Bioinformatics

Can Keşmir
Theoretical Biology/Bioinformatics, UU

2013
## Contents

1 Introduction to Systems Biology and Bioinformatics .............................................. 1  
   1.1 Exercises (pen and paper) ................................................................. 5  

2 Sequence Alignment ......................................................................................... 15  
   2.1 Similarity between protein sequences .................................................. 16  
   2.2 Similarity between nucleotide sequences ........................................... 19  
   2.3 Sequence alignment .............................................................................. 20  
   2.4 Multiple Alignments ............................................................................. 24  
   2.5 DNA Alignments .................................................................................. 25  
   2.6 An application: Finding regulatory elements in DNA sequences .......... 26  
   2.7 Summary .............................................................................................. 29  
   2.8 Exercises (pen and paper) ................................................................. 30  

3 Local Alignments and Database Searches ......................................................... 33  
   3.1 Smith-Waterman .................................................................................... 34  
   3.2 Heuristic methods ................................................................................ 35  
   3.3 Expectation Values ............................................................................. 40  
   3.4 Summary .............................................................................................. 41  
   3.5 Exercises (pen and paper) ................................................................. 44  

4 Molecular Evolution and Phylogeny ................................................................ 47  
   4.1 What is a phylogenetic tree? ................................................................. 48  
   4.2 Data selection ...................................................................................... 50  
   4.3 Constructing a phylogenetic tree ......................................................... 52  
   4.4 Horizontal gene transfer ..................................................................... 59  
   4.5 Self-test ............................................................................................... 61  
   4.6 Exercises (pen and paper) ................................................................. 65  

5 Hidden Markov Models and Sequence logos ...................................................... 71  
   5.1 From regular expressions to HMMs ..................................................... 72  
   5.2 Profile HMMs ....................................................................................... 75  
   5.3 Information content in biological sequences ..................................... 78  
   5.4 Information Carried by Biological Sequences ................................... 79  
   5.5 Logo Visualization of Entropy ............................................................. 81  
   5.6 An application ...................................................................................... 81  
   5.7 Exercises (pen and paper) ................................................................. 82  

6 Analysis of gene expression data ..................................................................... 89  
   6.1 DNA microarrays ............................................................................... 89
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2</td>
<td>Normalization of microarray data</td>
<td>91</td>
</tr>
<tr>
<td>6.3</td>
<td>Hierarchical Cluster Analysis</td>
<td>92</td>
</tr>
<tr>
<td>6.4</td>
<td>Visualization of data</td>
<td>94</td>
</tr>
<tr>
<td>6.5</td>
<td>An example</td>
<td>95</td>
</tr>
<tr>
<td>6.6</td>
<td>Exercises (pen and paper)</td>
<td>99</td>
</tr>
</tbody>
</table>

**Practice exam questions** 103

**Answers to the exercises** 111

**Glossary** 129
Chapter 1

Introduction to Systems Biology and Bioinformatics

If you type in “systems biology” in Google, you will notice that the definition of the term has been discussed extensively. This is mainly because the term “systems biology” has emerged only in the last decade, while the research that is meant by the term had been around for much longer under several different names. What many people want to emphasize with this term is the fact that to fully understand the functioning of cellular processes, whole cells, organs, and even organisms, it is not enough to simply assign functions to individual genes, proteins and other cellular components. Instead, we need to analyze the organization and control of the system in an integrated way by looking at the dynamic networks of genes and proteins and their interactions with each other. These pathways are complex dynamic systems, and often behave in a nonlinear way.

This information on its own is not new: the scientists have known for decades (if not for centuries) that the living cells are composed of dynamic networks. What new is the amount of data we have on these dynamic networks. Thanks to the high-throughput technology biological data are being produced at an enormous rate. Many databases are doubling their sizes every 15 months. Complete genomes of many organisms are available today, ranging in size from the smallest known virus, to the human species. We will not even attempt here to make a list of examples where the biological data is blooming, because due to the “big bang” in biology, almost every subfield is expanding rapidly. As genome sequencing projects continue to advance, the emphasis progressively switches from the accumulation of data to its interpretation. Our ability in the future to make new biological discoveries will depend strongly on our ability to combine diverse data sets in such a way that allows information flow and eventually results in better understanding of the biological processes. Sequence data will have to be integrated with structure and function data, with gene expression data, with pathways data, with phenotypic and clinical data, and so forth.

In today’s biology, the challenge is no longer to get the data, but much more on analyzing and integrating the data to understand the biological system as a whole. Therefore, the term systems biology has emerged as a new term for an old concept. This part of the systems biology course will teach you ways to deal with “new” biological data by giving an introduction into Bioinformatics. Bioinformatics was first defined by Hogeweg & Hesper (1978) as the
study of informatic processes in biotic systems. Today the word bioinformatics is often associated only with the application of computational techniques to analyze (molecular) biological data. We will teach you, amongst other things, how to do sequence alignments, database searches and phylogenetic analysis with the ultimate aim that you can critically use such methods whenever you encounter them in your future biological research. In other words, you will learn how to discover many patterns hidden in this enormous amount of data using bioinformatic analysis. We will stress that there are always many different methods for each particular situation, and will focus on how to be able to choose among them.

Protocol: A simple search for Yeast across all NCBI databases using Entrez

1. Start at the NCBI home page: http://www.ncbi.nlm.nih.gov/ and in the text box type Yeast. Click the adjacent Go button, or press 'Enter' on your keyboard. You will now reach Entrez cross-database search page: http://www.ncbi.nlm.nih.gov/gquery/ (see Fig. 1.1). Next to each of the database names and icons in the lower part of the screen will be a number (or, in a few cases 'none'). This indicates the number of entries in each of these databases that contained, anywhere within it, the word 'Yeast'.

2. Many of these records will not relate to yeast itself. For example, some will describe proteins from other species that are noted as interacting with, or being similar to, yeast proteins.

---

1Some protocols are adapted from Dear (2007).
3. We therefore want to limit the search to entries in which the organism is yeast. To do this, repeat the query, this time using the text **Yeast[ORGANISM]**. A list of fields that can be searched is given at [http://www.ncbi.nlm.nih.gov/entrez/query/static/help/Summary_Matrices.html#Search_Fields_and_Qualifiers](http://www.ncbi.nlm.nih.gov/entrez/query/static/help/Summary_Matrices.html#Search_Fields_and_Qualifiers). Many fields can be abbreviated; for example, ‘ORGN’ can be used in place of ‘ORGANISM’.

4. The page will be updated, this time giving the number of entries (in each of the databases) in which ‘Yeast’ is in the ‘Organism’ field.

5. Clicking on any of the results will take you to the respective results page, listing all of the ‘Yeast’ entries found. For example, clicking on **Protein** or the adjacent icon will take you to the start of a list of over 73,000 yeast protein sequence entries.

### Protocol: Searching for protease genes in *P. falciparum*

This search could be started from the Entrez home page (searching all NCBI databases and then selecting those hits from the Gene database). Alternatively, as here, we can navigate to the Entrez Gene page to search only that database.

2. Click on the Gene link or adjacent icon to go to the Entrez Gene page.
3. Into the text box at the top of the screen, type:

   **Plasmodium falciparum[ORGANISM] protease**

![Figure 1.2: Entrez Gene page.](image-url)
4. How many genes did your query return? The chromosome location is given in the summary list. Are all the hits on the same chromosome?

5. Is item PF14_0348 among the genes on your list? Click on the link for PF14_0348 to get a detailed summary of its annotation. Study this new page (see Fig. 1.2). There is a lot of information that is quite easy to understand, while others are more complicated. Click to FASTA link under Genomic regions, transcripts, and products. The output you get is the sequence in FASTA format (see Fig. 1.3).

6. Under Bibliography you will see three articles listed. These are the references that identified this gene. Also on the right hand side of the page, under Links, if you choose for PubMed, you will get the links to the same articles.

7. Which article studied the function of this gene? (tip: look for GenRIFs, Gene Ref Into Function, section)

8. In which other organisms has this gene homologs? (tip: look for Homology section)

Figure 1.3: FASTA Format.

**Obligatory reading**

a. from Campbell & Reece (2008)
   - Chapter 21
   - Review/repeat chapter 20 (Sections 20.2 and 20.4)
   - Review/repeat chapter 5 (Sections 5.4 and 5.5).


**Optional reading**

a. Metzker et al. (2002)
1.1 Exercises (pen and paper)

Self-test Biological Background
(adapted from Higgs & Attwood (2005) and Campbell & Reece (2008))

1. What is the difference between a linkage map and a physical map?
   A) For a linkage map, markers are spaced by recombination frequency, whereas for a physical map they are spaced by numbers of base pairs.
   B) For a physical map, the complete nucleotide sequence of a genome needs to be determined, but not for the linkage map.
   C) For a linkage map, it is shown how each gene is linked to every other gene.
   D) For a physical map, the distances must be calculable in units such as nanometers.
   E) There is no difference between the two except in the type of pictorial representation.

2. How is a physical map of the genome of an organism generated?
   A) using recombination frequency
   B) using very high-powered microscopy
   C) using restriction enzyme cutting sites
   D) using shotgun sequencing
   E) using DNA fingerprinting via electrophoresis

3. Which of the following most correctly describes a shotgun technique for sequencing a genome?
   A) genetic mapping followed by sequencing
   B) physical mapping followed by sequencing
   C) cloning large genome fragments into very large vectors such as YACs, followed by sequencing
   D) cloning fragments of various size into various size vectors followed by sequencing and assembly
   E) cloning the whole genome directly, from one end to the other

4. One of the problems with the shotgun technique is its tendency to underestimate the size of the genome. Which of the following might best account for this?
   A) some of the clones are not sequences at all
   B) some of the overlapping regions of the clones cannot be sequenced
   C) counting some of the overlapping regions of the clones twice
   D) highly similar sequences are sometimes included only a single time in the final assembly

5. What is proteomics?
   A) the linkage of each gene to a particular protein
B) the study of the full protein set encoded by a genome
C) the totality of the functional possibilities of a single protein
D) the study of how amino acids are ordered in a protein
E) the study of how a single gene activates many proteins

6. Fragments of DNA have been extracted from the remnants of extinct wooly mammoths, amplified, and sequenced. These can now be used to
   A) introduce into relatives, such as elephants, certain mammoth traits.
   B) clone live wooly mammoths.
   C) study the relationships among wooly mammoths and other wool-producers.
   D) understand the evolutionary relationships among members of related taxa.
   E) appreciate the reasons why mammoths went extinct

7. Which of the following seems to be the upper and lower size limits of the sequenced genomes (excluding viruses)?
   A) 1 - 2900 Mb (million base pairs)
   B) 1,500- 40,000 Mb
   C) 1 - 150,000 Mb
   D) 100 - 120,000 Mb
   E) 100 - 200,000 Mb

8. Which of the following is a representation of gene density?
   A) Humans have 3,200 Mb per genome.
   B) C. elegans has approximately 20,000 genes.
   C) Humans have approximately 25,000 genes in 3,200 Mb.
   D) Humans have 27,000 bp in introns.
   E) Fritillaria has a genome 40 times the size of a human.

9. Why might the cricket genome have 11 times as many base pairs as that of Drosophila melanogaster?
   A) The two insect species evolved in very different geologic eras.
   B) Crickets have higher gene density.
   C) Drosophila is more complex organism.
   D) Crickets might have more non-coding DNA.

10. What is it about short tandem repeat DNA that makes it useful for DNA fingerprinting?
    A) The number of repeats varies widely from person to person or animal to animal.
    B) The sequence of DNA that is repeated varies significantly from individual to individual.
    C) The sequence variation is acted upon differently by natural selection in different environments.
    D) Every racial and ethnic group has inherited different short tandem repeats.
11. How might identical and obviously duplicated gene sequences have gotten from one chromosome to another?
   A) by normal meiotic recombination
   B) by normal mitotic recombination between sister chromatids
   C) by transcription followed by recombination
   D) by chromosomal translocation
   E) by deletion followed by insertion

12. Several of the different globin genes are expressed in humans, but at different times in development. What mechanism could allow for this?
   A) exon shuffling
   B) intron activation
   C) pseudogene activation
   D) differential translation of mRNAs
   E) differential gene regulation over time

13. What is it that can be duplicated in a genome?
   A) DNA sequences above a minimum size only
   B) DNA sequences below a maximum size only
   C) entire chromosomes only
   D) entire sets of chromosomes only
   E) sequences, chromosomes, or sets of chromosomes

14. In comparing the genomes of humans and those of other higher primates, it is seen that humans have a large metacentric pair we call chromosome #2 among our 46 chromosomes, while the other primates of this group have 48 chromosomes and any pair like the human #2 pair is not present; instead the primate groups each have two pairs of midsizeacrocentric chromosomes. What is the most likely explanation?
   A) The ancestral organism had 48 chromosomes and at some point a centric fusion event occurred.
   B) The ancestral organism had 46 chromosomes, but primates evolved when one of the pairs broke in half.
   C) At some point in evolution, human ancestors and primate ancestors were able to mate and produce fertile offspring, making a new species.
   D) Chromosome breakage results in additional centromeres being made in order for meiosis to proceed successfully.
   E) Transposable elements transferred significantly large segments of the chromosomes to new locations.

15. In order to determine the probable function of a particular sequence of DNA in humans, what might be the most reasonable approach?
   A) Prepare a knockout mouse without a copy of this sequence and examine the mouse phenotype.
B) Genetically engineer a mouse with a copy of this sequence and examine its phenotype.

C) Look for a reasonably identical sequence in another species, prepare a knockout of this sequence in that species and look for the consequences.

D) Prepare a genetically engineered bacterial culture with the sequence inserted and assess which new protein is synthesized.

E) Mate two individuals heterozygous for the normal and mutated sequences.

16. Bioinformatics includes all of the following except

A) using computer programs to align DNA sequences.

B) analyzing protein interactions in a species.

C) using molecular biological techniques to express specific gene products.

D) development of computer-based tools for genome analysis.

E) use of mathematical tools to make sense of biological systems.

17. Which of the following has the largest genome and the fewest genes per million base pairs?

A) Haemophilus influenzae (bacterium)

B) Saccharomyces cerevisiae (yeast)

C) Arabidopsis thaliana (plant)

D) Drosophila melanogaster (fruit fly)

E) Homo sapiens (human)

18. Multigene families are

A) groups of enhancers that control transcription.

B) usually clustered at the telomeres.

C) equivalent to the operons of prokaryotes.

D) sets of genes that are coordinately controlled.

E) identical or similar genes that have evolved by gene duplication.

19. Two eukaryotic proteins have one domain in common but are otherwise very different. Which of the following processes is most likely to have contributed to this phenomenon?

A) gene duplication

B) RNA splicing

C) exon shuffling

D) histone modification

E) random point mutations

20. There are 20 different amino acids. What makes one amino acid different from another?

A) different carboxyl groups attached to an alpha carbon

B) different amino groups attached to an alpha carbon

C) different side chains (R groups) attached to an alpha carbon
21. How many different kinds of polypeptides, each composed of 12 amino acids, could be synthesized using the 20 common amino acids?
   A) 412
   B) $12^{20}$
   C) 125
   D) 20
   E) $20^{12}$

22. The alpha helix and the beta pleated sheet are both common polypeptide forms found in which level of protein structure?
   A) primary
   B) secondary
   C) tertiary
   D) quaternary
   E) all of the above

23. The tertiary structure of a protein is the
   A) bonding together of several polypeptide chains by weak bonds.
   B) order in which amino acids are joined in a polypeptide chain.
   C) unique three-dimensional shape of the fully folded polypeptide.
   D) organization of a polypeptide chain into an alpha helix or beta pleated sheet.
   E) overall protein structure resulting from the aggregation of two or more polypeptide subunits.

24. What would be a consequence of changing one amino acid in a protein consisting of 325 amino acids?
   A) The primary structure of the protein would be changed.
   B) The tertiary structure of the protein might be changed.
   C) The biological activity or function of the protein might be altered.
   D) Only A and C are correct.
   E) A, B, and C are correct.

25. Altering which of the following levels of structural organization could change the function of a protein?
   A) primary
   B) secondary
   C) tertiary
   D) quaternary
   E) all of the above

26. Of the following functions, the major purpose of RNA is to
A) transmit genetic information to offspring.
B) function in the synthesis of protein.
C) make a copy of itself, thus ensuring genetic continuity.
D) act as a pattern or blueprint to form DNA.
E) form the genes of higher organisms.

27. Which of the following best describes the flow of information in eukaryotic cells?
   A) DNA → RNA → proteins
   B) RNA → proteins → DNA
   C) proteins → DNA → RNA
   D) RNA → DNA → proteins
   E) DNA → proteins → RNA

28. Which of the following statements best summarizes the structural differences between DNA and RNA?
   A) RNA is a protein, whereas DNA is a nucleic acid.
   B) DNA is a protein, whereas RNA is a nucleic acid.
   C) DNA nucleotides contain a different sugar than RNA nucleotides.
   D) RNA is a double helix, but DNA is single-stranded.
   E) A and D are correct.

29. If one strand of a DNA molecule has the sequence of bases 5’ATTGCA3’, the other complementary strand would have the sequence
   A) 5’TACCGT3’.
   B) 3’TACCGT5’.
   C) 5’UAACGU3’.
   D) 3’UAACGU5’.
   E) 5’UGCAAU3’.

30. A new organism is discovered in the forests of Costa Rica. Scientists there determine that the polypeptide sequence of hemoglobin from the new organism has 72 amino acid differences from humans, 65 differences from a gibbon, 49 differences from a rat, and 5 differences from a frog. These data suggest that the new organism
   A) is more closely related to humans than to frogs.
   B) is more closely related to frogs than to humans.
   C) may have evolved from gibbons but not rats.
   D) is more closely related to humans than to rats.
   E) may have evolved from rats but not from humans and gibbons.

31. Enzymes are
   A) carbohydrates.
   B) lipids.
C) proteins.
D) nucleic acids.

32. DNA microarrays have made a huge impact on genomic studies because they
   A) can be used to identify the function of any gene in the genome.
   B) can be used to introduce entire genomes into bacterial cells.
   C) allow the expression of many or even all of the genes in the genome to be compared at once.
   D) allow physical maps of the genome to be assembled in a very short time.
   E) dramatically enhance the efficiency of restriction enzymes.

33. Which of the following peptides would have the largest positive charge in a solution at neutral pH (use Fig. 1.4)?
   A) LYAIRT
   B) RTKPLH
   C) VEMDAS
   D) PHRYLD

34. Consider the following DNA oligomers. Which two are complementary to one another? All are written in the 5’ to 3’ direction. (i) TTAGGC (ii) CGGATT (iii) AATCCG (iv) CCGAAT
   A) (i) and (ii)
   B) (ii) and (iii)
   C) (i) and (iii)
   D) (ii) and (iv)

35. Which of the following statements about transcription is correct?
   A) Transcription is initiated at a start codon.
   B) Transcription is carried out by aminoacyl-tRNA synthetases.
   C) Eukaryotic RNA sequences must be spliced prior to transcription.
   D) Transcription involves the complementary pairing of a DNA strand and an RNA strand.

36. Which of the following components of a cell does not contain RNA?
   A) The nucleus
   B) The ribosome
   C) The spliceosome
   D) The cell membrane

37. Which of the following statements about ribosomes is correct?
   A) During translation, a ribosome binds to a messenger RNA near its 5’ end.
   B) Ribosomes are essential for DNA replication.
C) Ribosomes perform splicing in eukaryotes.
D) Inside a ribosome there is one tRNA for each type of amino acid.

38. Which of the following statements about the genetic code is correct (use Fig. 1.5)?

A) Cysteine and tryptophan have only one codon in the standard genetic code.
B) Serine and arginine both have 6 codons in the standard genetic code.
C) The fMet tRNA is a special tRNA that binds to the UGA stop codon.
D) All of the above.

39. Which of the following statements about polymerases is correct?

A) RNA polymerase has a much lower error rate than DNA polymerase due to proof-reading.
B) DNA polymerases move along an existing DNA strand in the 5’ to 3’ direction.
C) None of the above.

**Question 1.1. Evolution**

The average rate of change of DNA is approximately $10^{-9}$ per base pairs per year. Consider a piece of DNA region that is approximately 1000 base pairs, and two species that diverged from each other 10 million years ago:

a. which fraction of sites should approximately differ between these sequences today (keep this calculation as simple as possible)?

b. If some sites in this specific region were more difficult to mutate than others, would the fraction of sites that differ increase or decrease?

c. what is the probability that one sequence (1000 bp) remains unchanged in one year?
1.1 Exercises (pen and paper)

d. what is the probability that one sequence will stay unchanged in $10^7$ years (use your answer to the previous question)?
e. what is the probability that both sequences are still the same after $10^7$ years?

**Question 1.2. DNA sequences**

Assume all nucleotides occur with same frequency and independent of each other in DNA sequences.

a. How frequently would you find the nucleotide sequence GGATATCCGC (5' → 3' direction) by chance in a DNA molecule?
b. On average, how many times do you expect to find a specific 20 nucleotide sequence in a genome with a total size of $4 \times 10^9$ base pairs?

<table>
<thead>
<tr>
<th>first position</th>
<th>second position</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>U</strong></td>
<td>UUU Phe (F)</td>
</tr>
<tr>
<td></td>
<td>UUC Phe (F)</td>
</tr>
<tr>
<td></td>
<td>UUA Leu (I)</td>
</tr>
<tr>
<td></td>
<td>UUG Leu (I)</td>
</tr>
<tr>
<td><strong>C</strong></td>
<td>CUA Leu (I)</td>
</tr>
<tr>
<td></td>
<td>CUC Leu (I)</td>
</tr>
<tr>
<td></td>
<td>CUA Leu (I)</td>
</tr>
<tr>
<td></td>
<td>CUG Leu (I)</td>
</tr>
<tr>
<td><strong>A</strong></td>
<td>AUA Ile (I)</td>
</tr>
<tr>
<td></td>
<td>AUG Ile (I)</td>
</tr>
<tr>
<td></td>
<td>AUA Ile (I)</td>
</tr>
<tr>
<td></td>
<td>AUG Met (N)</td>
</tr>
<tr>
<td><strong>G</strong></td>
<td>GGU Val (V)</td>
</tr>
<tr>
<td></td>
<td>GUC Val (V)</td>
</tr>
<tr>
<td></td>
<td>GUA Val (V)</td>
</tr>
<tr>
<td></td>
<td>GUC Val (V)</td>
</tr>
</tbody>
</table>

Figure 1.5: Genetic code
Chapter 2

Sequence Alignment

The concept of protein families is based on the observation that, while there are a huge number of different proteins, most of them can be grouped on the basis of similarities in their sequences into a limited number of “families”. Proteins or protein domains belonging to a particular family generally share functional attributes, are derived from a common ancestor, and will most often be the result of gene duplication (i.e. making a copy of a gene that was already present).

Gene duplication is an obvious way for genomes to acquire new genes. During DNA replication duplications of DNA regions can occur. Sometimes these contain a whole gene and as a result the genome will contain two copies of a gene after the replication event. Since both gene sequences are subject to genetic events like mutations, insertions and deletions, the two sequences will no longer be identical after a while. Most of the time one of the genes accumulates deleterious mutations and would not be kept in the genome. Occasionally, one of the two copies acquires a new function and both genes are kept in the genome. Also mutations in the upstream or downstream regions can cause differential regulation without actually changing the function of the gene. Such homolog genes in one genome are called paralogs.

Similarly, orthologs are genes in different species that diverged from a common ancestral sequence as a result of the speciation. We are very much interested in finding orthologs of a gene, because orthology often suggests the same or a similar function. One of the most challenging questions in bioinformatics is “function prediction”. Although the human genome has been sequenced for some time now, we still do not know the function of the majority of the genes. By finding their orthologs in other species one can get a first clue of their function (obviously, if the function of the ortholog is known).

Having emphasized on how much information one can extract from homolog sequences, let us now move into the first step in searching for homology, namely the methods of measuring similarity between biological sequences.
Figure 2.1: Dotplot showing similarities between a short name (Dorothy Hodgkin) and a full name (Dorothy Crowfoot Hodgkin). Path shown with arrows is the best alignment. Figure is taken from Lesk (2002)

2.1 Similarity between protein sequences

The simplest similarity scoring is the relative amount of identical entities, also called % identity, or %ID. **Dotplot** method is one of the simplest ways to compare sequence similarity graphically, and is based on %ID. The dotplot is a simple table or matrix, where the rows correspond to the residues of one sequence and the columns to the residues of the other sequence. The positions are left blank if the two residues are different and filled if they match. Stretches of similar sequences show up as diagonals. Fig. 2.1 demonstrates how a dotplot is generated. A dotplot gives a quick view of a relation between two sequences. Similarity, if it exists, is rather easy to catch in such a plot. Moreover, one can also pinpoint which regions of the sequences are most similar. Often the similarity appears in parallel diagonals (as in Fig. 2.1). The parallel shifts indicate that there are insertions/deletions. One can filter the noise in a dotplot by not showing any “dots” unless they are within a region where both sequences are rather identical. Dotplots can also be used to find repeats within one sequence: by using the same sequence for the rows and the columns of the dotplot, one can detect intrasequence repeats in either proteins or nucleic acids.

This simple approach is actually too simple because amino acids share many physical-chemical properties, and similar amino acids can more easily be exchanged with each other than with unrelated amino acids. So to use the easiest scoring (substitution) matrix, **identity matrix**, e.g. where all the matches are scored as 1 and mismatches as $-1$ (or 0), is not satisfactory for many cases. Thus a scoring system that treats different substitutions differently is a much better approach. The most useful concept has been to estimate how often a given amino acid is exchanged for another in already aligned similar sequences.

**Scoring (Substitution) Matrices**

Dayhoff *et al.* (1978) calculated the original **percentage accepted mutations** (PAM) matrices using a database of amino acid changes in groups of closely related proteins. From these changes they derived the accepted “rates” of mutations. Each change was entered into a matrix listing all the possible amino acid changes. Each entry in the matrix gives the “relative mutability” of all amino acids, *i.e.* how often a given amino acid is changed to one other. The
information about the individual mutations, and about the relative mutability of the amino acids were then combined into one (symmetric) "mutation probability matrix."

The rows and columns of this matrix represent amino acid substitution pairs, i.e. the probability that the amino acid of the column will be replaced by the amino acid of the row after a given evolutionary interval. A matrix with an evolutionary distance of 1 PAM would correspond to roughly 1% divergence in a protein (one amino acid replacement per hundred amino acids). PAM 1 matrix would have numbers very close to 1 in the main diagonal, and small off diagonal numbers. A matrix with an evolutionary distance of 0 PAMs would have only ones on the main diagonal and zeros elsewhere. Assuming that proteins diverge as a result of accumulated and uncorrelated mutations, a mutational probability matrix for a protein sequence that has undergone N percent accepted mutations, i.e. a PAM-N matrix, can be derived by multiplying the PAM-1 matrix by itself N times. The result is a whole family of scoring matrices. A PAM250 matrix, which corresponds to an evolutionary distance of 250 substitutions per hundred residues (each residue can change more than once), works well for general sequence similarity calculations. Remember that not all the residues of a sequence have to change at this evolutionary distance: if some positions get substitutions very frequently, they will change several times, while other positions, e.g. the ones that are crucial for the function, will not change.

