Bioinformatics as a discipline has come of age, and there are now numerous databases and tools that are widely used by researchers in the biomedical field. However, successful development of future bioinformatics applications will depend on an appropriately formalised representation of domain knowledge.

This book provides a timely and first-of-its-kind collection of contributed chapters on anatomy ontologies. It is interdisciplinary in its approach, bringing together relevant expertise from computing and biomedical studies, and covering both theoretical and applied aspects, with an emphasis on newer work relevant to the emerging Semantic Web.

Topics and Features:
• Provides a comprehensive discussion of the foundations of anatomical ontologies and the state of the art in existing computational tools and applications
• Considers a number of fundamental modelling principles
• Includes chapters about research on algorithms to systematically align anatomy ontologies and to mine data in the literature, using anatomy terms
• Explains recent efforts to develop a common anatomy reference ontology
• Discusses anatomy in the context of spatio-temporal biomedical atlases
• Describes systems and tools for linking anatomy ontologies with each other and with other on-line resources, such as the biomedical literature
• Highlights the challenges of dealing with anatomy-based information on the Semantic Web

Although primarily written for readers who will be involved in developing the next generation of IT applications in the areas of life sciences, biomedical sciences and health care, this unique volume will be of interest to anyone who will further develop anatomy ontologies, who will use them, and who will be involved in the actual development of relevant (semantic) web applications.
Preface

Overview and goals

This book describes how to visualize and compare bacterial genomes. Sequencing technologies are becoming so inexpensive that soon going for a cup of coffee will be more expensive than sequencing a bacterial genome. Thus, there is a very real and pressing need for high-throughput computational methods to compare hundreds and thousands of bacterial genomes.

It is a long road from molecular biology to systems biology and in a sense this text can be thought of as a path bridging these fields. The goal of this book is to provide a coherent set of tools and methodological framework for starting with raw DNA sequences and producing fully annotated genome sequences, and then using these to build up and test models about groups of interacting organisms within an environment or ecological niche.

Organization and features

The text is divided into four main parts: Introduction, Comparative Genomes, Transcriptomics and Proteomics, and finally Microbial Communities. The first five chapters are introductions of various sorts. Each of these chapters represents an introduction to a specific scientific field, to bring all readers up to the same basic level before proceeding on to the methods to compare genomes. First, a brief overview of molecular biology and the concept of sequences as biological information are given. The equivalent in the post-genomics era of the ‘Central Dogma’ of molecular biology (DNA makes RNA makes protein) is that the genome makes the transcriptome, which makes the proteome. Before going on to the details of this, a historical background is provided that pictures the scene of the origins of molecular biology and biological sequences. After this introduction, Chapter 2 describes sequence alignment, the most common procedure to compare biological sequences. Instead of going into technical details of how exactly these alignments are calculated, the text focuses on their practical use. Chapter 3 introduces bacterial genomes and Chapter 4 deals with the most important databases, whilst Chapter 5 is an introduction to the computational background of the tools necessary to analyze all of this information.

The second part, on Comparative Genomics (Chapters 6-8), describes some basic methods to compare genomes. This section introduces various atlases building up to the ‘Genome Atlas’, which is our standard visualization tool for representing
the DNA sequence of a chromosome in a single figure, mapping the most relevant DNA properties along the chromosome. We have found such atlases very useful for mapping newly sequenced genomes and quickly visualizing regions of potential interest. The value of atlas projections is illustrated by the examples provided.

Part three (Chapters 9-11) takes the reader from genome sequences to RNA sequences (transcriptomics) to proteins (proteomics), and regulation of gene expression. An important overview of experimental results can be obtained by mapping back and visualizing the transcriptomic and proteomic data onto physical chromosomal maps. Examples illustrate how important chromosome location is, and which features can be predicted by careful analysis of genes and their surrounding sequences.

The final part (Chapters 12-14) deals with microbial communities. In a sense this can be thought of as ‘population genomics’ (as opposed to the more traditional ‘population biology’ which often focuses on only one or a few genes). First the concept of ‘pan-genome’ and ‘core-genome’ is introduced (chapter 12), followed by metagenomics (chapter 13), and then evolution of microbial communities (chapter 14). From a larger perspective, population genomics can provide a framework for modeling ecosystems in terms of interacting biological systems.

Target audiences and required background knowledge

The reader should have basic knowledge about computers and be able to use web servers. For programmers, some general knowledge on microbiology is assumed, but it is our hope that both programmers and more ‘biology-oriented’ readers will find this book helpful. Details on programming were deliberately left out; instead, the text concentrates on the use and interpretation of publicly available web tools. This book has grown out of lectures for the course in Comparative Microbial Genomics1, which DWU taught since 2001 as a full semester length course at the Technical University of Denmark, and one-week workshop versions, given in Bangkok, Thailand, in Petropolis, Brazil, and in Oslo, Norway.

This book is in a sense about merging different scientific languages. The three authors have different scientific and national backgrounds. DWU is from the U.S., studied biochemistry, worked in molecular biology and for the last 10 years lead a group in bioinformatics and genomics. SB is from Italy, studied quantum chemistry with focus on scientific programming, data standardization and software integration, whereas TMW studied biochemistry and worked in molecular biology, and later as consultant in microbiology. These different backgrounds actually helped to develop a common language in science. The subject area of this text-

1 http://www.cbs.dtu.dk/dtucourse/programme2744.php
book is extremely interdisciplinary, covering, (bio)chemistry, physics, biology, microbiology, mathematics and computational science, and by introduction concepts (and some jargon) from these different disciplines, the different languages used by specialists are bridged.

This book is meant mainly for people studying bacterial genomes, although of course nearly all of the methods described in the text would work for viral, Archaeal or Eukaryotic genomes as well. There are two main target audiences. The first is the microbiologist who wants to get the best out of a bacterial genome sequence. These could be university students or experienced laboratory microbiologists who enter the field of genomics. This book enables one to get a handle on how to use high-throughput computational methods to compare only a few, or hundreds of sequenced genomes. The second audience comprises the computer programmers who assist these microbiologists in actually carrying out the analyses. From experience we know there can be communication problems between the experimental bacteriologist who is more laboratory-oriented, and the computer scientist who wants to do everything on computers. Both disciplines are essential in present-day research. This book aims to explain to the computational scientist why and how we want to study bacterial genomes, and what questions we hope to answer. At the same time, it explains to the biologist some of the basics behind bioinformatic tools that are necessary for research in the field. Bringing those two worlds, scientific interests, and languages together is our ultimate goal.

Notes to the instructor

There are no exercises or questions at the end of the chapter, although textboxes present descriptions of essential methods used at the end of most chapters. From experience we can say that giving small groups of students a project where they can choose a recently sequenced bacterial genome and compare this to other similar genomes, can produce surprisingly successful results. It is very motivating to work with recently published data (new genome sequence papers are being published on an almost daily basis now), and sometimes they produce important observations that the authors of the scientific papers had missed! In some occasions, such activities have resulted in a real scientific publication by the students, illustrating how ‘easy’ it is to do these kinds of analyses, as long as one asks relevant questions.

Supplemental resources

A number of web links are mentioned in the book and since web addresses are not always stable, a dedicated web page is put up on which all web pages presented in the book are summarized, and as necessary, updated. This can be found at http://comparativemicrobial.com.
Acknowledgements

This book contains input from many people, including our research team and external collaborators. We are grateful for all advice, assistance and help we received throughout this project. We thank all current and former members of the Comparative Microbial Genomics group at CBS, in particular Peter F. Hallin for his excellent programming skills and help with development of many of the programs mentioned in this book; Flemming Hansen for his work on bacterial replication and his vast knowledge of *E. coli*; Henrik J. Nielsen for his help with *E. coli* genomics; Kristoffer Kiil for his help with phylogeny and work on protein function; and Carsten Friis for his assistance with various analyses and for keeping the group running whilst we were writing.

We thank former group members whose work also contributed to this book, including Tim T. Binnewies for his work with *Vibrio* genomes and secretion systems and Hanni Willenbrock for her work with developing pan-genome microarrays.

We are grateful to external collaborators, notably Thomas Quinn from Denver University for his work on phylogenetic trees whilst on sabbatical in our group; Karin Lagesen from CMBN, Institute of Medical Microbiology, Rikshospitalet University Hospital in Oslo who has helped with the rRNA and tRNA searches; and Jon Bohlin from the Norwegian School of Veterinary Science who has helped with analysis of oligonucleotide usage patterns in bacterial genomes.

We would also like to acknowledge help from the many people at CBS, which is currently one of the largest bioinformatics groups in Europe. In particular, we thank Hans Henrik Stærfeldt from the CBS systems administration group, who wrote the original code for the GeneWiz program that is used to construct the atlas plots, and for his help and support over the past 10 years in updating and maintaining GeneWiz. Jannick D. Bendtsen helped us on the secretome, and Thomas Blicher kindly provided wonderful pictures of protein structures...Finally, Søren Brunak, center director for CBS has established a wonderful place to work (including an excellent coffee machine!) and has been supportive of our group since it was formed in 1998.

David would like to thank his students over the years in his Comparative Microbial Genomics course for their many helpful suggestions and comments. He would also like to thank his wife for helpful editorial comments and her support during the writing of this book.

Stefano would like to thank his parents, Paola Marani and Walter Padovani, for the constant support and trust, and his dear friends Paolo Soriani and Ruggero Paratelli for a life-long sharing of experiences and fun.

Trudy would like to thank her son Martijn for inventing the analogy of a road to explain DNA strand direction, both of her sons for their understanding and patience and her husband for his constant support.
Much of the work described in this textbook has been funded by grants over the past decade from the Danish National Research Foundation (Danmarks Grundforskningsfond), Danish Research Councils and the EU. Many of the calculations presented in this book have been made on our large computer system at CBS, funded in part by money from the Danish Center for Scientific Computing.

