matrix, compare the score of this hexapeptide with the score of the hexapeptide VDFSAE.

4.20 (a) Make an inventory of the region from residues 90–95, similar to the table on page 189. What contribution would the following sequences aligned to these residues make to a simple profile score using inventories as weights? (b) ISSAVK (c) FVGAKE.

4.21 (a) Is the following pair of trees identical in topology?



(b) Is the following pair of trees identical in topology?



4.22 Draw all possible rooted trees relating three taxa. How many are there?

4.23 For the final graph in Example 4.7, how was the branch length 1.5 of the nodes joining the clusters {ATCC, ATGC} and {TTCG, TCGG} arrived at?

4.24 Using the original matrix of distances, and the final tree, in Example 4.7, for each pair of species compare the original distance between them with the sum of the lengths of the paths joining them in the tree.

4.25 Draw an example of a completely connected graph that is not a tree.

4.26 Mitochondrial DNA sequences of European, African and Asian cattle suggest that European and African breeds are more closely related to each other than to Indian breeds. To exclude the appearance of this result as the spurious result of differential rates of evolution in the two lineages, suggest a reasonable choice of a species as an outgroup.

4.27 Draw the final tree in Example 4.7 to scale, with sizes of the edges proportional to the assigned branch lengths.

4.28 For the dynamic programming method for alignment of two sequences of length n, we noted that the execution *time* requirements scale as n^2. In a naive implementation of the algorithm, how would the storage *space* requirements scale with n : (a) If we want to determine an optimal alignment, so that traceback information must be stored? (b) If we want only the score and not an alignment, so that traceback need not be stored? (Note: subtle ways of implementing the algorithm substantially reduce the space requirements over the naive implementation.)

Problems

4.1 Draw a dotplot of the following sequence from the Wheat dwarf virus genome: `ttttcgtgagtgcgcggaggctttt` against itself. In what respects is it not a perfect palindrome?

4.2 (a) How would you change the algorithm in the section on dynamic programming (starting on page 176) to find the optimal matches of a relatively short pattern $A = a_1a_2 \ldots a_n$ in a long sequence $B = b_1b_2 \ldots b_m$ with $n \ll m$. (No gap penalty in the regions of B that precede and follow the region matched by A.) This corresponds to motif matching as described in Chapter 1. (b) Redo the calculations of Example 4.6 for aligning the strings `ggaatgg` and B = `atg` as a motif matching problem, using the same scoring scheme: match = 0, mismatch = 20, *internal* gap initiation 25, gap extension 22. (c) How do the results that you get differ from those derived in the example?

4.3 How could you modify the profile method to retain its ability to pick up non-mammalian thioredoxins if a large number of additional mammalian, closely-related, sequences were added to the table? Consider (a) methods that attempt to remove redundancy by ignoring certain sequences, and (b) methods that retain all the sequences but include a weighting scheme to balance the representation of the closely-related ones.

4.4 Write a PERL program to make profile inventories from a multiple sequence alignment, and to score the matching of query sequences using BLOSUM62. Assume that the query sequence has already been aligned before it is presented to the program.

4.5 (a) Write a PERL program to read in two character strings and report all matches of contiguous five-character regions. Test this on the strings:

`My.care.is.loss.of.care,.by.old.care.done,` and
`Your.care.is.gain.of.care,.by.new.care.won`

(b) Develop this program to extend and combine the matches found to the longest regions that contain perfectly-matching 5-mers, without gaps, with no more than 25% mismatches overall.

4.6 Extend the previous problem by writing a PERL program to illustrate, in a form based on dotplots, the progress of a BLAST-type algorithm at the stages when it (a) detects all matching substrings of length 5, (b) extends them to maximal contiguous matches, (c) combines them to form a match with no more than k mismatches. You may make use of the PERL program for dotplots in the text.

4.7 Write a program to *animate* the progress of a BLAST-type algorithm as described in the previous problem. As background, look on the Web for examples of animation of string search algorithms. (This problem requires a relatively high level of experience with computing.)

4.8 Single-stranded RNAs, such as tRNA, adopt conformations containing *stem-loop* regions, in which a region of the chain loops back on itself to form a double-stranded helix from complementary base pairs, with antiparallel strands. How

would a program that would detect palindromes be useful in analysing RNA sequences to detect regions capable of forming perfect (i.e. no mismatched bases) stem-loop structures?