The probabilities of substitutions can be very small numbers. To avoid working with these very small numbers, one actually uses so-called “log-odds” in sequence comparisons (see Chapter 5). The odds matrix is constructed by taking the elements of the probability matrix and dividing each component by the frequency of the replacement residue. In this way each component gives the odds of replacing a given amino acid with another specified amino acid. Finally, the log of this ratio is used as the weights in the matrix. The PAM250 matrix is shown in Fig. 2.2.

The blocks substitution matrix (BLOSUM) matrix, described by Henikoff & Henikoff (1992), is another widely used amino acid substitution matrix. To calculate the BLOSUM matrix only very related blocks of amino acid sequences (conserved blocks) were considered. Originally these were taken from the BLOCKS database of pre-aligned sequence families (Henikoff &
Sequence Alignment

Henikoff, 1991). For each block the number of amino acids in each position of the sequence is summed to get a frequency table of how often different pairs of amino acids occur in these conserved regions, i.e. this frequency table gives the likelihood of accepted (tolerated) mutations. A pair of amino acids (with frequencies $p_i$ and $p_j$) should occur with the expected frequency $e_{ij} = p_i^2$ if $i = j$ and $e_{ij} = 2p_ip_j$ if $i \neq j$ because there are two ways in which one can select two different residues. Every value in BLOSUM matrices is calculated as $s_{ij} = \log_2(q_{ij}/e_{ij})$, where $q_{ij}$ is the observed frequency of the pair $ij$. One can also use $\ln$ instead of $\log_2$ to calculate $s_{ij}$; as long as it is consistent the nature of the logarithms used does not matter. If the observed number of differences between a pair of amino acids is equal to the expected number then $s_{ij} = 0$. If the observed frequency is less than expected then $s_{ij} < 0$ and if the observed frequency is greater than expected $s_{ij} > 0$.

One of the key aspects in generating BLOSUM matrices is to weight the sequences to try to reduce any bias in the data. This is necessary because the sequence databases are highly biased toward certain species and types of proteins, which means that there are many very similar sequences present. The weighting involves clustering the most similar sequences together, and different matrices are produced according to a threshold, $C$, used for this clustering. Sequences are clustered together if they have $\geq C\%$ identity, and the substitution statistics are calculated only between clusters, not within them. Weights are determined according to the number of sequences in the cluster. For a cluster of $N$ sequences, each sequence is assigned a weight of $1/N$. The weighting scheme was used to obtain a series of substitution matrices by varying the value of $C$, with the matrices named as BLOSUM-62, for example, in the case where $C = 62\%$. The BLOSUM62 matrix is shown in Fig. 2.2.

Choice of Substitution Matrices depends on the problem

With many substitution matrices available, it is hard to know which one to use. Since the initial PAM1 matrix was made from very similar sequences (they could fall all in a single cluster for $C \leq 85\%$), the evolutionary distances between those are very short, and most changes captured will be single base mutations leading to particular types of amino acid substitutions, e.g. substitutions requiring more than one base mutation will be very rare. Even the multiplications done to expand this matrix to longer evolution time cannot compensate for this (Gonnet et al., 1992) and therefore the BLOSUM matrices perform better when used for the alignment of relatively diverged sequences. The matrices are in a format where you can sum up the scores for each position in the sequence to obtain a total alignment score, and the alignment resulting in the highest score is then considered to be optimal.

BLOSUM matrices with higher numbers and PAM matrices with low numbers are designed for comparisons of closely related sequences. BLOSUM matrices with low numbers and PAM matrices with high numbers are designed for comparisons of distantly related proteins (see Fig. 2.3).

How to use scoring matrices: an example

It is rather complicated to understand how these matrices are made, however, to use them in proper context, you need to understand the differences between them. Once we have these
2.2 Similarity between nucleotide sequences

Imagine we have the following two short sequences:

\[
\begin{align*}
\text{KAWSADV} \\
\text{KDWSAEV}
\end{align*}
\]

Using the identity matrix (e.g. match=1 and mismatch=-1), the similarity between these two sequences would be \(1 - 1 + 1 + 1 + 1 - 1 + 1 = 3\). Using the BLOSUM62 matrix, the similarity becomes:

\[
\]

Here \(S(i, j)\) values come from the BLOSUM62 matrix given in Fig. 2.2. If we now take \(S(i, j)\) values from PAM250 matrix (also in Fig. 2.2) the similarity of the sequences becomes:

\[
5 + 0 + 17 + 2 + 2 + 3 + 4 = 33.
\]

These values allow us to decide which sequences are closer to each other. Obviously, this closeness depends on which scoring matrices we use. Take a third sequence KYWSAEV. According to the BLOSUM62 scoring matrix, the distance between KAWSADV and KYWSAEV is 28, i.e. the third sequence KYWSAEV is as similar to KAWSADV as KDWSAEV. However, with the PAM250 matrix, the similarity between KAWSADV and KYWSAEV is 30, and KAWSADV is less similar to KYWSAEV than to KDWSAEV.

2.2 Similarity between nucleotide sequences

Let us start very simple: Let \(D\) be the number of non-matching sites divided by the total number of sites in a sequence. The distance between two nucleotide sequences \((d)\) can simply be \(D\). However this is not a sufficient measure in most cases. As was the case for amino acids, some nucleotide substitutions are more likely to occur than others. For example, two sequences that differ by an A and G do not have the same quality of difference as two sequences that differ by an A and a T. The former substitution is a transition (a purine becomes another purine), which happens frequently, while the latter is a transversion (a purine becomes pyrimidine) which occurs far less frequently. Hence it would be desirable to treat these substitutions in a different fashion.

There are also subtler problems. Assume for the moment that all mutations occur with equal frequency. You might think that the difference between two sequences is calculated simply by counting the number of nucleotide differences between the sequences. This is true only initially: the initial rate of change is approximately twice the product of the mutation rate and time, because both sequences are diverging from a common ancestor. When the time of divergence between two sequences increases, the probability of a secondary substitution at any one nucleotide site that has already changed also increases. Thus the number of sites that differ between two sequences, \(D\), increases more slowly than expected. This makes \(D\) an
undesirable measure of distance. In Chapter 4 we will summarize possible solutions to this problem.

### 2.3 Sequence alignment

Within a protein sequence some regions are more conserved than others during evolution. These regions are generally important for the function of a protein and/or the maintenance of its three-dimensional structure, or other features related to its localization or modification. By analyzing constant and variable properties of such groups of similar sequences, it is possible to derive a signature for a protein family or domain, which distinguishes its members from other unrelated proteins. Alignment of sequences (as exemplified in Fig. 2.5) allows us to discover these signatures.

Alignment is the task of locating equivalent regions of two or more sequences to maximize their similarity. Sequence alignment is the oldest and probably the single most important tool in bioinformatics. Although one of the basic techniques within sequence analysis, alignment is far from simple, and the analytic tools are still not perfect. Furthermore, the question of which method is optimal in a given situation strongly depends on which question we want to answer. The most common questions are: How similar (or different) are these groups of sequences, and which sequences in a database are similar to a specific query sequence. The reasoning behind the questions might, however, be important for the choice of the algorithmic solution. Why do we want to know this? Are we searching for the function of a protein/gene, or do we want to obtain an estimate of the evolutionary history of the protein family? Issues like the size of database to search, and available computational resources also influence our selection of an alignment tool.

From the early days of protein and DNA sequencing it was clear that sequences from highly related species were highly similar, but not necessarily identical. Aligning closely related sequences is a trivial task and can be done even manually (see e.g. Fig. 2.4A). In cases where genes are of different sizes and the similarity is low, alignments become more difficult to construct.

### Gap Penalties

Mutations between different types of nucleotides or amino acids are not the only changes that appear in sequences during evolution. The sequences can also lose or gain nucleotides (deletions or insertions, respectively). This also affects how similar two sequences are. For example, the alignments in Fig. 2.4 reveal that there are two parts of the human proteasomal subunit with a high number of identical amino acids, but without inserting or deleting letters in one of the sequences they cannot be aligned simultaneously. Using the identity matrix only the first identical part is aligned (in Fig. 2.4B given with lower case letters) and with BLOSUM30, both identical parts are aligned (in Fig. 2.4C given with lower case letters). This leads obviously to the necessity of inserting gaps in the alignments.

A gap in one sequence represents an insertion in the other sequence. First, to avoid having
2.3 Sequence Alignment

(A)

<table>
<thead>
<tr>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>huDSS1</td>
<td>MSEKQPVdGLLLEEDDEFFEPASEDPWAGLDEDEDADHVWEDNWDDONVEDDFSQNLRAELEKIGYKMTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZebFDSS1</td>
<td>MSEKQTVdGLLLEEDDEFFEPASEDPWAGLDEDEDADHVWEDNWDDONVEDDFSQNLRAELEKIGYKMTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(B)

<table>
<thead>
<tr>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>huDSS1</td>
<td>MSEKQPVdGLLLEEDDEFFEPASEDPWAGLDEDEDADHVWEDNWDDONVEDDFSQNLRAELEKIGYKMTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AnophDSS1</td>
<td>MSDKENKDKPKLdGLLLEEDDEFFEPASEDPWAGLDEDEDADHVWEDNWDDONVEDDFSQNLRAELEKIGYKMTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(C)

<table>
<thead>
<tr>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>huDSS1</td>
<td>MSEKQPVdGLLLEEDDEFFEPASEDPWAGLDEDEDADHVWEDNWDDONVEDDFSQNLRAELEKIGYKMTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AnophDSS1</td>
<td>MSDKENKDKPKLdGLLLEEDDEFFEPASEDPWAGLDEDEDADHVWEDNWDDONVEDDFSQNLRAELEKIGYKMTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.4: The human proteasomal DSS1 subunit aligned against the zebra fish homolog (A) or mosquito homolog (B) using the identity matrix. “:” means a match, ‘.’ is a mismatch. (C) shows the optimal alignment with the mosquito homolog using BLOSUM30 matrix. Here ‘.’ means a mismatch, however among similar amino acids. No symbol implies mismatch with dis-similar amino acids. Lower case letters show the regions that are aligned correctly (see main text).

Figure 2.5: Figure 26.8 in Campbell & Reece (2008).

gaps all over the alignment, gaps have to be given penalties, just like unmatching amino acids. This penalty (i.e. the probability that a given amino acid will be deleted in another
related sequence) cannot be derived from the database alignments used to create the PAM and BLOSUM matrices, since these matrices were derived from ungapped alignments. Instead, a general gap insertion penalty is determined, usually empirically. Having only one score for any gap inserted is called a “linear gap” cost, and will lead to the same total penalty for three single gaps at three different positions in the alignment as having a single stretch of three gaps. This does not make sense biologically, however, since insertions and deletions often involve a longer stretch of DNA in a single event. For this reason two different gap penalties are usually included in the alignment algorithms: one penalty for having a gap at all (gap opening penalty), and another, smaller penalty, for extending already opened gaps. This is called an “affine gap penalty”, and is actually a compromise between the assumption that the insertion, or deletion, is created by one or more events. Furthermore, it is possible to give no penalty for gaps appended at the ends of the sequences, since insertions at the ends will have a much greater chance of not disrupting the function of a protein. For a more detailed discussion on how to set gap penalties, see Vingron & Waterman (1994).

Alignment by Dynamic Programming

Introducing gaps greatly increases the number of different comparisons between two sequences and in the general case it is impossible to do them all. To compensate for that, several shortcut optimization schemes have been invented. One of the earliest schemes was developed by Needleman & Wunsch (1970) and works for global alignments, i.e. alignments covering all residues in both sequences.

The general idea of the algorithm is the following: Assume we have two sequences with length $N_1$ and $N_2$ that we want to align. For any $0 < i \leq N_1$ and $0 < j \leq N_2$, we wish to calculate the score for partial alignment ending in $a_i$ and $b_j$. There are only three things that can happen: i) $a_i$ and $b_j$ are aligned with each other, ii) $a_i$ is aligned with a gap, or iii) $b_j$ is aligned with a gap. Let $S(a_i, b_j)$ be the similarity of $a_i$ and $b_j$, e.g. from the PAM250 matrix. The highest alignment score, $H(i, j)$ then becomes:

$$H(i, j) = \max \left\{ \begin{array}{ll}
H(i-1, j-1) + S(a_i, b_j) & \text{diagonal} \\
H(i, j-1) - g & \text{horizontal} \\
H(i-1, j) - g & \text{vertical} 
\end{array} \right. \quad (2.1)$$

where $g$ is the gap penalty and $H(0,0) = 0$.

Let us explain how the algorithm works with an example, where we align two very short sequence stretches, SPEARE and SHAKE, using PAM250 matrix given in Fig. 2.2. Let the gap penalty $g = 6$. We need to calculate the $H$ matrix given by Eq. 2.1. The steps we need to follow are:
2.3 Sequence alignment

Fill in $S(i,j)$ values from the PAM250 matrix, to obtain the matrix given on the left. This will help us to calculate the $H$ matrix because we need $S(a_i, b_j)$ values in Eq. 2.1. This matrix shows only the pairwise amino acid scores.

Draw the $H$ matrix by putting one sequence on the rows and the other on the columns.

Set $H(0,0) = 0$ and give all the cells at the edge of the $H$ matrix will have scores that are multiples of $-6$, which is the gap penalty $g$.

At $H(1,1)$, we have three options: i) diagonal: $0 + S(S,S) = 2$, ii) horizontal: there will be a gap introduced in the beginning of SHAKE sequence, $-6 + (-6) = -12$ iii) vertical: there will be a gap introduced in the beginning of SPEAR sequence $-6 + (-6) = -12$. We choose the maximum, 2. We draw a diagonal arrow from this cell to indicate that the best option was to move diagonally.

Now let us find $H(1,2)$. Note that $S(S,P) = 1$. Our options are: i) diagonal: $-6 + 1 = -5$, ii) horizontal: $2 - 6 = -4$ and iii) vertical: $-12 - 6 = -18$. We choose for the second possibility, and $H(1,2) = -4$. Since we had to introduce a gap, we draw a horizontal arrow to indicate this.
Using Eq. 2.1 we can fill up the rest of $H$ matrix (Fig. 2.6). At any field, we will finally have a score. This score is the maximal alignment score you can get coming from the upper left diagonal and to the position in the sequences matching that field. Sometimes there are two possibilities because different options have the same score.

When the matrix is filled out completely, the final alignment score is in the lower right corner of the score matrix. In the above example the final alignment score is then 6. The alignment is reconstructed using the traces we kept. To reconstruct the alignment, start in the lower right corner of the final trace matrix (Fig. 2.6). Following the directions written in the fields, the alignment is now reconstructed backward. Here *diagonal arrow* means a match between the two last residues in each sequence (E match E), and a move diagonal up-left. Similarly, an arrow up or right means introducing gaps. The resulting alignment will be:

```
S-HAKE
SPEARE
```

This way to produce an alignment is called dynamic programming, and is used in major alignment software packages (*e.g.* the ALIGN tool in the FASTA package uses the Needleman-Wunsch algorithm for global alignments). Try now to do the same alignment using BLOSUM62 matrix, are the two alignments different? What happens if gap penalty is lower/higher? The optimal alignment is only optimal for the chosen substitution scores and gap penalties, and there is no exact way to tell in a particular example if one set of scores gives a more “correct” alignment than another set of scores.

### 2.4 Multiple Alignments

So far we discussed only pairwise alignment methods. However, we often want to align a set of sequences, because multiple alignments of protein sequences are important tools in studying proteins. Multiple alignments allow for identification of conserved sequence regions. This is very useful in designing experiments to test and modify the function of specific proteins, in predicting the function and structure of proteins, and in identifying new members of protein families.
Conceptually, there is no reason why a Needleman-Wunsch algorithm cannot be performed with more than two sequences. The matrix simply becomes multi-dimensional and the algorithm would work successively through each dimension. However, it becomes computationally very expensive when the number of sequences increases: For more than eight proteins of average length, the problem is uncomputable given current computer power. Currently, the most widely used approach is to exploit the fact that homologous sequences are evolutionarily related (first introduced by Hogeweg & Hesper (1984)). One can build up a multiple alignment progressively by a series of pairwise alignments, following the branching order in a phylogenetic tree (see Chapter 4 for this). First we align the most closely related sequences, gradually adding in the more distant ones. This approach is sufficiently fast to allow alignments of virtually any size. Further, in simple cases, the quality of the alignments is excellent, as judged by the ability to correctly align corresponding domains from sequences of known secondary or tertiary structure. In more difficult cases, the alignments give good starting points for further automatic or manual refinement.

ClustalW (Thompson et al., 1994) is probably the most popular multiple alignment program at the moment and it uses this progressive alignment principle. ClustalW consists of three main steps:

a. all pairs of sequences are aligned separately in order to calculate a distance matrix giving the distance between each pair of sequences.

b. a guide tree (phylogenetic trees are the subject of Chapter 4, so at the moment I am only mentioning it!) is calculated from the distance matrix;

c. the sequences are progressively aligned according to the branching order in the guide tree. The basic procedure at this stage is to align larger and larger groups of sequences, i.e. larger alignments should be made by aligning two alignments. With a sequence-to-sequence alignment, a similarity matrix such as BLOSUM62 is used to obtain a score for a particular substitution between the pairs of aligned residues. In aligning alignments, however, each of the two input alignments are treated as a single sequence, and you calculate the score at aligned positions as the average substitution matrix score of all the residues in one alignment vs. all those in the other. If you have two alignments with \( m \) and \( n \) sequences in each, the score at any position is the average of all the \( m \times n \) scores of the residues compared separately. For example, the score for aligning P/A with I/R would be \( (S(P, I) + S(P, R) + S(A, I) + S(A, R))/4 \), where \( S \) is the score. Any gaps that are introduced are placed in all of the sequences of an alignment at the same position.

All gaps in the ends of the sequences are free of penalties. This might give some artifacts, especially when sequences of different length are aligned.

### 2.5 DNA Alignments

Until now only protein alignments have been described. The basic algorithms and programs used for DNA alignment are the same as for proteins. DNA sequences are more difficult to align since at each position, we can have one of only four different bases as opposed to one of twenty in peptide alignments. So we will not have a specific substitution matrix like BLOSUM or PAM but rather take a step back and use a general substitution score for any match or mismatch but still using affine gap penalties. This makes the probability of any
Sequence Alignment

given substitution equally high, and thus the significance of the final alignment will be lower. Some nucleotide matrices, however, do have different substitution scores for transitions and transversions. Dealing with DNA/RNA sequences from coding regions, gives an opportunity to shortcut the alignment by actually aligning the translation products, rather than the actual DNA sequences. This approach has been implemented in most alignment software packages. In this basic, but strong approach, gaps in the aligned DNA sequences will only occur in multiples of triplets. This will, however, not catch events causing frame-shifts, leading to major changes of larger or smaller parts of the translated protein. For investigating possible frame-shifts the programs GenA1 (Hein & Støvlbaek, 1994, 1996) and COMBAT (Pedersen et al., 1998) can be used, however, they can only perform pairwise alignments of DNA sequences.

Multiple DNA alignments are especially useful for investigating the molecular evolution. Such alignments allow us to examine exactly which positions in the DNA are more or less likely to undergo mutations that survive and are transferred to the progeny. We can also calculate the probability that a given codon has mutations that will not lead to an amino acid change (silent mutations or synonymous mutations), and compare it to the frequency of substitutions leading to an amino acid change (nonsynonymous mutations). This is often called the $dn/ds$ ratio and it is used to determine selection pressures on biological sequences.

2.6 An application: Finding regulatory elements in DNA sequences

Throughout this chapter we have been focusing on genes and proteins. However, especially in eukaryotic genomes, the non-coding sequences are as essential as coding sequences. An obvious and truly dramatic difference between prokaryotic and eukaryotic genomes is in the amount and fraction of non-coding DNA. It seems that the powerful pressure of selection for genome compactness, which keeps intergenic distances as short as possible in prokaryotes and apparently also in unicellular eukaryotes, had been somehow removed in multicellular eukaryotes. What is all this non-coding DNA for?

In 1980s it was suggested that the non-coding DNA is actually junk DNA, i.e. not good for anything, but the organism can somehow not get rid of this DNA. However, recent comparisons of the non-coding DNA sequences in two nematode species and in humans and mice have shown that 20–30% of the non-coding DNA evolves under purifying selection, i.e. is not junk at all. It is now believed that non-coding DNA is full of regulatory elements and these regulatory elements define the program of development. Basically, the non-coding regions give rise to the differences between organisms, as similarity in the gene sets of different animals is amazing: a worm, a fly, and a mammal do indeed share many of the same essential cell types. However, beyond a few such basic features they seem radically different in their body structure.

To a large extent, the instructions needed to produce a multicellular animal are contained in the non-coding, regulatory DNA that is associated with each gene. Each gene in a multicellular organism is associated with thousands of nucleotides of non-coding DNA. This DNA may contain dozens of separate regulatory elements or enhancers — short DNA segments that serve as binding sites for specific complexes of gene regulatory proteins. Roughly speaking, the presence of a given regulatory module of this sort leads to expression of the gene whenever
the complex of proteins recognizing that segment of DNA is appropriately assembled in the cell (in some cases, an inhibition or a more complicated effect on gene expression is produced instead). If we could decipher the full set of regulatory modules associated with a gene, we would understand all the different molecular conditions under which the product of that gene is to be made.

The regulatory elements can be very short and also rather noisy. Below you see an alignment of transcription binding sites for the heat shock protein, hsIV, from \textit{E.coli}. Heat shock proteins are a class of functionally related proteins whose expression is increased when cells are exposed to elevated temperatures or other stress and then downregulated when the stressing situation comes to an end. The changes in expression is transcriptionally regulated. The extensive upregulation of the heat shock proteins is a key part of the heat shock response. These proteins are found in virtually all living organisms, from bacteria to humans.

\begin{verbatim}
Haemophilus influenzae atctc AAAAAATGATCAACTTACATTTT ttatt
Pseudomonas aeruginosa ggttg AAAAAAGCGCGAATCGCCCTATAT ctccc
Shewanella putrefaciens ctgcc GAAATAGCGCCAACGGCAGCTTTT gcgta
Salmonella typhi tgg  AAAAACTCAAATCTCCCCCATCT atact
Escherichia coli  tgg  AAAAACTCAAATCCCCCATCT ataat
Vibrio cholerae cccca AAAAAAGCTGATGTTGATCAGGCTTTT tttgt
Yersinia pestis t GAAAAGTCTTAATCCCCCATCT tataa
\end{verbatim}

From this alignment one can generate a position weight matrix (PWM). In this representation an $l$ long sequence motif is represented with $l \times 4$ matrix, where each possible base at each position in the binding score is represented with a score. The score of any specific sequence is just the sum of the position scores from the weight matrix corresponding to that sequence. An entire genome can be scanned with such a matrix and the score at every position obtained. High scores obtained by these matrices along the upstream regions of a gene suggest possible regulatory sites for a specific gene. We will study this subject more in depth in Chapter 5.

**Protocol: ClustalW**

1. From the course website under Files link (theory.bio.uu.nl/BPA/Data) download the file \texttt{DHFR.txt}. This file contains, in FASTA format, the protein sequences of dihydrofolate reductase from chicken, human, \textit{Pneumocystis} (a fungus), and \textit{Pseudomonas} (a bacterium).

2. Go to the CLUSTALW server at EBI: www.ebi.ac.uk/clustalw and paste the contents of the DHFR.fasta file into the large text box in the screen.

3. Do not tick “get results by email” box, instead wait and view the results when they are ready. Leave all other options at their default settings and click \textbf{Run}. Always use ClustalW’s slow or full alignment algorithm, and not the fast one. In some servers default is the fast algorithm, so do not forget to change this.

4. After a few moments, the results page will appear (see Fig. 2.7). This page contains sequence alignment and links to numerous other files. The alignment of the chicken and
human sequences to each other is clear; note the introduction of gaps in these sequences and that of *Pseudomonas*, primarily to align them with the longer *Pneumocystis* sequence. The alignment between the vertebrate and the other sequences is less strong than between two vertebrates, but there are clear “islands” where all sequences align, indicating the presence of conserved motifs and suggesting that the alignment is biologically meaningful. Fully conserved positions are indicated with “*”. Substitutions can fall into three categories: between very similar amino acids (“:”), between relatively similar amino acids (“.”) and between non-similar ones (indicated without any symbol).

5. Click on Results summary. Here scores table gives a simple measure of the degree of alignment (as the percentage of identical amino acids in the alignment, see Fig. 2.8) between each pair of sequences. Not surprisingly, it is the largest between chicken and human (74%), and lower between all other pairs. This is a crude reflection of the phylogenetic relationships, phylogeny methods will be discussed more in depth in Chapter 4.

6. You can explore the many options that are offered. In particular Jalview option (started by a button towards the top of the page in Result Summary page) provides a useful view of the alignment, highlighting the quality of the alignment and the presence of conserved regions, giving a consensus sequence.

7. In the main ClustalW page in you can change the parameters of your alignment job. The labels of most of the ClustalW options at the EBI website are links to the relevant parts of the ClustalW help pages.

8. Remember that ClustalW will attempt to align all sequences globally, even if they are unrelated or have only very short local similarities.
2.7 Summary

Efficient algorithms exist to align several related sequences. These algorithms make use of a score based on similarity of amino acids and gap penalties (one for initiating a gap and one for extending a gap). Alignment algorithms use a technique called dynamic programming, which relies on breaking down the problem into smaller pieces. The solution is built up progressively from the solution of smaller problems. Multiple alignment algorithms often start with a pairwise alignment of all sequences, and then make use of a phylogenetic tree (guide tree) to align clusters of already aligned sequences. In many cases manual improvements are required to correct the errors made by alignment algorithms, however. Newer multiple alignment algorithms implemented in programs such as T-Coffee (Notredame et al., 2000), DIALIGN (Morgenstern, 1999) and MUSCLE (Edgar, 2004) perform better and better, but the algorithms behind these methods are beyond the scope of this course.

Obligatory Reading

- Eddy (2004c): Where did the BLOSUM62 alignment score matrix come from? (Link to be found in the course webpage)
- Eddy (2004b): What is dynamic programming? (Link to be found in the course webpage)

Suggested Reading

2.8 Exercises (pen and paper)

**Question 2.1. PAM and BLOSUM**

Take a good look at BLOSUM62 and PAM250 matrices and remember how these matrices were made.

a. which amino acids are most conserved?

b. which amino acids are least conserved?

c. Do the matrices agree on which amino acids need to be most conserved in a multiple sequence alignment?

**Question 2.2. Similarity between sequences**

You are given the following three sequences:

1. VWEDNWDDD
2. VSEDNRDDD
3. VWDDNWDED

Calculate similarity scores between these sequences using:

a. Identity matrix (assume mismatch=0, and match=1)

b. BLOSUM62

What difference does it make to use BLOSUM62 matrix?

**Question 2.3. Gap penalty**

You are given the following two alignments:

(i) -CGCATG (ii) CGCATG
   
   ACG-A-G   ACGA-G

Using the identity matrix (match=1 and mismatch=-1), for which values of a linear gap penalty, $g$, the alignment in (i) is better than the alignment in (ii)? HINT: Try to write it as an equation and solve. Does it make sense to have positive gap penalties?

**Question 2.4. Needleman-Wunsch**

Align DEVD and DEEEVW using the Needleman-Wunsch algorithm, the identity matrix
2.8 Exercises (pen and paper)

(match=1 and mismatch=-1), and a gap penalty of $-2$. Hint: have a good look at the example explained in Fig. 2.6. You can practice Needleman-Wunsch algorithm more intensively at baba.sourceforge.net!