David Ussery, Stefano Borini and Trudy Wassenaar
May 2008
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asn</td>
<td>Asparagine</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Adenine</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Alanine</td>
<td></td>
</tr>
<tr>
<td>aa</td>
<td>Alanine amino acid</td>
<td></td>
</tr>
<tr>
<td>Ala</td>
<td>Alanine</td>
<td></td>
</tr>
<tr>
<td>Arg</td>
<td>Arginine</td>
<td></td>
</tr>
<tr>
<td>Asp</td>
<td>Aspartic acid</td>
<td></td>
</tr>
<tr>
<td>ATP</td>
<td>Adenosine-triphosphate</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>G or T or C</td>
<td></td>
</tr>
<tr>
<td>BLAST</td>
<td>basic local alignment search tool</td>
<td></td>
</tr>
<tr>
<td>BLASTN</td>
<td>BLAST for DNA sequences searching DNA sequences</td>
<td></td>
</tr>
<tr>
<td>BLASTP</td>
<td>BLAST for protein sequences searching protein sequences</td>
<td></td>
</tr>
<tr>
<td>BLASTX</td>
<td>BLAST for DNA sequences searching protein sequences</td>
<td></td>
</tr>
<tr>
<td>bp</td>
<td>base pair(s)</td>
<td></td>
</tr>
<tr>
<td>BSE</td>
<td>bovine spongiform encephalopathy</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Cytidine</td>
<td></td>
</tr>
<tr>
<td>CBS</td>
<td>Center for Biological Sequences</td>
<td></td>
</tr>
<tr>
<td>CLUSTAL</td>
<td>cluster-all (multiple alignment program)</td>
<td></td>
</tr>
<tr>
<td>CLUSTALW</td>
<td>CLUSTAL with a command line interface</td>
<td></td>
</tr>
<tr>
<td>CLUSTALX</td>
<td>CLUSTAL with a graphical user interface</td>
<td></td>
</tr>
<tr>
<td>CML</td>
<td>Chemical markup language</td>
<td></td>
</tr>
<tr>
<td>COG</td>
<td>cluster of orthologous genes</td>
<td></td>
</tr>
<tr>
<td>CRISPR</td>
<td>clustered, regularly interspersed short palindromic repeats</td>
<td></td>
</tr>
<tr>
<td>C-terminal</td>
<td>Carboxy-terminal</td>
<td></td>
</tr>
<tr>
<td>Cys</td>
<td>Cysteine</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Aspartic acid</td>
<td></td>
</tr>
<tr>
<td>DBMS</td>
<td>database management system</td>
<td></td>
</tr>
<tr>
<td>DDBJ</td>
<td>the DNA database of Japan</td>
<td></td>
</tr>
<tr>
<td>DNA</td>
<td>deoxyribonucleic acid</td>
<td></td>
</tr>
<tr>
<td>dsDNA</td>
<td>double strand DNA</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Glutamic acid</td>
<td></td>
</tr>
<tr>
<td>EBI</td>
<td>European Bioinformatics Institute</td>
<td></td>
</tr>
<tr>
<td>EC</td>
<td>enzyme commission</td>
<td></td>
</tr>
<tr>
<td>eggNOG</td>
<td>evolutionary genealogy of genes: non-supervised orthologous groups</td>
<td></td>
</tr>
<tr>
<td>EMBL</td>
<td>European Molecular Biology Laboratories</td>
<td></td>
</tr>
<tr>
<td>ExPASy</td>
<td>expert protein analysis system</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Phenylalanine</td>
<td></td>
</tr>
<tr>
<td>FASTA</td>
<td>fast-all (pairwise alignment program)</td>
<td></td>
</tr>
<tr>
<td>FIS</td>
<td>[binding site for Fis protein]</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Guanine</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Glycine</td>
<td></td>
</tr>
<tr>
<td>GCDML</td>
<td>genomic contextual data markup language</td>
<td></td>
</tr>
<tr>
<td>GEI</td>
<td>genomic islands</td>
<td></td>
</tr>
<tr>
<td>GIS</td>
<td>geographical information system</td>
<td></td>
</tr>
<tr>
<td>Gln</td>
<td>Glutamine</td>
<td></td>
</tr>
<tr>
<td>Glu</td>
<td>Glutamic acid</td>
<td></td>
</tr>
<tr>
<td>Gly</td>
<td>Glycine</td>
<td></td>
</tr>
<tr>
<td>GOLD</td>
<td>genomes online database</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Histidine</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>A or C or T</td>
<td></td>
</tr>
<tr>
<td>H-bond</td>
<td>Hydrogen-bond</td>
<td></td>
</tr>
<tr>
<td>His</td>
<td>Histidine</td>
<td></td>
</tr>
<tr>
<td>HTML</td>
<td>hypertext markup language</td>
<td></td>
</tr>
<tr>
<td>HTTP</td>
<td>hypertext transfer protocol</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>Isoleucine</td>
<td></td>
</tr>
<tr>
<td>IHF</td>
<td>integration host factor</td>
<td></td>
</tr>
<tr>
<td>Ile</td>
<td>Isoleucine</td>
<td></td>
</tr>
<tr>
<td>INSDC</td>
<td>International nucleotide sequence database collaboration</td>
<td></td>
</tr>
<tr>
<td>IQR</td>
<td>inter-quartile range</td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>Insertion Sequence</td>
<td></td>
</tr>
<tr>
<td>ISBN</td>
<td>International standard book number</td>
<td></td>
</tr>
<tr>
<td>ITS</td>
<td>internally transcribed spacer</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>Lysine</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>G or T</td>
<td></td>
</tr>
<tr>
<td>K_a</td>
<td>non-synonymous mutation rate</td>
<td></td>
</tr>
<tr>
<td>kDa</td>
<td>kilo Dalton</td>
<td></td>
</tr>
<tr>
<td>K_s</td>
<td>synonymous mutation rate</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>Leucine</td>
<td></td>
</tr>
<tr>
<td>Leu</td>
<td>Leucine</td>
<td></td>
</tr>
<tr>
<td>Lip</td>
<td>Lipoprotein secretion signal</td>
<td></td>
</tr>
<tr>
<td>Lys</td>
<td>Lysine</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>Methionine</td>
<td></td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>A or T</td>
<td></td>
</tr>
<tr>
<td>Met</td>
<td>Methionine</td>
<td></td>
</tr>
<tr>
<td>MIGS</td>
<td>Minimal information about a genome sequence</td>
<td></td>
</tr>
<tr>
<td>MIMS</td>
<td>Minimal information about a metagenomic sequence</td>
<td></td>
</tr>
<tr>
<td>mRNA</td>
<td>messenger RNA</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>any nucleotide</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>Asparagine</td>
<td></td>
</tr>
<tr>
<td>NCBI</td>
<td>National Center for Biotechnology Information</td>
<td></td>
</tr>
<tr>
<td>ncRNA</td>
<td>non-coding RNA</td>
<td></td>
</tr>
<tr>
<td>NEWT</td>
<td>[a taxonomy database]</td>
<td></td>
</tr>
<tr>
<td>NIH</td>
<td>National Institute of Health</td>
<td></td>
</tr>
<tr>
<td>NLM</td>
<td>National Library of Medicine</td>
<td></td>
</tr>
<tr>
<td>nt</td>
<td>nucleotide(s)</td>
<td></td>
</tr>
<tr>
<td>N-terminal</td>
<td>Amino-terminal</td>
<td></td>
</tr>
<tr>
<td>OH</td>
<td>Hydroxyl</td>
<td></td>
</tr>
<tr>
<td>ORF</td>
<td>open reading frame</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>Proline</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>promoter</td>
<td></td>
</tr>
<tr>
<td>PAI</td>
<td>pathogenicity island</td>
<td></td>
</tr>
<tr>
<td>PCR</td>
<td>polymerase chain reaction</td>
<td></td>
</tr>
<tr>
<td>PDB</td>
<td>Protein database</td>
<td></td>
</tr>
<tr>
<td>PEDANT</td>
<td>Protein extraction, description and analysis tool</td>
<td></td>
</tr>
<tr>
<td>Phe</td>
<td>Phenylalanine</td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td>genome project ID</td>
<td></td>
</tr>
<tr>
<td>PIR</td>
<td>Protein information resource</td>
<td></td>
</tr>
<tr>
<td>PMID</td>
<td>PubMed ID</td>
<td></td>
</tr>
<tr>
<td>Pro</td>
<td>Proline</td>
<td></td>
</tr>
<tr>
<td>PSD</td>
<td>Protein sequence database</td>
<td></td>
</tr>
<tr>
<td>pur</td>
<td>purine</td>
<td></td>
</tr>
<tr>
<td>pyr</td>
<td>pyrimidine</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>Glutamine</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Arginine</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>G or A</td>
<td></td>
</tr>
<tr>
<td>RCSB</td>
<td>Research collaboratory for structural bioinformatics</td>
<td></td>
</tr>
<tr>
<td>RNA</td>
<td>ribonucleic acid</td>
<td></td>
</tr>
<tr>
<td>rRNA</td>
<td>ribosomal RNA</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Serine</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>G or S</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Svedberg unit</td>
<td></td>
</tr>
<tr>
<td>Sec</td>
<td>secretion</td>
<td></td>
</tr>
<tr>
<td>Ser</td>
<td>Serine</td>
<td></td>
</tr>
<tr>
<td>SIB</td>
<td>Swiss Institute of Bioinformatics</td>
<td></td>
</tr>
<tr>
<td>SNP</td>
<td>single nucleotide polymorphism</td>
<td></td>
</tr>
<tr>
<td>SOA</td>
<td>service oriented architecture</td>
<td></td>
</tr>
<tr>
<td>SOAP</td>
<td>SOA protocol</td>
<td></td>
</tr>
<tr>
<td>SQL</td>
<td>structured query language</td>
<td></td>
</tr>
<tr>
<td>ssDNA</td>
<td>single-strand DNA</td>
<td></td>
</tr>
<tr>
<td>STRING</td>
<td>Search tool for the retrieval of interacting genes / proteins</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Threonine</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Thymine</td>
<td></td>
</tr>
<tr>
<td>T3SS</td>
<td>type III secretion system</td>
<td></td>
</tr>
<tr>
<td>T4SS</td>
<td>type IV secretion system</td>
<td></td>
</tr>
<tr>
<td>Tat</td>
<td>twin Arginine (secretion signal)</td>
<td></td>
</tr>
<tr>
<td>Thr</td>
<td>Threonine</td>
<td></td>
</tr>
<tr>
<td>tmRNA</td>
<td>transfer-messenger RNA</td>
<td></td>
</tr>
<tr>
<td>trEMBL</td>
<td>Translated EMBL database</td>
<td></td>
</tr>
<tr>
<td>tRNA</td>
<td>transfer RNA</td>
<td></td>
</tr>
<tr>
<td>Trp</td>
<td>Tryptophan</td>
<td></td>
</tr>
<tr>
<td>Tyr</td>
<td>Tyrosine</td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>Uracil</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>Uniform resource locator</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Valine</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>G or C or A</td>
<td></td>
</tr>
<tr>
<td>Val</td>
<td>Valine</td>
<td></td>
</tr>
<tr>
<td>VCR</td>
<td>Vibrio cholerae repeat</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>Tryptophan</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>A or T</td>
<td></td>
</tr>
<tr>
<td>WGS</td>
<td>Whole genome shotgun database</td>
<td></td>
</tr>
<tr>
<td>WSDL</td>
<td>Web services description language</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>any aa</td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td>Extensible markup language</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>Tyrosine</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>nucleotide T or C</td>
<td></td>
</tr>
</tbody>
</table>
# Table of Contents

Preface i

## PART 1: INTRODUCTION

1 Sequences as Biological Information: Cells Obey the Laws of Chemistry and Physics
   - Why study microbes? 1
   - What is biological information and where does it come from 2
   - How DNA sequences code for information 5
   - From DNA to protein: transcription and translation 8
   - DNA sequences: more than protein-coding genes 11
   - The two strands of DNA 11
   - From DNA to DNA: replication 12
   - Proteins: structure and function P13

2 Bioinformatics for Microbiologists: an Introduction
   - Identifying similarities: sequence comparison by means of alignments 18
   - Pairwise alignments: BLAST and FASTA 22
   - Multiple alignments: CLUSTALW 25
   - From alignments to phylogenetic relationships 28
   - Genome annotation: the challenge to get it right 30
   - Information beyond the single genome 32

## PART 2: BASICS of COMPUTATIONAL GENOMICS

3 Microbial Genome Sequences: A New Era in Microbiology
   - The first completely sequenced microbial genome 36
   - The importance of visualization 38
   - A race against the clock: the speed of sequencing 44
   - The first completely sequenced bacterial genome 45
   - Comparative bacterial genomics 47
   - The microbial genome: variation and diversity versus similarity and commonness 49

4 An Overview of Genome Databases
   - Introduction to databases 52
   - What is a database? 52
   - Three databases storing sequences and a lot more 56
Data files and formats 60
DNA Databases - How does one compare genome sequences? 63
RNA databases 63
Protein databases 64

5 The Challenges of Programming: a Brief Introduction 70
What’s in a name? Data and metadata 71
A look at the most common bioinformatic procedures 74
Data extraction for computational analysis 74
Data analysis: where the action is 77
Data presentation and storage 78
Achieving better automation 82
Implementing scalable processes 84
Going technical: the tools and the future 85
Programming Languages 85
Markup languages 88
Service Oriented Architecture 89
Specific tools for bioinformatic use 91

PART 3: METHODS to COMPARE GENOMES

6 Methods to Compare Genomes: the First Examples 94
Genomic comparisons: the size of a genome 94
Pairwise alignment of genomes 97
Comparing gene content and annotation quality 99
RNA Comparisons: a look at rRNAs 101
Proteome comparisons: what makes a family? 102

7 Genome Length and DNA Base Composition 109
Length of genomes: the “C-value paradox” 110
Genomic average base composition: the percentage AT 113
GC skew - bias towards the replication leading strand 118
Global chromosomal bias of AT content 122
DNA structures 125
The Structure Atlas 128
Bias in purines - A-DNA Atlases 129

8 Word Frequencies, Repeats, and Repeat-related Structures in Bacterial Genomes 133
Analyzing word frequencies in a genome 134
DNA repeats within a chromosome 136
Introduction to the DNA Repeat Atlas 142
Local DNA repeats are related to the percentage AT of
143 DNA Structures related to repeats in sequences
145 More on Structure Atlases
148 The Genome Atlas: our standard method for visualization of bacterial genomes

PART 4: TRANSCRIPTION and TRANSLATION of GENOMES

9 Transcriptomics: translated and untranslated RNA in Microbial Genomes
  Counting rRNA and tRNA genes
  A closer look at ribosomal RNA
  Genes encoding transfer RNA
  Genes encoding mRNA: comparing codon use between bacteria
  Other non-coding RNA: tmRNA

10 Expression of Genes and Proteins
  Comparing gene expression and protein expression
  Part 1: Regulation of translation
  Part 2: Regulation of translation
  Part 3: Protein modification and cellular localization
  Antigen and epitope prediction

11 Of Proteins, Genomes and Proteomes
  PART 1: Analysis of individual protein-encoding genes
  PART 2: How to annotate a complete genome
  Technical developments in genome sequencing
  PART 3: Proteome comparisons

PART 5: COMPUTATIONAL MODELS of BACTERIAL COMMUNITIES

12 Microbial Communities: Core and Pan-genomics
  Defining pan-genomes and core genomes
  Current data available for pan- and core genome analysis
  The pan- and core genome of Streptococcus
  The current Bacillus pan- and core genome
  The Burkholderia pan- and core genome

13 Metagenomics of Microbial Communities
  Metagenomics based on 16S rRNA analysis
  Metagenomics based on complete DNA sequencing
  Environmental influences on base composition
Visualization of environmental metagenomics data 237
Marine metagenomics 241
Other metagenomics applications 242

14 Evolution of Microbial Communities -or-
On the Origins of Bacterial Species 245
Where does diversity come from? 246
Evolution takes time 248
Evidence of evolution in a single genome 249
Genomic islands 252
Darwinian Evolution on a chip 255
Bacterial species and speciation: Vibrio cholerae 256
Can we predict evolution? Escherichia coli genome reduction 258

Index 261
1 Sequences as Biological Information: Cells Obey the Laws of Chemistry and Physics

Outline Molecular biology has revolutionized our understanding of life. Biological information is organized in a way that resembles text, meaning that biological information is based on the specific order of components (a sequence of building blocks) forming biological polymers. The building blocks are monomeric subunits of nucleic acids or amino acids, which form long polymers (DNA, RNA, and proteins). Their sequence determines their shape, and it is this shape (structure) that determines function. The Central Dogma of biology states that the flow of biological information is from DNA to RNA to proteins. The development of high-throughput methods to determine DNA sequences is revolutionizing our approach to study life. In the age of sequenced genomes, the flow of scientific information now frequently can be read from the genome to the transcriptome, proteome, and cellular components. Cellular processes obey the laws of chemistry and physics, and we can use information from biological sequences to model structures and extrapolate their functions, without the need to resort to unexplainable ‘vital life forces’ for the molecular basis of life.
Why study microbes?

Aristotle divided all life into three basic kingdoms: Plants, Animals, and Minerals. Minerals are no longer considered a kingdom of life, although diatoms living in the ocean, such as *Thalassiosira pseudonana*, have characteristics of plants, animals, and minerals (Armbrust et al., 2004). Three Kingdoms are still recognized today, as shown in Figure 1.1, although for historical reasons these are often referred to as ‘SuperKingdoms’. Two of these, Bacteria and Archaea, are based entirely on unicellular organisms that do not have a nucleus and are too small to see without the aid of a microscope. They are jointly referred to as prokaryotes. The third kingdom Eucarya (also spelled Eukarya), is characterized by cells that contain a nucleus; it includes all plants and animals but compared to microbes these represent only a tiny fraction of the overall diversity.