4.9 Compare the predictions of the residue limits of the transmembrane helices of bacteriorhodopsin by the Phobius method with those observed in the experimental structure (Fig. 4.7).

4.10 Suppose that you have a pair of dice, one red and one green. Define a *state* of the pair of dice as the following pair of numbers: the number appearing on the upper face of the red one followed by the number appearing on the upper face of the green one. Instead of rolling the dice, pass from state to state by tipping each of the dice by 90° in any direction with equal probability. Then a state in which 6 is up on one of the dice can be followed, with equal probability, by 2, 3, 4, or 5 up. (Dice are constructed so that the sum of the numbers on each of the three sets of opposite faces is 7. Therefore the probability that 1 follows 6 is zero, because this would require a 180° rotation.) The probability of the sequence 6, 2, 6, 4 is $(1/4)^4 = 1/256$. The probability of generating the sequence 6, 2, 5, 4 is zero, because the transition from $2 \rightarrow 5$ is not allowed, and the probability of the sequence 6, 6, 2, 3, 4 is zero because the system must change its state, so 6 cannot be followed by 6.

This procedure defines a first-order Markov process.

Write a program to answer the following questions: Suppose the initial state has a four at the top of the red die and a three at the top of the green die. (a) What is the probability of another state in which the numbers add up to 7 appearing within 5 moves? (b) If the initial state is an 8, what is the probability that another 8 appears before a 7?

4.11 Show that any (undirected) graph that has either of the following properties must also have the other: (1) There is a unique path between any two nodes. (2) The graph contains no cycles.

4.12 How many paths are there altogether from Start to Finish in Fig. 4.9? Count them in each of the following ways:

(a) Brute force—write down all the possibilities. This is actually less of a mindless exercise than it appears. It demonstrates that it is really not so difficult to do as it first seems. Second, it shows how you will sense patterns as you do it.

(b) In Fig. 4.9, count the number of paths from Start to A and from A to Finish. Multiply these numbers together to get the total number of paths from Start to Finish that pass through A. Then count the number of paths from Start to B and from B to Finish. What is the relationship between these numbers? Multiply them together to get the total number of paths from Start to Finish that pass through B. Compute the total number of paths from Start to Finish as the sum of the number of paths from Start to Finish that pass through A, B, C, and D.

(c) Recognize that to go from Start to Finish requires six steps, including exactly three left turns and three right turns (else you won't end up at the right place). Different choices of the order of right and left turns correspond to different

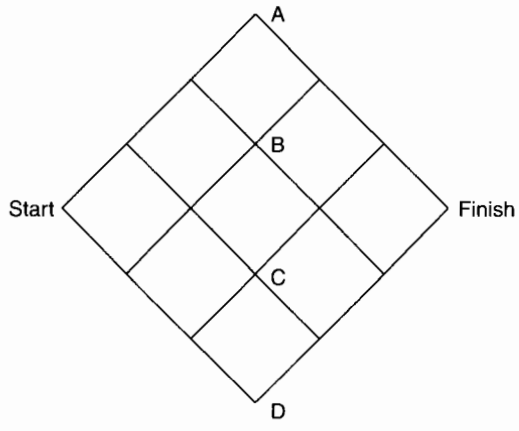


Fig. 4.9 Counting paths on a finite lattice.

paths. To count the number of paths, recognize that you need decide only how many ways there are to choose the three steps at which you turn left (for then you must turn right at the other three steps). To assign three left turns to six steps, first we can choose one of six steps for one left turn, then one of the five remaining steps for the next left turn, and then one of the four remaining steps for the final left turn. However, the product of these numbers overcounts the number of possibilities, because it includes the same sets of steps assigned in different order. Each triple can arise in six different ways, and it is necessary to correct for this. The result is equal to the binomial coefficient $\binom{6}{3} = 6!/(3!3!)$.

4.13 For the final tree in Example 4.7, derive possible ancestors at internal nodes chosen from a maximum parsimony criterion. Are there any ambiguities?

4.14 A convenient notation for trees uses nested parentheses to indicate the clusters. (a) Expand the following into a rooted tree: (A(BC)D). (b) Write the parenthesis notation for the trees shown in Exercise 4.21.

4.15 Add an adequate amount of comments to the PERL program for drawing trees.

4.16 Write a PERL program for the UPGMA method of deriving a phylogenetic tree from a matrix of distances. You may use the program for tree drawing to produce graphical output.