**Question 2.5. A multiple alignment**

The alignment below is part of multiple alignment of six proteins sequences (human, chimpanzee, mouse, rat, shark and chicken). Amino acids are shaded according to their conservation and physico-chemical properties.

a. Mark all conserved positions.
b. Are conserved positions distributed evenly over the entire length of the alignment? What does this imply for the relative importance of the different regions in the aligned proteins?
c. Do the amino acids at conserved positions in this alignment match with most conserved amino acids in PAM250 or BLOSUM62 (see above question)? What could this mean?
d. If the first sequence is from human and the third is rat, which one is chimpanzee and which one is mouse?
e. Is the shark sequence (5th sequence) closer to the human than the chicken is to human (use identity to calculate the distance)?

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLEVQVRESLAKSSQVAIEALSAMPTVRSFANEEGEAQKFRKSLQBIKTL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLEVQVRESLAKSSQVAIEALSAMPTVRSFANEEGEAQKFRKSLQBIKTL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLAVKQESLAKSTQVAIEALSAMPTVRSFANEEGEAQKFRKQLEEMKTL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLAVKQESLAKSTQVAIEALSAMPTVRSFANEEGEAQKFRKQLEEMKTL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLSEVQESLAKANDVAVETFLSMKTVRSFANEDGENSRRYGERLEDTFRL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALAPQSMGKAQARASEVAVETFQAMATVRSFANEDGAAAYQQRCLQSCHR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Question 2.6. DNA Alignments**

a. Which positions are most conserved in the below alignment?
b. Assume the first position in the below alignment is the first position of a codon (i.e. assume that translation starts from the first position). Give a biological reason for some positions being more conserved than others.

```
ATC---TAGGTG
ATCTTCTACGTG
ATCAATTTACGTG
ATA---TATGTC
ATT---TATGTC
```
Chapter 3

Local Alignments and Database Searches

Comparative sequence analysis is the first step to study sequence-structure-function relationship in protein and nucleotide sequences. Comparing the sequence of a certain protein with the ones in annotated protein databases often gives very important clues about the 3D structure of the protein and its possible function. Protein structure and function are still two unresolved major problems in molecular biology. Proteins have a large variety of 3D structures, and making sense of these observed structures is a challenge: how can we recognize the important similarities and differences between the protein structures and how does the structure relate to protein function?

<table>
<thead>
<tr>
<th>No</th>
<th>DB:ID</th>
<th>Description</th>
<th>Length</th>
<th>Match%</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SW:HS104_YEAST</td>
<td>HSP104.</td>
<td>908</td>
<td>100.0</td>
<td>3226</td>
</tr>
<tr>
<td>2</td>
<td>SW:HSP98_NEUCR</td>
<td>HSP98.</td>
<td>927</td>
<td>49.7</td>
<td>1838</td>
</tr>
<tr>
<td>3</td>
<td>SW:HS101_ARATH</td>
<td>HSP101.</td>
<td>911</td>
<td>44.3</td>
<td>1564</td>
</tr>
<tr>
<td>4</td>
<td>SW:HS101_ORYSA</td>
<td>HSP101.</td>
<td>912</td>
<td>42.8</td>
<td>1550</td>
</tr>
<tr>
<td>5</td>
<td>SW:CLPB_GLOVI</td>
<td>Chaperone clpB.</td>
<td>872</td>
<td>43.4</td>
<td>1530</td>
</tr>
<tr>
<td>6</td>
<td>SW:CLPB_AGRT5</td>
<td>Chaperone clpB.</td>
<td>874</td>
<td>44.4</td>
<td>1528</td>
</tr>
<tr>
<td>7</td>
<td>SW:CLPB1_SYNP7</td>
<td>Chaperone clpB 1.</td>
<td>874</td>
<td>41.7</td>
<td>1522</td>
</tr>
<tr>
<td>8</td>
<td>SW:CLPB_CAUCR</td>
<td>Chaperone clpB.</td>
<td>859</td>
<td>44.6</td>
<td>1521</td>
</tr>
<tr>
<td>9</td>
<td>SW:CLPB1_SYNEL</td>
<td>Chaperone clpB 1.</td>
<td>871</td>
<td>42.6</td>
<td>1521</td>
</tr>
<tr>
<td>10</td>
<td>SW:CLPB_RHOPA</td>
<td>Chaperone clpB.</td>
<td>879</td>
<td>43.4</td>
<td>1519</td>
</tr>
<tr>
<td>11</td>
<td>SW:CLPB2_ANASP</td>
<td>Chaperone clpB 2.</td>
<td>872</td>
<td>41.6</td>
<td>1518</td>
</tr>
<tr>
<td>12</td>
<td>SW:CLPB_ECO57</td>
<td>Chaperone clpB.</td>
<td>857</td>
<td>42.7</td>
<td>1517</td>
</tr>
<tr>
<td>13</td>
<td>SW:CLPB_ECOLI</td>
<td>Chaperone clpB.</td>
<td>857</td>
<td>42.7</td>
<td>1517</td>
</tr>
<tr>
<td>14</td>
<td>SW:CLPB_ECOL6</td>
<td>Chaperone clpB.</td>
<td>857</td>
<td>42.7</td>
<td>1517</td>
</tr>
<tr>
<td>112</td>
<td>SW:HSP78_CANAL</td>
<td>Heat shock protein 78</td>
<td>812</td>
<td>43.1</td>
<td>1385</td>
</tr>
<tr>
<td>119</td>
<td>SW:HSP78_YEAST</td>
<td>Heat shock protein 78</td>
<td>811</td>
<td>43.7</td>
<td>1353</td>
</tr>
</tbody>
</table>

Table 3.1: Selected parts of a Smith-Waterman search made with heat shock protein 104 of Yeast using the PAM300 scoring matrix and \( g_{open} = 12 \) and \( g_{ext} = 2 \). SW indicates that these entries are from SwissProt database.
**Local Alignments and Database Searches**

<table>
<thead>
<tr>
<th>No</th>
<th>DB:ID</th>
<th>Description</th>
<th>Length</th>
<th>Match%</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SW:HS104_YEAST</td>
<td>HSP104</td>
<td>908</td>
<td>100.0</td>
<td>9411</td>
</tr>
<tr>
<td>2</td>
<td>SW:HSP98_NEUCR</td>
<td>HSP98</td>
<td>927</td>
<td>53.4</td>
<td>3424</td>
</tr>
<tr>
<td>3</td>
<td>SW:HS101_ARATH</td>
<td>HSP101</td>
<td>911</td>
<td>49.0</td>
<td>2546</td>
</tr>
<tr>
<td>4</td>
<td>SW:HS101ORYSA</td>
<td>HSP101</td>
<td>912</td>
<td>48.8</td>
<td>2486</td>
</tr>
<tr>
<td>5</td>
<td>SW:CLPB_CGRTB5</td>
<td>Chaperone clpB.</td>
<td>874</td>
<td>50.2</td>
<td>2472</td>
</tr>
<tr>
<td>6</td>
<td>SW:CLPB_NEIMB</td>
<td>Chaperone clpB.</td>
<td>859</td>
<td>50.3</td>
<td>2464</td>
</tr>
<tr>
<td>7</td>
<td>SW:CLPB_NEIMA</td>
<td>Chaperone clpB.</td>
<td>859</td>
<td>50.3</td>
<td>2457</td>
</tr>
<tr>
<td>8</td>
<td>SW:CLPB2_SYNY3</td>
<td>Chaperone clpB.</td>
<td>872</td>
<td>50.1</td>
<td>2439</td>
</tr>
<tr>
<td>9</td>
<td>SW:CLPB_RHOPA</td>
<td>Chaperone clpB.</td>
<td>879</td>
<td>49.8</td>
<td>2432</td>
</tr>
<tr>
<td>10</td>
<td>SW:CLPB_CACUR</td>
<td>Chaperone clpB.</td>
<td>859</td>
<td>50.7</td>
<td>2414</td>
</tr>
<tr>
<td>11</td>
<td>SW:CLPB_BRAJA</td>
<td>Chaperone clpB.</td>
<td>879</td>
<td>49.7</td>
<td>2407</td>
</tr>
<tr>
<td>12</td>
<td>SW:CLPB_RHIME</td>
<td>Chaperone clpB.</td>
<td>868</td>
<td>49.4</td>
<td>2406</td>
</tr>
<tr>
<td>13</td>
<td>SW:CLPB_GLOVI</td>
<td>Chaperone clpB.</td>
<td>872</td>
<td>47.3</td>
<td>2404</td>
</tr>
<tr>
<td>14</td>
<td>SW:CLPB_BIFLO</td>
<td>Chaperone clpB.</td>
<td>889</td>
<td>48.7</td>
<td>2400</td>
</tr>
<tr>
<td>66</td>
<td>SW:HSP78_CANAL</td>
<td>HSP78</td>
<td>812</td>
<td>46.0</td>
<td>2237</td>
</tr>
<tr>
<td>85</td>
<td>SW:HSP78_YEAST</td>
<td>HSP78</td>
<td>811</td>
<td>46.2</td>
<td>2188</td>
</tr>
</tbody>
</table>

Table 3.2: Selected parts of a Smith-Waterman search made with the same heat shock protein using the PAM50 scoring matrix and $g_{open} = 40$ and $g_{ext} = 7$.

The global alignment scheme described in Chapter 2 is very good for comparing and analyzing the relationship between two selected sequences. Proteins, however, are often comprised of different domains, where each domain may be related to a certain function. Thus when it comes to searching for functionally related sequences it is often beneficial to look for small parts of the sequences that are similar to each other. Such a search consists of making a pairwise alignment of your “query” sequence to all the sequences in the database, and order the resulting alignments by the alignment score. The “hits” are the top few sequences from the database that have a high score alignment with the query sequence. Here we explain a few algorithms that perform such database searches.

### 3.1 Smith-Waterman

Smith & Waterman (1981) further developed the dynamic programming approach we reviewed in Chapter 2 to make local alignments. The Smith-Waterman algorithm is like Needleman-Wunsch, except that the traces only continue as long as the scores are positive. Whenever a score becomes negative it is set to 0 and the corresponding trace is empty. Thus in the Smith-Waterman algorithm:

$$H(i, j) = \max \left\{ \begin{array}{ll} H(i-1, j-1) + S(a_i, b_j) & \text{diagonal} \\ H(i, j-1) - g & \text{horizontal} \\ H(i-1, j) - g & \text{vertical} \\ 0 & \text{start again} \end{array} \right. \quad (3.1)$$

The backtracing procedure for local alignments begins at the highest-scoring point in the matrix, and follows the arrows back until a 0 is reached. The highest scoring cell does not need to be at the bottom right-hand corner, it could be anywhere in the matrix.
One web server that implements this algorithm is at http://www.ebi.ac.uk/MPsrch. Take as an example a yeast chaperone protein (heat shock protein 104). If we query this sequence against Swiss-Prot database using the PAM300 matrix for sequence similarity and default gap open and extension penalties, we get the results summarized in Table 3.1. The top hit is the query sequence itself. Among the top 15 hits we find homologous sequences from other fungi, other eukaryotes, and bacteria. One homolog (HSP78_CANAL) in Candida Albicans (another fungus) is ranked much lower than e.g. the plant homologs (HS101_ORYSA and HS101_ARATH). If we increase the gap penalty and use a less distant PAM matrix (see Table 3.2) the rank of HSP78_CANAL improves. The second heat shock protein from yeast (heat shock protein 78) has rather diverged from 104. The Smith-Waterman algorithm is nevertheless able to find it with both parameter setups. Finally, in both cases the search returns also sequences that are not functionally related (these sequences were omitted from Table 3.1 and 3.2). These unrelated hits often have a non-significant score (see below). Mimi virus also has a homolog to yeast 104 heat shock protein, but this homolog is not found by the Smith-Waterman search.

3.2 Heuristic methods

This was an exhaustive search, i.e. all possible pairwise alignments were made for this database search. The dynamic programming algorithm ensures that the optimal alignment will always be found, given specific gap penalties and substitution scores. With increasing number of sequences in the databases, this approach is no longer realistic: even with present-day computer power this algorithm is far too slow to search today’s sequence databases. Instead, we need to perform a heuristic search, where we look for similar sequences in the database, without being certain that we use exactly optimal alignments. These heuristic methods can identify many similar sequences in a short time. We can subsequently use an exact alignment program to align these hits with the query sequence.

BLAST

A very popular local alignment tool is BLAST (Altschul et al., 1990, 1997; Altschul & Gish, 1996). The basic BLAST algorithm starts with scanning the database for words of length \( w \) that have a similarity score of at least \( T \) to the query sequence. As a default \( w = 3 \) for proteins and \( w = 11 \) for DNA. The local alignment around a hit is then extended in each direction. The extension stops when the score starts falling drastically. This corresponds to saying this route looks so bad that there is no point in continuing in this direction. The first version of BLAST extended every hit it found. The newest version requires two non-overlapping hits within a certain distance of each other before it extends a hit. The locally optimal alignments are called high-scoring segment pairs (HSPs). “Gapped BLAST” attempts to search gapped extension (by dynamic programming) to find best HSP’s. To speed up the calculations this phase is only continued until the score falls a certain amount below the best score seen so far. Fig. 3.1 explains the parameters we mentioned above.

Many nucleotide and amino acid sequences are highly repetitive in nature. If your query sequence contains regions of low complexity or repeats, you can end up with many non-
Local Alignments and Database Searches

Figure 3.1: This figure is taken from www.ncbi.nlm.nih.gov. The BLAST algorithm is a heuristic search method that seeks words of length $w$ (default $w = 3$ in BLAST for proteins) that score at least $T$ when aligned with the query and scored with a substitution matrix (in this case BLOSUM62). Words in the database that score $T$ or greater are extended in both directions in an attempt to find a locally optimal ungapped alignment or HSP.

related, high scoring sequences being found during BLAST searches. These hits are often biologically unrelated, and it might be useful to exclude them in database searches based on sequence similarity. The BLAST package includes programs that have been devised to filter out unwanted segments from a sequence. For example, filtering low complexity regions out of your query sequence before searching a database can markedly reduce the number of unrelated sequences that match by chance. Filtering works by running programs that identify regions containing particular types of sequences. These regions are replaced with a series of N’s (in the case of nucleotide sequences), or X’s (in the case of peptide sequences).

An important application of BLAST is to detect orthologs (see Chapter 2). The COG (Clusters of Orthologous Groups) database was set up to identify related groups of genes in complete genomes (Tatusov et al., 2003). Each COG is a set of genes composed of individual orthologous genes. For each gene in each of the complete genomes analyzed, BLAST was used to find the best hitting gene in each of the other genomes. The clustering procedure began by looking for triangular relationships of genes in three species, such that each is the best hit of other two. This is also called a “bidirectional BLAST” hit. Whenever two triangles of genes shared a side, these sets of genes were combined into one cluster. COG database for unicellular organisms (most updated version) today has 66 complete genomes, and 4872 COGs (see www.ncbi.nlm.nih.gov/COG).

Different BLAST programs

This section is partly adapted from NCBI-BLAST help pages.
3.2 Heuristic methods

Nucleotide sequences: MEGABLAST, Discontigous-megablast, Blastn

The best way to identify an unknown sequence is to see if that sequence already exists in a public database. If the database sequence is a well-characterized sequence, then one will have access to a wealth of biological information. MEGABLAST, discontiguous-megablast, and blastn all can be used to accomplish this goal. However, MEGABLAST is specifically designed to efficiently find long alignments between very similar sequences and thus is the best tool to use to find the identical match to your query sequence.

One of the important parameters governing the sensitivity of BLAST searches is the length of the initial words, or word size as it is called. The most important reason that blastn is more sensitive than MEGABLAST is that it uses a shorter default word size (11). Because of this, blastn is better than MEGABLAST at finding alignments to related nucleotide sequences from other organisms. The word size is adjustable in blastn and can be reduced from the default value to a minimum of 7 to increase search sensitivity.

A more sensitive search can be achieved by using the newly introduced discontiguous megablast page. Rather than requiring exact word matches as seeds for alignment extension, discontiguous megablast uses non-contiguous word within a longer window of template. This is established by (among others) focusing on finding matches at the first and second codon positions while ignoring the mismatches in the third position. Searching in discontiguous MEGABLAST using the same word size is more sensitive and efficient than standard blastn using the same word size. For this reason, it is now the recommended tool for this type of search. Alternative non-coding patterns can also be specified if desired.

Protein sequences: Blastp, PSI-BLASTand PHI-BLAST

Standard protein-protein BLAST (blastp) is used for both identifying a query amino acid sequence and for finding similar sequences in protein databases. Like other BLAST programs, blastp is designed to find local regions of similarity. When sequence similarity spans the whole sequence, blastp will also report a global alignment, which is the preferred result for protein identification purposes.

As described earlier, the scoring matrices represent the general evolutionary trends for mutations. However, in reality, allowed mutations are very much constrained by their physical context. For example, it is possible to insert, delete, or exchange a number of different amino acids in a flexible loop on the surface of a protein and still preserve the overall structure and function of the protein. The corresponding number of allowed substitutions would probably be much more limited in the core — or in a secondary structure rich — region of the protein. So if a general substitution matrix works well, a matrix representing the specific evolutionary trend for a given position in a given protein should work even better.

In the Position specific iterative-BLAST (PSI-BLAST) approach, first an ordinary BLAST search on the basis of the BLOSUM62 matrix is performed against the database. Second, a position-specific scoring matrix (PSSM) is calculated by considering the substitutions observed in pairwise alignments made between the query sequence and the good hits (i.e. the ones that are above a selected threshold). The details of how this matrix is generated is beyond the
Local Alignments and Database Searches

Figure 3.2: Example of a PSSM. Here the query sequence is 17 positions long and it is shown vertically. For every position the scoring is calculated using the good hits one gets in the initial BLAST search made with this query.

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

Figure 3.3: Flow of PSI-BLAST.

Pattern-Hit Initiated (PHI)-BLAST is designed to search for proteins that contain a pattern specified by the user AND are similar to the query sequence in the vicinity of the pattern. This dual requirement is intended to reduce the number of database hits that contain the pattern, but are likely to have no true homology to the query. Pattern syntax is explained at www.ncbi.nlm.nih.gov/blast/html/PHIsyntax.html. An example would be:

```
[LFYT] -G-E-x-[GAS] -[LIVM] -x(5,11) -R-[STAQ]-A-x-[LIVMA]-x-[STACV]
```
[LFYT] means one occurrence of L or F or Y or T, x(5,11) means 5 to 11 positions where any residue is allowed.

**Blastx**

Translated query vs protein database (blastx) is useful for finding similar proteins to those encoded by a nucleotide query. Translated BLAST services are useful when trying to find homologous proteins to a nucleotide coding region. Blastx compares translational products of the nucleotide query sequence to a protein database. Because blastx translates the query sequence in all six reading frames and provides combined significance statistics for hits to different frames, it is particularly useful when the reading frame of the query sequence is unknown or it contains errors that may lead to frame shifts or other coding errors. Thus blastx is often the first analysis performed with a newly determined nucleotide sequence and is used extensively in analyzing expressed sequence tags (EST) sequences. This search is more sensitive than nucleotide blast since the comparison is performed at the protein level.

**tblastn**

Protein query vs translated database (tblastn) is useful for finding protein homologs in unannotated nucleotide data. A tblastn search allows you to compare a protein sequence to the six-frame translations of a nucleotide database. It can be a very productive way of finding homologous protein coding regions in unannotated nucleotide sequences such as ESTs and draft genome records (HTG), located in the BLAST databases est and htgs, respectively.

ESTs are short, single-read cDNA sequences. They comprise the largest pool of sequence data for many organisms and contain portions of transcripts from many uncharacterized genes. Since ESTs have no annotated coding sequences, there are no corresponding protein translations in the BLAST protein databases. Hence a tblastn search is the only way to search for these potential coding regions at the protein level. The HTG sequences, draft sequences from various genome projects or large genomic clones, are another large source of unannotated coding regions.

Like all translating searches, the tblastn search is especially suited to working with error prone data like ESTs and draft genomic sequences from HTG because it combines BLAST statistics for hits to multiple reading frames and thus is robust to frame shifts introduced by sequencing error.

**tblastx**

Translated query vs translated database (tblastx) is useful for identifying novel genes in error prone nucleotide query sequences.

tblastx takes a nucleotide query sequence, translates it in all six frames, and compares those translations to the database sequences dynamically translated in all six frames. This effectively
performs a more sensitive blastp search without doing the manual translation.

tblastx gets around the potential frame-shift and ambiguities that may prevent certain open
reading frames from being detected. This is very useful in identifying potential proteins
encoded by single pass read ESTs. In addition, it can be a good tool for identifying novel
genes.

This type of search is computationally intensive and should be used only as last resort. Searching with large genomic queries is NOT recommended.

3.3 Expectation Values

An important issue for database searches is whether a given match score between two se-
quences is statistically significant, i.e. it is not occurring by chance. In this context, “chance”
can mean the comparison of (i) real but non-homologous sequences, (ii) real sequences that are
shuffled to preserve compositional properties, or (iii) sequences that are generated randomly
based upon a DNA or protein sequence model. Repeat question 1.2 from the first chapter.
For every DNA/amino acid sequence of length \( l \) there is a probability larger than 0 that it
could occur by chance in a database. With increasing \( l \), this chance becomes almost zero.

The statistics of high scoring segment pairs (HSP) scores are characterized by the length of
query and the total length of all the sequences in the database (say \( n \) and \( m \), respectively).
Most simply, the expected number of HSPs with score at least \( S \) is given by the formula:

\[
E = K \times m \times n \times e^{-\lambda S} \tag{3.2}
\]

We call this the E-value for the score \( S \). The parameters \( K \) and \( \lambda \) can be thought of as scaling
factors for the search space (database) and the scoring matrix, respectively. The E-value is the
number of different alignments with scores equivalent to or better than \( S \) that are expected to
occur in a database search by chance. The lower the E value, the more significant the score.
Often only E-values lower than 0.05 are interesting, however, if you are searching for very
distant homologs, you should also consider higher E-values. E-values become more accurate if
one uses the total length of all “non-redundant (i.e. non-identical) sequences in the database.

Eq. 3.2 makes intuitive sense. Doubling the length of either sequence should double the
number of HSPs attaining a given score. Also, for an HSP to attain the score \( 2x \) it must
attain the score \( x \) twice in a row, so one expects E to decrease exponentially with score.

Different alignment programs use different approaches to calculate the E-value of a given
database hit. BLAST uses a pre-made empirical curve to assign E-values to each alignment
returned from a database search. Figure Fig. 3.4 illustrates this process.

The substitution matrix used in a BLAST search directly affects the E-values obtained be-
cause, the score of the hit depends heavily on the matrix chosen. Several correction parameters
are used in E-value calculations to compensate for the matrix differences (e.g. \( \lambda \)) above. As
well as the degree of the evolutionary distance, the length of the query sequences must be
taken into account when choosing a substitution matrix. The length of the sequence is taken
Figure 3.4: A cartoon demonstrating the main principle behind BLAST statistics. The significance of an hit, E-value, is calculated based on a distribution of scores obtained using alignment of randomly chosen, i.e. unrelated, sequences. A hit is significant if there are only few random hits that have a score higher than the score of the actual hit. Adapted from Introduction to Bioinformatics for Molecular Biologists, UU, 2010 course notes.

into account when accessing the significance of the hit (see above): the shorter the sequence, the higher the score needs to be in order to be judged significant. For short sequences it is advised to use matrices designed for short evolutionary time scales such as PAM40, or BLOSUM80. Longer sequences of 100 residues or more can better use matrices intended for use with longer evolutionary time scales (such as PAM250, or BLOSUM50).

3.4 Summary

Pairwise alignment methods can be used to search databases for sequences that are similar to a query sequence. However, we can not use directly the pairwise methods we learned in Chapter 2, because full alignments in a large database would be too time consuming. Instead heuristic search programs are developed for this purpose. The most popular search tool is the Basic Local Alignment Search Tool (BLAST). BLAST finds regions of local similarity between sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. BLAST can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families. PSI-BLAST is a tool that produces a position-specific scoring matrix constructed from a multiple alignment of the top-scoring BLAST responses to a given query sequence. This scoring matrix produces a profile designed to identify the key positions of conserved amino acids within a motif. When a profile is used to search a database, it can often detect subtle relationships between proteins that are distant structural or functional homologues. These relationships are often not detected by a BLAST search with a sample sequence query.

Protocol: A typical search using BLAST

2. Click on protein blast.
3. Paste the \(\alpha\)-tubulin protein sequence from *M. musculus* in FASTA into the **Enter Query Sequence** box. To do this, you should first obtain the sequence from NCBI web site www.ncbi.nlm.nih.gov. From the pull-down menu titled Search at the top left, select protein and type “mus musculus[ORGANISM] alpha tubulin”. Click Go or press Enter on your keyboard. You will see a list of links in the next page. Choose the first one with correct annotation, i.e. the one that sounds most like the protein you were looking for. Dowload the sequence in FASTA format and paste it into the query box in BLAST page.

4. In BLAST page you will see all major protein databases listed to perform this search. Leave Database at its default, nr, which encompasses all non-redundant translations from Genbank or several other major databases (you can get more information if you use help option).

5. Leave all other settings at their defaults for now and click BLAST to start the search using standard BLASTP program.

6. Typically within few seconds the results of the BLASTP run will be displayed (see Fig. 3.5). At the top of the results page (after the references), the summary of the domains identified in this protein is given.

7. Following the domains summary, a graphic represents the extent and significance of hits against the query sequence. Moving the mouse over any of the colored bars will cause the details of that alignment to appear on the small text window above the graphic.

8. The hits are listed below the graphic (see Fig. 3.6) in order of increasing E-value (decreasing the significance).

9. The default threshold for E-value is 10, but can be adjusted by the user. Sequences beyond an E-value of 0.01 should be treated with caution, as they might indeed be unrelated.

10. First column of each of the hits point to the Entrez protein database entry for the protein. **U** and **G** symbols to in the last column link to the entries in UniGene or
3.4 Summary

Entrez gene. From all of these linked databases, there are numerous links to other resources for that protein.

11. Below the list of hits, each alignment is shown in detail (see Fig. 3.7). You can jump directly to the alignment by clicking to the link under Score or by clicking on the colored bar in the graphical interface. Note that redundant entries are listed here; only one of these would be shown in the graphic window.

12. On the query page (step 3), there are numerous options to alter the search parameters or limit the search. In particular, under Algorithm parameters (bottom), the E-value threshold (expect threshold), word size, and the gap penalties (Gap costs) can be adjusted as well as more complex parameters.

13. For example, Organism box (under Choose Search Set) can be used to limit the search to specific organisms or group of organisms. Try repeating the same BLASTP search using *Sus scrofa* (pig) as the organism. The result will be limited to only few hits.
3.5 Exercises (pen and paper)

Question 3.1. E-values

You have a query sequence, which is 40 residues long. We BLAST this sequence against a non-redundant protein database with 29318 sequences, and having total length of 6548585 amino acids. The best hit is in a sequence which is 420 amino acids long. Looking at the alignment of the best hit, you observe the following (for clarity only the HSP is shown):

<table>
<thead>
<tr>
<th>Query</th>
<th>27 NFSSSQ 32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sbjct</td>
<td>73 NFSTSQ 78</td>
</tr>
</tbody>
</table>

For this question we will use BLOSUM62 matrix.

a. What is the alignment score?

b. For this database search $K = 0.04$ and $\lambda = 0.27$ values are appropriate. Calculate the E-value for this HSP.

c. Why do you think this hit is significant / insignificant?