Figure 1.1. A phylogenetic tree displaying the genetic distances between members of the three 'SuperKingdoms' of life: Bacteria, Archaea and Eucarya. The represented bacterial genera will appear in examples throughout the book. The distance between bacterial genera is much larger than that of plants and animals, drawn on the same scale of genetic distance.
Though not generally appreciated, even the kingdom of eukaryotes is dominated by microscopic life. Most life is microscopic, in terms of the physical number of organisms, the number of species present in the environment, and, despite their small size, the biomass they represent on a worldwide scale. Even inside an animal, microbes are abundant: only one out of every 10 cells in a human body is actually human, whilst the other 9 cells are prokaryotic.

From an evolutionary perspective, bacteria and archaea have been around for more than 3 billion years; plants and animals are relatively recent ‘newcomers’ on the scene, arriving less than half a billion years ago. Since bacteria and archaea can divide rather quickly and have had much more time to evolve, their diversity by far exceeds that of eukaryotes (the members of Eucarya). Our human perception is that plants and animals are completely unlike each other, and so are, say, insects and mammals, as they are strikingly different even at first sight. The diversity of microbes, however, cannot be judged from their looks. Only when zooming in at their genetic material do we appreciate their diversity. In a phylogenetic tree, which depicts genetic lineages, the microbial world is dominant and so diverse that, on the drawn scale of diversity as shown in Figure 1.1, plants and animals actually group very close together.

What is biological information and where does it come from?

It is obvious that children often resemble their parents, and for thousands of years, humans have wondered about how hereditary traits are passed from one generation to the next. Several hundred years ago, it was thought that sperm cells contained ‘little people’ inside of them, which then somehow enlarged to become children. Although this concept proved to be incorrect, the subsequently proven nature of heredity builds on the underlying concept of an organism inheriting from its parents a complete ‘blueprint’ that determines its outcome. The ongoing debate about ‘nature vs. nurture’ demonstrates that the environment too, to some degree, determines the outcome of reproduction. Original models of genetic blueprints did not allow for such environmental effects, and although the impact of environmental factors acting on embryonic development is still investigated, it is clear that life is a dance of interactions between the genetic material and the environment.

The physical basis of heredity

In 1866, the Czech monk Gregor Mendel proposed that there were physical units of inheritance. These were responsible for attributes such as how tall or short an
organism was, as well as other characteristics. Mendel proposed a theoretical unit of inheritance, called a gene. The chemical structure for the four bases present in DNA was determined a few years later, by Albrecht Kossel, although at the time there was no link between DNA and these so-called genes. It took many decades of experimental detective work to determine that the physical basis of the genes causing Mendel’s traits were due to the activity of specific proteins, which were again encoded by DNA. Mendel’s ideas were ignored and largely forgotten, and the search for the basis of heredity took a few detours before the impact of his observations was realized.

The first clue that genes were something real, rather than just a theoretical unit, came from studies of fruit flies. In the early 1900s, photographs of cells from Drosophila showed some densely staining structures, called chromosomes, when the cells were getting ready to divide. Careful analysis of these structures showed a correlation with different characteristics of the fruit flies, and eventually it was proposed that chromosomes contained the hereditary information. How this information was stored in the chromosomes was not known, but for the next few decades, it was thought that proteins were responsible for this storage of information. They were a more likely candidate than DNA, since proteins were known to contain many more different building blocks (i.e., 20 different amino acids) than DNA, which for many years was considered to be a boring polymer of repeats of the four different nucleotide subunits.

**What is the genetic material?**

During the 1930s, George Beadle and Edward Tatum proposed the “one gene, one enzyme” hypothesis, stating that a gene somehow correlates with an enzyme. At the time enzymes were known to consist of proteins, and proteins were still assumed to be the basis of genetic material. However, a few years later, Oswald Avery, Maclyn McCarty and Colin MacLeod demonstrated experimentally that DNA was the material of inheritance. The initial reaction to this was skepticism, although this work was inspirational for James Watson and Francis Crick, who in 1953 published a model of the DNA double helix. Francis Crick was also inspired by what the physicist Erwin Schrödinger had written in 1943: “We believe a gene - or perhaps the whole chromosome fibre - to be an aperiodic solid”. Schrödinger further compared the gene to Morse code, which can encode information by using different combinations of dots and dashes. From this came the idea that somehow the DNA sequence was the genetic material inherited from one generation to the next and that this contained information on how to make proteins.

In the 1950s, Watson proposed the “General Idea” of molecular biology, which consisted of three parts: *The Sequence Hypothesis* - describing that the amino acid
sequence in proteins is specified from DNA via RNA sequences; *The Central Dogma*, which states that once the information flows from DNA to RNA to protein, it can’t flow backwards; *The Structure / Function Relationship*, which states that the sequence of DNA (or RNA or protein) determines its shape, and the shape determines its function. All three parts have proven largely to be correct.

**Cells obey the laws of chemistry**

Despite the protests of the Intelligent Design community, more than 40 years after the first publication of Watson’s “Molecular Biology of the Gene”, it is still clear that cells obey the laws of chemistry and physics. We can understand the flow of biological information in terms of coding sequences, which can be used to model structures and then functions, with no need to resort to some sort of unexplainable vital life force for the molecular basis of life. Maybe it is easier to understand or accept this when dealing with microbes than with complex eukaryotes. However, the basic biological processes taking place in, say, a mammalian nerve cell are the same as in a bacterial cell. We can’t yet completely understand complex biological processes such as thought or memory, but the underlying biology obeys the laws of chemistry, just like the movement of a microbe towards a food source is ruled by chemical processes.

In conclusion, despite their apparently mysterious characteristics, the mechanisms behind heredity, evolution, protein expression and synthesis have clear and predictable behavior, which can be traced down at the molecular level, and is in compliance with the rules of chemistry. “Until recently, heredity has always seemed the most mysterious of life’s characteristics. The current realization that the structure of DNA already allows us to understand practically all its fundamental features at the molecular level is thus most significant. We see not only that the laws of chemistry are sufficient for understanding protein structure, but also that they are consistent with all known hereditary phenomena.” from *Molecular Biology of the Gene*, by James D. Watson (W.A. Benjamin Inc., New York, 1965), page 67.

**How DNA sequences code for information**

The building blocks and chemical and physical properties of DNA are the same in all thee SuperKingdoms of life (Figure 1.2). The order in which these four DNA *nucleotides* are connected, *i.e.*, their *sequence*, is the code by which information is stored. DNA has a double role of coding (amongst other information) for protein sequences, and ensuring that the offspring of a cell can continue to use that infor-
mation by passing on an exact replica of itself. The elegance of the system is that only one copy of a DNA molecule is enough to fulfill both roles. Proteins, on the other hand, are both far more abundant in a cell and more diverse. Graphical representation of the chemical structure of nucleotides and DNA is a rather elaborate way to represent the sequence that codes for the information stored in DNA. Instead, the one-letter abbreviation of the four DNA nucleotides, A, G, C and T are normally used.

These names were more or less arbitrarily chosen after the base part, which determine their nature: for instance, the base Guanine was first isolated from bird’s droppings used as fertilizer, called guano after Latin-American Spanish, huanu, dung. A different way of representing DNA nucleotides would have been equally sufficient, like 4 different musical tones, or 4 differently colored dots. The one-letter convention is shorthand for the complex structure that DNA represents.

![Nucleotides](Image)

**Figure 1.2.** Nucleotides, the building blocks of DNA and RNA, consist of a base, a sugar and one to three phosphate groups (top). As the arrow indicates, the sugar of DNA lacks a hydroxyl group, hence its name ‘deoxy’. The base is either a purine or a pyrimidine. DNA combines the sugar deoxyribose with either Adenine, Guanine, Cytosine or Thymine. The sugar of RNA is always ribose, and its bases are the same as in DNA, with the exception of Uracil, replacing Thymine. Uracil misses the methyl group at carbon 5. An example of a DNA and an RNA nucleotide is given at the bottom.
The order in which the 20 building blocks of proteins, the amino acids, are connected determines the shape and function of proteins. Proteins are the ‘working horses’ of a cell, responsible for structural tasks, metabolism, transport, regulation, signal propagation and many activities. To produce proteins from DNA, RNA is required to ensure the DNA code message can be translated into protein code. These messenger RNA molecules (mRNA) are specifically produced upon demand, for the production of specific proteins. The structure of RNA resembles that of DNA in many respects, but the sugar building blocks in ribonucleotides are slightly different (Figure 1.2). One ribonucleotide is different from its DNA counterpart and hence bears a different name: Uracil in RNA is the equivalent of Thymine in DNA.

The key feature of nucleotide polymers is that they tend to form hydrogen bonds between the base moieties. In the so-called canonical Watson-Crick pairing, base A only pairs with T, and G only with C (Figure 1.3).

**Figure 1.3.** Graphic of a DNA double helix showing the base pairs holding the two strands together. The ribose-phosphate backbone is colored green, the bases are black and the hydrogen bonds are blue. A G-C base pair is held together by three hydrogen bonds, whereas an A-T base pair only has two. The two strands run in opposite directions, from 5'-phosphate to 3'-hydroxyl. Each strand can only be extended on the 3'-OH group, given in red.
The result is that two polymers are aligned to form a double-stranded molecule (dsDNA) that spins around its axis like a slightly crooked helical ladder, with the base pair (bp) forming the steps. The hydrogen bonds between the base pairs are much weaker than the covalent bonds connecting the individual nucleotides within a strand. The two strands can disassemble relatively easy, a process called ‘melting’ or ‘denaturation’. Such local separation of the two DNA strands into localized single-stranded DNA allows the bases of the nucleotides to pair with loose nucleotides or alternative polymers, both events being essential steps to important processes of life.

**From DNA to protein: transcription and translation**

Production of protein from DNA in the cell is a two-step event (Figure 1.4). First, during transcription the code of DNA sequence is transcribed into mRNA. This step ensures that only a fraction of the DNA code is used, which specifically encodes for the protein that is needed. Double-strand DNA will locally melt, and the now freely available DNA nucleotides can pair with ribonucleotides. With the help of the enzyme RNA polymerase, these ribonucleotides are added and linked together one after another to form an mRNA polymer. Only one DNA strand serves as a template. The hetero-duplex DNA/RNA complex is less stable than dsDNA, so that eventually the DNA helix closes again and the now completed mRNA is separated. The result is an RNA copy of part of the DNA strand that is then called the coding strand. DNA and RNA both have 4 building blocks so their code is translated on a 1 to 1 basis.

![Figure 1.4. From DNA to RNA to protein: transcription and translation. Key components are indicated in the scheme on the left. On the right is a schematic representation showing how RNA polymerase slides along the locally melted DNA, reading one strand and producing mRNA. A ribosome then attaches to the mRNA and ‘reads’ it with the help of tRNAs while attaching amino acids (aa), to the growing protein chain. The protein is produced from the N-terminus, often beginning with Methionine (M) and grows towards the C-terminus.](image-url)
During translation (the second step), the mRNA is translated into the specified amino acid sequence, using cellular complexes called ribosomes, which are able to connect the amino acids by peptide bonds. The ribosome needs transfer RNA molecules (tRNAs) that function as decoding entities to choose the correct amino acid dictated by the mRNA sequence. Proteins have 20 building blocks but RNA only has four. Therefore, triplets of nucleotides, also called codons, are used as a code for each amino acid. A stretch of DNA starting with a translation start codon (mostly ATG) followed by a number of codons and ending with a stop codon is called an open reading frame, abbreviated as ORF. The transcribed mRNA contains the open reading frame sequence (and a bit more flanking sequence), which the ribosome decodes with help of tRNAs. These molecules provide the actual physical link between nucleotides and amino acids: a tRNA contains three ribonucleotides that pair with the codon (the anticodon sequence) and are ‘loaded’ with the correct amino acid by specific enzymes to ensure the ribosome can attach this to the growing protein strand.

The maximum of 64 possible triplet combinations is more than sufficient to encode 20 amino acids, plus a signal that tells the ribosome that the end of a protein is reached. Thus, all but two amino acids are encoded by more than one triplet and the stop signal comes in three variations. The link between nucleotide triplets and amino acids is called the genetic code (Table 1.1). It should be pointed out that technically this is more a genetic ‘cipher’ or look-up table than code.

<table>
<thead>
<tr>
<th>Table 1.1. The standard genetic code</th>
</tr>
</thead>
<tbody>
<tr>
<td>UUU</td>
</tr>
<tr>
<td>UUC</td>
</tr>
<tr>
<td>UUA</td>
</tr>
<tr>
<td>UUG</td>
</tr>
<tr>
<td>CUA</td>
</tr>
<tr>
<td>CUC</td>
</tr>
<tr>
<td>CUA</td>
</tr>
<tr>
<td>CUG</td>
</tr>
<tr>
<td>AUU</td>
</tr>
<tr>
<td>AUG</td>
</tr>
<tr>
<td>GUU</td>
</tr>
<tr>
<td>GUC</td>
</tr>
<tr>
<td>GUA</td>
</tr>
<tr>
<td>GUG</td>
</tr>
<tr>
<td>GAU</td>
</tr>
<tr>
<td>GUC</td>
</tr>
<tr>
<td>GUA</td>
</tr>
<tr>
<td>GUG</td>
</tr>
<tr>
<td>GTU</td>
</tr>
<tr>
<td>GGU</td>
</tr>
<tr>
<td>GTA</td>
</tr>
<tr>
<td>GUA</td>
</tr>
<tr>
<td>GUG</td>
</tr>
</tbody>
</table>
For protein sequences two shorthand representations exist, a three-letter and a one-letter code. As the one-letter naming is shorter, this is usually used in bioinformatics. The one and three-letter names are also given in Table 1.1, alphabetically sorted for both, as the one-letter is not always the first letter of their name. A complete set of 61 different tRNAs to cover all possible codons is not required in a living cell: the third base of a codon is less important in codon-anticodon recognition, and can vary to some extent (the ‘wobbling’ base), so that one tRNA can recognize various codons. Thus, the genetic code is redundant to a certain degree.