Weblems

4.1 Retrieve the gene sequence of mitochondrial ATPase subunit 6 from Atlantic hagfish (*Myxine glutinosa*). Draw a dotplot against the homologous gene from sea lamprey (*Petromyzon marinus*). Comment on the similarity observed and compare with the similarity between the lamprey and dogfish sequences shown on page 162.

4.2 Submit the amino acid sequence of papaya papain to a BLAST search and to a PSI-BLAST search. Which of the homologues appearing in Fig. 4.2 are successfully detected by BLAST? Which by PSI-BLAST?

4.3 Submit the amino acid sequence of papaya papain to a PSI-BLAST search (see previous weblem). In the results for the match to human procathepsin L, indicate on a photocopy of the dotplot (Fig. 4.2b) the regions of local matches reported.

4.4 Find structures of thioredoxins appearing in the alignment table in Plate III. from organisms other than *E. coli*. On a photocopy of the alignment table, indicate the regions of helix and strands of sheet as assigned in the Protein Data Bank entries, and compare with the helices and strands of *E. coli* thioredoxin.

4.5 Align the amino acid sequence of papaya papain and the homologues shown in Fig. 4.2 using CLUSTAL-W or T-Coffee. Compare the results with the alignment table in Pfam based on Hidden Markov Models, and with the structural alignments in Fig. 4.2.

4.6 Can PSI-BLAST identify the homology between immunoglobulin domains and the domains of *Cellulomonas fimi* endogluconase C and *Streptococcus agalactiae* IgA receptor?

4.7 (a) Can PSI-BLAST identify the relationship between *Klebsiella aerogenes* urease. *Pseudomonas diminuta* phosphotriesterase and mouse adenosine deaminase? (b) Compare the alignments of these three sequences produced by DALI and by CLUSTAL-W or T-Coffee.

4.8 The growth hormones in most mammals have very similar amino acid sequences. (The growth hormones of the Alpaca, Dog, Cat, Horse, Rabbit, and Elephant each differ from that of the Pig at no more than 3 positions out of 191.) Human growth hormone is very different, differing at 62 positions. The evolution of growth hormone accelerated sharply in the line leading to humans. By retrieving and aligning growth hormone sequences from species closely related to humans and our ancestors, determine *where* in the evolutionary tree leading to humans the accelerated evolution of growth hormone took place.

The next series of weblems is designed to place the human species in its biological context by analysis of sequences from near and distant relatives, and to illustrate some of the variety of genetic information that has been used to investigate phylogenetic relationships.

4.9 The living species most closely related to humans are apes and monkeys. Alu elements are a type of SINE (Short INterspersed Element) useful as species markers. Although part of the repetitive noncoding portion of the genome, some Alu elements function in gene regulation. On the basis of Alu elements that regulate the genes for parathyroid hormone, the haematopoietic cell-specific FcεRI-γ receptor, the central nervous system-specific nicotinic acetylcholine receptor α3, and the T-cell-specific CD8α, derive a phylogenetic tree for human, chimpanzee, gorilla, orangutan, baboon, rhesus monkey and macaque monkey.

4.10 Humans are primates, an order that we, apes, and monkeys share with lemurs and tarsiers. On the basis of the β-globin gene cluster of human, a chimpanzee, an old-world monkey, a new-world monkey, a lemur, and a tarsier, derive a phylogenetic tree of these groups.

4.11 Primates are mammals, a class we share with marsupials and monotremes. Extant marsupials live primarily in Australia and neighbouring islands, except for the oppossum, found in North and South America. Extant monotremes are limited to three species: the platypus and two related species of echidna. Collect the nuclear genes for mannose 6-phosphate/insulin-like growth factor II receptor from mammalian species including placentals, marsupials and monotremes. From them draw an evolutionary tree, indicating branch lengths. Are monotremes more closely related to placental mammals or to marsupials?

4.12 Mammals are vertebrates, a subphylum that we share with fishes, sharks, birds and reptiles, amphibia, and primitive jawless fishes (example: lampreys). For the coelacanth (*Latimeria chalumnae*), the great white shark (*Carcharodon carcharias*), skipjack tuna (*Katsuwonus pelamis*), sea lamprey (*Petromyzon marinus*), frog (*Rana pipens*), and Nile crocodile (*Crocodylus niloticus*), using sequences of cytochromes *c* and pancreatic ribonucleases, derive evolutionary trees of these species.