Question 3.2. Smith-Waterman

For the sequences GYYI and DFKYIW calculate the Smith-Waterman alignment using the BLOSUM62 matrix. Take a gap penalty of $-12$. You can practice Smith-Waterman algorithm more intensively at baba.sourceforge.net!

Question 3.3. BLAST, I

Imagine you have sequenced a novel fungus genome. There are 6500 predicted genes in this genome. You do a BLAST search among known fungal genomes using the translation of predicted genes.

a. Of the 6500 proteins, 5500 have a unique match in other species of fungi (S. cerevisiae, S. pombe, or N. crassa) with a very low E-value. What can you say about these proteins?

b. The remaining 1000 proteins each have a best match with an E-value larger than 10. What can you say about these proteins?

c. Of these last 1000 proteins, can you make another BLAST search to verify your conclusion? What else can you try?
3.5 Exercises (Pen and Paper)

Question 3.4. BLAST, II

Site 1:
>gb|AAC60279.1| ubiquitin/ribosomal protein [Gallus gallus]
Length=156
Score = 47.8 bits (112), E-value = 1e-04
Identities = 47/95 (49%), Positives = 50/95 (52%), Gaps = 36/95 (37%)
Query 1
IRKETTLHKVLRLWGGAYKDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXKKKSY 60
I+KE+TLH VLRL GGA K
Sbjct 61
IQUESTILHLVLRLRGAKK-----------------------------------RRKKSY 85
Query 2
TMPXXXXXXXXXXX-AVLPYYKIDEYGKISRFRRE 94
T P
Sbjct 86
TPPKKNHKKRKVKVKLAVLKYYKVDENGKISRLRE 120

Site 2:
>sp|P42568|AF9_HUMAN Protein AF-9 (Myeloid/lymphoid or mixed-lineage leukemia associated)
Length=568
Score = 68.9 bits (167), E-value = 5e-11
Identities = 40/52 (76%), Positives = 44/52 (84%), Gaps = 0/52 (0%)
Query 2
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSKKSYTMPKKNHKKHKK 72
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSKPHKLMKEHKE 205
Sbjct 154
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSTSFSKPHKLMKEHKE 205

We do a BLAST search to predict the function of our human query protein on two different internet sites that provide a BLAST search tool. The alignments of best hits are given above.

a. Which hit is statistically more significant? Explain.

b. What is the reason for the difference between the two BLAST results? Which of the two hits do you think is most likely to be a true homolog? Explain.

Question 3.5. BLAST, III

We have a short protein segment from chicken:

FGGHNAITYPPGVSLAVGHFFSEWAFAFGDPLYRSSSSSSSSSSSSSTENKLAFGTHRDRDVGHFFCKAAGAAEFK

Sequences producing significant alignments:

| gi|76638832|ref|XP_60  similar to SRp25 nuclease [Bos taurus] | 40.8 | 0.017 |
| gi|6649242|gb|AAF21439.1| splicing coactivator subunit SRm300 | 40.0 | 0.028 |
| gi|66828915|ref|XP_647811.1| hypothetical protein DDB0206273 | 40.0 | 0.028 |
| gi|66358726|ref|XP_626541.1| hypothetical protein cgd2_3540 | 39.7 | 0.037 |
| gi|66910579|gb|AAH97374.1| ADP-ribosylation factor-like 6 int... | 39.7 | 0.037 |
| gi|66816197|ref|XP_642108.1| hypothetical protein DDB0204407 | 39.7 | 0.037 |
Local Alignments and Database Searches

46

We do a BLAST search to predict its function. Top 10 hits are given above.

a. Can you predict the function of this protein based on this output?
b. What can you do to improve this search?

Question 3.6. BLAST, IV

a. We have determined the genome sequence of a bacterium. How can we use BLAST to identify protein-coding genes in this genome if we only have access to protein sequence databases?
b. A BLASTP search has not returned any hits at all. Would it be useful to do a PSI-BLAST using the same settings as the original BLASTP?

Question 3.7. BLAST, V

a. We have the DNA sequence of a gene, and we want to identify distant homologs of this sequence. Can we use MEGABLAST, or should we use BLASTN?
b. We have the amino acid sequence of a well-studied enzyme from *E. coli*. We have been unable to identify a homologous sequence in *B. subtilis* (another bacterium) using BLASTP, but experimental evidence shows that *B. subtilis* is capable of doing this enzyme reaction. What type of BLAST could you do to identify the gene responsible for the reaction in *B. subtilis*? Assume that the genome and all predicted proteins of *B. subtilis* are in the databases.
c. Below you will find part of the output of a BLASTP search of the protein AP012432 from zebrafish against a database containing all proteins from yeast. Predict the function of AP012432.

Sequences producing alignments:

ref|YP_00168034| two-component sensor histidine kinase 173 4e-78
ref|ZP_02948675| two-component system sensor histidine kinase 173 4e-77
ref|LP_03036538| sensory transduction histidine kinase 163 3e-75
ref|YP_00128034| unknown extracellular enzyme 25 0.6
ref|ZP_01238675| matrix protein homologous to env4 21 1.1
ref|LP_02134538| DNA helicase precursor protein 21 1.1

d. What could you do to find out if AP012432 and YP00168034 are orthologs?
Chapter 4

Molecular Evolution and Phylogeny

(The introduction to phylogeny here was adopted from Snel et al. (2005) with permission from Berend Snel.)

Baffled by the variety in life, one of man’s first biological activities has been to classify it. Since Darwin’s theory of evolution, the ultimate goal is to obtain a hierarchical classification that matches the evolutionary relations between species. This makes the construction of phylogenies one of the central activities of biologists, not only to reconstruct the history of life, but also to understand it because “nothing in biology makes sense except in the light of evolution” (Dobzhansky, 1973). Traditionally, phylogenies were constructed from phenotypic characteristics, and phenotypic characteristics continue to play a dominant role in the analysis of data such as fossils. However, with the advent of sequencing technologies, it has become possible to construct trees on the basis of nucleotide and amino acid sequences. Sequence-based trees such as the ones based on the ribosomal RNA molecules have become the golden standard in areas where phenotypic data are scarce. Sequence-based analyses have yielded surprising observations, such as the close phylogenetic relationship between archaea and eukaryotes relative to bacteria, between fungi and animals relative to plants. Furthermore, sequences can be used for organisms for which we do not have phenotypic data, or for which we do not even know exist, as in the case of e.g. the environmental sampling of ribosomal RNA from uncultured bacteria that live in hydrothermal vents at the deep-sea bottom. Reconstructing the evolutionary history of genes is also of great importance for the bioinformatic prediction of the function of genes when their function has not been experimentally determined. In fact, both the prediction of gene function as well as the discovery of the evolutionary relationships

Figure 4.1: Figure 8.1. in Higgs & Attwood (2005) Rooted trees with a time axis. The tree in (a) can be converted into that in (b).
4.1 What is a phylogenetic tree?

Phylogenies reveal evolutionary relationships between organisms and specific sequences (e.g. genes and proteins). One often studies a set of sequences that are likely to have evolved from a common ancestor to find out which species diverged from each other and in which order. The evolutionary relations between sequences (i.e. when, and in what order, did the sequences diverge) can be depicted in a tree-like drawing where the branch length correspond to time or evolutionary change. Such a drawing is called a phylogenetic tree. There are several ways of reconstructing such a tree, but let us first focus on how the information in a phylogenetic tree is to be interpreted, using a tree of species as an example.

Fig. 4.1a shows a simple rooted tree. This diagram shows that species A diverged 30 million years ago (Ma) from species B, C, and D. B diverged from C and D 22 Ma, and C and D diverged from each other 7 million years ago. The root of the tree is the most common ancestor of all four species. We get a clear picture of the evolutionary relationships between these species: Species C and D are evolutionarily close to each other, while A and B are more distant from C and D. The lengths of the horizontal lines do not mean anything: they basically make the tree readable. It is possible to “swing” branches in a tree, i.e. the tree in Fig. 4.1b is equivalent to the tree in Fig. 4.1a. But we cannot swap around all branches, e.g. we can swap C and D around, but not B and C. The grouping of C and D together is often referred to as a clade: i.e. C and D form a clade.

Often it is not possible to estimate divergence times reliably from molecular data. If your data set does not allow for such an estimate, the final tree branches do no longer represent Ma’s. If we know that the sequences under study evolve with similar rates, the branch lengths can again be drawn to scale, and representing evolutionary distance (e.g. the number of nucleotides that have been substituted) instead of Ma. Fig. 4.2 shows a rooted tree with branch lengths scaled according to the amount of evolutionary change thought to have happened on each branch. The branch leading to species C is longer than to D, meaning that the rate of evolution has been faster on the line leading to C than to D (e.g. more nucleotides have changed in C than in D). Note that Fig. 4.2 does not tell us the precise time of speciation, but it is still very
4.1 What is a phylogenetic tree?

informative about the order of events that presumably have taken place.

In rooted trees, a single node is designated as a common ancestor and a unique path leads from it to all other species studied. An unrooted tree, on the other hand, only specifies the relationship between nodes and gives no information about the direction of the evolution. Fig. 4.3a shows an unrooted tree of four species. Here the internal nodes (represented by $i$ and $j$) represent ancestors of which we do not have data. The branch lengths are again scaled with evolutionary distance, e.g. we can say that there had been a lot of changes on the branch leading to species A compared to species D. However, we can not say anything about whether or not the branching at $i$ occurred earlier or later than in $j$. When a tree is sketched as in Fig. 4.3a, it is clearly an unrooted tree. However, sometimes unrooted trees are drawn as phenograms, i.e. as in Fig. 4.3b-c. In this case read the figure caption carefully to make sure you understand whether the tree is rooted or not. When the root is not known, it is always a better practice to draw a radially branching tree as in Fig. 4.3a.

Roots can usually be assigned to trees using an outgroup, i.e. a species for which we know that it has separated the earliest from the other species being studied. For example, if we are studying the relationship between leopard, turtle, salamander, tuna and lamprey (a jawless aquatic vertebrate), a proper outgroup would be to use the lancelet, a small animal that lives in mudflats and (like vertebrates) is a member of the Chordata. Unlike the vertebrates, however, the lancelet does not have a backbone. The root of the tree would then be somewhere along the branch connecting lancelet to the common ancestor of all other vertebrates (see Fig. 4.4).

In both rooted and unrooted trees, the leaves are grouped in clusters. This grouping depends heavily on the algorithm used to construct the tree. Some algorithms just give one of potentially many, more or less equally probable, outputs. Other approaches actually calculate many different solutions and give the most probable outcome with some indication of how reliable a particular solution is. If we have $n$ sequences, the actual number of possible rooted trees, $N_R$, is $N_R = (2n - 3)!! = (2n - 3) \times (2n - 5) \times (2n - 7) \times \ldots \times 3 \times 1$ and the number
of unrooted trees, \( N_U \), is 
\[
N_U = (2n - 5)!! = (2n - 5) \times (2n - 7) \times \ldots \times 3 \times 1
\]
(the notation !! means product of odd numbers). So even if we have only 10 sequences in our data set, the number of possible rooted trees is more than 34 million. Therefore, several methods have been developed to identify the most likely trees among all possible ones. We will review these methods shortly at the end of this chapter.

### 4.2 Data selection

The starting point of any phylogenetic work is a collection of sequences that are evolutionarily related, i.e. that are homologous. Such a set could be extracted from public databases using some of the tools described previously (for example BLAST), or it could be data from one’s own work. It is important to realize that there are at least two types of trees: **gene trees** and **species trees**. If a phylogenetic tree is based on the divergence observed within a single homologous gene, than it represents only the evolutionary history of that gene, *i.e.* it is a gene tree. Species trees are often made using different sources of information, such as multiple genes, gene order, or, as was traditionally the case, using morphological characters (Snel *et al.*, 2002; Van Noort *et al.*, 2003). Discrepancies between gene trees and species trees can arise because genes have their own dynamics within populations and genomes. For example, divergence within genes typically occurs prior to the splitting of populations during speciation. This occurs particularly frequently in genes, for which (allelic) diversity in population is advantageous. A prominent example of this is the major histocompatibility complex (MHC) which has the largest degree of polymorphism among mammalian proteins. If MHC molecules alone were used to determine species trees, many humans would be grouped with gorillas rather than with other humans, because the origin of polymorphism they carry pre-dates the split that gave rise to the two lineages (see Fig. 4.5).

Gene duplication within the genome is another reason for differences between gene trees and species trees. Gene duplication means that a copy of a gene is inserted somewhere else on the chromosome. This process affects trees over much large time-scales than the allelic divergence. Yet they are similar in the sense that one species can harbor multiple variants of a homologous gene and that thus a tree of genes from these species can display multiple genes per species. Gene duplications become especially confusing when they are followed by gene loss, *i.e.* a gene is completely deleted or deteriorates after a nonsense mutation. We have illustrated the effect on the interpretation of a gene tree in the light of the occurrence of gene duplications and
4.2 Data selection

Figure 4.5: The first tree is a tree of alleles (gene variants) from humans, chimps and gorillas. The second tree is a likely explanation for this tree in terms of allelic divergence and loss of alleles. Allelic divergence means that a mutation generates a new variant for a gene in the population and that the old and new variant (i.e. alleles) remain present in that population. The explanation uses the generally accepted phylogeny of primates wherein human and chimp are more closely related to each other than either is to gorilla.

gene losses in Fig. 4.6. The occurrence of multiple copies of the hypothetical gene from the figure in human, mouse and pufferfish already suggests that one or more gene duplications have occurred. We can make a reconciliation of the gene tree with the species trees based on i) the inference of one gene duplication in the ancestor of the mammals followed by the loss of one of the duplicate genes in rat and ii) one gene duplication in the ancestor of the two teleost fish followed by the loss of one of the copies in zebrafish.

Figure 4.6: Species trees versus gene trees. Panel (a) depicts the species tree and panel (b) depicts a reconstructed gene tree. Panel (c) depicts how we can explain the gene tree by drawing a reconstruction of likely past events on an adjusted version of the gene tree. The gene tree is fictional and was thought up specifically for this example.
4.3 Constructing a phylogenetic tree

The calculation of the grouping and the branch lengths of a tree is done in two major steps. First, using clustering methods based on distance matrix (calculated from a multiple alignment, see Chapter 2) an initial tree is constructed. The clustering methods do not guarantee the optimal solution. Second, the “optimal tree” is found by a search in the tree-space. Here we will explain two optimization methods for optimal tree searches: parsimony and maximum likelihood.

Clustering using distance matrix

From a multiple alignment one can compute a matrix of the evolutionary divergence between each pair of sequences in the alignment. Such a matrix is called “distance matrix” and it is symmetric: the distance of sequence \( i \) to sequence \( j \) is the same as the distance of sequence \( j \) to sequence \( i \). The distance between nucleotide sequences, \( d \), can be calculated by several methods. For example, \( d \) can be the number of non-matching sites divided by the total number of sites. However this measure has several problems as mentioned in Chapter 2. Jukes & Cantor (1969) defined the distance between two nucleotide sequences as

\[
d = -\frac{3}{4} \ln(1 - \frac{4}{3}D),
\]

where \( D \) is the number of non-matching sites divided by the total number of sites. This is called **Jukes-Cantor** model of sequence evolution. The derivation of this formula is rather complex. However it behaves as follows: If we align two random sequences containing equal frequencies of the four bases, on average \( 3/4 \) of sites will differ between the two sequences. According to this formula as \( D \) approaches \( 3/4 \), i.e. if two sequences are almost random, the distance between the sequences approaches infinity. If \( D \) is small, then \( \ln(1 - \frac{4}{3}D) \approx -\frac{4}{3}D \) and thus \( d \approx D \), whereas if \( D > 1/2 \), i.e. if half of the positions are non-matching, \( d \) is much greater than \( D \).

This formula fails to correct for differences in the rates of transition and transversion. To do this we can use what is called the Kimura 2-parameter correction. Kimura (1980) suggested that

\[
d = -\frac{1}{2} \ln(1 - 2S - V) - \frac{1}{4} \ln(1 - 2V)
\]

where \( S \) is the number of sites that differ by a transition and \( V \) is the corresponding number for transversion and \( D = S + V \). Since each of four nucleotides can change into any of the other three, 12 substitutions are possible. Thus, the two parameter model (transversion and transition only) can be extended to 12-parameter models, however, simulation results often show that one or two parameter models give more reliable results.

For protein sequences it is preferable to use a distance that is based on substitution rates rather than on identities. Such a distance can be based on the PAM matrix and we can compute the fraction of sites that differ between two sequences at a PAM distance of \( n \) using:

\[
D = \sum_i \pi_i(1 - M_{ii}^n).
\]
4.3 Constructing a phylogenetic tree

Figure 4.7: Fig 26.13 in Campbell & Reece (2008). All branches in this tree have the same total length from the base of the tree.

Here $M_{ii}$ is the diagonal in the substitution probability matrix from which PAM-1 matrix is generated. Remember that the values in the PAM matrices are log-odds of the substitution probabilities. $M_{nn}^{ii}$ is the diagonal in the substitution probability matrix after $n$ PAM units of time, and $\pi_i$ is the frequency of amino acid $i$. Thus the equation above gives the probability that the amino acid in the first sequence is an $i$ multiplied by the probability that the amino acid in the second sequence has not remained an $i$ after $n$ PAM units of time. Subsequently we need to obtain a similar correction for proteins like Jukes-Cantor performs for nucleotides, i.e., a correction that accounts for multiple substitutions. Kimura showed that one can compute the real evolutionary distance between two protein sequences, $d$, from the fraction of sites that differ, $D$, by

$$d = -\ln(1 - D - 0.2D^2).$$

A distance matrix is the input for clustering algorithms. These algorithms work as the following:

- Join the closest two clusters to form a single larger cluster.
- Recalculate distances between all clusters and the “new” cluster.
- Repeat the above steps until all sequences are joined within one cluster.

We will here discuss two algorithms that are very widely used: UPGMA (see also Chapter 6) and neighbor-joining.

The UPGMA (unweighted pair group method with arithmetic mean) method is one of the simplest examples of a hierarchical clustering method. The distance between two clusters is calculated as the average of the distances from each data point in the first cluster to each other data point in the second cluster. The UPGMA method assumes that there is a strict molecular clock and that all species evolve at the same rate. Thus, if you use UPGMA to draw a tree, all data points are put on a (horizontal) line and the nodes combining clusters are put at a height equal to half the distance between the clusters, because we are assuming that half of the changes separating the two species occurred on the branch leading to one species and half occurred on the other.

If we pick any three species from an UPGMA tree, there are always two that are more closely related to each other than to the third. The distances from the third species to the first and
to the second are equal. The distance from the first to the second is always smaller than this. This property is called **ultrametricity**. An example of an UPGMA tree is given in Fig. 4.7.

Take three species: fish, bird and human. Bird and human are more closely related to each other than they are to fish. From fish to bird is the same as the distance from fish to human. The distance between bird and human is smaller than the fish-bird (or fish-human) distance. In reality many species might evolve with different rates, e.g. a line leading to mouse might be slower than one leading to rat. Therefore, the ultrametric trees generated by UPGMA are often incorrect. However, UPGMA is the simplest clustering algorithm and equal amounts of chronological time can be represented by these ultrametric trees. For example, fish and human had a common ancestor 542 million years ago and the evolution in one branch could be faster than the other, but nevertheless there had been 542 million years of evolution in both branches. UPGMA is also widely used for hierarchical clustering of numeric data (see Chapter 6).

The distance between two clusters in UPGMA method is defined as the following. Consider you have a cluster $X$ containing $N_X$ sequences and cluster $Y$ containing $N_Y$ sequences. Initially, each cluster would only contain a single sequence. The distance between cluster $X$ and $Y$ is defined as the arithmetic mean of the all pair distances between the sequences in $X$ and $Y$:

$$d_{XY} = \frac{1}{N_X N_Y} \sum_{i \in X, j \in Y} d_{ij} \tag{4.1}$$

where $i$ labels all sequences in $X$ and $j$ labels all sequences in $Y$, and $d_{ij}$ is the distance between sequences $i$ and $j$. When $X$ and $Y$ are combined into a new cluster $Z$, the new distances can be defined using existing cluster-to-cluster distances without using sequence pair distances:

$$d_{ZW} = \frac{N_X d_{XW} + N_Y d_{YW}}{N_X + N_Y} \tag{4.2}$$

We have adopted a figure to demonstrate how this would work (see Fig. 4.8).

The neighbor-joining (NJ) algorithm can be seen as an extension of the UPGMA algorithm to produce unrooted trees. Two species in a tree are called neighbors if they are connected via a single node. In Fig. 4.3a, A and B are neighbors, but A and C are not. The NJ method starts with a star-like tree where all tip nodes representing the sequences in our data set are connected to a central node. The method chooses nodes $i$ and $j$ and connects them by introducing a new internal node $k$. Let us assume that distances are additive in the distance matrix, e.g. $d_{AC} = d_{Ai} + d_{ij} + d_{jC}$ in Fig. 4.3a. Since all distances are additive, with the new internal node $k$, we have $d_{ij} = d_{ik} + d_{kj}$. To any other sequence $m$: $d_{im} = d_{ik} + d_{km}$ and $d_{jm} = d_{jk} + d_{km}$. Solving these three equations gives: $d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij})$. Now we have to define $d_{ik}$ and $d_{jk}$. If there are $N$ sequences in total we can define $d_{ik} = (d_{ij} + r_i - r_j)/2$, $d_{jk} = (d_{ij} - r_i + r_j)/2$, where $r_i = \frac{1}{N-2} \sum_m d_{im}$ and $r_j = \frac{1}{N-2} \sum_m d_{jm}$.

But how to choose nodes $i$ and $j$ in the first place. UPGMA chooses the two nodes that are closest to each other. But we are not always sure if the nodes that have the closest distance to each other are also neighbors. Take B and D in Fig. 4.3a. They are closest to each other, however, they are not neighbors, so it would be wrong to connect them. The NJ method computes all modified distances $d^*_{ij} = d_{ij} - r_i - r_j$ and chooses the nodes that have the smallest modified distance. It has been shown that the nodes that are chosen in this way...
4.3 Constructing a phylogenetic tree

must be neighbors (Durbin et al., 1998b). After each iteration of the algorithm the number of unprocessed nodes decreases by one, and the algorithm stops when all the nodes are connected to a single tree. Neighbor-joining generally yields a much better phylogenetic reconstruction than UPGMA. One of the reasons for this improvement is that neighbor-joining, in contrast to UPGMA, does not assume equal rates of evolution.

We have adopted a figure from Zvelebil & Baum (2008) to demonstrate how the NJ algorithm works (see Fig. 4.9).

How to search for all possible trees?

Once we generate an initial tree using clustering methods, we have to search for the optimal tree in the tree space. Since the number of tree topologies is enormous (number of unrooted trees with \( N \) species is \((2^N - 5)!!\) and number of rooted trees is \((2N - 3)!!\), it is not possible to test every possible tree exhaustively if we have more than, say, eight species. Instead we have to make a heuristic search in the tree space. In an heuristic search rather than trying all possible trees, you try and focus on trees that seem to be getting you nearer to the optimal tree. Of course, you generally can not be sure that you are really near the optimal tree. But we might be able to have a good guess. Heuristics are used to help us make that guess.

Suppose we have an initial guess at an optimum tree. This could be the tree produced by distance-matrix methods, for example. This tree is possibly not “too bad”, i.e. it is wise to
Figure 4.9: Figure 8.7 in Zvelebil & Baum (2008): a worked example of the NJ method of phylogenetic construction for five sequences.
4.3 Constructing a phylogenetic tree

start our search for optimal trees from the trees that are similar to this initial tree. If we find a neighboring tree (i.e. a tree that has only a single subgroup different than our initial tree) that is better, then we move to this new tree and search for its neighbors for the optimal tree. We keep doing this until a tree is the local optimum, i.e. it has no nearest neighbors that are better than the tree itself. We have now no guarantee that this local optimum tree is also the global optimum one. One way of checking this is to have several different initial trees and repeat the process of finding the local optimum tree. Sometimes long-range moves (i.e. testing a tree where some species are randomly ordered) can improve the search, and search programs typically use a combination of nearest neighbors and few long range moves.

Parsimony

Parsimony is probably the most widely used of all tree building algorithms. The main idea is to find the tree which can explain the observed sequences with minimum number of substitutions. It assigns a cost to a tree and search through all trees (or finds a clever way of doing it, see below) to find the “best” tree.

Each column in the sequence alignment is treated as a “site”. Since we do not know what the ancestral sequence looked like, there is no distinction between loss or gain of a site. Let us use Fig. 4.10 to explain the algorithm. Assume that a site is a C in human, chimpanzee and gorilla, while it is a T in orangutan and gibbon. The tree in Fig. 4.10a needs one substitution to occur at “***”, whereas in Fig. 4.10b we need two substitution event to occur. Based on this site tree in (a) is preferable over (b). Not all sites are informative for parsimony approach. Obviously, if a site is identical in every sequence it is non-informative. Moreover, the sites shown in Fig. 4.10c-d are also non-informative, because only a single species has a substitution and it does not help us to distinguish between trees. In general a position is informative (independent of number of sequences being used) if it has at least two different characters (nucleotides or amino acids) and each of these occur at least twice in the data set. Another example of how parsimony works is given in Campbell & Reece (2008) in Figure 26.15.

Figure 4.10: Figure 8.14 in Higgs & Attwood (2005). In the text we use this example to explain the principle of molecular parsimony.
The parsimony approach suffers from a so-called “long-branch attraction”: if two branches are long due to rapid evolutionary rates along them, many tree construction methods tend to put them together. Parsimony is a good method when we know very little about the evolutionary events that took place. However, if we have a good quantitative model of evolution, maximum likelihood methods are superior to parsimony.

Maximum Likelihood

The method of maximum likelihood (ML) attempts to reconstruct a phylogeny using an explicit model of evolution, e.g. assume that all sites in the sequences will be selected neutrally and that they mutate at a constant rate per gamete per generation. Using such an evolutionary model we can calculate the likelihood that the known sequences would evolve on a tree. ML criteria chooses the tree that maximizes this likelihood.

There are some advantages of ML methods over other methods:
- they have often lower variance than other methods (i.e. ML methods are least affected by sampling error)
- they tend to be robust to many violations of the assumptions in the evolutionary model
- even with very short sequences they tend to outperform alternative methods such as parsimony or distance methods.
- ML methods are statistically well founded
- they evaluate different tree topologies

Bootstrapping

Bootstrapping is a widely used technique to measure the reliability of reconstructed trees. This approach creates a number of re-sampled versions of the original alignment in order to establish how firmly the data, i.e. the alignment, supports the conclusion, i.e. the tree. Each resampling consists of randomly drawing positions (columns) from the original alignment and placing them in a new “resampled alignment” (as exemplified in Fig. 4.11). When we draw a position we do not remove it from the original alignment and hence it can be drawn again. The same number of positions is drawn as are present in the original alignment. This procedure thus results in a bootstrap alignment that is of equal length as the original alignment, but which contains a randomly re-weighted sub-sample of the original alignment.

When we construct a tree with this resampled alignment, we are not guaranteed to get the same tree as from the original alignment. If the phylogenetic signal is strong in the original alignment, there are many positions that support it and it will be detected even in a resampled alignment. If however the phylogenetic signal is weak, resampling will diminish it. Another way to look at this is to say that if the reconstruction of the initial phylogenetic tree depended on just few “noisy” positions in the alignment (especially if the alignment is short), the bootstrap resampling should reveal this, because it will often reconstruct a different phylogeny based on resampling that left the “noisy” positions out.