The ‘standard genetic code’ of Table 1.1 is used by most bacteria, archaea and eukaryotes in the nucleus but it is not conserved in all life forms. For instance, *Mycoplasma* and *Spiroplasma* use UGA for Trp. Ciliates and some yeasts use two of the stop codons for Gln. Eukaryotic mitochondria, which have their own DNA, use other alterations, and produce their own tRNA to use their specific code. Further differences in transcription and translation exist between eukaryotes and bacteria, but the eukaryotic details are mostly outside the scope of this book. Archaea, on the other hand, mostly use the same processes of transcription and translation as bacteria. The division between archaea and *eubacteria* (the latter term is used to differentiate ‘true’ bacteria from archaea) will be mentioned in Chapter 9.

**mRNA sequences are longer than their genes**

The mRNA doesn’t start at the beginning of the coding sequence, but a bit further ‘upstream’, before the actual start of the coding sequence. The so-called *transcription start* is largely dictated by the DNA sequence regulating transcription, which is called the *promoter*. The promoter is not transcribed, but is recognized by sigma factors, specific proteins that direct RNA polymerase to the position where it should start. The mRNA also extends beyond the end of the protein-coding region of a gene, and downstream DNA sequences dictate where RNA polymerase should finish (the transcription stop or *termination*).

Since the mRNA of a gene starts before the actual gene-coding part, not all of the mRNA molecule will be translated. The ribosome scans the front end of mRNA for a specific sequence, the *ribosome binding site*. A little bit downstream from this, translation usually starts with the codon for methionine and this is recognized by the ribosome. Alternative translation *start codons* can be used by most bacteria. In prokaryotes transcription and translation occur more or less simultaneously: the ribosome can start translating even when the mRNA is not completely finished yet. The end of translation is encoded by the *stop codon* given in the genetic code table. Figure 1.5 illustrates the position of mRNA relative to the coding gene, with transcription and translation start and stop signals for prokaryotes.
Figure 1.5. mRNA in prokaryotes. The position of a promoter P and of regulatory sequences is indicated on the DNA. The transcription start is dictated by the promoter whereas the transcription stop depends on structural features of the DNA. Positions of translation start and stop are indicated on the mRNA. Three translation start and stop codons can be used by most bacteria, although CUG and UUG are less frequently used for start than AUG.

DNA sequences: more than protein-coding genes

Protein-coding genes are not the only genes stored in DNA sequences. Some genes code for RNA only, and do not require translation to proteins. Thus, the biological information flow from DNA to RNA to protein sometimes only needs the first step (for some viruses that use RNA as their genetic content the flow of information from RNA to protein suffices). The genes coding for tRNAs are an obvious example of genes not producing proteins. In addition, ribosomes are built up of protein and RNA; in the case of bacteria a ribosome contains 3 different molecules of ribosomal RNA (rRNA). These, too, are encoded by genes that require transcription but not translation. Relatively recent discoveries are additional DNA sequences that encode short RNA molecules (non-coding RNA or ncRNA) that play a role in regulation of transcription.

DNA also codes for information that regulates when and how much of a gene should be transcribed. Precious cellular resources are saved when proteins are only produced when and where they are needed. In a multicellular organism, all cells contain the same DNA but obviously a liver cell requires different proteins compared to a skin cell. In unicellular organisms, different proteins may be expressed under different conditions. This is possible because gene expression is tightly regulated. Regulatory proteins can bind to specific DNA sequences and by doing so they ensure that a particular gene is expressed, or not, under given conditions. The DNA sequences to which regulatory proteins bind are blocks of information that dictate whether genes are ‘on’ (allowing the gene product to be formed) or ‘off’. In most cases, they are found in the vicinity of the gene they regulate. Chapter 10 deals with regulation of gene expression.
The two strands of DNA

DNA in the cell is normally double-stranded (dsDNA). Since the chemical structure of one strand is asymmetrical, each strand has a 'direction' comparable to a two-way street. While driving on the correct side of a street you can read the signs on your side, but not those meant for upcoming traffic. Similarly, information stored on DNA is only meaningful when you read either strand in the right direction. Going backwards on that strand is not meaningful. However, knowing one strand, it is easy to deduce the complementary strand from the base pairs. It is therefore sufficient to represent only one strand of DNA by single letter codes.

For individual genes the coding strand is usually written down. The complementary strand would not produce meaningful information, even when read in its correct direction. It would be like reading this paragraph backwards, and the cell can’t interpret backwards sequences. This is exemplified in Figure 1.6. Nevertheless, both strands of DNA are used in the cell, though rarely at the same location (at least, in bacteria). Thus, in a genome, some genes are on one strand and others are coded for on the complementary strand.

Figure 1.6. A fragment of dsDNA is shown in the top panel where the direction of information is indicated by black arrows. The start of a protein-coding gene (given in capitals for the coding strand) is depicted at the top strand (left). Reading the complementary strand (top right) doesn’t provide meaningful information. At the bottom, the line represents dsDNA with individual genes indicated as blocks above or below, depending on which strand they are encoded. The distance between genes can vary. Their transcripts are symbolized by the open arrows. Messengers can contain multiple genes. Genes can even overlap for a short distance, but the two strands are hardly ever coding for proteins simultaneously for more than a few amino acids.

From DNA to DNA: replication

For the life cycle of an organism to be complete, offspring need to be produced. Reproduction has all to do with producing a copy of DNA. Bacteria reproduce
asexually, producing a complete copy of themselves without the need of two parental cells, though there are a number of possibilities to exchange DNA between cell. Following the production of a second copy of the DNA the cellular contents will be redistributed and two cells are formed from one. The enzyme DNA polymerase produces DNA from DNA, not unlike RNA polymerase produces RNA. The difference of course is that DNA polymerase uses DNA nucleotides. Moreover, the replication process will produce a copy of both DNA strands, though DNA polymerase can only work in one direction. Thus, one strand, the leading strand, is made as a continuous product, while the other (the lagging strand) is produced as short fragments from the opposite direction, after which the enzyme ‘jumps’ to the next bit. These fragments are later fused into one strand.

DNA replication is perhaps not that interesting from a bioinformatical point of view, as it only produces identical copies of DNA. However, the origin of replication (Ori) is a signal on the DNA of a cell that has a large impact on DNA structure and even gene distribution, as will be seen in Chapters 7 and 10. Moreover, mistakes made by DNA polymerase that are not corrected by the cell will become part of the DNA of a daughter cell, and this is one of the processes that underlie evolution. Together with other processes, including DNA exchange between cells, such changes ensure that DNA is not constant over time, but subject to minor (and not so minor) changes. Evolution is the combination of variation and selection. Since bacteria have a short life cycle and reproduce fast, they are excellent organisms to study evolution. As will become apparent, bioinformatic tools are essential in the study of microbial evolution.

**Proteins: structure and function**

The final step in the ‘flow of biological information’ is to finish the produced protein. The entire process of producing of a completely finished protein from a DNA sequence is called protein expression. Once the protein is produced it may start to function immediately, but more likely it requires further modification before becoming functional. In that case the gene encodes a ‘precursor’ protein, which may require truncation, folding or modification like the addition of a co-factor, sugar or lipid parts. Several protein subunits may need to be united into a protein complex before functioning. These events are all post-translational processes that can be delicate and essential for protein function.

Proteins play a critical role in thousands of processes in the cell, from catalyzing reactions in metabolism, to producing cellular structures such as membranes and even directly providing mechanical support. The individual amino acid sequence of proteins can allow for millions of different conformations or shapes, which can have almost as many functions. Nevertheless, particular structural fea-
tures are frequently encountered, of which the alpha-helix and beta-sheet are the most common. Some of these 3-dimensional structures form automatically as the protein chain grows, as if coils ‘spring’ into position. Other structures can only be formed with the help of specific ‘folding’ proteins, such as chaperones, which bend and twist various parts until hydrogen or covalent bonds are formed to keep the protein correctly folded.

Protein folding can have dramatic effects on gene function. For instance, prions are proteins that work as infective agents when incorrectly folded, with the causative agent of BSE (‘mad cow disease’) as an infamous example. Correctly folded, however, they are normal components of the cell. Finally, some proteins are required to pass the cellular membrane in order to be functional, and specific signals are frequently present on the precursor protein to aid their secretion. Such signals may be chopped off during the export process, and the protein may only fold in its functional form after that.

Figure 1.7. Protein structure of the Integration Host Factor with a piece of DNA wrapped around it. The protein is a dimer of two identical subunits. Each subunit contains a loop wrapped as an ‘arm’ around the DNA (indicated by yellow arrows). Structural features of the protein are indicated by helices (for alpha helix structures) and arrow-headed sheets (for beta sheet structures). The DNA bends around the protein like a horseshoe or, as in this picture, an inverse ‘C’. Courtesy of Protein Data Bank (PDB at NCBI).
Bioinformatic tools have been developed to model protein folding. Figure 1.7 displays one such example, of the *E. coli* Integration Host Factor protein (IHF). The protein is represented by ribbons and DNA by sticks and balls. Note that the protein wraps around the backside of the double helix, and strongly bends the DNA. In this figure, 20 bp of DNA (which represents about 2 helical turns) is bent by nearly 180°. IHF is one of a number of small abundant histone-like proteins, which are responsible for compaction of bacterial DNA.

Besides relative straightforward protein folding, protein secretion and their signal peptides can also be predicted with sufficient accuracy. Other post-translational modifications are harder to model. For instance, where a bacterial protein will be glycosylated is hard to predict, although there are prediction models that work well for eukaryotic proteins. Even whether a protein requires post-translational modification in order to function is not directly obvious from its gene; all such modifications are the result of other proteins acting on the precursor, and their signals of recognition are not always known. Predicting protein folding and post-translational modification is a challenge for bioinformatics and reliable tools are still under development.

**Concluding remarks**

The surprising finding of molecular biology that biological information is stored as ‘text’, *i.e.*, strings of characters representing simple monomeric units, has revolutionized our understanding of life. This view of the molecular basis of life can be both fascinating and overwhelming. The study of sequences has in a way become a novel branch of biology, investigating what they code for and how they interact to form larger complexes, eventually resulting in whole organisms. Even though we still do not fully understand how many of the complicated cellular processes work (let alone how they have evolved), we do know enough to realize that cells do indeed obey the laws of chemistry and physics, and that there is no need to resort to supernatural explanations of ‘vital life forces’ or an ‘intelligent designer’ to explain the molecular basis for life.

**References:**

Books on the History of DNA / Genetics


Erwin Schrödinger, “What is Life?”, (Cambridge University Press, 1944)


2 Bioinformatics for Microbiologists: an Introduction

Outline  Bioinformatics is the study of biological information using computational approaches. It depends both on knowledge of the underlying biology and of physical chemical information. It is important for the microbiologist to understand the basic methodologies used in bioinformatics, in order to be able to successfully apply available tools and correctly interpret the results from the lab. Of these tools, sequence alignment methods, such as Blast, are of crucial importance. This chapter will mainly focus on commonly used alignment tools for sequence-based methods of comparison. The chapter is clearly not an introduction to bioinformatics in general, as many aspects of the field are ignored. Instead, we provide here the basics that are tailored for use in later chapters, where bioinformatic tools are applied to the comparison of bacterial genomes.
Identifying similarities: sequence comparison by means of alignments

The basic idea behind a sequence alignment is quite simple. The essence is to align two (or more) sequences and score the positions that are identical. In order to find a possible function of a new gene, for example, one can compare the query sequence against those of known genes in a database, in case a very similar gene with known function has already been described by someone else. Figure 2.1 shows an example of an alignment, using strings of text from abstracts of two published papers on genome sequences (Kawarabayasi et al., 1999, 2000). In order to do an alignment of two different texts, the Query sequence is compared to an identified similar sentence, the Subject.

**Figure 2.1.** Text alignment of two early archaeal genome sequence papers. Note that although they are quite similar in many places, there are regions of ‘divergence’ where the specific genome being sequenced is discussed.

There are several similarities in the wording, indicating a common origin (they are from the same laboratory). Note that the first six words of the abstract are identical in both, they are conserved, and although the next few words are different, it is clear from the context that in both texts these words describe the method, both appearing in the same position of the sentence. The second sentence is almost identical in both abstracts, only the numbers are different. The third sentence contains an insertion: the word “genome” is absent in the query, but present in the subject. A gap was introduced in the sentence (solid line) to ensure the rest of the text matched. Looking at the time of publication, it is apparent that the subject originated a year earlier than the query, thus implying that the word “genome” is miss-
ing in the query due to a deletion. Not knowing which text came first, the identified anomaly could be either an insertion, or a deletion, which is described by the neutral word indel. The next sentences are again identical but for some numbers. Thus, if a literary scholar were to examine these two manuscripts, it would be safe to conclude that they are quite similar, and likely to be related to each other, perhaps by the same author, or that the later author was likely to be aware of the earlier copy. The query could have been derived from the subject, or alternatively, both could have been derived from a common template. Further linguistic analysis might even be able to show that the query text is actually more recent or descended from the subject.

The above example illustrates the general idea behind an alignment. DNA, RNA or protein sequences can be aligned just like text. The alphabet of protein sequences contains 20 letters, not that different to the English alphabet (though without empty spaces) but DNA or RNA only contains 4 letters, which largely influences the chance that a particular position matches with a query sequence. By introducing enough gaps, one could match nearly any two DNA sequences but of course that would be meaningless: the introduction of a gap has some cost. Notably, longer gaps are more costly than shorter gaps, but this is less influential than the introduction of the gap itself. If the four nucleotides of two DNA sequences were randomly distributed, alignment would result in approximately 25% similarity as every position has a 25% chance to be conserved in the other sequence. For a random protein sequence, the chance of an amino acid pairing with an identical amino acid at any given position is only 5%. However, neither DNA nor protein sequences are random. As in the text example above, the presence of particular ‘words’ or patterns increase the likelihood for other patterns in their vicinity. If someone not too familiar with sequencing methods wanted to know the meaning of ‘the whole genome shotgun method’ in the example of Figure 2.1, it could be guessed that ‘assembling the sequences’ was somewhat similar. Thus, alignment can identify particular meaning of a pattern from its content, even if the two sequences are not completely identical. This illustrates the power of alignments.