4.13 Tetrapods are gnathostomes, a superclass that we share with fishes. The traditional view of fish → tetrapod evolution is that jawed vertebrates split into one group containing cartilaginous fishes, *Chondrichthyes*, including sharks and rays, and another containing both ray-finned fishes, *Actinopterygii*, including modern bony fishes such as cod and salmon, and lobe-finned fishes, *Sarcopterygii*, including coelacanths, lungfishes and tetrapods (see Fig. 4.10a). Test this hypothesis using at least 12 mitochondrial protein-coding genes from at least 30 species including sharks, lungfish, bony fishes, amphybia, reptiles, birds, and mammals, using a lamprey as an outgroup. Of the three groups—cartilaginous fishes, bony fishes and tetrapods—which pair appears to be most closely related: bony fishes and tetrapods as in the traditional view, or a different pair? Consider specifically a phylogeny according to which the tetrapods split off first, and cartilaginous fishes, bony fishes, and even lungfishes are sister taxa (see Fig. 4.10b).

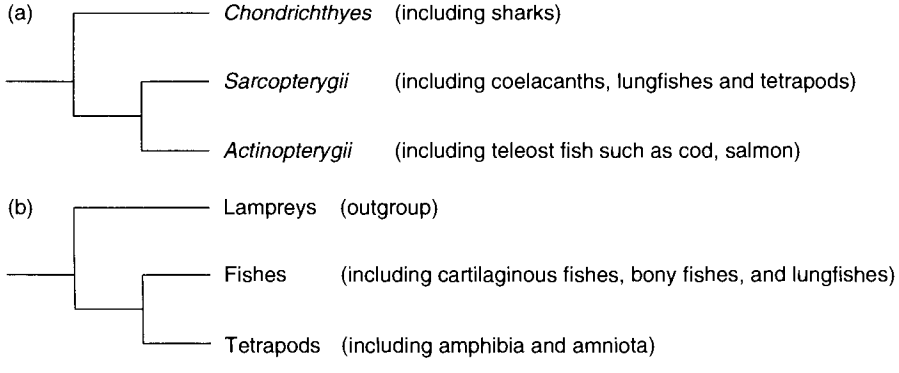


Fig. 4.10 (a) Traditional view of gnathostome evolution: The first split was between cartilaginous fishes such as sharks and others; then tetrapods emerged from a group containing coelecanths, lungfishes and tetrapods split off. An alternative to be considered is that the split leading to the tetrapods was the earliest. (b) Alternative tree, rooted using lamprey as an outgroup, in which the lineage leading to tetrapods split off first.

4.14 Vertebrates are chordates, a phylum that also includes lancelets (small fish-like marine animals; example: amphioxus), and jawless vertebrates (lacking a true vertebral column (example: lamprey). As in other organisms with bilateral symmetry (including insects), vertebrate HOX genes encode a family of DNA-binding proteins. The expression of these genes varies along the head-to-tail body axis, and controls the setting out of the body plan. Indeed there is an amazing mapping between the order of the genes on the chromosome, the order of their action along the body, and the relative times during development of the onset of their activity.

During the course of vertebrate evolution there have been large scale genomic duplications, associated presumably with the development of greater complexity of body architecture, as presciently suggested by S. Ohno in 1970. The genomes of insects and amphioxus have a single HOX cluster. Zebrafish have seven HOX clusters, interpretable in terms of a series of duplications: $1 \rightarrow 2 \rightarrow 4 \rightarrow 8$ followed by loss of one to reduce $8 \rightarrow 7$.

Find the number of HOX clusters in the human and the lamprey, perform a multiple sequence alignment to assign correspondences among the individual genes, and derive from the results a phylogenetic tree for amphioxus, lamprey, fishes, and mammals.

4.15 Chordates are deuterostomes (see Fig. 1.3), a grouping we share with urochordates (example: sea squirts), hemicordates (example: amphioxus), and echinoderms (example: starfish). There are systematic differences between these four phyla in their mitochondrial genetic codes. Determine, for examples of organisms in each phylum, the amino acids that correspond to the codons ATA and AGA. Derive from these results a phylogenetic tree of the four deuterostome phyla.