To carry out a bootstrap analysis, the resampling process is usually repeated 100 to 1000 times and for each resampled alignment a tree is generated. Then we look for each group of
Figure 4.11: This figure illustrates how alignments are resampled by the bootstrap procedure. The X, Y and Z in the first column are the names of the sequences, and the 1,2,3,4 and 5 in the top row are the position numbers in the original alignment. Each resampled alignment has random compositions resampled from the positions of the original alignment (given in the upper row). Note that in a real bootstrap at least a 100 resamples are created, in contrast to the three shown in this example.

species in the original tree and determine what percentage of the randomized trees contain this same group. This percentage gives us a measure of confidence that those species really do form a related group. Often values greater than 70% are thought to be reasonably strong evidence for a “true” clade.

4.4 Horizontal gene transfer

Tree-like presentations for taxonomic classification have an illustrious history in evolutionary biology, but they do not provide a complete representation of life’s history, especially for prokaryotes. Some genes within an organism have tree-like histories that differ from the histories of other genes within the same organism, owing to horizontal gene transfer (HGT). Horizontal gene transfer is defined as which information travels through the generations as the cell divides. It is most often thought of as a mechanism for the mobilization of chromosomal DNA among bacterial cells.

Three main mechanisms of HGT have been described: natural transformation, the uptake of free DNA in competent species, exhibited by about 1% of validly described bacterial species; transduction, the transfer of DNA between an infected cell and its infecting agent; and conjugation, the transfer of mobile genetic elements by specialized structures assembled between two adjacent cells. Although all types of genes can be susceptible to horizontal transfer, different types of genes and groups of organisms vary in their propensity for HGT.

Keeping the species tree in mind, it is possible to detect HGT in gene trees. In Fig. 4.12 an example of a gene (N-acetylneuraminate lyase) transfer is demonstrated. The vertebrate version of this gene clusters very convincingly with Vibrio cholerae and Yersinia pestis genes, indicative of lateral gene transfer involving bacteria and eukaryotes.
Figure 4.12: Phylogeny of the gene encoding N-acetylneuraminate lyase (adapted from Andersson et al. (2001)). Bacteria donated this gene to the protozoan parasite T. vaginalis. Vertebrates (human, mouse, and pig) together with two bacterial lineages (Vibrio and Yersinia) also show a branching pattern indicative of gene transfer, although it is not possible to infer the direction of the transfer.

**Protocol Phylogenic Trees**

1. From the course website under the Data link (http://theory.bio.uu.nl/BPA/Data/) download the file Phlogeny.txt. The file contains protein sequences of histone acetyltransferases from a number of vertebrates (dog (CANFA), human, rat, mouse, frog (XENTR), green puffer fish (TETNG), puffer fish (FUGRU), stickleback (GASAC), cow (BOVIN), chicken and opossum (MONDO)), sea squirts (CIOSA and CIOIN) and insects (fruit fly (DROME) and mosquito (AEDAE)).

2. Note that this file contains unaligned protein sequences. To construct a phylogenetic tree, you first need to align the sequences. Make an alignment using ClustalW (http://www.ebi.ac.uk/clustalw) as in protocol in Chapter 2: paste the sequence in the large textbox, use default settings and make sure that slow (or full) algorithm is chosen. Click Submit.

3. For the next step you need the alignment file which you get by clicking the Download Alignment File button.

4. Open the EBI Phylogeny website (http://www.ebi.ac.uk/Tools/phylogeny/clustalw2/) and paste the multiple sequence alignment into the large textbox. Leave the settings at their default values and make sure that the clustering method is neighbor-joining. Click the Submit button.

5. The page with results will load within few moments. On top, titled Phylogenetic tree, there is a text representation of our gene tree. You can find a more intuitive graphic representation at the bottom of the page.

6. The tree is unrooted. If we wanted to root the tree, we would use insects as an outgroup. The length of branches reflects the rate of divergence between protein sequences. In other words, the sequence which changed more is represented with a longer branch.
7. Clicking on the tree will bring up the menu where you can modify how the tree looks. You can, for example, change the color of the branches with *line colour*. *View → Reload* will reset the modifications you have made.

**Obligatory Reading**

- Campbell & Reece (2008) chapter 26

**Optional Reading**


### 4.5 Self-test

(adapted from Campbell & Reece (2008))

1. Which mutation should least require realignment of homologous regions of a gene that is common to several related species?
   
   A) 3-base insertion  
   B) 1-base substitution  
   C) 4-base insertion  
   D) 1-base deletion  
   E) 3-base deletion

Use the tree on the next page for the following questions.
2. A common ancestor for both species C and E could be at position number
   
   A) 1.
   B) 2.
   C) 3.
   D) 4.
   E) 5.

3. The two currently living species that are most closely related to each other are
   
   A) A and B.
   B) B and C.
   C) C and D.
   D) D and E.
   E) E and A.

4. Which species are extinct?
   
   A) A and E
   B) A and B
   C) C and D
   D) D and E
   E) cannot be determined from the information provided

5. If this evolutionary tree is an accurate depiction of relatedness, then which of the following should be correct?
   1. The last common ancestor of species B and C occurred more recently than the last common ancestor of species D and E.
   2. Species A is the direct ancestor of both species B and species C.
   3. The species present at position 3 is ancestral to C, D, and E.
   
   A) 3
   B) 2
   C) 1 and 2
   D) 1 and 3
   E) 1

6. A taxon of which all members have the same common ancestor is
   
   A) paraphyletic.
   B) polyphyletic.
   C) monophyletic.
The following questions refer to the table below, which compares the % sequence similarity of four different parts (2 introns and 2 exons) of a gene that is found in five different eukaryotic species. Each part is numbered to indicate its distance from the promoter (e.g., Intron I is that closest to the promoter). The data reported for Species A were obtained by comparing DNA from one member of species A to another member of Species A. For the other species (B, C, D, and E), the similarity to species A is indicated.

<table>
<thead>
<tr>
<th>Species</th>
<th>Intron I</th>
<th>Exon I</th>
<th>Intron VI</th>
<th>Exon V</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>B</td>
<td>98%</td>
<td>99%</td>
<td>82%</td>
<td>96%</td>
</tr>
<tr>
<td>C</td>
<td>98%</td>
<td>99%</td>
<td>89%</td>
<td>96%</td>
</tr>
<tr>
<td>D</td>
<td>99%</td>
<td>99%</td>
<td>92%</td>
<td>97%</td>
</tr>
<tr>
<td>E</td>
<td>98%</td>
<td>99%</td>
<td>80%</td>
<td>94%</td>
</tr>
</tbody>
</table>

7. Which of these is the best explanation for the high degree of sequence similarity observed in Exon I among these five species?

A) It is the most-upstream exon of this gene.
B) Due to alternative gene splicing, this exon is often treated as an intron.
C) It codes for a polypeptide domain that has a crucial function.
D) These five species must actually constitute a single species.
E) This exon is rich in G-C base pairs; thus, is more stable.

8. Which of these is the best explanation for the relatively low level of sequence similarity observed in Intron VI?

A) Mutations that occur here are neutral; thus, are neither selected for nor against, and thereby accumulate over time.
B) Its higher mutation rate has resulted in its highly conserved nature.
C) The occurrence of molecular homoplasy explains it.
D) This intron is not actually homologous, having resulted from separate bacteriophage-induced transduction events in these five species.

9. Which of these is the best explanation for Intron I’s relatively high sequence similarity among these five species?

A) It is the most-upstream of this gene’s introns.
B) It was once an exon, but became intronic in the common ancestor of these five species.
C) Due to alternative gene splicing, it is often treated as an exon in these five species; as an exon, it codes for an important part of a polypeptide.

10. Which of these four gene parts should allow the construction of the most accurate phylogenetic tree, assuming that this is the only part of the gene that has acted as a reliable molecular clock?

A) Intron I
B) Exon I
C) Intron VI
11. The four-chambered hearts of birds and the four-chambered hearts of mammals evolved independently of each other. If one were unaware of this independence, then one might logically conclude that

A) the birds were the first to evolve a 4-chambered heart.
B) birds and mammals are more distantly related than is actually the case.
C) early mammals possessed feathers.
D) the common ancestor of birds and mammals had a four-chambered heart.
E) birds and mammals should be placed in the same family.

12. A researcher wants to determine the genetic relatedness of several breeds of dog (Canis familiaris). The researcher should compare homologous sequences of ________ that are known to be ________.

A) carbohydrates; poorly conserved
B) fatty acids; highly conserved
C) lipids; poorly conserved
D) proteins or nucleic acids; poorly conserved
E) amino acids; highly conserved

13. Species that are not closely related and that do not share many anatomical similarities can still be placed together on the same phylogenetic tree by comparing their

A) plasmids.
B) chloroplast genomes.
C) mitochondrial genomes.
D) homologous genes that are poorly conserved.
E) homologous genes that are highly conserved.

14. Which kind of DNA should provide the best molecular clock for gauging the evolutionary relatedness of several species whose common ancestor became extinct billions of years ago?

A) that coding for ribosomal RNA
B) intronic DNA belonging to a gene whose product performs a crucial function
C) paralogous DNA that has lost its function (i.e., no longer codes for functional gene product)
D) mitochondrial DNA
E) exonic DNA that codes for a non-crucial part of a polypeptide

15. Which statement represents the best explanation for the observation that the nuclear DNA of wolves and domestic dogs has a very high degree of homology?

A) Dogs and wolves have very similar morphologies.
B) Dogs and wolves belong to the same order.
C) Dogs and wolves are both members of the order Carnivora.
D) Dogs and wolves shared a common ancestor very recently.
E) Convergent evolution has occurred.

16. What is true of gene duplication
   A) It is a type of point mutation.
   B) Its occurrence is limited to diploid species.
   C) Its occurrence is limited to organisms without functional DNA-repair enzymes.
   D) It is most similar in its effects to a deletion mutation.
   E) It can increase the size of a genome over evolutionary time.

17. Paralogous genes that have lost the function of coding for a functional gene product are known as “pseudogenes.” Which of these is a valid prediction regarding the fate of pseudogenes over evolutionary time?
   A) They will be preserved by natural selection.
   B) They will be highly conserved.
   C) They will ultimately regain their original function.
   D) They will be transformed into orthologous genes.
   E) They will have relatively high number of mutations because of not having selection pressure.

4.6 Exercises (pen and paper)

**Question 4.1. Jukes-Cantor**

Make a graph of the Jukes-Cantor model (i.e. $d$, distance between two sequences, as a function of $D$, the fraction of sites that differ between two sequences) using your graphical calculator. Explain how $d$ depends on $D$. If two sequences evolve by random drift and they are observed to differ 20% of sites, what is the Jukes-Cantor distance between these sequences?

**Question 4.2. UPGMA**

Construct a phylogenetic tree using the UPGMA method from the following distance matrix:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td>B</td>
<td>0.2</td>
<td>-</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>C</td>
<td>0.3</td>
<td>0.4</td>
<td>-</td>
<td>0.5</td>
</tr>
<tr>
<td>D</td>
<td>0.4</td>
<td>0.4</td>
<td>0.5</td>
<td>-</td>
</tr>
</tbody>
</table>

Calculate branch lengths.
Question 4.3. A simple tree

Four species (A,B,C,D) are characterized by the respective homologous sequences ATCC, ATGC, TTCG and TCGG. Use the number of substitutions/mismatches as the distance between each pair of sequences, and the UPGMA method to derive a phylogenetic tree. Calculate branch lengths.

Question 4.4. Tree reading

Look at the four trees below. The branch lengths are not drawn to scale.

a. Assume the trees were rooted, which trees would be equivalent?
b. Consider them unrooted and redraw them as such (i.e. star representation). Which trees are then equivalent?

Question 4.5. Parsimony, I

Given the alignment of a protein coding region below (assume that 1st position is also 1st position in the codon):

a. Do synonymous substitutions occur more often than non-synonymous substitutions?
b. Do transversions occur more often than transitions in this alignment? (see Section 2.2)
c. Where are the informative sites for parsimony method?
d. Draw a phylogenetic tree using parsimony method.

<table>
<thead>
<tr>
<th>Hedgehog</th>
<th>GTGAATGAATGGGCTTTCCAGAAGTGAACTG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>GTTAATGAGTGTTTTCAGAAGTGAACTG</td>
</tr>
<tr>
<td>Hare</td>
<td>GTTAACGAGTGTTTTCAGAAGTGAAATG</td>
</tr>
<tr>
<td>Wombat</td>
<td>GTTAATGAGTGTTATCCAGAAGTGAGATA</td>
</tr>
<tr>
<td>Opossum</td>
<td>GTTAATGAGTGTTATCCAGAAGTGAGATA</td>
</tr>
</tbody>
</table>
Question 4.6. Parsimony, II

Use the Parsimony method to derive an unrooted phylogenetic tree for the following alignment.

<table>
<thead>
<tr>
<th>Species</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fugu</td>
<td>ATGTC GGGAA ACCT</td>
</tr>
<tr>
<td>ZebraFish</td>
<td>ATGTC GGGAA ACTGT</td>
</tr>
<tr>
<td>Trout</td>
<td>ATGGC AAGGA ACTCT</td>
</tr>
<tr>
<td>Salmon</td>
<td>ATGGC AAGGA ACTCT</td>
</tr>
<tr>
<td>Halibut</td>
<td>ATGGC AGGGA AATCT</td>
</tr>
<tr>
<td>Shark</td>
<td>ACTGA TGCGG GATCA</td>
</tr>
</tbody>
</table>

Question 4.7. IL-11

Interleukin-11 (IL-11) is a key cytokine in the regulation of proliferation and differentiation of hematopoietic progenitors and is also involved in bone formation, and protection of mucosal epithelium from pathogens. Huising et al. (2005) studied the evolution of this gene. Below is part of their phylogenetic analysis. Explain from this tree where gene duplication, gene loss and horizontal gene transfer might have taken place.

Question 4.8. Cyclin-dependent kinase

A phylogenetic tree of the cyclin-dependent kinase (CDK) family in eukaryotes was constructed by the neighbor-joining method and is given below (S. pombe is a fungus, C. elegans is a worm, Xenopus is a frog, Arabidopsis is a plant species). This figure is taken from Hughes et al. (2001).

a. Use this tree to predict the number of CDK genes in the common ancestor of eukaryotes.

b. Mention two gene loss and two gene duplication events.

c. Looking at the bootstrap values, how often do you expect a clade with only the Human CDKH gene and the Xenopus CDKH1 gene?
Question 4.9. Bootstrapping, I

In a bootstrap experiment, where you generate in total 100 trees, you observe the following:

72 trees 19 trees 9 trees

Draw the most likely tree with bootstrap values.

Question 4.10. Bootstrapping, II

Given the tree with bootstrap values below:

a. How often are W+X not together as a clade (i.e. grouping of two species) in the 100 bootstrap trees from the resampled alignments?

b. Could the clade W+V be present 14 times in the 100 bootstrap trees?

c. Could the clade W+Y+Z be present 14 times in the 100 bootstrap trees?
**Question 4.11. COX genes**

An unrooted phylogenetic tree of the COX genes was constructed by the neighbor-joining method and is given below (the tree is adapted from Zou *et al.* (1999)).

**a.** Draw this unrooted tree in rooted form. Use your biological knowledge to decide where the root can be. Make sure that your tree is the most parsimonious one.

**b.** Indicate where possible gene duplications, and gene losses have occurred.

**c.** We are performing 100 bootstrap experiments, and find that 25 times Human COX1 and Mouse COX1 gene form a clade alone. Based on this add a single bootstrap value to the COX gene tree.

**d.** In a more recent study the ortholog of human COX1 gene was found in trout. Does this finding change any of your answers in A, B, and C? If yes, what would be the new answers?
Chapter 5

Hidden Markov Models and Sequence logos

The first part of this chapter is adapted from Krogh (1998) with permission from Anders Krogh.

Very efficient programs text searching are available on many computers. The same methods can be used for searching for patterns in biological sequences, but often they fail. This is because biological ‘spelling’ is much more sloppy than English spelling: proteins with the same function from two different organisms are almost certainly spelled differently, that is, the two amino acid sequences differ. It is not rare that two such homologous sequences have less than 30% identical amino acids. Similarly in DNA many interesting patterns vary greatly even within the same genome. Well-known examples are ribosome binding sites and splice sites, but the list is long. Fortunately there are usually still subtle similarities between two such sequences, and the question is how to detect these similarities.

The variation in a family of sequences can be described statistically, and this is the basis for most methods used in biological sequence analysis, see Durbin et al. (1998a) for a presentation of some of these statistical approaches. For pairwise alignments, for instance, the probability that a certain residue mutates to another residue is used in a substitution matrix, such as one of the PAM matrices. For pattern finding in DNA, e.g. splice sites, a weight matrix is very often used. This is simply a position specific score calculated from the frequencies of the four nucleotides at all the positions in known cases. Similarly, methods for finding genes use, almost without exception, the statistics of codons or di-codons in some form or other.

A method that is widely used in biology for detection patterns is a hidden Markov model (HMM). An HMM is a statistical model, which is very well suited for many tasks in molecular biology, although they have been mostly developed for speech recognition since the early 1970s. The most popular use of the HMM in molecular biology is as a ‘probabilistic profile’ of a protein family, which is called a profile HMM. From a family of proteins (or DNA) a profile HMM can be made, that can subsequently be used to search for a database for other members of the family.

The HMM can be applied to other types of problems. It is particularly well suited for problems
Hidden Markov Models and Sequence logos

with a simple ‘grammatical structure,’ such as gene finding. In gene finding several patterns must be recognized and combined into a prediction of exons and introns, and the prediction must follow various restrictions to make it a reasonable gene prediction, e.g. a gene can not stop before it starts. An HMM can recognize several patterns, and it can be made such that the predictions always follow the restrictions of a gene structure.

Since much of the literature on HMMs is a little hard to read for many biologists, we will attempt in this chapter to give a non-mathematical introduction to HMMs. First, HMMs are introduced by an example and then profile HMMs are described. Then an HMM for finding eukaryotic genes is sketched, and finally pointers to the literature are given.

5.1 From regular expressions to HMMs

Regular expressions can also be used to characterize protein families, which is the basis for the PROSITE database (Bairoch et al., 1997). This is a very elegant and efficient way to search for some protein families, but difficult for other. As already mentioned in the introduction, the difficulties arise because protein spelling is much more free than English spelling. Therefore the regular expressions sometimes need to be very broad and complex. Imagine a DNA motif like this:

\[
\begin{align*}
A & \ C & A & - & - & - & A & T & G \\
T & C & A & A & C & T & A & T & C \\
A & G & A & - & - & - & A & T & C \\
A & C & C & G & - & - & A & T & C
\end{align*}
\]

A regular expression for this is:

\[
[AT] \ [CG] \ [AC] \ [ACGT]^* \ A \ [TG] \ [GC]
\]

meaning that the first position is A or T, the second C or G, and so forth. The term ‘[ACGT]^*’ means that any of the four letters can occur any number of times. The problem with the above regular expression is that it does not in any way distinguish between the highly implausible sequence

\[
T \ G \ C \ T & - & - & A & G & G
\]

which has the exceptional character in each position, and the consensus sequence

\[
A \ C \ A \ C & - & - & A \ T \ C
\]

with the most common character in each position (the dashes are just for aligning these sequences with the previous ones). What is meant by ‘exceptional’ sequence can of course
Figure 5.1: A hidden Markov model derived from the alignment discussed in the text. The transitions are shown with arrows whose thickness indicate their probability. In each state the histogram shows the probabilities of the four nucleotides.

be debated, although most would probably agree that the first sequence is not likely to be the same motif as the 5 sequences above. It is possible to make the regular expression more discriminative by splitting it into several different ones, but it easily becomes messy. The alternative is to score sequences by how well they fit the alignment.

To score a sequence, we say that there is a probability of \( \frac{4}{5} = 0.8 \) for an A in the first position and \( \frac{1}{5} = 0.2 \) for a T, because we observe that out of 5 letters 4 are As and one is a T. Similarly in the second position the probability of C is \( \frac{4}{5} \) and of G \( \frac{1}{5} \), and so forth. After the third position in the alignment, 3 out of 5 sequences have ‘insertions’ of varying lengths, so we say the probability of making an insertion is \( \frac{3}{5} \) and thus \( \frac{2}{5} \) for not making one. To keep track of these numbers a diagram can be drawn with probabilities as in Fig. 5.1.

This is a hidden Markov model. A box in the drawing is called a state, and there is a state for each term in the regular expression. All the probabilities are found simply by counting in the multiple alignment how many times each event occurs, just as described above. The only part that might seem tricky is the ‘insertion,’ which is represented by the state above the other states. The probability of each letter is found by counting all occurrences of the four nucleotides in this region of the alignment. The total counts are one A, two Cs, one G, and one T, yielding probabilities \( \frac{1}{5}, \frac{2}{5}, \frac{1}{5}, \) and \( \frac{1}{5} \) respectively. These probabilities are called emission probabilities. After sequences 2, 3 and 5 have made one insertion each, there are two more insertions (from sequence 2) and the total number of transitions back to the main line of states is 3 (all three sequences with insertions have to finish). Therefore there are 5 transitions in total from the insert state, and the probability of making a transition to itself is \( \frac{2}{5} \) and the probability of making one to the next state is \( \frac{3}{5} \). These probabilities are called transition probabilities.

It is now easy to score the consensus sequence. The probability of the first A is \( \frac{4}{5} \). This is multiplied by the probability of the transition from the first state to the second, which is 1. Continuing this, the total probability of the consensus is:

\[
P(ACACATC) = 0.8 \times 1 \times 0.8 \times 1 \times 0.8 \times 0.6 \times 0.4 \times 0.6 \times 1 \times 1 \times 0.8 \times 1 \times 0.8 \approx 4.7 \times 10^{-2}.
\]
Table 5.1: Probabilities and log-odds scores for the 5 sequences in the alignment and for the consensus sequence and the ‘exceptional’ sequence.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Probability × 100</th>
<th>Log odds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consensus</td>
<td>4.7</td>
<td>6.7</td>
</tr>
<tr>
<td>Original sequences</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCAC</td>
<td>3.3</td>
<td>4.9</td>
</tr>
<tr>
<td>TCAACTAC</td>
<td>0.0075</td>
<td>3.0</td>
</tr>
<tr>
<td>ACCAAGC</td>
<td>1.2</td>
<td>5.3</td>
</tr>
<tr>
<td>AGAAC</td>
<td>3.3</td>
<td>4.9</td>
</tr>
<tr>
<td>ACCAG</td>
<td>0.59</td>
<td>4.6</td>
</tr>
<tr>
<td>Exceptional</td>
<td>0.0023</td>
<td>-0.97</td>
</tr>
</tbody>
</table>

Making the same calculation for the exceptional sequence (TGCT–AGG) yields only $0.0023 \times 10^{-2}$, which is roughly 2000 times smaller than for the consensus. This way we achieved the goal of getting a score for each sequence, a measure of how well a sequence fits the motif.

The same probability can be calculated for the four original sequences in the alignment in exactly the same way, and the result is shown in Table 5.1. The probability depends very strongly on the length of the sequence. Therefore the probability itself is not the most convenient number to use as a score, and the log-odds score shown in the last column of the table is usually better. It is the logarithm of the probability of the sequence divided by the probability according to a null model. The null model is one that treats the sequences as random strings of nucleotides, so the probability of a sequence of length $L$ is $0.25^L$. Then the log-odds score is

$$\log\text{– odds for sequence } S = \log \left( \frac{P(S)}{0.25^L} \right) = \log P(S) - L \log 0.25$$

We have used the natural logarithm in Table 5.1. Logarithms are proportional, so it does not really matter which one you use; it is quite common to use the logarithm base 2. One can of course use other null models instead. Often one would use the over-all nucleotide frequencies in the organism studied instead of just 0.25.

When a sequence fits the motif very well the log-odds is high. When it fits the null model better, the log-odds score is negative. Although the raw probability of the second sequence (the one with three inserts) is almost as low as that of the exceptional sequence, notice that the log-odds score is much higher than for the exceptional sequence, and the discrimination is very good. Unfortunately, one cannot always assume that anything with a positive log-odds score is ‘a hit’, because there are random hits if one is searching a large database.

If the alignment had no gaps or insertions we would get rid of the insert state, and then all the probabilities associated with the arrows (the transition probabilities) would be 1 and might as well be ignored completely. Then the HMM works exactly as a weight matrix of log-odds scores.
5.2 Profile HMMs

A profile HMM is a certain type of HMM with a structure that in a natural way allows position dependent gap penalties. A profile HMM can be obtained from a multiple alignment and can be used for searching a database for other members of the family in the alignment. The structure of the model is shown in Fig. 5.2. The bottom line of states are called the main states, because they model the columns of the alignment. In these states the probability distribution is the frequency of the amino acids or nucleotides as in the above model of the DNA motif. The second row of diamond shaped states are called insert states and are used to model highly variable regions in the alignment. They function exactly like the top state in Fig. 5.1, although one might choose to use a fixed distribution of residues, e.g. the overall distribution of amino acids, instead of calculating the distribution as in the example above. The top line of circular states are called delete states. These are a different type of state, called a silent or null state. They do not match any residues, and they are there merely to make it possible to jump over one or more columns in the alignment, i.e., to model the situation when just a few of the sequences have a ‘-’ in the multiple alignment at a position. Let us turn to an example.

FRY--IQGL
FRVIPVITL
GRVPVTIGK
GRF--TQTK
GRFPITITL
RRV-IVQGK
RRVL-TQGL
RRYL-TTTL

Suppose you have a multiple alignment as the one shown above. There are a few ways of constructing a profile HMM for this alignment. Here is one of them, in which we will model positions 4 and 5 as insertions (as in Fig. 5.1). The profile HMM should model the sequences, not the alignment. The alignment is a guide for us to make HMM model. Therefore, we do not need to worry about the order of insertions that occur at the 4th and 5th position.
Figure 5.3: A profile HMM made from the alignment given in the text. This is only one of the possible models. Transitions with probability zero are not shown.

The HMM will look like the one given in Fig. 5.3. Within each box the emission probabilities of amino acids are shown. The amino acids with zero emission probabilities are not shown for clarity. The transitions are depicted by the arrows, and their probabilities are shown above them. If the transition probability is 1, it is not shown.

The rectangular boxes represent the main states, that occur at position 1, 2, 3, 6, 7, 8. From the third state, we notice that only two sequence of the 8 continues to the main state in position 6 (sequence number 1 and 4), and such a transition is represented with probability 2/8. In the other 6 sequences we enter the insertion state. Here we have to count the number of times a sequence remains in the insertion state (a transition from insertion state back to insertion state). For example:

Sequence FRVIPVITL remains in the insertion state for one more cycles. This is because we already have taken into account that the first insertion (I) would occur with 6/8 likelihood (seen in the transition from main state 3 to insertion state). Sequence RRVL-TQGL on the other hand, goes through the loop only once (1 insertion, L), which occurs after V. By counting up all times that the sequences remain in the insertion state, we arrive at a total of 3 insertion-insertion transitions. So, the probability of remaining in the insertion node is, (and noticing that all sequences will have to go back to the main state eventually, i.e. 6 times):

\[
\frac{\text{transition}_{\text{ins} \rightarrow \text{ins}}}{\text{transition}_{\text{ins} \rightarrow \text{ins}} + \text{transition}_{\text{ins} \rightarrow \text{main}}} = \frac{3}{3 + 6} = \frac{1}{3}
\]

Following the same reasoning, we obtain the transmission probability of returning to the main state: \(\frac{2}{3}\).