Aligning DNA versus protein sequences

In Chapter 1 we explained that DNA is usually represented as one strand only, since the sequence of the complementary strand can be easily deduced. Figure 2.2 illustrates what happens if the wrong strand is compared. The two sequences at the

---

1 In this case we know the date of the publications, but often a query finds similarities against database entries without knowledge what came first. An earlier entry in a database is no evidence that a given sequence evolved earlier than the query. Always consider the possibility that a common ancestor resulted in both, instead of one resulting in the other.
top appear not related. However, if the subject is read from the other strand, as in the second alignment, their similarity is obvious. The sequence below gives both strands of the subject to illustrate that really the same sequence is compared, paying attention that DNA is always represented from 5’ to 3’. The same result would be obtained if the complementary strand of the query were used. DNA alignment programs check both strands of any given DNA sequence.

Query.  5’ GGCTAGTAACCCATAGCTATACACCCGGATA 3
      :             :            :
Subject. 5’ TAACCCGGTTTATAAGCTATGGGCTAGGCCC 3

Query.  5’ GGCTAGTAACCCATAGCTATACACCCGGATA 3
      :             :            :
Subject. 5’ GGCTACTAACCCATAGCTATAACCCGGTTA 3

Subject (ds)  5’ TAACCCGGTTTATAAGCTATGGGCTAGGCCC 3
            3’ ATTCGCCAAATATCCGATACCCCATCATCGG 5

Figure 2.2. Alignment of two DNA sequences at the top does not display similarity. When the complementary strand of the subject is used (the second alignment) the similarity is apparent. At the bottom both strands of the subject are given.

A protein sequence can be deduced from a DNA sequence using the genetic code (see Chapter 1). Since the genetic code is redundant to some degree, several DNA sequences can code for the same protein sequence. As a consequence, the similarity of two protein-coding DNA sequences may appear lower than their translated protein sequence, as illustrated in Figure 2.3.

Query   AGGCAAAGTCGCCATGCTGCCATCTGTATG      Query.    RQSRHAICM
      :             :             :
Subject CGTGAGAGCAGGCACTACGCAATAAGCATG      Subject.  RQSRHYAISM
Identity: 15 in 30 positions (50%) 8 in 10 positions (80%)

Query   AlaLysSerProCysCysHisLeuTyr     (Frame 3)
      :             :
Subject ArgGluSerArgHisTyrAlaIleSerMet     (Frame 1)
Identity: 1 in 10 positions (10%)
The reading frame of a DNA sequence may not always be known, and shifting it with one position has dramatic effects on the translated amino acid sequence. The similarity at the protein level can be completely destroyed, as Figure 2.3 illustrates. Thus, alignments of protein-coding sequences performed at DNA and amino acid levels do not always give the same results.

It should be pointed out that a protein sequence can be deduced from a DNA or RNA sequence. However, from one protein sequence, several possible DNA sequences could be predicted, since it cannot be known which codons were used to obtain the protein sequence. Thus, when working with protein sequences, a degree of information is lost that was present in the DNA or RNA sequence. This is in agreement with the “Central Dogma” of molecular biology.

For a DNA sequence, nucleotides are either the same or they are different. For proteins, however, there is a third category, as amino acids can be similar though not identical. These are amino acids that have a similar chemical structure, for instance serine and threonine, which both have hydroxyl (-OH) groups. Leucine and isoleucine also have similar chemical properties, and glutamate and aspartate are both acidic. Replacing Asp for Glu in a protein requiring an acidic amino acid in its active site would likely not completely alter the function, although substitution in the same location with a large aromatic amino acid, such as Tyr, could well destroy the enzyme activity. Thus it would be appropriate to score Glu as ‘similar’ to Asp, but both as ‘different’ to Tyr.

Amino acids can be placed in groups that can be considered similar, and taking this into consideration in alignments produces two scores: an identity score and a similarity score (Figure 2.4). However, determining which amino acids are similar is not always as clear as it might seem, as there are different degrees of similarity, depending on the context. For example, alanine, isoleucine, leucine, and valine are all aliphatic amino acids, and in many cases these can be substituted for each other in a globular protein without much difference in the overall shape. However, their size is different enough to have significant impact if substituted in an active site of an enzyme. In some cases, it only matters that an amino acid is charged, although whether it is a positive or negative charge is not important. But in other cases,
when ionic bonding stabilizes a structure for example, charge is crucial. So depending on which list one consults, amino acids can sometimes appear as similar, sometimes not. Because of this ambiguity in definitions of similarity, it is our opinion that more weight should be given to the percentage identity of an alignment score of two sequences than to the percentage similarity, as well as the length of the alignment, as a fraction of the query sequence.

Pairwise alignments: BLAST and FASTA

Alignments of sequences are commonly performed using the Basic Local Alignment Search Tool (BLAST; Altschul et al., 1990). BLAST can be quite fast, and there are several automated servers available on the web, where one can paste a sequence in a form and quickly search for similarity to genes or sequences stored in a database. GenBank, a public database storing DNA and protein sequences, allows to specifically search all or particular selections of microbial genomes. This and other databases will be discussed in Chapter 4.

BLASTN is the program to search a DNA query against DNA, whereas BLASTP searches a protein sequence against a protein database. BLASTN is set up to automatically search for homologies on either strand present in the database, so that similarities such as in Figure 2.4 will not be missed. BLASTX uses a DNA query and translates this in all 3 reading frames, for both strands, and performs 6 BLASTP searches in addition to BLASTN. BLASTP uses various similarity matrices to determine which amino acids are similar. Complete textbooks are written about the use of BLAST. A few points will only be addressed here that have puzzled some BLAST users.

The output of BLAST gives a considerable amount of information about the alignment. In addition to the sequence alignment, with identity and similarity scores, it also produces a bit score and an expectation value. The bit score is a measure of the statistical significance of the alignment; the higher the score, the more similar the two sequences. The expectation value (E-value) is also a statistical measure: it is the number of times the hit may have occurred by chance. If the number is very low, it means it is very unlikely the finding occurred just by chance, so the lower the E-value, the more significant the score is. An E-value of 10 means that one would expect to have 10 such hits in the searched database by chance, so it is quite likely that the hit is not significant. An E-value of $10^{-58}$ would make it very unlikely the alignment happened by chance, so this is a good score. However, the obtained E-value is dependent on the length of the match, and the size of the database, as well as of course the content of the searched database. The

---

2 http://www.ncbi.nlm.nih.gov/sutils/genom_table.cgi
score is based on the (false) assumption of a completely random database. For example, if the sequences searched against are dominated by *E. coli* and related gamma-Proteobacteria, for example, the chances of getting a hit when searching with an *E. coli* protein are much better than the E-value might predict. In that case, relatively high E-values (normally a sign for findings that are not significant) might still be meaningful.

The strength of BLAST is that it is able to identify a local stretch of similarity in a longer sequence. This is excellent for identifying a protein domain with a particular function, such as an ATP-binding region for enzymes that require energy. However, it is important to keep an eye on where an aligned segment is located in the protein. If, for instance, an enzyme typically contains a particular domain in its amino-terminal region (away from the ATP binding region, for the sake of the argument), finding similarity to a small region towards the C-terminal end of a long protein may be a coincidence, and not biologically meaningful. The output of BLAST shows the alignments with their scores; a glance at the position numbering of both the query and the hit can be useful to determine how relevant a finding is.

Figure 2.5 shows parts of a BLASTX search, using as the query a sequence that was generated from a cloning procedure. The graphical representation shows that the sequence contains two parts with similarity to different hits. This can be a hint of a chimeric sequence (two sequences that were artificially or naturally combined). The list provides brief information on the first 12 alignments, with their score and E-value. From the descriptive line (which, unfortunately, is incomplete and cannot be shown in full at the NCBI site from which this example is taken) it is obvious that the query sequence has highest similarity to *Campylobacter jejuni* sequences (a Gram-negative pathogen causing enteric diseases). The first 5 hits suggest that sequence similarity is detected to an oxidoreductase subunit (an enzyme). The annotation of the 6th hit doesn’t reveal the function. The next 6 hits suggest similarity to flagellin (a structural component of flagella). Flagellin and oxidoreductase are very different proteins, and the fact that similarity to either class of proteins is clearly divided between the halves of the query again suggest a chimeric sequence. The graphical representation of the 6th hit suggests that there may be at least one entry in the searched database of such a chimera (though the junction is not conserved). Closer inspection shows that the two are different sequence entries in the database (generated from different strains) and do not belong together.

Although this search was performed in the ‘non-redundant’ database, many of the hits were generated with identical results. This is illustrated with the database entries that produced the alignment as shown in the bottom of Figure 2.5. These are mostly (but not all) produced from different strains and are present in the non-redundant database as they are regarded as independent entries. Thus, there is still
quite a level of redundancy in the ‘non-redundant’ database. From this analysis it was concluded that the generated sequence was a chimera; subsequent PCR analysis confirmed that two fragments had been introduced in one clone that did not belong together on the genome. The chimera was the result of a cloning artifact. Thus a fairly simple BLAST search revealed an error in laboratory results, suggested a possible explanation, and pointed out which experiments could confirm (or dismiss) this explanation. One limitation of BLAST is that it can only perform comparisons of 2 sequences at a time. The results are reported as query/subject scores, for each alignment identified in the search. When the query sequence is similar to 2 domains in one sequence, these will be presented as 2 separate hits. If so, it is important to visually inspect the alignment location to verify this possibility.

**Figure 2.5. Results of a BLASTX search with a DNA sequence generated from a cloned DNA fragment.** A shows the graphical representation (produced by BLAST at NCBI) for the first 12 hits. B shows the one-line header for these 12 hits, with their E-value. C shows the alignment of the 3rd hit but in fact, this alignment was obtained with 8 database entries some of which were from the same strain of the organism. By clicking on the link one can inspect the database entry, which may reveal such redundancy (not shown here).
BLAST is not the only alignment tool (although it is probably the most commonly used). Another well established program is FASTA (FAST All), which uses an alternative algorithm to detect sequence similarities (Lipman and Pearson, 1985). FASTA is more sensitive than BLAST, and when it was developed more than 20 years ago it was quite fast. Today, however, the size of the databases has grown so much that this method can take quite a bit of time, and often BLAST searches are considerably quicker. FASTA is now less frequently used than BLAST.

The term ‘FASTA’ lives on as a format for sequences that is accepted by many sequence analysis programs. Instead of just entering your sequence, it is often advantageous to give it an identifier (a name, number or description). But this name should not be ‘read’ by the program as part of the sequence itself. The FASTA format reserves the first line for this, and has to start with a greater-than-sign (‘>’). The line finishes with a hard return, so that everything from the 2nd line onwards is read as sequence (this can be DNA, RNA, or protein). An example is shown below, for the H-NS sequence (a histone-like protein) from Salmonella. Note that the first line ends with a soft return only, to be continued on the next line. The end of line is indicated with a hard return, indicated by the ‘¶’ symbol.

>gi|7800486|gb|AAF70002.1|AF250878_163 'DNA binding protein, H-NS-like' [Salmonella typhi]
MSEALKSLNNITLRAQRGRELPLEILEELKSLSVVVEERRQEESSKEAEELKARKEIESLRQLMLE
DGIDPEELLSFSKSGAPKVRKAKAEMEHKADKIFK

**Multiple alignments: CLUSTALW**

Multiple alignments in which several sequences are compared to each other are very informative, as they can identify regions that are less variable or more variable within a set of genes. For multiple alignments CLUSTALW (Thompson et al., 1994) is a frequently used program3. This program first calculates the highest similarity for each possible pair of combinations, and then estimates the optimal multiple alignments for all (it is based on the same algorithm for similarity as FASTA). CLUSTALW is much slower than BLAST, and is more suitable for the input of short sequences of which a degree of similarity was already established. CLUSTALW is not suitable to search databases. A better approach is to first search for hits in GenBank with a query gene, and then to take a selection of these hits and combine them in a multiple alignment together with the query sequence. This way one can identify regions of higher or lower degrees of conservation, for instance to identify a constant region that can be used for PCR primer design, or a variable region that may be a target for a typing procedure.

3 [http://www.ebi.ac.uk/Tools/clustalw](http://www.ebi.ac.uk/Tools/clustalw) (for example)
Chapter 1 ended with a figure of the Integration Host Factor (IHF), wrapped around a short piece of DNA. Figure 2.6 shows an example of a multiple alignment of DNA sequences. These represent different IHF binding sites: the exact location of where the protein binds had been experimentally determined. By aligning the sequences, it is obvious that sequences in the middle are quite strongly conserved, whilst the flanking regions are less conserved. The alignment is shown for 20 sequences and a consensus sequence is added to the bottom. A new letter is introduced here to represent a certain ambiguity: W for A or T. There are single-letter codes for all degrees of uncertainty, (Table 2.1) although most bioinformatic tools accept GATC only, plus in some cases N for unknown.

**Table 2.1. The DNA alphabet**

<table>
<thead>
<tr>
<th>No ambiguity</th>
<th>1 ambiguity</th>
<th>2 ambiguities</th>
<th>3 ambiguities</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>S = G or C</td>
<td>H = A or C or T</td>
<td>N = A or G or C or T</td>
</tr>
<tr>
<td>A</td>
<td>W = A or T</td>
<td>B = G or T or C</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>R = G or A</td>
<td>V = G or C or A</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Y = T or C</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M = A or C</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>K = G or T</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: S stands for 'Strong' as G and C share three hydrogen bonds; A and T only share two H-bonds, thus W for 'Weak'. R stands for puRine and Y for pYrimidine.
There is another way of visualizing the conserved binding site region, based on the occurrence frequency at each position in the alignment: a so-called sequence logo plot. An example is shown in Figure 2.7, for the same IHF binding site alignment but this time based on experimental data obtained in the laboratory. In this figure the size of the letter is a measure for its frequency. The logarithmic scale, in bits, comes from information theory and represents the amount of information conveyed. It is clear that the centrally positioned TG pair is strongly conserved. In fact, the dinucleotide TG is responsible for the ‘bend’ in the double helix that allows the DNA to bend around the IHF protein (as shown in the structure in the previous chapter, Figure 1.7). Sequence logo plots are further applied in Chapter 10.

![Figure 2.7. Sequence logo for the IHF consensus binding sites. The highest value on the bits scale is 2 bits, representing a 100% conserved nucleotide.](image)

From alignments to phylogenetic relationships

Multiple alignments can also be done for proteins, and Figure 2.8 shows an example of several IHF proteins aligned. For simplicity, only part of the sequence is shown in the alignment. Again, colors are introduced for easy optical inspection. The strongly conserved block represented here happens to be the DNA binding motif, the protein part that recognizes the DNA sequences shown in Figure 2.7. But from other segments in the alignment, it can also be seen that some sequences (derived from various organisms) are more alike each other, and others cluster to different groups. The proteins in Figure 2.8 were originally arranged in alphabeti-
cal order, by the first name of the bacterial species from which the protein originated. In the figure, though, the most similar sequences are grouped together, to illustrate more clearly the clusters one can identify (this is what CLUSTAL, a program designed to perform multiple alignments, normally does). This illustrates that the alignment conservation hints at how closely the proteins are related to each other.

Figure 2.8. Multiple sequence alignment for 19 different IHF alpha proteins produced using CLUSTAL. For sake of clarity, only the first part of the protein sequence is shown. The original FASTA file contained the proteins in alphabetical order by the species from which they were derived (starting with A_baumannii for Acinetobacter baumannii). The aligned version puts the proteins most similar to each other (and to the consensus) at the top, with the least similar (in this case from Silicibacter) at the bottom. The plot underneath shows the relative conservation, quality and consensus for each position. Amino acids were color-coded according to similarity groups.

Multiple alignment analysis is used to identify gene similarity and to define how diverse two genes might be to still consider them similar. This is important, for instance, in designing probes used in microarray analysis; genes we consider ‘similar’ should be recognized by one probe, and their hybridization signal should be treated as equal. Probe design of microarrays is quite complex and this branch of bioinformatics will not be covered directly in this book. Sketching out the difficulties briefly, we recognize that designing probes specific to conserved regions only will result in loss of information, as the true variety in genetic content is not assessed. Probes designed for variable regions, however, may also result in loss of information if they are too specific (as variants may be present but no longer hybridize), whereas less specific sequences may result in false positive findings.

Another commonly used method to visualize similarity of the sequence of proteins is to use a tree plot, as shown in Figure 2.9. Notice in this figure that now there are two main clusters: a fairly tight cluster of gamma-Proteobacteria (E. coli and relatives), and a more loosely set of ‘other organisms’, which are taxonomically more diverse. There are several methods for producing a tree plot, and many web sites offer a service where one can paste in a FASTA file containing multiple sequences, do the alignment using CLUSTALW, and then have the program draw a tree.
Phylogenetic trees have been around for nearly 150 years; an evolutionary tree is one of the few illustrations in Darwin’s *The Origin of Species*. However, the more modern ‘molecular based’ trees have only been around since the 1960s, and it has been estimated that there are more than 3,000 papers about inferring phylogenies based on sequences (Felsenstein, 2004). There are several different types of tree plots. They can be *rooted*, with a single ancestral organism implied, as the one shown in the top part of Figure 2.9, or *unrooted*, with no clear origins, as shown at the bottom. Most biologists (including Darwin) tend to think of trees as rooted.

To produce a rooted tree, one can add a known sequence as an outlier, in order to anchor or *root* the tree. Effectively this means one must know in advance something about the phylogeny, as one must know that a particular sequence truly rep-
resents an outlier. Another method for rooting the tree is to use the molecular clock assumption, which also has problems in that it is assuming perhaps a more homogeneous rate of mutation than exists. Some biological variation, though, can best be captured in an unrooted tree to describe the underlying relationships (for example, clonal expansion of a bacterial population with increasing diversity). There are ways of calculating the reliability of branch positions in the tree, but these are beyond the scope of this chapter.

**Genome annotation: the challenge to get it right**

The general term *genome annotation* is used for the description of all genes identified in a genome, their location, possible function and sometimes closest similarity to other known genes. Genome annotation can be rather minimal (a gene name, start and end nucleotide numbers, and a short description) or very verbose, explaining on what evidence a particular predicted gene function was based. The richness of a sequenced genome lies largely in the accuracy of its annotation, and it is a challenge to get this right.

How much M13 cloning vector DNA is present in *Helicobacter pylori*? And how much IS10 sequences (an insertion sequence typical of prokaryotes) would be found in plants? Not much, one would think, but a search in the database can reveal some unexpected findings. Both mistakes stem from sequencing the ‘wrong’ DNA: in the first example vector DNA instead of the cloned insert was sequenced, and in the second example bacterial DNA rather than plant DNA was most likely sequenced. There are several examples resulting from contamination in the laboratory, so that the wrong DNA was sequenced and an incorrect annotation was entered in the database. It is estimated that such errors are present in 0.27% of all database entries. Presently, contamination of DNA during genome sequencing is a major concern (as the shotgun cloning procedure, introduced in the next chapter, is sensitive to contamination) and even sequence databases can get mixed up by wrong computational activity. When these mistakes remain unnoticed, wrong annotations in the public databases are the result.

In addition, the problem of a ‘similarity chain’ can occur. When protein A is similar to B, and B is similar to C, can A then be considered similar to C? Sometimes, but not always, as the example with chimeric sequences presented in Figure 2.5 reveals. What if the cloned DNA of that example had been a naturally occurring chimera? It would show good similarity to both an oxidoreductase and flagellin. If an oxidoreductase sequence had been in the database first, followed by this unfortunate chimera, the latter would have been annotated as ‘similarity to oxidoreductase’. A next query of a newly discovered, unknown protein (in this case
flagellin) produces good similarity to our chimera and would be given the annotation of ‘similarity to oxidoreductase’ where this would be absolutely incorrect.

The example is theoretical but not far-stretched. The database is littered with such ‘related to something related to something else’ trains, producing inaccurate or absolutely incorrect annotations. Complete Ph.D. projects have been ruined by such false information. One way to steer away from this cliff is to do a multiple alignment with a few of the BLAST hits you’ve obtained with your query gene (choose hits with various scores, E-values and annotations) and to inspect where the similarity is located.

Genome annotation after the genome sequence has been obtained actually consists of three challenges: correct assembly of the fragmented pieces of sequence obtained from the sequencing process (explained in the next chapter), identifying where genes are located (gene finding) and finding clues what these genes could code for (gene annotation). Although now it is possible to sequence a bacterial genome in a few hours, it is still not easy to correctly assemble it, and even once it is put into one contiguous piece, finding all the protein encoding genes and the rRNA, tRNA and other non-coding RNA genes is a real challenge. Finding all ORFs is not a problem, but an ORF is not synonymous to a gene: every protein-coding gene is an open reading frame, but not every ORF is a protein-coding gene. Unfortunately, the distinction is not always made. For example, if one were to just extract the proteins from the GenBank file for Aeropyrum pernix (a hyperthermophile found in hot springs) there are about twice as many proteins in this genome as in related organisms with similar sized genomes. The reason is that all possible ORFs have been included. This is an example of over-annotation. Moreover, genes encoding RNA are not ORFs, so even if all ORFs were (incorrectly) annotated, genes could still be missed. Further, not only are there far too many ‘genes’ in this organism, many of the true genes found in proteomics experiments are not annotated (Yamazaki et al., 2006), meaning that this genome also suffers from under-annotation.

It is the challenging task of an annotation team to filter out, with the help of computer programs, the ‘real’ genes (at least, the best candidates) from the background noise. Alignments are suitable tools to assist in this task: identifying similarities to existing, recognized genes by BLAST and other alignment tools help to screen which ORF is a good gene candidate. However, if we would only annotate those ORFs as genes that have been discovered in other organisms already, we wouldn’t be making much progress.

Novel genes are bound to be present in a novel genome sequence, so how to recognize which ORFs encode for the ‘unknown’ genes, and which are not genes at all? This task is best performed by programs that ‘learn’ on the spot: they need to be primed for what, in a given genome sequence and based on prior knowledge, we can be certain is a gene, and then make best guess predictions about unknown
ORFs. Such machine learning approaches are frequently based on Artificial Neural Networks or Hidden Markov Networks, the details of which are beyond the scope of this book.

As explained above, gene finding is not the same as genome annotation: it is only the first step. The result of a gene finding program is merely a list of locations in the genome where protein-coding genes are likely to be found. These results require a coordinated systematic computerized approach to provide the possible function and potential gene names for all identified putative genes. Nowadays, the time required for assembly (dealt with in Chapter 3) and genome annotation by far exceeds that required to do the actual sequencing, although it is possible to automate these procedures. As we will see in Chapter 11, it is now possible to deliver a bacterial genome sequence in less than 2 days – starting with a purified DNA preparation and producing a draft of an annotated genome sequence file. Genome assembly becomes an impossible task when sequences are obtained from DNA isolated from environmental samples containing lots of bacterial species. In Chapter 13 we will see that there are limits of what is currently technically possible.

**Information beyond the single genome**

Once a genome sequence is available and annotated, the hard work begins for the microbiologist, trying to make sense of this wealth of information. Undoubtedly, to have a genome sequence available for quick reference makes life a lot easier for a lot of researchers. One can quickly check if a gene is present that could be responsible for a phenotype one encounters, or, when a gene was identified and partially sequenced for any reason, a genome sequence quickly tells you what the rest of the gene might look like, what neighbors might be present, and so on. In that way, a genome sequence works as a catalyst of ongoing research. Genome sequences also have generated, apart from *in silico* research (the analogy for computer-based research, to complement *in vivo* and *in vitro* analysis), a whole new area of wet-lab research that was unthinkable of in the past.

The possibilities don’t end here. A single genome sequence bears a wealth of information that is ready to be explored, but start comparing different genomes and a complete extra level of biological information is added. The final chapters of this book are dedicated to such studies, although there are examples of multiple-genome comparisons throughout the book.

In Chapter 12 analyses will be introduced that have only become possible now that multiple genome sequences are available per species (or per genus). Only now do we recognize the true degree of genetic diversity amongst bacteria, even mem-
bers of the same species. We can now define the so-called *pan-genome*, considering all genes that can be possibly present in a given isolate, which can easily be more than twice as many as can be found in any individual genome of that species.

Another approach to look at genomic information beyond the genome scale is to investigate all (bacterial) DNA that is present in a particular ecological niche. This so-called *metagenomic* approach is still relatively novel, and thus a lot of bioinformatical work is still in the experimental phase, presented in Chapter 13. Finally, in the last chapter we will see that evolution left its marks on bacterial genomes, which can be read as a book full of patches and overprints.

**Concluding remarks**

A proper bioinformatic analysis of available information can save months of unnecessary experimental laboratory work. The two fields are of course complementary, and findings from one approach can strengthen or dismiss hypotheses derived from the other. It is not that bioinformatics is meant to replace work in the laboratory, but rather, bioinformatics has become an essential tool to greatly enhance the possibility of the experimentalist. Although it is exciting and rewarding to ‘play’ with sequences at a computer, bioinformatic analysis has most strength when applied in a hypothesis-driven manner. Otherwise, it will produce lots of findings with a high ‘so what’ character. Microbiological technology-driven lab work can, when the experiments work, produce lots of results that don’t really produce insights. But experiments don’t always work. Computational analyses do always work, and produce lots of output. Quite likely, though, the output can’t be interpreted in a biologically meaningful way. This is why insights both from an informational and a biological viewpoint are needed to produce data that help microbiology progress.

**References:**


**Books on Bioinformatics**


Microbial Genome Sequences:  
A New Era in Microbiology

Outline Microbiological research has changed significantly in the last decade, now that complete bacteriological genomes can be generated. In this chapter we take a brief look at the historical developments that allowed this achievement to take place. The first genome to be sequenced was that of a virus. Since then, technological development has improved significantly, reducing the time, efforts and costs by several orders of magnitude. The first bacterial genomes took years to finish and cost more than a million dollars; currently a bacterial genome can be sequenced in a few hours, costing a few thousand dollars. Now that complete bacterial genomes are available for analysis, visualization is extremely important to interpret the vast amounts of data generated, and hence visualization plays a major role in bioinformatics. Several examples will be introduced here, to be further explained in subsequent chapters. Bacterial genomes come in various shapes (linear vs. circular) and number of molecules (up to three chromosomes can be present). In addition, plasmids may be present. Not all bacteria are like *E. coli*, and this diversity makes comparative genomics so interesting.
The first completely sequenced microbial genome

In the previous two chapters we have introduced the idea of the flow of biological information through sequences, and we introduced some of the methods used to analyze those sequences. In this chapter we introduce the first whole genomes, that is all the DNA sequences stored in a cell from a given organism. A genome can tell a lot about the organism it was derived from, if we use the proper analysis tools. A few examples will be presented here, to be further explored in Chapters 6 to 11. Comparing different genomes will be the main topic of Chapters 12 to 14. We will start with a major breakthrough in microbiology, dating back approximately 3 decades.

The first complete genome to be sequenced was not that of a bacterium, but rather of a bacterial virus. It was bacteriophage φX174 (pronounced ‘fie ex one seven four’, a phage infecting *E. coli*) whose genome was completely sequenced in 1978 (Sanger et al., 1978). This major achievement had been performed by subcloning and sequencing mapped fragments, after which the genome was pieced together. The sequence was produced using a method developed by Fred Sanger, based on the incorporation, into a synthetic DNA strand produced with DNA polymerase, of nucleotides with a missing 3′OH group (dideoxynucleotides). This meant that in a subset of products the next nucleotide could not be attached, resulting in termination of the product. Products were then separated by gel electrophoresis and visualization of bands allowed the sequence to be ‘read’. Through the use of small amounts of each dideoxynucleotide in 4 separate reactions for each of the four bases and radioactive $^{32}$P isotopes, it was possible to run 4 different lanes on the gel and get various lengths of fragments. By tedious sequencing of subclones, the 5386 bp DNA sequence could eventually be pieced together. It took more than a year to sequence the φX174 genome, and to do honor to the work we show the entire sequence (the only genome sequence that will be presented in this book) in Figure 3.1. Sanger shared the Nobel Prize in chemistry for this work with Paul Berg and Walter Gilbert (his second, the first he won in 1958 for his work on protein structure).