**Searching a database**

Above we saw how to calculate the probability of a sequence in the alignment by multiplying all the probabilities (or adding the log-odds scores) in the model along the path followed by that particular sequence. However, this path is usually not known for other sequences which
5.2 Profile HMMs

The distribution of log-odds scores from a search of Swissprot with a profile HMM of the SH3 domain. The dark area of the histogram represents the sequences with an annotated SH3 domain, and the light those that are not annotated as having one. This is for illustrative purposes only, and the sequences with log-odds around zero were not investigated further.

are not part of the original alignment, and the next problem is how to score such a sequence. Obviously, if we can find a path through the model where the new sequence fits well, then we can score the sequence as before. We need to ‘align’ the sequence to the model. This resembles very much the pairwise alignment problem, where two sequences are aligned so that they are most similar, and indeed the same type of dynamic programming algorithm can be used.

For a particular sequence, an alignment to the model (or a path) is an assignment of states to each residue in the sequence. There are many such alignments for a given sequence. For instance an alignment might be as follows. Let us label the amino acids in a protein as A1, A2, A3, etc. Similarly we can label the HMM states as M1, M2, M3, etc. for main states, I1, I2, I3 for insert states, and so on. Then an alignment could have A1 → M1, A2 and A3 → I1, A4 → M2, A5 → M6 (after passing through three delete states), and so on. For each such path we can calculate the probability of the sequence or the log-odds score, and thus we can find the best alignment, i.e., the one with the largest probability. Although there are an enormous number of possible alignments it can be done efficiently by the above mentioned dynamic programming algorithm, which is called the Viterbi algorithm. The algorithm also gives the probability of the sequence for that alignment, and thus a score is obtained.

The log-odds score found in this manner can be used to search databases for members of the same family. A typical distribution of scores from such a search is shown in Fig. 5.4. As is also the case with other types of searches, there is no clear-cut separation of true and false positives.

An alternative way of scoring sequences is to sum the probabilities of all possible alignments of the sequence to the model. This probability can be found by a similar algorithm called the forward algorithm. This type of scoring is not very common in biological sequence comparison, but it is more natural from a probabilistic point of view. However, it usually gives very similar results.

Let us demonstrate use of profile HMMs in database searches by giving an example. Pfam (pfam.wustl.edu) is a database of multiple alignments of protein domains or conserved protein
Hidden Markov Models and Sequence logos

regions. The alignments represent some evolutionary conserved structure which has implications for the protein’s function. From each alignment a profile HMMs is built. These models are very useful for automatically recognizing that a new protein belongs to an existing protein family, even if the homology is weak.

Now imagine we have a new frog protein, and we want to predict which domains it has. Submitting this protein to Pfam HMMs might generate an output like:

<table>
<thead>
<tr>
<th>Model</th>
<th>from</th>
<th>to</th>
<th>Score</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cytochrome_B</td>
<td>10</td>
<td>205</td>
<td>430.2</td>
<td>2.6e-126</td>
</tr>
</tbody>
</table>

This output means that our sequence has a cytochrome b domain, because from position 10 to 205 it fits very well (score 430.2, the chance that this score is obtained randomly is $2.6 \times 10^{-126}$) to Pfam HMM build for cytochrome B (the proteins you analyzed during web exercises for multiple alignments) protein sequences. Here the Viterbi algorithm is used for finding the best path in more than 8000 profile HMMs, each one describing one protein family. As an output we get the best matching model.

5.3 Information content in biological sequences

Evolutionary change has been observed in the fossil record, in the field, in the laboratory, and at the molecular level in DNA and protein sequences. A general method for quantifying these changes is the well-established mathematics of information theory (Shannon, 1948). This theory can be used to measure information content of protein, DNA or RNA sequences. Moreover, changes in this measure provides us a tool to estimate the speed of evolution. The information content is especially important in supervised bioinformatics (see Chapter 5), because a prediction method can only be developed if the existing data contains sufficient information about the biological question we study.

Here we will use the binding motif of major histocompatibility complex (MHC) molecules as an example to explain the concept of information in biological sequences. MHC molecules present protein fragments (peptides) generated in the cell. This presentation enables the immune system to see (intracellular) pathogens, in the form of small peptides. MHC molecules have the largest polymorphism (diversity) known in mammals so far, i.e. most individuals carry a combination of different MHC molecules. On a population basis, hundreds of alleles (alternative versions of a gene) have been found for most of the MHC encoding locus (a fix position on a chromosome). On an individual basis, only one (homozygous) or two (heterozygous) of these alleles are expressed per locus. A collection of sequences known to bind a certain MHC molecule can be used to construct a simple algorithm to predict MHC binding. Table 5.2 shows a set of peptide sequences known to bind to the A*0201 human MHC molecule.

One of the possible binding motifs based on the data given in Table 5.2 is:

$X_1[LMIV]_2X_3X_4X_5X_6X_7X_8[MNIV]_9$,

where $X_i$ indicates that all amino acids are allowed at position $i$, and $[LMIV]_2$ indicates that only the specified amino acids L, M, I, and V are allowed at position 2. According to this...
5.4 Information Carried by Biological Sequences

A number of different information measures can be used to characterize a motif in a biological sequence. Most widely used are entropy and relative entropy.

Entropy

The entropy of a variable is a measure of the uncertainty of the variable (Cover & Thomas, 1991). For any alignment, a probability distribution showing the chance of observing a specific amino acid

The entropy $H$ (also called the Shannon entropy) of an amino acid distribution $p$ is defined as

$$ H(p) = - \sum_{i=1}^{20} p_i \log_2(p_i) , \quad (5.1) $$

where $p_i$ is the probability of the $i$th amino acid at a particular position in an alignment. The above expression means $H(p) = -(p_A \log_2(p_A) + p_C \log_2(p_C) + p_D \log_2(p_D) + p_E \log_2(p_E) + ... + p_Y \log_2(p_Y))$, where the subscripts of $p$ correspond to the one letter amino acid codes. The entropy is $-20 \frac{1}{20} \log_2(1/20) \approx 4.3$ if all amino acids are equally probable, and is zero if only one amino acid is observed at a given position (assuming $0 \log(0) = 0$). For the data shown in Table 5.2 $H(2) \approx 1.36$ and $H(9) \approx 2.12$.

For DNA sequences composed of nucleotides, the entropy is calculated similarly:

$$ H(p) = - \sum_{i=1}^{4} p_i \log_2(p_i) , \quad (5.2) $$
where this time $p_i$ indicates the probability of the $i$th nucleotide at a particular position in an DNA alignment. Opening up the above expression, it becomes $H(p) = -(p_A \log_2(p_A) + p_C \log_2(p_C) + p_G \log_2(p_G) + p_T \log_2(p_T))$.

Relative Entropy

The relative entropy can be seen as a distance between two probability distributions, and is used to measure how an amino acid distribution $p$ differs from some background distribution $q$. The relative entropy is also called the Kullback-Leibler distance $D$ and is defined as

$$D(p|q) = \sum_a p_a \log_2 \left( \frac{p_a}{q_a} \right). \quad (5.3)$$

For the background distribution one often takes the distribution of amino acids in proteins in a large database of sequences. Alternatively, $q$ and $p$ can be the distributions of amino acids among sites that are known to have, or not have, some property, like glycosylation, phosphorylation, or MHC binding.

For nucleotides, the (relative) entropy is calculated very similarly: the only difference is that the sums run from one to four, instead of 20, as we have an alphabet of four letters in DNA/RNA sequences.
5.5 Logo Visualization of Entropy

One can visualize the characteristics of sequence patterns/motifs with the so-called sequence logo technique (Schneider & Stephens, 1990). The information content at each position in the sequence motif is indicated by the height of a column of letters, each representing amino acids or nucleotides. For proteins the information content is often defined as the relative entropy between the amino acid distribution in the motif, and a background distribution where all amino acids are equally probable. This gives the following relation for the information content:

\[
I = \sum_{i=1}^{20} p_i \log_2 \frac{p_i}{1/20} = \log_2(20) + \sum_{i=1}^{20} p_i \log_2 p_i .
\]  

(5.4)

This information content is a measure of the degree of conservation and has a value between zero (no conservation; all amino acids are equally probable) and \( \log_2(20) \approx 4.3 \) (full conservation; only a single amino acid is observed at that position). \( I \) is sometimes given in bits (as in Fig. 5.5).

In the logo plot, the height of each letter within a column is proportional to the frequency \( p_a \) of the corresponding amino acid \( a \) at that position. An example of a logo is shown in Fig. 5.5. When another background distribution is used, the logos are normally called Kullback-Leibler logos, and letters that are less frequent than the background are displayed upside down. An example of a tool to generate logo plots can be found on the web http://www.cbs.dtu.dk/~gorodkin/appl/plogo.html.

In logo plots, the amino acids are normally colored according to their properties:
- Acidic [DE]: red
- Basic [HKR]: blue
- Hydrophobic [ACFILMPV]: black
- Neutral [GNQSTY]: green

5.6 An application

In Chapter 2 we have discussed regulatory elements and used the alignment of transcription binding sites for a heat shock protein as an example (given also below). Fig. 5.6 gives a logo presentation of this alignment.

Haemophilus influenzae atctc AAAAAATGATCAAACACTCTTTTT ttatt
Pseudomonas aeruginosa ggttg AAAAAAGCGCCGATCGCCCTATAT cttccc
Shewanella putrefaciens ctgcc GAAAAATCGCAACCGAGCTTTTT gcgta
Salmonella typhi tg AAAAACTCAAATCTCCCCCATCT atact
Escherichia coli tg AAAAACTCAAATCCCCCATCT ataat
Vibrio cholerae cccca AAAGCTGTATGGATGACCCCTTT tttgt
Yersinia pestis t GAAAAATCTTTAATCTCCCCATT tataa
Figure 5.6: The sequence logo of Heat Schock Protein, hslV, transcription binding sites.

Obligatory reading

- Eddy (2004a): What is a hidden Markov model? (Link in the webpage of the course.)

Suggested reading

- Higgs & Attwood (2005) pg. 234-243
- Eddy (1998)

5.7 Exercises (pen and paper)

Question 5.1. Profile HMMs, I

Develop a profile HMM for the following alignment. Find the emission and transition probabilities. Note that there are multiple correct answers: you can model positions 4,5,6 and 9,10,11 either as insertions or as deletions.

ATC---TACGTG
ATCTTCTATCGTG
ATCAAATAC---
ATA---TATGTC
ATT---TATGTG

Question 5.2. Profile HMMs, II

a. Build a profile HMM for the below short alignment, using at least one insertion state.
b. Build a profile HMM for the below short alignment, using only deletion and main states (i.e., do not use any insertion states).

LAFFGH---K
LAFFGHAALK
LFFFGHAL--K
LAFFGKL--K
-AFFGKAALK
Question 5.3. Profile HMMs, III

We are making database searches with the HMM model depicted above. The hits we got are represented as sequence logos. For each logo indicate whether or not it can be a sequence motif that is found by the above HMM. Explain your choices with one sentence.

a.

b.

c.
Question 5.4. PSI-BLAST
We are doing a PSI-BLAST with the sequence ERCWWSSPPPGGGWWCCCPT. The output is in the following page. Explain why in round 2 hit P5 is more significant than hit P4, while in round 1 this was the other way around.

--- Results from round 1 ---
>P1 Score = 42.7 bits (99), Expect = 3e-10
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCWWSSPPPGGGWWCCCPT
Sbjct: 1 ERCWWSSPPPGGGWWCCCPT 20

>P2 Score = 40.4 bits (93), Expect = 1e-09
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCW WW PGGGWWCCCPT
Sbjct: 1 ERCWWKKPPPGGGWWCCCPT 20

>P3 Score = 37.0 bits (84), Expect = 1e-08
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCWWSSPPP WWCCCPT
Sbjct: 1 ERCWWSSPPPFFFWWCCCPT 20

>P4 Score = 36.6 bits (83), Expect = 2e-08
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCWWSS GGGWWCCCPT
Sbjct: 1 ERCWWSSYYYGGGWWCCCPT 20

>P5 Score = 35.0 bits (79), Expect = 6e-08
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCWW PPP WWCCCPT
Sbjct: 1 ERCWWKKPPPFFFWWCCCPT 20

--- Results from round 2 ---
>P1 Score = 34.9 bits (79), Expect = 6e-08
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCWWSSPPPGGGWWCCCPT
Sbjct: 1 ERCWWSSPPPGGGWWCCCPT 20

>P2 Score = 34.9 bits (79), Expect = 7e-08
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCWWW GGGWWCCCPT
Sbjct: 1 ERCWWSSYYYYGGGWWCCCPT 20

>P3 Score = 34.9 bits (79), Expect = 7e-08
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCWWW PPP GGGWWCCCPT
Sbjct: 1 ERCWWSSYYYYGGGWWCCCPT 20

>P5 Score = 34.5 bits (78), Expect = 8e-08
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCWWW PPP WWCCCPT
Sbjct: 1 ERCWWSSYYYYGGGWWCCCPT 20

>P4 Score = 32.6 bits (73), Expect = 3e-07
Query: 1 ERCWWSSPPPGGGWWCCCPT 20
   ERCWWW GGGWWCCCPT
Sbjct: 1 ERCWWSSYYYYGGGWWCCCPT 20
**Question 5.5. Transcription factor binding sites**

In bacteria protein quality control is carried out by a protein network, comprising of chaperones and proteases. Central to this network is a protein called ClpB.

The above alignment shows predicted transcription factor binding sites for ClpB in *Haemophilus influenzae* (HI), *Vibrio cholerae* (VC), *Salmonella typhi* (ST), *Thiobacillus ferrooxidans* (TF), *Shewanella putrefaciens* (SP), *E. coli* (EC) and *Yersinia pestis* (YP). The information content is often used to predict and identify transcription binding sites.

**a.** What is the most common nucleotides per position (consensus sequence) in the above alignment?

**b.** Calculate the information content at each position. ($\log_2 x = 3.32 \log_{10} x$)

---

**Question 5.6. Signal peptides**

A well-known N-terminal sequence motif directing proteins to the secretory pathway is called **signal peptide**. The logo plot below shows this region in eukaryotes, gram positive and gram negative bacteria. The mature protein starts from position 1. The signal peptide is cleaved while protein is leaving the endoplasmic reticulum.

**a.** What are the differences between gram positive and gram negative bacteria logos?

**b.** What differentiates eukaryotic signal peptides from bacterial ones?

**c.** Based on these logos, we want to develop a prediction method to detect signal peptides in newly sequenced proteins. Which one of the three groups of signal peptides would be most difficult to predict? Why?
Asparaginyl Endopeptidase (AEP) is an enzyme that cleaves proteins (cell’s own proteins, or any bacterial or viral) into smaller fragments. The fragments originating from pathogens are used by the adaptive immune system to detect an infection. The above logo gives the information content of the known cleavage sites by AEP. This logo is based on the experimental data available from AEP cleavages in 18 proteins. The cleavage occurs between P1 and P1’ positions. P1’ - P3’ are the flanking regions on the right (C-terminal) of the cleavage site, while P2-P4 are flanking regions on the left (N-terminal). The true value of N in the middle is 4.3 but here it is scaled to one.

**a.** Which amino acid is preferred by AEP at the cleavage site? Are there any exceptions to this?

**b.** We know that N, asparagine, occurs in proteins with 6 % frequency. However, an average protein of 200 amino acids would have one or two cleavage sites by AEP. How can this be?
Chapter 6

Analysis of gene expression data

Gene expression data is a high-throughput data type (like DNA and protein sequences) that requires bioinformatic pattern recognition. The complete set of mRNAs that are transcribed in a cell is often called its transcriptome. Transcriptomes are at the moment studied with DNA microarray technology. The study of DNA microarrays has been booming within bioinformatics to such a degree that many now take bioinformatics as a synonym for the analysis of this type of data. DNA microarray data are even more valuable when integrated with other types of data.

Gene expression profiling is often used to find genes that are differentially used by the cell under particular circumstances. Moreover, the expression profiles can be used to predict molecular or cellular function of genes: in a list of similarly expressed genes, those with an unknown function can be assigned a function based on their pattern of shared regulation with genes whose function is known. However, the application of gene expression profiles is not limited to this. Used properly, such profiles can be a powerful tool to generate and test biological hypothesis. An important application is definition of genetic pathways. Monitoring gene expression in time can be used to order the activation and repression of transcription, suggesting which genes regulate the expression of other genes. The transcriptional variance is an essential tool in evolutionary genetics, allowing to determine differences between individuals, populations, and species under a variety of different environmental conditions. On the clinical site, gene expression profiles are being developed with regard to prognosis of cancer progression, classification of infections or predicting the side effects of drugs.

6.1 DNA microarrays

DNA microarrays are used to measure the concentration of different mRNAs in a biological sample (see Fig. 6.1). This can be done by spotting oligonucleotide sequences, known as probes, on a slide (also called a chip), with different sequences spotted on different locations. The mRNA from the biological sample is normally converted to complementary DNA (cDNA), by reverse transcription, and finally labeled and put on the glass slide. If a cDNA sequence from the sample is complementary to the oligonucleotide sequence of one of the probes on the slide it will hybridize to it (i.e. bind to its complementary part). By labeling the cDNA
sequences in the sample with a fluorescent dye the concentration of cDNA in the sample can be quantified by a scanner.

Two types of chips are often used. The first type is the custom chip, where a robot is used to spot cDNA on a glass slide, and two different fluorescent labels are used to distinguish between sample and control. The second type of array is a prefabricated oligonucleotide chip where the oligonucleotide sequences are synthesized on the chip using photo-lithography. The sample and reference are hybridized on two different chips. The most common vendor is Affymetrix. The conventional chips of both types typically cover predefined genes from an entire genome. In the newest versions the entire set of exons from a complete organism, or even the complete genomic sequence, is covered. Other, more flexible technologies exist where researchers themselves can produce custom-made DNA chips. For example, NimbleGen makes DNA microarrays based on micromirror technology (used in data projectors), where the user can define the exact sequences of the probes.

The experimental procedure in all cases consists of six main steps:

a. Extracting mRNA from sample and control.
b. Converting the sample to cDNA or cRNA, amplification and labeling with fluorescent dye.
c. Hybridization of the sample to the probes on the chip.
d. Washing.
e. Scanning of the chip.
f. Image processing of scanned image on a computer.

After the image processing the results are often normalized by adjusting the expression levels relative to a gene or group of genes that are assumed to have a constant expression level between samples (Quackenbush, 2002). Normally, household genes, which are presumed to be equally expressed under all conditions, or the total amount of mRNA in the sample, are used
for this normalization. We will now explain how to normalize custom chips. The normalization of Affymetrix chips follows the same principle.

6.2 Normalization of microarray data

The raw data arising from a microarray analysis consists of red and green fluorescence intensities for every spot. Let $R_i$ and $G_i$ represent the red and green fluorescence in spot $i$ and let the red dye be used for the sample and the green dye for the control. These intensities can vary a lot from one spot to the other. Ideally, this variation only reflects the biological signal. However, the complex nature of sample and array preparation can introduce many biases and noise. These biases can cause significant variation among the spots that has nothing to do with the biology of the sample. Therefore the data analysis is often done in terms of ratios, or fold change, relative to a reference (control). For each gene $i$ the fold-change can be calculated as $M_i = R_i/G_i$. If the ratio $M_i$ is larger than one, the gene is upregulated with respect to the reference gene. Similarly if the ratio is less than one, it is considered to be down-regulated. A doubling of expression corresponds to $M_i = 2$ (and similarly halving corresponds to $M_i = 0.5$). In practice the ratios are converted to a logarithmic scale (often base 2) in order to prevent overflows during computation.

Looking at the raw data, one often sees that majority of spots in a microarray might have a $M_i$ less than 1, suggesting down-regulation. For an organism like yeast having more than 6000 genes, it seems very unlikely (although not impossible) that 5000 genes will be downregulated. Since this is a rather unlikely biological situation, we should look for a systematic bias in the experiment instead. One easy way of normalizing the microarray data is to use housekeeping genes which are assumed to be expressed at constant levels. So we need to shift the data so that $M_i = 1$ for house keeping genes. However, often it is difficult to decide which genes are keeping their expression levels constant.

A simple normalization method is to find mean (or median) $M$ values and use this mean value to directly normalize gene expression. In this case the normalized values would be $M_i^N = M_i/M$, where $M = \sum M_i / N$, $N$ being number of genes in the experiment. Note that in the ideal case where there is no dye bias, one should obtain $M = 1$. The mean (or median) of intensity ratios, $M$, are sometimes called scaling factors. Such a normalization is global because all the points are shifted by the same factor. In case of a dye bias, e.g., if the efficiency of labeling of the two DNA populations are different, such a global approach could be useful. However, if the bias depends on the strength of the signal, e.g., when the signal is high the bias is high, every intensity needs an individual or local normalization.

For mRNA derived from closely related samples, a significant fraction of the genes would be expected to be expressed at similar levels. In a scatter plot of intensities (or their logarithms), these genes would cluster along a straight line, the slope of which would be one if the experimental quality is the same for both samples. One can calculate the best-fit slope using regression techniques to normalize such data, e.g., by adjusting the intensities so that the calculated slope is one. In many experiments, the intensities are nonlinear, and local regression techniques are more suitable, such as LOWESS (LOcally WEighted Scatterplot Smoothing) regression. Today, LOWESS or related/similar methods are most commonly used for normalization of microarray data.
To establish the significance of the results, the experiment must be repeated and the statistical significance can then be established, for example, by using a t-test. The t-test assesses whether the means of two groups are statistically different from each other. This analysis is appropriate whenever you want to compare the means of two groups. t-test (and other statistical tests) produces a “p-value” that states the chance of finding the observed difference in the analysed data even if there was no difference between the two groups, i.e. if the means of two or more groups were indeed the same. Often a level of \( p = 0.05 \) is taken to define significant differences: if the observed difference between the means of two groups is likely to occur less than five times out of a hundred by “chance”, one can conclude that the two groups have different means. Since in a single microarray experiment many different mRNA levels are compared, it is important to correct significance values for multiple testing. A number of methods to do this have been developed. The simplest and the most stringent is the Bonferroni correction. All the \( p \)-values are corrected by multiplying them with the number of tests which are performed. For an experiment to be significant with \( p = 0.05 \) when 1000 different probes are compared, the uncorrected \( p \)-value must be smaller than \( 0.05/1000 = 0.00005 \) (this is called Bonferroni correction). In most microarray data, however, the Bonferroni correction is too strict.

### 6.3 Hierarchical Cluster Analysis

When DNA microarray data are available for several conditions, they can be used to define clusters of genes that behave similarly (i.e. are up- or downregulated) in different experiments. Such genes may be part of a common pathway or be co-regulated. If one of the genes in a group of co-regulated genes is known to be associated with a disease it may indicate that other genes in the group are also associated with that disease. Even if some related genes are missed (false negatives) and unrelated genes are picked up (false positives) the general concept of “guilt by association” seems to work quite well (Quackenbush, 2003; Stuart et al., 2003).

Data clustering algorithms can be hierarchical or partitional. Hierarchical algorithms find successive clusters using previously established clusters, whereas partitional algorithms determine all clusters at once. Hierarchical clustering is a general method, of which many applications exist in ecology. We will here focus on only hierarchical clustering algorithms and their application in microarray data analysis. A key step in a hierarchical clustering is to select a distance measure.

**Distance metrics**

Each gene can be represented by \( N \) expression levels, i.e. an \( N \)-dimensional vector of numbers, when \( N \) experiments are done. That is, we can represent a gene \( x \) with the following vector: \( \bar{x} = (x_1, x_2, ... x_N) \). The similarity between different genes can be calculated as a distance in this \( N \)-dimensional space. A very simple measure is the Euclidean distance. The Euclidian distance between two genes \( x \) and \( y \) is given as \( \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \).

In addition to the Euclidean distance, there are various distance measures that are used in the analysis of gene-expression data. The Manhattan distance (or city block distance) is an
example of a non-Euclidean metric distance measure. The name comes from the distance one would travel in crossing a large city, such as Manhattan, in which the streets are laid out in a regular, rectangular grid. In most cases, this distance measure yields results similar to the simple Euclidean distance. The Manhattan distance is calculated as the sum of the absolute distances between the components of each expression vector:

\[ d = \sum_{i=1}^{N} |x_i - y_i|. \]

The most commonly used distance measure in the analysis of gene expression data is the Pearson correlation coefficient (also known as the centered Pearson correlation coefficient), \( r \):

\[ r = \frac{\sum_{i=1}^{N} (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{N} (x_i - \mu_x)^2} \sqrt{\sum_{i=1}^{N} (y_i - \mu_y)^2}}, \]

where \( \mu_x \) and \( \mu_y \) are the mean values of expression values for genes \( x \) and \( y \).

The values of the Pearson correlation coefficient range between \(-1\) and \(+1\), with \( r = 1 \) when the two vectors are identical (perfect correlation), \( r = -1 \) when the two vectors are exact opposites (perfect anti-correlation), and \( r = 0 \) when the two vectors are completely independent (uncorrelated or orthogonal vectors). In contrast to Euclidean and Manhattan distances, the Pearson correlation coefficient (PCC) measures the extent to which two patterns are similar to each other. Therefore, the PCC is very useful if the “shape” of the expression vector is more important than its magnitude, as is often the case in microarray data. PCC is proven to be effective to detect similarities between gene expressions. However, PCC is not always robust to outliers and may generate false positives. If two gene profiles have a common peak or valley at a single experimental condition, the PCC will be dominated by this condition, although the profiles at remaining conditions may not be similar at all.

If the relative gene expression level is important, it is better to use the uncentered Pearson correlation coefficient, \( r_{un} \):

\[ r_{un} = \frac{\sum_{i=1}^{N} x_i y_i}{\sqrt{\sum_{i=1}^{N} (x_i - \mu_x)^2} \sqrt{\sum_{i=1}^{N} (y_i - \mu_y)^2}}, \]

Figure 6.2: The differences among clustering algorithms lie on the way they define the distance between two clusters: Single linkage (left), Complete linkage (middle) and UPGMA (right).
Clustering algorithms

Different algorithms have been used to define clusters. Some commonly used clustering algorithms are (see also Fig. 6.2):

- **Single linkage**: In this method the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters. This rule will string objects together to form clusters, and the resulting clusters tend to represent long “chains”.

- **Complete linkage**: Here the distances between clusters are determined by the largest distance between any two objects in the different clusters (i.e. by the “furthest neighbors”). This method performs quite well in cases when the objects actually form naturally distinct “clumps”. If the clusters tend to be elongated, or are of a “chain” type nature, this method is inappropriate.

- **UPGMA (unweighted pair group method with arithmetic mean)**: The distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters. This method is also very good when the objects form natural distinct “clumps”. However, it performs equally well with elongated, “chain” type clusters. An example on the differences of these clustering methods is given in Quackenbush (2001) Fig. 3.