What does this sequence tell about the organism? Just by looking at the DNA sequence as it is presented in Figure 3.1, it doesn’t reveal very much. If the four nucleotides were represented in four colors, one could perhaps identify regions with particular patterns, not knowing though what those patterns mean. It would help to have a look at the GenBank file (accession number J02482, more about this in Chapter 4), and for example to inspect the regions that encode proteins. These regions are not apparent from the DNA file. Unfortunately, a DNA sequence by itself doesn’t easily reveal its message to the human eye.

Figure 3.1 (opposite). The entire nucleotide sequence of the φX174 genome in FASTA format.
The importance of visualization

Figure 3.2 shows two alternative views of the \( \Phi \)X174 genome. Now it is obvious is that the genome is circular, which is not visible from the sequence, since it was artificially reproduced as a linear sequence. The figure on the left displays some of the restriction enzyme cutting sites that were used to produce the subclones required for sequencing, which were then put in the right order again with the help of a map. The figure on the right shows superimposed the 11 protein encoding genes (named ‘P1’ through ‘P11’, in green), and the outer circles show the predicted mRNA transcripts (in red).

Note that all genes in this genome are going in the same direction. There is a biological reason for this in this particular virus, although in many genomes the orientation of genes can vary between the two strands. The replication origin is labeled in orange in the figure, at roughly the ten o’clock position of the chromosome. This plot reveals information about the gene content, but there is little information here on the DNA sequence itself, which the sequence in Figure 3.1 contains. In case we want to do a thorough analysis on base composition of this DNA, we need to visualize both in a single, circular plot.

![Figure 3.2](image-url)
**Base Atlases to visualize base composition features**

Figure 3.3 is an ‘absolute’ Base Atlas, or a graphical representation of the entire φX174 DNA sequence plotted on a single figure (for the positive strand, the one represented in Figure 3.11). Since we are interested in base composition analysis, the densities of the four bases are plotted by color intensity (the four outer circles of the top panel). It is obvious that this DNA is quite T-rich, as there is far more red (T’s) than green (A’s), turquoise (G’s) or violet (C’s). It would be a challenge to see this at one glance from Figure 3.1.

![Base Atlas Diagram](image)

**Figure 3.3.** Absolute DNA Base Atlas of the nucleotide sequence of the φX174 genome. The legend to the right explains what is represented from the outer to the inner circle. Shown are the fraction of each nucleotide along the genome (first four circles counting inwards), the coding sequences on the positive (clockwise) strand, the AT and GC skew and the percent AT. In an ‘absolute’ Atlas all lanes are plotted with a fixed range.

Continuing to read this plot from outside to inside, the coding sequences are plotted next, and since they are all on one strand only one color is needed here (in case there are coding sequences on the strand complementary to the strand that is published, we color them red). The next circle is called AT skew, and is a measure of the bias of A’s towards one strand (and T’s towards the other). As will be discussed in Chapter 7, for some bacteria, the A’s are biased towards the replication

---

1 Phage φX174 is a virus that packs its DNA as single strand DNA (ssDNA) in viroid particles so it only contains this positive strand in viroid form.
leading strand, but in other bacterial chromosomes, including *E. coli*, which this phage normally infects, the A’s are biased towards the replication lagging strand. The strong red color in this lane means that T’s are biased towards the strand represented by the sequence, implying that this is the leading replication strand. The next circle shows the GC skew, and since the scale is the same as that of the AT skew (+/- 0.20), the absence of dark colors indicates the bias of G’s towards one strand or the other is not as strong as for the A’s in the previous circle. AT and GC skew are further explained in Chapter 7. Finally, the deviation of AT content from the chromosomal average percentage AT is plotted, ranging from 40% to 60% AT, with 50% AT in the middle; thus bright red regions contain lots of A’s or T’s, and the blue regions are GC-rich. There are four dark red regions in the innermost circle that are much more AT rich than the rest of the chromosome.

**Figure 3.4. Relative Base Atlas of the φX174 genome.** In this atlas the colors represent the regions where the base density varies with more than 3 standard deviations from the genomic average. To the right of each scale is indicated whether fixed average or 3 standard deviations are plotted. The numbers below the scales indicate how color intensity was chosen. This relative Base Atlas (and not the absolute version of Figure 3.3) is the default Base Atlas used in the remainder of the book.

The plot of Figure 3.4 shows the same data as in Figure 3.3, but now as a ‘relative’ Base Atlas: the data are normalized to the genomic average for the values in each lane; only values greater than three standard deviations above the average are colored. At first sight, this is a rather bleached version of the previous figure, but it does reveal different information. For instance, there is a region where A’s are highly overrepresented compared to the global A content (around 3.5 k) and a
relatively small stretch where G’s are overrepresented (around 1k). These are not as easy to identify in the previous, absolute Base Atlas. Atlas figures can display either absolute ranges, or show regions that deviate by more than 3 standard deviations from the chromosomal average, or a combination of these. The way to tell the scale is to look at the legend, which is always oriented with the outermost circle on the top, going towards the innermost circle at the bottom. At the right of each scale in the legend, a ‘fix’ indicates a fixed range, while ‘dev’ means that the average is in the middle value (usually light grey) and the extreme ends represent plus or minus 3 standard deviations from the average.

**Genome Atlases to visualize chromosomes**

The analysis of DNA base composition is interesting in itself, but a Base Atlas displays only a fraction of the type of information a genomic atlas can provide. The next step is to combine this with presence of genes, and to also indicate regions containing repeats in DNA sequences. Structural features of the DNA can also be plotted. That way, we start to produce what we call a **Genome Atlas**, providing a quick overview of some of the most important and informative features in a microbial chromosome, plasmid or phage. Figure 3.5 represents a Genome Atlas of the φX174 genome.

![Genome Atlas](image)

**Figure 3.5.** Genome Atlas of the nucleotide sequence of the φX174 genome.
Of the circles of the Base Atlas of Figure 3.4 we have chosen to represent only AT skew (as a fixed average) and percent AT (as deviation). Three outer circles have been added to the atlas, representing DNA structural properties: intrinsic DNA curvature in the outermost, followed by stacking energy and position preference. These three parameters provide important insights in the physical and mechanical properties of the DNA molecule, which will affect how the molecule is folded. This again can potentially influence gene expression, the likelihood of genome rearrangements and even the occurrence of evolutionary hotspots. The more technical details of these parameters are explained in Chapter 7.

The two other circles added to the Genome Atlas of the φX174 genome represent the location of two kinds of repeats: perfect palindromes and inverted repeats. In an inverted repeat, the same piece of DNA (read from 5’ to 3’) is repeated on the opposite strand. In this case, the repeated sequences are found relatively close to each other (within 100 bp). These are called global repeats. Palindromic repeats (or palindromes) are also inverted repeats, but now the inverted repeat is in fact the complement of the original repeat unit. Palindromes are a special kind of local inverted repeats. Chapter 8 will explain repeat sequences in more detail, and also introduce other types of repeats. To calculate the local inverted repeats represented in the Genome Atlas in Figure 3.5, we performed a sliding window analysis. This means a short stretch of DNA (a window, in this case of 15 bp long) was analyzed, looking for a match on the other strand of DNA, within a larger window (here: 100 bp). This would identify a palindrome or repeat of at least 15 nucleotides long with a spacer of at maximum 70 nucleotides (that would still fit the 100 bp window). The odds of finding a long perfect palindrome in a piece of DNA are quite low, but as the length of the searched palindrome goes down, the probability goes up. The sliding window analysis is explained in more detail in Chapter 8.

Atlas representations of bacterial genomes were developed several years ago (Jensen et al., 1999, Pedersen et al., 2000) and we have set a standard for representation of various levels of information on a Genome Atlas. There are several ways of visualizing genomic information in an atlas, and it depends on the question that is being addressed which is best. Currently, our web pages present seven different types of atlases for each sequence. Subsequent chapters of this book will introduce features in detail that can be plotted on various types of atlases. An example of a Genome Atlas of a plasmid is shown in Figure 3.6, where the plasmid of Escherichia coli O157:H7 is shown twice. This plasmid gives the serotype O157 of E. coli its particular virulent properties. It was sequenced twice by two research groups from two different samples of O157.

2 As we will see in Chapter 9, a standard Genome Atlas displays different kind of repeats but since those are absent from this small genome, we deviated from our standard this time.

3 http://www.cbs.dtu.dk/services/GenomeAtlas
Figure 3.6. Genome Atlas of pO157, the virulence plasmid of *Escherichia coli* O157:H7. The top panel represents the sequence generated by Burland et al., 1998, and at the bottom is the sequence generated by Makino et al., 1999, of the plasmid isolated from strain Sakai. The position of the origin, various genes and the *toxB* region have been indicated.
At first sight, the two atlases look different, but note that the one is a tilted version of the other. Remember this plasmid is circular, whereas the sequence in a database is entered in as a linear sequence. The circle has to be ‘cut open’ somewhere, and the convention is to do this at the origin of replication. For plasmids it is not always easy to identify the origin of replication. In fact, in Chapter 7 we will see that even for bacterial chromosomes this is not always correctly done. As a result of ‘incorrect’ (though always artificial) opening of the circle, the numbering of the one sequence file is different from that of the other. Atlases will always put number 1 of the published sequence at the top. Ignoring the difference in orientation, though, the two Genome Atlases look very similar. That similarity would not at all be obvious if the two plasmids were viewed just as a sequence of bases.

A race against the clock: the speed of sequencing

Bacterial genomes are on average about a thousand times larger (typically in the range of a few million bp) than most viruses or plasmids. This means that, using the Sanger method at the same rate as it was used to sequence the φX174 genome, it would take a thousand years to sequence a bacterial genome, though the method has evolved to become more efficient with time. Large collaborating teams made use of improved sequencing methodology to sequence within a year the *Bacillus subtilis* genome (Kunst et al., 1997) and the first *Escherichia coli* genome (Blattner *et al*., 1997), which were amongst the earliest bacterial genomes to be published.

Before we get to the first sequenced bacterial genome, it is worth taking a short digression on the technology that made this possible. For sequencing of all φX174 DNA, it was digested with restriction enzymes and individual fragments were cloned in a vector. These inserts were cloned (i.e., inserted in a vector replicating in *E. coli*), sequenced, and the use of overlapping fragments allowed the genome to be assembled. The term shotgun cloning was coined when a library of DNA fragments of varying (but defined) length and identity were cloned in individual vectors. Either case, after transformation colonies were selected; putative positive clones would produce a white color on the plates, allowing for relatively fast screening. Very large Petri plates were needed (at first cafeteria trays were used) to clone the fetal gamma-globin gene from the entire human genome (Blattner *et al*., 1978). Within a few years, the shotgun cloning method had been applied to sequencing, called shotgun DNA sequencing (Messing *et al*., 1981). Another breakthrough in genome sequencing was to produce a library of cloned fragments from a complete chromosome and sequence these ‘at random’. The challenge was to assemble all these short sequences into a full chromosome, for which novel assembly programs had to be developed. This technology set the stage for sequencing larger DNA molecules, eventually including bacterial genomes.
A few years after the \( \Phi X174 \) genome was published, scientists at the Los Alamos National Laboratory in New Mexico began discussions of sequencing the human genome. At the time the plan was highly ambitious: if it would take a thousand years to sequence a bacterial genome, it would take a million years to sequence the 3 billion bp human genome. Clearly the speed of sequencing was the limiting step. The U.S. Department of Energy decided to invest in technology to facilitate the speed of sequencing, with the goal of eventually being able to sequence the human genome in a practical time span. The Human Genome Sequencing project started in 1985, with a goal of investing $200,000,000 per year in technology to improve the speed of sequencing.

Meanwhile, fluorescent labels were being applied to sequencing reactions to replace radioactivity, and together with automated reading devices, high throughput gel electrophoresis and robotic machines performing the sequence reactions the process became more and more efficient. Most of these methods were a direct spin-off from the Human Genome Sequencing project. Development is ongoing to improve efficiency even further, so that soon we may be able to sequence a human genome for less than $1000 U.S., in less than a day; this means that a bacterial genome sequence would cost only one dollar! This is worth keeping in mind, when we think about the first bacterial genomes that were sequenced, where we are today and where we are headed in the near future.

### The first completely sequenced bacterial genome

The task of sequencing the first bacterial genomes, over 10 years ago, was quite a bit different from today’s practice. The technology was still being pioneered and was expensive and time-consuming: a genome cost more than a million dollars, and took years to. It was highly appropriate that the first genomes were published in high profile scientific journals. The projects required heavy investment from a company, which patented the resulting genomes hoping to get some sort of return for their money. The first bacterial genome sequence to be published, in 1995, was that of *Haemophilus influenzae*, an opportunistic human pathogen (Fleishmann et al., 1995, U.S. patent number 6,528,289). This species has a relatively small genome of 1.8 megabases (Mb) and sequencing the shotgun clones took approximately a year to complete. The publication of the first bacterial genome sequence was a breakthrough. The article in *Science* proudly announced “The *H. influenzae* Rd sequence” (Rd denotes the strain from which the DNA was isolated), but more accurately, it should have stated “*An H. influenzae* sequence”.

---

4 We consider this investment in sequencing speed increase as quite successful; the first human genome cost $3 billion U.S., and took 15 years to finish; the second human genome (Craig Venter’s genome, sequenced by Celera), cost a ‘mere’ $100 million, and took 9 months to finish; James Watson’s genome took only 2 months, and cost $900,000.
Figure 3.7. Genome Atlases of the nucleotide sequence of two different Haemophilus influenzae genomes. Some striking differences are indicated by arrows and circles for the bottom genome.
At the time few people appreciated the enormous diversity of bacterial genomes even within a species, as will be discussed in later chapters in this book. In retrospect, it appears somewhat ironic that the *H. influenzae* strain Rd was chosen to be sequenced to exemplify the genome of a ‘pathogen’, as serotype d strains are not pathogenic (the ‘R’ stands for rough, a particular colony morphology). Nevertheless, many databases present this genome sequence as an example of a pathogen. Ten years later, a closely related, truly pathogenic strain of *H. influenzae* has been sequenced to set things right (Harrison et al., 2005). Figure 3.7 shows a Genome Atlas for these two *H. influenzae* genomes.