### 6.4 Visualization of data

Gene expression data is often multi-dimensional, because the expression is, for example, measured at several time points, or under different experimental conditions. For example, consider the response of yeast to glucose starvation. To find out which genes are down- and up-regulated, we measure the gene expression every half hour during the first 6 hours. This makes every gene a data point in a 12-dimensional space. It is obviously not possible to visualize this data using all 12 dimensions. Data always contains redundant information. For example, during the starvation experiment the response of yeast might be delayed, e.g. maybe in the first hour nothing is changing. The gene expressions at $t = 0$ and $t = 30$ minutes then are more closely related than the others. In this case, we could ignore some of the redundant experiments, or use some average of the information, without a significant loss of information.

Often principal component analysis (PCA) is used to reduce the dimension of our data set down to two or three, so that graphical visualization is possible. PCA (also called singular value decomposition) is a mathematical technique that picks out patterns in the data while minimizing the loss of information. As a result the dimensions of gene-expression space are reduced. The mathematics behind PCA is rather complex, but the basic principles are as follows. Imagine taking a three-dimensional cloud of data points and rotating it so that you can view it from different perspectives, i.e. have different 2D views on your 3D cloud of data points. You might imagine that certain views would allow you to better separate the data into groups than other views. PCA finds those views that give you the best separation of the data. In Fig. 4 of Quackenbush (2001) we see how the genes clustered by hierarchical methods form still clusters in two or three dimensional space, i.e. by going from nine to two dimensions we do not lose essential information, and at the same time we can visualize every individual gene to see how related they are.
In most implementations of PCA, it is difficult to define accurately the precise boundaries of distinct clusters in the data, or to define genes (or experiments) belonging to each cluster. The well-defined clusters is a rare situation. However, PCA is a powerful technique for the analysis of gene-expression data when used with another classification technique, such as k-means clustering, that requires the user to specify the number of clusters.

6.5 An example

An interesting application of gene expression data is experimental evolution. Ferea et al. (1999) cultured yeast cells in glucose-limiting media and characterized changes in the pattern of gene expression in three evolved strains by comparing mRNA levels for virtually every yeast gene using DNA microarrays. To help identify similarities and differences in the expression profiles among the evolved and the parental strains, the gene expression patterns were subjected to cluster analyses. Fig. 6.3 shows two representations of the results for the 88 genes whose transcript levels differed from those in the parental strain by at least 2-fold (some down- and others up-regulated). In Fig. 6.3a, the genes are ordered by means of a hierarchical clustering algorithm, by which genes with the most similar expression patterns are arranged adjacent to each other. In Fig. 6.3b, genes with similar functional descriptors (as assigned by the Saccharomyces Genome Database) are grouped together.

Surprisingly, the number of mutations causing these changes in expression levels was extremely small: there had been only 6 selective sweeps during 250 generations. Thus, among this small set of mutations that accounted for the increased fitness of the evolved strains, some directly affected the regulation of genes that had altered expression, i.e. they might take place in a few key regulators of these genes.

The cells adapt to the glucose-limiting media by reducing their dependence on glucose fermentation and switching to oxidative phosphorylation as a more efficient means of generating energy. About the half of the genes shown in Fig. 6.3 were previously characterized, and most of these have roles in respiration, fermentation and metabolite transport. For the other half, i.e. the genes without a known function, it is now possible to assign a possible functional role. In short, this experiment allowed us:

- to identify which genes are taking place in response to glucose limitation in short time scale
- to predict a functional role for almost 50 genes
- to demonstrate how very few key mutations can change the expression of a large set of genes

Protocol: Using NCBI Expression databases

During aging process, the brain’s gene expression changes. Investigating these changes will help scientists to understand the effects of aging as well as related diseases. In this protocol, we will examine the differences in gene expression between young and old human subjects. By

---

1Adapted from NCBI tutorial for GEO database.
Figure 6.3: This figure is adapted from Ferea et al. (1999). (a) Hierarchical clustering of gene expression. Relationships in expression patterns among the genes are represented by the tree at the left of the display, with branch lengths indicative of the magnitude of the differences between gene expression patterns. (b) Expression patterns organized according to functionally defined categories.
performing microarray analyses on samples of human brain from both young and old subjects, we will learn how to access microarray data in general and how we can identify genes that have specific patterns of gene expression.

Figure 6.4: DataSet Record in GEO database.

1. Find the article by Lu et al. (2004) in PubMed using the keywords “Gene regulation and DNA damage in the ageing human brain”. The micro array data analyzed in this article can be accessed through GEO, the Gene Expression Omnibus (www.ncbi.nlm.nih.gov/GDS/). When you find the article, go to ‘Related information’ menu on the right hand side and then select ‘GEO DataSets’.

2. There are two datasets associated with this publication. We will focus on the first one. To access the data, click on the Dataset link (GDS707).

3. The new page (see Fig. 6.4) displays the DataSet Record, a detailed description of the experiment. On the bottom of the page you will see the section where you can compare two sets of samples. A query function is available for identifying genes that exhibit specific patterns of expression among subsets of interest. In Step 1, choose “Value means difference \( A \text{vs} B \) to 6 fold higher or more. In step 2, assign group A to subjects aged 20-39 (both male and female), and assign group B to subjects ranging in age from 80-99 (both male and female). Click OK. We can then Query to find genes for which the expression level in young people (group A) are 6-fold higher than in older group (group B) (see Fig. 6.5).

4. If you click on Query button, you get a list of genes whose expression is 6-fold higher in young people than in older people. To examine the expression pattern detected for MAP1B gene, for example, click on the Profile Chart.

5. In the profile chart (Fig. 6.6) each cluster of bars indicates the expression level of MAP1B for a given sample, with Samples arranged by gender and age. The red bars represent normalized data Values for expression, and the blue bars reflect the relative abundance of this gene when compared with all of the other genes detected in the sample. Click on sort by age to see that the level of expression is much higher in young people than in older people, with variable levels of expression of intermediate ages.
Analysis of gene expression data

Figure 6.5: Generating two groups with different age categories.

6. To find other genes with an expression pattern similar to that of MAP1B, select ‘Profile Neighbors’ in the list of genes (in the list of genes as in step 4), next to MAP1B item.

Figure 6.6: Gene expression profile for MAP1B gene. To sort it with respect to age, click on age on the left of the screen.

Obligatory Reading

- D’haeseleer (2005): How does gene expression clustering work?

Optional Reading

- Quackenbush (2002)
Data for the above 10 genes are obtained from a microarray (custom chip) where each number represents the fluorescence intensity for the red (R) or green (G) dye. Red dye is used for the sample and green for the control (reference).

**a.** Find $R/G$ ratio for each gene.

**b.** Calculate average $R/G$ ratio. Is this average lower or higher than 1? What does this mean? Which dye is more intense in general?

**c.** Which genes in the sample are expressed twice as much as the control? Which genes are down-regulated to half?

**d.** This data is normalized as explained in the beginning of this chapter (by dividing the individual $R/G$ ratios by the average ratio) and we found that for gene E, the normalized $R/G$ ratio is 2.041. That is gene E is expressed twice as much in the sample relative to the control. Does this result fit to your answer above, and if not how can you explain it?

**Question 6.2. Cancer**

The following table shows the gene expression values for eight genes for three types of cancer (C1-C3).

<table>
<thead>
<tr>
<th>Gene ID</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>D</td>
<td>8</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>E</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>G</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
a. Use Manhattan distance and single linkage to cluster these genes.
b. Which genes can be used to diagnose (discriminate) the three cancer types?
c. What information do you need to decide whether or not these genes are all cancer related?

Question 6.3. Aging
In a microarray experiment analysing the effect of aging in yeast cells, the gene YGL163C is differentially expressed. The annotation of this gene in GenBank is: "DNA-dependent ATPase, stimulates strand exchange by modifying the topology of double-stranded DNA; involved in the recombinational repair of double-strand breaks in DNA during vegetative growth and meiosis". This gene is upregulated in aged yeast cells.

a. How can you find the human genes that are similar to YGL163C?
b. How would you study if similar regulation occurs in aging human cells? (Assume that you have microarray already experiments available from aged and young human cells)
Question 6.4. Amino acid starvation

In a microarray experiment we want to see the effect of amino acid starvation on the gene expression of yeast cells. Below you see clustering of a selected set of genes (all gene names start with "Y") with UPGMA and with the single linkage method. Red color is used for up-regulated genes during amino acid starvation and green for down-regulated ones. The relative expression values are found with respect to t=0 gene expression.

a. Which clustering is the result of UPGMA and which one of single linkage clustering? Why?

b. Name three genes that are down-regulated very fast during amino acid starvation? Which clustering method identifies these genes most easily?

c. How can you use this yeast data set if you want to identify human genes that would alter their expression during amino acid starvation?
Practice exam questions

Question 7.1.

Werner syndrome (WS) is a premature aging disease that begins in adolescence or early adulthood and results in the appearance of old age by 30-40 years of age. This disorder is inherited. Cells from WS patients have a shorter lifespan in culture than normal cells do. It is known that the Werner syndrome is caused by defects in the WRN gene. We search for other gene(s) that might be responsible for WS using gene expression data. We have cells available from four healthy controls and three patients with WS and we measure the gene expression in G1 phase of the cell cycle using micro arrays. Above figure shows a clustering of absolute gene expression from 28 genes from healthy individuals and WS patients (H1-H4: healthy individuals, P1-P3: patients). Red colors indicate high gene expression, black close-to-average gene expression and green low gene expression.

**a.** Assume that all patients (P1-P3) have the same defect in the promoter of WRN gene. What is the effect of this defect in the WRN expression?

**b.** According to this clustering of gene expression, which gene(s) other than WRN can be responsible for Werner syndrome? Explain in one sentence.

**c.** We design a knock-out experiment in yeast to test whether the genes you identified in **b)** can be related to Werner syndrome. In a knock-out experiment an organism is engineered to lack (e.g. by deletion) the expression and activity of one or more genes. Which yeast genes would you delete in these knock-outs?
Question 7.2.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C</th>
<th>T</th>
<th>C</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Based on the completed dynamic programming matrix above, answer the following:

a. Is this a Needleman-Wunsch (global) or Smith-Waterman (local) alignment?

b. What is the score of the best alignment?

c. What was the gap penalty used?

d. What was the nucleotide match score? Assume that any match (e.g., A-A, C-C, G-G, or T-T) receives the same score.

e. What was the nucleotide mismatch score? Assume that any mismatch receives the same score.

f. What is the alignment suggested by this matrix?

Question 7.3.

We have a short protein fragment of which we would like to know the protein it belongs to. First we do a BLAST search using the BLOSUM62 matrix, then we make 3 iterations in PSI-BLAST. The alignment of one of the hits is shown below:

BLAST OUTPUT:

>gi|75856671|ref|ZP_00764295.1 Tryptophan synthase beta chain [Vibrio sp.]
Length=407

Score = 74, E-value = 3.1
Identities = 14/27 (51\%), Positives = 18/27 (66\%), Gaps = 0/27 (0\%)

Query 2 QNITAGTTNTLYLKREDLLHGGAHTN 28
   +N++   +YLKREDL H CAH N
Sbjct 70 ENLSKKYADIYLKREDLNHTGAHKIN 96

PSI-BLAST OUTPUT:

>gi|75856671|ref|ZP_00764295.1 Tryptophan synthase beta chain [Vibrio sp.]
Length=407

Score = 120, E-value = 2e-05
Identities = 14/27 (51\%), Positives = 18/27 (66\%), Gaps = 0/27 (0\%)

Query 2 QNITAGNNTTLYLRDELHGGAHKTN 28
+N++ +YLKREDL H GAHK N
Sbjct 70 ENLSKGYGADIYLRDELHGTKH 96

Explain briefly why the score of the alignment increased and the E-value has improved.

**Question 7.4.**

We have a short protein fragment from *E. coli* (bacteria) of which we would like to know its function. First we do a BLAST search using the BLOSUM62 matrix and get the following four hits as the best ones (Salmonella is also a bacteria):

<table>
<thead>
<tr>
<th>SeqA_Ecoli</th>
<th>replication initiation regulator SeqA [E. coli]</th>
<th>36.2</th>
<th>0.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>HypoProt_Salmonel</td>
<td>hypothetical protein CKO_02472 [Salmonella]</td>
<td>35.0</td>
<td>0.23</td>
</tr>
<tr>
<td>ATPBind_Human</td>
<td>ATP-binding prot. involved in DNA repair [Human]</td>
<td>33.9</td>
<td>0.62</td>
</tr>
<tr>
<td>MGC78798_Frog</td>
<td>MGC78798 protein [Frog]</td>
<td>25.0</td>
<td>1.7</td>
</tr>
</tbody>
</table>

The alignment of the best hit is:

Score = 36.2, Expect = 0.10
Identities = 29/40 (72\%), Positives = 30/40 (75\%), Gaps = 2/40 (5\%)

Query 8 SQPAAPVTKEVRVASPAIVEHIHKTI--YVRDREQSIS 45
SQPAAPVTKEVRVASPAIVE VKTI VR RE +S
SeqA_Ecoli 84 SQPAAPVTKEVRVASPAIVEAKPVKTIKDKVRAMRELLLS 123

**a.** Can you predict the function of this protein based on the results of BLAST search? Explain with max 3 sentences.

When we make 3 iterations in PSI-BLAST, the alignment of the hit with SeqA_Ecoli is shown below:

Score = 39.3
Identities = 29/40 (72\%), Positives = 30/40 (75\%), Gaps = 2/40 (5\%)

Query 8 SQPAAPVTKEVRVASPAIVEHIHKTI--YVRDREQSIS 45
SQPAAPVTKEVRVASPAIVE VKTI VR RE +S
SeqA_Ecoli 84 SQPAAPVTKEVRVASPAIVEAKPVKTIKDKVRAMRELLLS 123

**b.** Will the E-value of this alignment be higher or lower than 0.10?
We get a different set of best hits with the PSI-BLAST. (Yersinia pestis is a bacterium)

<table>
<thead>
<tr>
<th></th>
<th>Similar to DNA repair protein</th>
<th>[Salmonella]</th>
<th>Score</th>
<th>E-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepA_Salmonella</td>
<td>(Salmonella)</td>
<td>58.1</td>
<td>1e-11</td>
<td></td>
</tr>
<tr>
<td>MaxD_Salmonella</td>
<td>DNA repair protein, MaxD</td>
<td>57.4</td>
<td>1e-10</td>
<td></td>
</tr>
<tr>
<td>MRP_Yersinia</td>
<td>MRP-like prot. (DNA synthesis) [Yersinia Pestis]</td>
<td>53.2</td>
<td>5e-8</td>
<td></td>
</tr>
<tr>
<td>ATPBind_Human</td>
<td>ATP-binding prot. involved in DNA repair[Human]</td>
<td>43.9</td>
<td>6e-6</td>
<td></td>
</tr>
</tbody>
</table>

c. Do these extra hits change your prediction for the function of this protein? Explain with max 3 sentences.
d. Explain in 1 sentence why we get a significant hit in humans.

Question 7.5.

a. Calculate an UPGMA tree for the following distance matrix, showing the relationship between genes X, Y, Z, W, Q.

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>W</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>-</td>
<td>0.16</td>
<td>0.6</td>
<td>0.7</td>
<td>0.65</td>
</tr>
<tr>
<td>Y</td>
<td>0.16</td>
<td>-</td>
<td>0.8</td>
<td>0.75</td>
<td>0.7</td>
</tr>
<tr>
<td>Z</td>
<td>0.6</td>
<td>0.8</td>
<td>-</td>
<td>0.17</td>
<td>0.19</td>
</tr>
<tr>
<td>W</td>
<td>0.7</td>
<td>0.75</td>
<td>0.17</td>
<td>-</td>
<td>0.08</td>
</tr>
<tr>
<td>Q</td>
<td>0.65</td>
<td>0.7</td>
<td>0.19</td>
<td>0.08</td>
<td>-</td>
</tr>
</tbody>
</table>

b. The tree below shows the evolutionary relationship of Kinesin-7 (left) and Kinesin-1 (right) molecules in Drosophila (sequences starting with Dm), human (sequences starting with Hs), C. elegans (sequences starting with Ce) and arabidophsis(sequences starting with At). Kinesis molecules are involved in chromosome movement during mitosis. Mark possible nodes where gene duplication or gene loss events could have occured.
c. It is still an ongoing debate whether Drosophila is closer to mammals than C. elegans is. What does the Kinesin-13 tree below suggest? Explain. (Human sequences start with Hs, Drosophila with Dm, C. elegans with Ce, Arabidopsis with At and Leishmania with LM)

A rooted phylogenetic tree of the MAP gene was constructed by the neighbor-joining method and is given below (Fugu is a fish, Drosophila is the fruit fly, and Anopheles is a mosquito, numbers after species name indicate sequence identifiers).

a. Indicate the places in the tree where gene duplication and gene deletion might have occurred.

b. Use this tree to predict the number of MAP genes in the root of the tree (the last common ancestor). Explain how you make this prediction in maximally 3 sentences.

c. Looking at the bootstrap values, how many times at most (out of 100 bootstrap experiments) can the Drosophila MAP gene have a fish gene as its closest relative?

Answers to practice exam questions
Question 7.1.
a. WRN expression is decreased in all three patients compared to the healthy individuals.
b. ARF, RecQ and ABHD. These are the only three genes that consistently behave differently in patients compared to healthy individuals.
c. Using BLAST we should find yeast homolog of WRN (to verify that yeast is a suitable model organism) and the homologs of three genes identified in b. These homologs can be deleted in the knock-outs.

Question 7.2.
a. This is a Smith-Waterman alignment; a Needleman-Wunsch alignment would also include negative values (for example on the top row).
b. 14.
c. The gap penalty is 2: scores decrease by 2 when moving vertically or horizontally through the matrix.
d. The match score was 4: when moving diagonally to a correctly aligned residue, the score is increased by 4.
e. The mismatch score was -1: when moving diagonally to an incorrectly aligned residue, the score is decreased by 1.
f. The alignment suggested by the matrix:

```
TCTCA
TC-CA
```

Question 7.3.
PSI-BLAST calculates scores based on a scoring matrix determined by the alignments generated in earlier iterations. Because of this, highly conserved residues can get higher scores than they would get from the BLOSUM62 matrix, which in turn leads to a higher score (and thus a more significant, i.e. lower, E-value) of the PSI-BLAST hit.

Question 7.4.
a. We cannot predict the function. The E-values are all quite high (i.e. not very significant), and the hits all have similar scores but do not have the same function.
b. The E-value will be lower: the Score is higher, making the hit more significant.
c. Yes: the hits are more significant and consistently point to a function in DNA repair.
d. DNA repair is important in both bacteria and eukaryotes, and apparently the sequence of these proteins is highly conserved. Here horizontal gene transfer (HGT) is less likely, because we need to then assume that there had been at least two HGT, as the homologs are found in diverse bacteria (Salmonella and Yersinia). Note that HGT could not happen between human cells and the ancestor of Yersinia and Salmonella due to the fact that Yersinia and Salmonella diverged from each other much earlier than speciation of humans.

Question 7.5.
a. The shortest distance is Q - W. Grouping these two and recreating the matrix gives us:

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>WQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td>0.16</td>
<td>0.6</td>
<td>0.675</td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
<td>0.8</td>
<td>0.725</td>
</tr>
<tr>
<td>Z</td>
<td></td>
<td></td>
<td></td>
<td>0.18</td>
</tr>
<tr>
<td>WQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The shortest distance is now X - Y. The new matrix:

<table>
<thead>
<tr>
<th></th>
<th>XY</th>
<th>Z</th>
<th>WQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>XY</td>
<td></td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>Z</td>
<td></td>
<td></td>
<td>0.18</td>
</tr>
<tr>
<td>WQ</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The shortest distance is now Z - WQ. The new matrix:

<table>
<thead>
<tr>
<th></th>
<th>XY</th>
<th>ZWQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>XY</td>
<td></td>
<td>0.7</td>
</tr>
<tr>
<td>ZWQ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The tree:
b. The squares depicted on the tree below are duplication events. There could be a gene loss in Arabidopsis (At) and in C. elegans (Ce). It is also likely that there had not been any gene losses in At, but that this gene could have been an invention after diversification of plants.

c. The Kinesin-13 tree suggests that drosophila is closer (assuming the tree is rooted to the left): when we head from the root of the tree towards human, C. elegans diverges before drosophila.

Question 7.6.

a. The squares indicate duplications. There had been possibly a gene loss even in Zebra fish and in rat.

b. The ancestor had 1 MAP gene: fly and mosquito both have a single gene, and there are no duplication or loss events between these two species and the root of the tree.

c. 0 times: the bootstrap value joining mosquito and fly is 100.
Answers to the exercises

Chapter 1: Self-test

Question 1.1. Evolution
a. 
\[ \text{rate} = 10^{-9}/[\text{b.p.}]\text{[year]} \]
fraction = [rate] \times [time] = 10^{-9} \times 10^7 = 10^{-2}=1\%.

But, because we are analyzing the difference between two sequences, we can consider the time of divergence to be the “double” (both sequences are evolving), so we multiply the last answer by two, and thus, expect a difference of 2%.

b. Decrease.

c. For each loci, there is a chance of mutation of $10^{-9}$ happening in one year, so the probability that one site will not change is $(1 - 10^{-9})$. Because the sequence is 1000bp long, we multiply these probabilities together 1000 times:

\[ (1 - 10^{-9})^{1000} = 0.999999 \]

Which means that in one year, there is a 99.9999% chance that the sequence does not change.

d. We have to multiply the above answer by itself the number of years which pass:

\[ (\text{chance of staying the same in one year})^{\text{number of years}} \]
\[ (0.999999)^{10^7} = 4.539 \times 10^{-5} \]

e. The probability of BOTH sequences to be the same after $10^7$ years:

\[ 4.53 \times 10^{-5} \times 4.53 \times 10^{-5} = 2.06 \times 10^{-9} \]

which is even much more improbable than finding just one of the sequences unchanged!
Question 1.2. DNA sequences

a. Each nucleotide is an equal and independent event, so the chance of anyone to appear at a point in the sequence is of \( \frac{1}{4} \). The sequence is 10 nucleotides long, meaning that the probability of it occurring is:

\[
\left(\frac{1}{4}\right)^{10}.
\]

b. To calculate the average chance, we multiply the probability of finding a specific 20 nucleotide sequence with the size of the whole genome. Thus

\[
\left(\frac{1}{4}\right)^{20} \times (4 \times 10^9) = 0.0036
\]

Question 2.1. PAM and BLOSUM

Comparing PAM250 and BLOSUM62

a. most conserved amino acid: W, and second most conserved C for both matrices. Third most conserved is Y for PAM250 and H for BLOSUM62.

b. least conserved amino acids:

A, I, L, S, V (BLOSUM62);

A,N,S (PAM250)

c. Often non-diagonal values agree (at least in sign). Thus, the two matrices agree for the majority of substitutions.

Question 2.2. Similarity between sequences

Note the number of mismatches between sequence 1 and 2 is the same as between sequence 1 and 3.

a. Using identity, sequence 1 is as similar to sequence 2 and sequence 3, because both sequence 2 and 3 have 2 mismatches with respect to sequence 1. This is true independent of which identity matrix you use, e.g. if you assume match=2, mismatch=-2, then still sequence 1 is as similar to sequence 2 and 3.

b. With BLOSUM62:

Score of similarity between 1 and 2 = 4-3+5+6+6-3+6+6+6 = 33
Score of similarity between 1 and 3 = 4+11+2+6+6+11+6+2+6 = 54
Score of similarity between 2 and 3 = 4-3+2+6+6-3+6+2+6 = 26
Sequence 1 is more similar to sequence 3 than 2.

Question 2.3. Gap penalty

The score for alignment in (i) = 4 + 3g and in (ii) = g − 1. For (i) to be a better alignment: 4 + 3g > g − 1 or g > −2.5. So, g = −2 or g = −1 are the values of g that makes the alignment
of (i) better than (ii).

**Question 2.4. Needleman-Wunsch**

<table>
<thead>
<tr>
<th></th>
<th>j=0</th>
<th>j=1</th>
<th>j=2</th>
<th>j=3</th>
<th>j=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>i=0</td>
<td>0</td>
<td>←</td>
<td>←</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td>i=1</td>
<td>D</td>
<td>↑</td>
<td>←</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
</tr>
<tr>
<td>i=2</td>
<td>E</td>
<td>↑</td>
<td>←</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-4</td>
<td>-1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>i=3</td>
<td>E</td>
<td>↑</td>
<td>←</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-6</td>
<td>-3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>i=4</td>
<td>E</td>
<td>↑</td>
<td>←</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-8</td>
<td>-5</td>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>i=5</td>
<td>V</td>
<td>↑</td>
<td>←</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-10</td>
<td>-7</td>
<td>-4</td>
<td>-1</td>
</tr>
<tr>
<td>i=6</td>
<td>W</td>
<td>↑</td>
<td>←</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-12</td>
<td>-9</td>
<td>-6</td>
<td>-3</td>
</tr>
</tbody>
</table>

This means 3 different alignments are possible.

1

```
D E - - V D
D E E E V W
```

2

```
D - E - V D
D E E E V W
```

3

```
D - E V D
D E E E V W
```

**Question 2.5. Multiple alignment**

a. 2,11,16,17,19,24,26,27,28,29,30,31,32,33,35,44,50.

b. No, they are grouped. This implies that certain parts (i.e. the well-conserved parts) of the sequence are functionally more important than others.

c. Not always. It seems that what is more important is where an amino acid is. If an amino acid is within a conserved region, it is more likely that it will be conserved independent of which amino acid it is.

d. second (chimp) and fourth (mouse)

e. We use match=1; mismatch=0 (alternative is match=1; mismatch=-1): Score(shark, human) = 25; Score(human, chicken) = 23. So, shark sequence is closer to human than chicken is to human if we use identity.
Question 2.6. DNA alignments

a. 1, 2, 7, 8, 10, 11.

b. 3rd base pair in a codon is often redundant, i.e., different nucleotides in the 3rd base pair would still code for the same amino acid. Therefore the conservation is low in the third position. Variants of a codon on the first and second position almost always (except Serine) translate for a different amino acid, and thus these positions are highly conserved in an alignment of related sequences.

Question 3.1. E-values

a. See Chapter 2 for the BLOSUM62 matrix

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>F</th>
<th>S</th>
<th>S</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>S</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

score: 26

b. The formula for the expectation value is

\[ E = K \times m \times n \times e^{-\lambda S} \]

(see also chapter 3). The parameters are \( K = 0.04 \) and \( \lambda = 0.27 \). \( n \) and \( m \) are the lengths of the query sequence and the total database length (40 and 6,548,585 respectively). For \( S \) we use the score from question a. So we get

\[ E = 0.04 \times 40 \times 6,548,585 \times e^{(-0.27 \times 26)} = 9365.3 \]

c. This hit is not significant as its E-value is much higher than 0.05. Because the HSP is so short it is possible to encounter this sequence by chance in other sequences in this relatively large database.

Question 3.2. Smith-Waterman

Again see Chapter 2 for the BLOSUM62 matrix. The gap penalty is 12.

Now we are going to do it a little bit different than in the Needleman Wunsch algorithm. Instead of starting at the bottom right corner, we look for the highest score in the matrix (i=5, j=4) We follow the arrows until the score becomes 0, then we stop. Doing this we get:

Y I
Y I
Remember that this method does not give you a full alignment. It is not necessary to fill in the arrows if the score is zero because you will not take them into account. Note also that it is very well possible that the highest score could show up in the middle of the matrix.

**Question 3.3. BLAST, I**

**a.** If you have 5500 proteins that each have significant (low E-values) hits with single proteins from other fungi it is very likely that these proteins are orthologues (i.e. these genes were present in the common ancestor of fungi). These proteins might perform important functions like housekeeping, DNA duplication, etc.

**b.** If the E-values of found protein matches are larger than 10 (a very bad match) it is most likely that the protein from your predicted gene is specific to your fungus. This set of genes are called “lineage specific expansions”.

**c.** You can BLAST against other genomes. Other possibilities: Because the gene prediction method you have used will not always be correct, you could use the DNA sequence in your BLAST search to circumvent the problem of the different reading frames. You could also try other gene prediction methods to test if your predicted gene set is robust. Since we do not have any significant hits, we can not try PSI-BLAST here, because to build an PSSM, we need to have few reasonable hits from BLAST search.

**Question 3.4. BLAST, II**

**a.** We have to look at E-values. Since the match from site 2 has a lower E-value (5 × 10\(^{-11}\) versus \(10^{-4}\)), the second match (with AF9_HUMAN protein) is more significant.

**b.** In the first search the low complexity filter is on, and in the second it is off. The main match in the second search is at polyS (SSSSSSSSSSSSSS) region. Since repeats can occur often in genomes, this finding does not say much about whether or not the protein found (AF9_HUMAN) can be a homolog to our query sequence. However, in the first search the match is outside the repeat region, and implies “real” sequence similarity. The corresponding E-value is low enough, thus this match with chicken ubiquitin protein is more likely. Notice that homologous proteins can easily come from different organisms, i.e., the organism itself is not always a good criteria to evaluate the quality of the database search.

**Question 3.5. BLAST, III**

**a.** We can not predict the function of this protein from this BLAST search, because i) hits seem equally likely (i.e., they have similar E-values) and ii) each hit suggests a different function (like enolase, nuclease, splicing coactivator subunit, etc) or no defined function (hypothetical protein).

**b.** We can set the low complexity filter on if it is not yet set, because there is a repeat in this sequence. Also we can try PSI-BLAST, which would use a position-specific substitution matrix, and could identify distant homologs.

**Question 3.6. BLAST, IV**
a. We can use BLASTX: translated nucleotide query against protein database. The program will translate our nucleotide sequence in six different reading frames and search against the protein sequence databases.

b. No, because in order to use PSI-BLAST, we need to build a PSSM. PSSMs are build from initial BLAST hits. If we do not have any BLAST hits, then we do not have enough information to build a position specific substitution matrix either.

Question 3.7. BLAST, V

a. We should use BLASTN which searches for shorter and less perfect hits rather than MEGABLAST (because the word size and match/mismatch scores are different for these two programs, check NCBI BLAST web page for exact value of these parameters).

b. First we can try tblastn, *i.e.* search with amino acid sequence of *E. coli* enzyme in *B. subtilis* genome sequence. Next, we can try to identify the homologs of this enzyme in other bacteria, build a PSSM with those hits, and use PSI-BLAST to identify *B. subtilis* homolog.

c. Histidine kinase (possibly important for the sensory system).

d. BLAST YP_00168034 against zebrafish genome. If AP_012432 is the best hit, then these two proteins are bidirectional hits and are therefore likely orthologs.

Chapter 4: Self-test


Question 4.1. Jukes-Cantor

Jukes-Cantor model,

\[ d = -\frac{3}{4} \ln(1 - \frac{4}{3} D) \]  (D.1)

looks like this: [Graph of the Jukes-Cantor model equation]
We do not have to plot this to see the behavior though, for we could simply draw the main features of such a function. First we observe that

\[ D \to \frac{3}{4}, \quad d \to \infty. \] This singularity can be seen in the graph at the value \( D = 0.75 \).

When the fraction of sites that differ between the sequences is zero (\( D = 0 \)), we notice that \( d = 0 \) as well. Moreover, when difference between the sequences are small, the distance “\( d \)” is increasing linearly with \( D \), because at these initial times there is a low probability of several mutations per site. We can see this by the slope of the curve at small values of \( D \) (slope 1).

If \( D = 0.20 \), then \( d = 0.23 \).

**Question 4.2. UPGMA**

\[
\begin{array}{c|cccc}
 & A & B & C & D \\
\hline
A & - & 0.2 & 0.3 & 0.4 \\
B & 0.2 & - & 0.4 & 0.4 \\
C & 0.3 & 0.4 & - & 0.5 \\
D & 0.4 & 0.4 & 0.5 & - \\
\end{array}
\]

\[
\begin{array}{c|cc}
\{A,B\} & C & D \\
\hline
\{A,B\} & - & 0.35 & 0.4 \\
C & 0.35 & - & 0.5 \\
D & 0.4 & 0.5 & - \\
\end{array}
\]

The last matrix is not necessary to calculate in order to form the clades, but only to calculate the distances. But, remember that it is necessary to use the values of the original matrix, namely \((A,D)+(B,D)+(C,D)/3\).

**Question 4.3. A simple tree**

- First the dissimilarity matrix is calculated:
The smallest distance (=1) cluster is formed \{seq1,seq2\}.

A new dissimilarity matrix is calculated, using the mean distance to seq1 and seq2:

<table>
<thead>
<tr>
<th></th>
<th>seq1</th>
<th>seq2</th>
<th>seq3</th>
<th>seq4</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>seq2</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>seq3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>seq4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

The next cluster is formed corresponding to the smallest distance (=2): \{seq3,seq4\}.

Repeat last dissimilarity matrix calculation (not needed, but interesting if you want to know the values of these distances, which are drawn into the tree below).

<table>
<thead>
<tr>
<th></th>
<th>{seq1,seq2}</th>
<th>seq3</th>
<th>seq4</th>
</tr>
</thead>
<tbody>
<tr>
<td>{seq1,seq2}</td>
<td>0</td>
<td>2.5</td>
<td>3.5</td>
</tr>
<tr>
<td>seq3</td>
<td>2.5</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>seq4</td>
<td>3.5</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Question 4.4. Tree reading

**rooted:** iii and iv are equal

**Unrooted:** all are equal. This can be seen if you draw all the trees as unrooted, i.e. in star shape.

Question 4.5. Parsimony

**a.** As a total, we count 13 substitutions. From these, we notice that most (9) substitutions occur at the third position of the codons. See your reader for definition of synonymous mutations. A mutation at the third position usually does not lead to a change of the amino acid, so the tendency seems to be that there exists more synonymous substitutions. However, if we carefully analyse the substitutions, we arrive at the following list of mutations:
• (1x) AAT ↔ AAC, corresponds to: N ↔ N
• (3x) GAG ↔ GAA, corresponds to: E ↔ E
• (1x) GTG ↔ GTT, corresponds to: V ↔ V
• (1x) TTT ↔ CTT, corresponds to: F ↔ L
• (2x) TTT ↔ TTA, corresponds to: F ↔ L
• (1x) TCC ↔ TTC, corresponds to: S ↔ F
• (2x) ATG ↔ ATA, corresponds to: M ↔ I
• (2x) ATG ↔ CTG, corresponds to: M ↔ L

(observe that the last four mutation occur in the last codon, and that we have considered here mutations on the consensus sequence ATG.)

This means that only 5 mutations are synonymous.

b. From the 13 substitutions, 8 are transitions and 5 are transversions. (see your reader for definition of transitions/transversions).
c. The most informative sites are those in which at least two substitutions to the same nucleotide occur; sites 15, 27, 28, 30. (see your reader).
d.

Question 4.6. Parsimony, II

Here the informative sites are 4, 6, 7, 9 and 12.
Question 4.7. IL-11

Because almost all fish species have IL11a and an IL11b, their common ancestor of fish probably had a gene duplication. Trout, carp (both missing IL-11b) and halibut (missing IL-11a) probably lost a gene, because they are the only ones of the fish that only have one of the IL-11’s. The mammalians have only one IL-11 as well, they probably did not experience the gene duplication. In the figure below gene duplications are indicated with squares and deletions with circles.

![Gene duplication diagram]

Question 4.8. Cyclin-dependent kinase

a. The common ancestor of eukaryotes had probably 1 CDK gene, because yeast, pombe and arabidopsis have still one CDK gene.

b.

c. Never, because bootstrap value of Xenopus CDKH1 and CDKH2 clade is 100.

Question 4.9. Bootstrapping, I

In all trees, D and E are found to form a clade. Also, in all cases, A, B, and C come together in clade (though in different ways). From the three possibilities of clustering A, B and C, we choose to draw the most probable one, as indicated below in the figure:
Note, that by drawing this tree with the probabilities observed, we loose some information, namely, that 19 times B was left alone while A and C formed a clade, and that 9 times A was on its own, while B and C formed a clade. The only information we can see now, is that in 72 cases A and B remained together.

Question 4.10. Bootstrapping, II

a. 14. We know that 86 times W and X DID form a clade, so 14 times this did not occur.
b. No. We know from the tree that 94 times W,X,Y and Z form a branch, meaning that there is a maximal number of 6 times that W could form a clade with V.
c. YES. Because the only information that is guaranteed is that W and X come together 86 times (and that all 4 - W,X,Y,Z - form a clade in 94 cases). So, it is possible that in the other 14 cases, W, Y and Z form a clade.

Question 4.11. COX genes

a. See tree above for the gene duplication. Deletions are Sheep COX2, Rat COX1, and
Chicken COX1.

**c.** See tree above.

**d.** This finding suggests that gene duplication occurred earlier. Most likely the bootstrap value would not change. The most parsimonious tree is:

![Tree Diagram]

The deletions do not change, either.

**Question 5.1. Profile HMMs, I**

This is a relatively simple HMM to construct. One of the possibilities is to model conserved positions as main states, and in between use transitions to deletion states. The emission probabilities for the main states are shown in the below figure. After position 3, 3 out of 5 sequences are going to a deletion state, making this transition probability 3/5 and the transition probability to the next main state 2/5. Once in the deletion state, none of the sequences return to the main state immediately, but they continue for another 2 deletion states. At position 9, 1 out of 5 sequences is again going through a 3 nucleotide deletion. The model ends with the end state. The transition probabilities that are 1 are not indicated in the drawing.

![HMM Diagram]

**Question 5.2. Profile HMMs, II**

**a.** The HMM using only insertion states (other models are also possible):
b. The HMM using only deletion states (other models are also possible):

Question 5.3. Profile HMMs, III

a. The sequences giving rise to this logo can be detected by the HMM given in the question. The insertion and deletion states are not used, the nucleic acids in each position are possible according to the main states of the HMM.

b. No. Because on the 9th position "A" is not allowed according to the HMM.

c. Yes. The insertion state after position 2 is used.

Question 5.4. PSI-BLAST

The first round of a PSI-BLAST search is a regular BLAST. P4 has 3 mismatches with the query sequence while P5 has 5 mismatches; combined with the values in the BLOSUM62 matrix, this results in a higher score for hit P4 (and thus a lower E-value, i.e. a more significant hit). In the second round of the search, PSI-BLAST does not use the generic BLOSUM62 matrix, but builds a position-specific scoring matrix based on the results of the first round. As a result the 3 Proline residues at positions 8 to 10 become more important than positions 6, 7, 11, 12 and 13, because positions 8-10 are conserved in 4 out of 5 cases (P4 is the only hit in which they are not conserved), while the latter five positions are non-conserved in two of the five hits. Using this PSSM P5 becomes more significant than P4 in this second round of searching.

Question 5.5. Transcription factor binding sites
a. The consensus sequence looks as follows:

position 1-10:  T C C A [C/G] T [T/A] G A A

On position 5, there is equal chance for having a C or a G. On position 7, there is equal chance for having a T or a A.

b. You can calculate the information content on each site as follows (see your reader):

\[ I = \sum_{i=1}^{20} p_i \log_2 \frac{p_i}{1/20} = \log_2(20) + \sum_{i=1}^{20} p_i \log_2 p_i. \]  

(E.2)

Pay attention, that we here have nucleotide sequences and not protein sequences. Therefore \( i = 1,2,3,4 \), instead of \( i = 1,...,20 \) and the probability for all nucleotides being as frequent as each other is \( 1/4 \) instead of \( 1/20 \). \( p_i \) is the frequency of each nucleotide that you observed at the site. For example for the first site we observe 7 times (out of 8) a T. Therefore \( p_T = 7/8 = 0.875 \). For G we get \( p_G = 1/8 = 0.125 \). For A and C you easily see that \( p_A = p_C = 0 \). Remember that we assumed \( 0 \log_2(0) = 0 \). So the formula for the information content of the first site is:

\[ I = \log_2(4) + p_T \times \log_2(p_T) + p_G \times \log_2(p_G) \]

(E.3)

\[ = 2.0 + 0.875 \times \log_2(0.875) + 0.125 \times \log_2(0.125) \]

(E.4)

\[ \approx 1.5 \]  

(E.5)

The same way you can calculate the information content for each site and you get approximately the following values:

pos. 1-10:  1.46  0.59  1.05  0.59  1.00  2.00  1.00  2.00  1.05  2.00

**Question 5.6. Signal peptides**

a. The first main difference between the logo plots of gram-positive and gram-negative bacteria is the length of the hydrophobic region (with a lot of A and L). For gram-negative bacteria the hydrophobic region is approximately from position -6 to position -16, while for gram-positive bacteria it is from position -6 to position -21, i.e. it is longer.

If you look at the basic region (left of the hydrophobic region), you can see more M and K occurring in the gram-negative bacteria, while in the gram-positive ones you see more K and L occurring.

Another difference can be found in position -6, where the probability of having an A is much higher in gram-negative bacteria. Further minor differences can be found in positions 2 and
-5.

b. A major difference between eukaryotes and bacteria is the hydrophobic region. In eukaryotes, L is the most frequent amino acid in this region, while in bacteria L and A are equally likely.

Further, on positions -1 and -3 bacteria almost always have an A, while eukaryotes do not have such a strong preference. The basic region (position -21 to -25) in eukaryotes consists rather of M, while bacteria more often have a K.

c. On one hand bacterial signal peptides are easy to predict because of their high preference to have an A on position -1 and -3. On the other hand eukaryotic signal peptides can be identified quite easily using their hydrophobic region. Therefore it is hard to say anything conclusive! One has to try and see.

**Question 5.7. AEP**

a. N (Asparagine) is preferred by AEP at the cleavage site, and there are no exceptions to this.

b. This means that AEP is not using all N’s in a protein to make a cleavage.

**Question 6.1. Normalization**

a. See table.

b. The average $R/G$ is 0.95, indicating that overall fluorescence is generally 5% higher in the Green dye.

c. See table. No genes have $R/G > 2$ and genes B and I appear to have $R/G < 0.5$. i.e., the expression of these genes are downregulated to half.

d. See table. In the column normalized we report the adjusted $R/G$ ratios, which are simply the original $R/G$ values divided by the average $R/G$ value. For gene E, the $R/G$ value was close to 2, 1.947. When we take into account that the green dye is in general more intense, we indeed get $R/G$ value higher than 2, meaning that the gene E is expressed twice as much in the sample.

<table>
<thead>
<tr>
<th>Gene ID</th>
<th>Red</th>
<th>Green</th>
<th>$R/G$</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1378</td>
<td>1687</td>
<td>0.817</td>
<td>0.856</td>
</tr>
<tr>
<td>B</td>
<td>331</td>
<td>797</td>
<td>0.415</td>
<td>0.435</td>
</tr>
<tr>
<td>C</td>
<td>1045</td>
<td>1001</td>
<td>1.044</td>
<td>1.094</td>
</tr>
<tr>
<td>D</td>
<td>3663</td>
<td>2557</td>
<td>1.433</td>
<td>1.502</td>
</tr>
<tr>
<td>E</td>
<td>259</td>
<td>133</td>
<td>1.947</td>
<td>2.041</td>
</tr>
<tr>
<td>F</td>
<td>1882</td>
<td>2264</td>
<td>0.831</td>
<td>0.871</td>
</tr>
<tr>
<td>G</td>
<td>604</td>
<td>738</td>
<td>0.818</td>
<td>0.858</td>
</tr>
<tr>
<td>H</td>
<td>2611</td>
<td>2671</td>
<td>0.978</td>
<td>1.025</td>
</tr>
<tr>
<td>I</td>
<td>528</td>
<td>1092</td>
<td>0.484</td>
<td>0.507</td>
</tr>
<tr>
<td>J</td>
<td>458</td>
<td>593</td>
<td>0.772</td>
<td>0.81</td>
</tr>
</tbody>
</table>

**Question 6.2. Cancer**
The distance matrix (smallest distance is bold):

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>17</td>
<td>24</td>
<td>14</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>16</td>
<td>23</td>
<td>13</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>10</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>-</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a. Since A-B pair has the minimum distance, we will start by clustering A and B in one cluster. The distance of this cluster to an other point X is $\min(M(A, X), M(B, X))$, where M stands for Manhattan distance. So you do not need to calculate a new distance matrix after adding an element to a cluster, just disregard a part of it (the part to ignore after clustering A and B is Italic in matrix). In the second step C and D are clustered together and so on, until you end up with grouping all the genes. The final tree is shown below. For clarity, we give the tables for each step:

<table>
<thead>
<tr>
<th></th>
<th>A+ B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+B</td>
<td>-</td>
<td>16</td>
<td>23</td>
<td>13</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>10</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>-</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A+ B</th>
<th>C+ D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+B</td>
<td>-</td>
<td>16</td>
<td>23</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>C+D</td>
<td>-</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>10</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>-</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A+ B</th>
<th>C+ D+F</th>
<th>E</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+B</td>
<td>-</td>
<td>15</td>
<td>23</td>
<td>8</td>
</tr>
<tr>
<td>C+D+F</td>
<td>-</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A+ B</th>
<th>C+ D+F +G</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+B</td>
<td>-</td>
<td>8</td>
<td>23</td>
</tr>
<tr>
<td>C+D+F +G</td>
<td>-</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A+ B</th>
<th>C+ D+F +G +E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+B</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>C+D+F +G +E</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
b. The genes that are differently expressed for one or more type(s) of cancer can be used, in this case B, C, D, F.
c. You need a control expression value taken from normal (i.e. not a tumor) cells to compare these values with, whether and if so which genes are up- or downregulated.

**Question 6.3. Aging**

a. By making a BLAST search with yeast gene YGL163C against the human proteome, one can find human ortholog of this yeast gene.
b. Once we find an ortholog, we can compare its expression in young and old human cells using the microarray technology. If the human ortholog of YGL163C is also upregulated in cells from elderly, than you have good evidence for similar regulation occurring in human cells as well.

**Question 6.4. Amino acid starvation**

a. Clustering I is the result of single linkage clustering. This figure shows a more ”stairs like” pattern: clusters are increasing in size by the addition of single genes (in contrast to adding clusters of genes).
b. Three genes that are strongly down regulated at t=2h: YOL157C, YCL025C, YGR055W. These are identified most easily with clustering II.
c. Identify yeast genes with altered expression using the array and look for their orthologs in human.
Glossary

Alignment  The process of lining up two or more sequences to achieve maximal levels of identity (and conservation) for the purpose of assessing the degree of similarity and the possibility of homology.

Algorithm  A fixed procedure embodied to solve a problem (often used for computer programs).

Allele  In genetics, an allele (pronounced al-eel or al-e-ul) is any one of a number of viable DNA codings occupying a given locus (position) on a chromosome. Usually alleles are DNA (deoxyribonucleic acid) sequences that code for a gene, but sometimes the term is used to refer to a non-gene sequence. An individual’s genotype for that gene is the set of alleles it happens to possess. In a diploid organism, one that has two copies of each chromosome, two alleles make up the individual’s genotype.

Bioinformatics  The study of informatic processes in biotic systems.

BLAST  A sequence comparison algorithm optimized for speed used to search sequence databases for optimal local alignments to a query. The initial search is done for a word of length ”W” that scores at least ”T” when compared to the query using a substitution matrix. Word hits are then extended in either direction in an attempt to generate an alignment with a score exceeding the threshold of ”S”. The ”T” parameter dictates the speed and sensitivity of the search.

BLASTn  BLAST search for nucleotide sequences.

BLASTp  BLAST search for protein sequences.

BLOSUM  Blocks Substitution Matrix. A substitution matrix in which scores for each position are derived from observations of the frequencies of substitutions in blocks of local alignments in related proteins. Each matrix is tailored to a particular evolutionary distance. In the BLOSUM62 matrix, for example, the alignment from which scores were derived was created using sequences sharing no more than 62% identity. Sequences more identical than 62% are represented by a single sequence in the alignment so as to avoid over-weighting closely related family members.

Conservation  Changes at a specific position of an amino acid or sequence that preserve the physico-chemical properties of the original residue.

Consensus  A sequence of nucleotides or amino acids in common between regions of homology in different but related DNA or RNA or protein sequences.
**Domain** A discrete portion of a protein assumed to fold independently of the rest of the protein and possessing its own function.

**E value** Expectation value. The number of different alignments with scores equivalent to or better than $S$ that are expected to occur in a database search by chance. The lower the E value, the more significant the score.

**FASTA** The first widely used algorithm for database similarity searching. The program looks for optimal local alignments by scanning the sequence for small matches called "words". Initially, the scores of segments in which there are multiple word hits are calculated ("init1"). Later the scores of several segments may be summed to generate an "initn" score. An optimized alignment that includes gaps is shown in the output as "opt". The sensitivity and speed of the search are inversely related and controlled by the "k-tup" variable which specifies the size of a "word".

**Filtering** Also known as Masking. The process of hiding regions of (nucleic acid or amino acid) sequence having characteristics that frequently lead to wrong high scores.

**gap** A space introduced into an alignment to compensate for insertions and deletions in one sequence relative to another. To prevent the accumulation of too many gaps in an alignment, introduction of a gap causes the deduction of a fixed amount (the gap score) from the alignment score. Extension of the gap to encompass additional nucleotides or amino acid is also penalized in the scoring of an alignment.

**Global Alignment** The alignment of two nucleic acid or protein sequences over their entire length.

**Homology** Similarity attributed to descent from a common ancestor.

**HSP** High-scoring segment pair. Local alignments with no gaps that achieve one of the top alignment scores in a given search.

**Identity** The extent to which two (nucleotide or amino acid) sequences are invariant.

**Identity Matrix** A scoring system in which only identical characters receive a positive score.

**K** A statistical parameter used in calculating BLAST scores that can be thought of as a natural scale for search space size.

**lambda** A statistical parameter used in calculating BLAST scores that can be thought of as a natural scale for scoring system.

**Local Alignment** The alignment of some portion of two nucleic acid or protein sequences

**Low complexity regions** Regions of biased composition including homopolymeric runs, short-period repeats, and more subtle overrepresentation of one or a few residues.

**Masking** Also known as Filtering. The removal of repeated or low complexity regions from a sequence in order to improve the sensitivity of sequence similarity searches performed with that sequence.

**Motif** A short conserved region in a protein sequence. Motifs are frequently highly conserved parts of domains.
Multiple Sequence Alignment An alignment of three or more sequences with gaps inserted in the sequences such that residues with common structural positions and/or ancestral residues are aligned in the same column. Clustal W is one of the most widely used multiple sequence alignment programs.

Orthologous Homologous sequences in different species that arose from a common ancestral gene during speciation; may or may not be responsible for a similar function.

P value The probability of an alignment occurring with the score in question or better. The p value is calculated by relating the observed alignment score, S, to the expected distribution of HSP scores from comparisons of random sequences of the same length and composition as the query to the database. The most highly significant P values will be those close to 0. P values and E values are different ways of representing the significance of the alignment.

PAM Percent Accepted Mutation. A unit introduced by Dayhoff et al. to quantify the amount of evolutionary change in a protein sequence. 1.0 PAM unit, is the amount of evolution which will change, on average, 1% of amino acids in a protein sequence. A PAM(x) substitution matrix is a look-up table in which scores for each amino acid substitution have been calculated based on the frequency of that substitution in closely related proteins that have experienced a certain amount (x) of evolutionary divergence.

Paralogous Homologous sequences within a single species that arose by gene duplication.

Profile A table that lists the frequencies of each amino acid in each position of protein sequence. Frequencies are calculated from multiple alignments of sequences containing a domain of interest. See also PSSM.

Proteomics Systematic analysis of protein expression of normal and diseased tissues that involves the separation, identification and characterization of all of the proteins in an organism.

PSI-BLAST Position-Specific Iterative BLAST. An iterative search using the BLAST algorithm. A profile is built after the initial search, which is then used in subsequent searches. The process may be repeated, if desired with new sequences found in each cycle used to refine the profile.

PSSM Position-specific scoring matrix; see profile. The PSSM gives the log-odds score for finding a particular matching amino acid in a target sequence.

Query The input sequence (or other type of search term) with which all of the entries in a database are to be compared.

Raw Score The score of an alignment, S calculated as the sum of substitution and gap scores. Substitution scores are given by a look-up table (see PAM, BLOSUM). Gap scores are typically calculated as the sum of G, the gap opening penalty and L, the gap extension penalty. For a gap of length n, the gap cost would be $G + Ln$. The choice of gap costs, G and L is empirical, but it is customary to choose a high value for G (10-15) and a low value for L (1-2).

Similarity The extent to which nucleotide or protein sequences are related. The extent of similarity between two sequences can be based on percent sequence identity and/or conservation.
**Substitution**  The presence of a non-identical amino acid at a given position in an alignment. If the aligned residues have similar physico-chemical properties the substitution is said to be “conservative”.

**Substitution Matrix**  A substitution matrix containing values proportional to the probability that amino acid $i$ mutates into amino acid $j$ for all pairs of amino acids. Such matrices are constructed by assembling a large and diverse sample of verified pairwise alignments of amino acids. If the sample is large enough to be statistically significant, the resulting matrices should reflect the true probabilities of mutations occurring through a period of evolution.

**Supervised bioinformatics**  Algorithms where methods that map inputs to desired outputs are developed.
Credits

Writing this reader the book of Higgs & Attwood (2005) and Zvelebil & Baum (2008) has been an enormous source of inspiration. I used also several figures from these books. I suggest these books to anybody who would like to start and advance in bioinformatics. Chapter 5 was adapted from Krogh (1998). I had been using several images from internet for this reader. I hope they are all acknowledged, that was my intention. Also much of text and figures are from our recent book on Immunological Bioinformatics (Lund et al., 2005).

To make a bioinformatics course for first year biology students is a big challenge. Without the help of several people I would not be able to make it.

Jos Boekhorst revised all the pen and paper exercises and computer exercises in 2009 and had been the second “docent” in this course for two years. His contributions made it much more clear why bioinformatics can be fun.

Rob de Boer had been critically reading the reader and gave me a lot of ideas/suggestions on how to make a course for first year biology students.

Joost Beltman had been reading the whole reader from the perspective of a first year student and gave many tips to make it more clear.

Veronica Grieneisen had been going through all pen and paper exercises carefully, had very good suggestions to improve them and prepared the answers.

Berend Snel had been revising the Phylogeny chapter, and made it more correct and clear.

Levien van Zon had prepared excellent ”hints” for web exercises.

Lidija Berke had been proof-reading the reader in 2011, and found several typos that were hidden in the text in the last five years.

Also other assistants in 2006 had been working much more than for an average course and they deserve special thanks for all their suggestions/corrections/critics: Marian Groenenboom, Ellen Evers, Boris Schmid, John van Dam, and Like Fokkens.

Paulien Hogeweg had given many ideas while I was preparing this course. Ludo Pagie had been reading critically the chapter on micro-array analysis and gave suggestions to improve the exercises.
Bibliography