There are some striking features about these atlases, in comparison to the smaller sequences that we’ve seen so far. Note that there are several strong, brightly colored regions present, corresponding to repeats and significant structural properties. Some of these are the rRNA operons, which have now been added in turquoise to the annotations (tRNAs are added as light green). There is also a region near the ten o’clock position of the chromosome, which shows a more GC-rich region that will tend to take more energy to melt (e.g., dark green in the stacking energy lane). As we will discuss in later chapters, it is common to find regions throughout the genome with variations in base composition, so the landscape along the chromosome is not as homogeneous as one might think.

The year 1995 also saw the second bacterial genome published, of *Mycoplasma genitalium* (Fraser et al., 1995, U.S. patent number 6,537,773), an intracellular human pathogen that is sexually transmitted. With this publication the field of comparative bacterial genomics was born: the two genome sequences were naturally compared and contrasted. With only 580,000 bp the genome of *M. genitalium* belongs to the smaller bacterial genomes. However, once again, things look different now, since even smaller genomes have been sequenced since, such as *Nanoarchaeum equitans* (a parasitic archaea living together with another archaea at extremely high temperatures) that only has 490,000 bp, and *Carsonella ruddii*, which is only 160,000 bp.

**Comparative bacterial genomics**

Figure 3.7 represents our first example of Comparative Genomics, as we are comparing the genomes of two bacterial strains to each other. The Genome Atlas provides a quick overview of a sequenced bacterial genome in a single figure. Scanning the two atlases in Figure 3.7 shows that these genomes appear to have a quite similar organization. For instance, the position of rRNA and tRNA genes is conserved, although there are a few differences, notably a high stacking energy region (the second circle counting inwards) that is missing in the genome of pathogenic nontypeable *H. influenzae* 86-028NP, shown in the bottom of the fig-
ure. Some other differences (such as in the positions of repeats between the two genomes) are also indicated in the lower genome. Atlases are, in our opinion, an excellent way of comparing small numbers of genomes to each other. However, since 1995, there has been an explosion in the number of sequenced genomes. Research grants are currently written not to sequence a genome, but rather to sequence an entire strain collection, with hundreds or thousands of bacteria represented. What this will do to the tools for bioinformatic analysis, databases to store the information, and scientists trying to cope with this avalanche of information cannot presently be envisaged.

In the years since 1995, many genome sequences were generated subsequently following a general pattern: bacteria with relatively small genomes and mostly species of clinical or economic relevance were sequenced by shotgun cloning and high throughput dideoxy sequencing, and novel genome sequences were compared with published ones. Although pathogens remained highly overrepresented, the first archaea was sequenced in 1996 (Methanocaldococcus jannischii, a methane-producing thermophile) and apathogenic bacteria soon followed (as mentioned above, B. subtilis and E. coli K-12 were both published in 1997). A novelty was the publication of a second genome for one bacterial species, in 1999. The honor went to Helicobacter pylori, a pathogen living in the human stomach. For the first time genomes could be compared within a species, and only now did it become apparent how much variation could occur between organisms that according to classical bacteriology were ‘identical’. The number of bacterial species being sequenced has since increased exponentially (Figure 3.8). Currently, multiple genomes per species are no longer an exception, and that this is not a waste of resources will be discussed in Chapter 13.

![Figure 3.8](image)

**Figure 3.8.** The number of bacterial and archaeal genome sequences (blocks) and their sequenced nucleotides (graph) have increased exponentially over the last decade, and are expected to rise further (last column represents genomes in progress).
The microbial genome: not all bacteria are like *E. coli*

The bacterium *Escherichia coli* (one of the few bacteria that are more commonly known by its genus abbreviation ‘E.’ than its full name) is considered in many microbiological textbooks as a ‘typical bacterium’. However, *E. coli* is not more typical for bacteria than, say, a hedgehog is typical for all mammals. The fact that *E. coli* is well studied and easy to grow in the laboratory doesn’t mean its features are always applicable to other bacteria. For instance, like *E. coli*, most bacteria that we know of have a circular chromosome, but some have a linear chromosome (like *Borrelia burgdorferi*, the causative agent of Lyme’s disease). *Vibrio cholerae* (causing cholera) has two chromosomes, some *Burkholderia* species (marine bacteria) have three and it is possible that there are bacteria out there with four or even more chromosomes.

*E. coli* DNA contains approximately 50% AT and 50% GC base pairs, but the AT content of bacteria can vary from 25% to over 75%, depending on the species. Some bacteria have smaller chromosomes than that of a complex virus. Other bacteria carry besides their chromosome megasize plasmids that are essential for replication and growth. In conclusion, the *E. coli*-based model of bacteria containing a single, circular chromosome where all essential genes are located, and non-essential plasmids that can be gained or lost, does not apply to all species.

Despite the described variation, bacterial genomes still have a lot in common to justify comparisons. All chromosomes contain an origin of replication and a terminus region (treated in Chapter 7). All genomes contain genes of which a large number (from 40% to 95% depending on the species) can be recognized to belong to gene families. For many of these families we have a good idea what function the encoded protein has. The number of genes present in a genome highly differs (as does genome size) and so does the gene density: how compact or diffuse the genes are distributed along the genome. These observations describe the basic essentials of comparative genomics: to compare identity and location of genes in a genome. Although genes are not the only features one can study while comparing genomes, they are the most significant in terms of evolution and biological functions. Gene-independent features of the genome sequence are covered in Chapters 7 and 8 such as frequency of base- and word counts, predicted three-dimensional structural variation, and repeats in sequences.

**Concluding remarks**

The genome of a bacterium must comprise all the DNA coding molecules present in the cell. For some bacteria, there is only one chromosome, so that the chromosome is equal to the genome. Other bacteria contain multiple chromosomes, and roughly a third of the sequenced bacterial genomes to date contain at least one
plasmid. DNA Base Atlases and Genome Atlases are visualization tools for looking at sequenced DNA molecules (chromosomes, plasmids or viruses), that can reveal regions of different nucleotide composition as well as regions of different structural and repeat properties, compared to the chromosomal average. The number of sequenced bacterial genomes is quickly increasing, due to the development of fast and inexpensive technologies for sequencing. All this DNA is stored in databases, that are can be searched for particular information and from which sequences can be retrieved. These are the subject of the next chapter.

References:


4 An Overview of Genome Databases

Outline Most genomic researchers are familiar with GenBank and its retrieval system ‘Entrez’, a portal to the protein and amino acid databases. Protein databases are overarched by UniProt, linking separate databases on protein function, structure, domains and much more. In addition there are ongoing efforts to develop dedicated databases to store additional relevant microbiological information. There are many genome databases available, oftentimes specialized for a few organisms, or enabling easy searching for specific features such as protein domains, special classes of RNA, or three-dimensional structures. The design and structure of a database have as much influence on their use as the data they store. The speed at which databases are updated is also an important area of concern. Databases of sequence and related information are still evolving, and the future may see further improvements and automated refinements, possibly at a cost of personal oversight.
Introduction to databases

This chapter focuses on genome databases. Most readers are probably familiar with GenBank, available through the U.S. government’s National Center for Biotechnology Information (NCBI) web pages. GenBank stores sequence information (protein and amino acid sequences) and the database is cross-linked to other NCBI databases storing information on scientific publications (PubMed), taxonomy, protein structure and more. This and other web-based databases are extremely powerful tools in scientific research and we cannot overemphasize their value. Some of the databases stored by NCBI that are less well known than PubMed and GenBank, will be introduced here. However, NCBI is not the only institute providing useful databases for the microbiologist.

We will examine some of these, although there are legions of databases available on the web in addition to those cited throughout this chapter. Some are dedicated to specific areas of interest, others cover broader fields. Some of these suffer from lack of curative efforts or are not as user-friendly as others. As many publicly available databases are the product of small research groups (if not individuals), many store incomplete or dated information. We emphasize that our selection is incomplete and omissions are by no means intended to reflect perceived lack of quality. The web addresses (URLs) of databases discussed here are provided. In order to facilitate ease in navigating these various sites, we provide a single web table\(^1\) that contains an updated list of all these URLs, as well as those mentioned elsewhere in the book. The reader is encouraged to visit this web page and bookmark it for future use.

What is a database?

There are (at least) three different meanings of the word database\(^2\). The first, and most common definition is that a database is an organized collection of information. The term can also refer to a computer program that is used to search through the records and retrieve specified information. In that meaning, ‘database’ is short for ‘DataBase Management System’ (DBMS), and examples include Structured Query Language (SQL) based systems such as MySQL. However, technically these are computer programs that manage a database, and are not the actual data-

\(^1\) http://comparativemicrobial.com

\(^2\) The word ‘data’ is plural for Latin ‘datum’ which means ‘that which is given’, in this case a piece of information (derived from ‘dare’: to give). In English ‘data’ is used both plural and singular. Since the synonym ‘databank’ is less frequently used in bioinformatics than ‘database’ we use the latter.
base itself. The third definition of a database that is commonly used (though mistakenly, in our opinion) is to describe a set of files that are stored on a computer. Thus for example, a folder containing multiple sequence files is considered by some to be a ‘database’. We prefer the first definition, which includes the word organized. For a database to be of added value beyond retrieving stored files, the data must be stored in a systematic way, such that a computer program can search particular features stored in the entered data. From this chapter it will become clear how important this organizing feature is.

Databases are extremely useful for the sorts of questions that microbiologists may want to investigate, using bacterial genomes as a resource of information. There is an awful lot of information in any one given genome sequence—of course, even more so for many genomes. Now, for the first time, it is possible to make queries across hundreds (and soon thousands) of genomes, asking questions such as “how many tRNA operons are there per genome?” or “what is the average coding density of these (or this subset of) bacterial genomes?”, “Which genome currently known is the most AT rich?”, “Which is the largest?”, etc. There are many properties of genomes that can be defined by one (or a few) variables. Using a database of genomes, it is possible to build up multiple lists of global properties. This book will illustrate that one database of genomes is not enough to cover all information. Many researchers active in the fields of bioinformatics or genomics spend a lot of time building their own databases that are tailor made to suit their needs. As setting up databases is a profession by itself, with many specialized books dedicated to the art, it is not extensively covered in this book, but some more background is presented in the next chapter.

As an example for the sorts of questions that can be addressed using databases of genomes, we will consider a simple case: investigating the total length of the genome, in terms of base pairs of DNA. Given the data entered in the database are complete genome DNA sequences, a simple sorting function can easily answer the question of which genome is the largest. When the data also list to what type of bacteria each genome belongs (giving species, genus, phylum etc.) one can also deduce the average size of genomes for a particular type of bacteria, for example for all bacteria belonging to one phylum.

When the stored data are linked to, say, a second database containing references to scientific literature, a search in one can identify an entry to further investigate in the other. Frequently, such links work only one-way. Within NCBI, all databases are linked to each other and some are linked to external databases (such as to publisher’s databases storing original publications), however, the latter ones do not generally link back into NCBI.

The organization of a database defines to a large extent what information can be retrieved, and how. Let’s consider setting up a database storing bacterial ge-
nemes. Before doing so, we should ask the question: what is a genome? This might seem a bit obvious, but from the perspective of a database record, it is extremely important to use a definition that is clear and used consistently in all data entries. The most general definition of a genome is all genetic information present in a given cell. As described in the previous chapter, bacterial genomes may contain multiple chromosomes or plasmids in addition to chromosomes. Only in cases where one chromosome covers all genetic information are the words chromosome and genome synonyms. Unfortunately, microbiologists frequently refer to the chromosome as the ‘genome’, irrespective of other DNA molecules present. More importantly, a data entry for our genome database should include all DNA related to that genome.

**How database organization affects its future use: A brief history of GenBank**

GenBank was first formed in 1982, initially housed at the Los Alamos National Laboratories, in New Mexico, and has grown tremendously since then. The first version of GenBank contained a total of 484,813 bp of DNA. GenBank’s original design focused on a single sequence file per data entry, a structure that has been preserved intact throughout the years: each genome entry contains one DNA molecule. As a consequence, multiple chromosomes per organism (or chromosome plus plasmids) are entered as separate files. At present, it is difficult to know from the individual GenBank files which plasmid and chromosome sequence were derived from the same strain, but when GenBank was originally developed, it contained sequence entries that covered far less than a complete genome. It was not an immediate requirement to provide contextual linkage to other sequence entries.

About ten years later, in 1992, GenBank was moved to its current location at the U.S. National Center for Biotechnology Information (NCBI), which is part of the U.S. National Library of Medicine, located just outside of Washington, D.C. Figure 4.1 shows the growth of GenBank with time. For comparison, Moore’s law is also plotted as a straight line.

Moore’s law states that approximately every two years the number of transistors that can fit inexpensively on a chip will double; this effectively doubles the computational power of computers. The numbers of nucleotides stored in GenBank increase over time every 18 months, therefore at a higher rate than the processing power. The plot for the number of nucleotides in GenBank ignores the ‘Whole Genome Shotgun’ (or ‘WGS’) section of GenBank, in which raw data of eukaryotic shotgun genome sequencing data are deposited. As of December 2007, this database was already larger than the rest of GenBank, and likely it will soon be twice its size.
Figure 4.1. Growth of total number of bp present in GenBank (curve). The blue line represents Moore’s law, describing the rate at which computer processing power increases, starting at the initial level of sequences in GenBank, and then doubling every two years. GenBank roughly increases every 18 months, a higher rate than computer processor speed. Not included in the represented GenBank versions is the Whole Genome section (WGS), which in December 2007 was already larger than GenBank (red dot).

This has major consequences to computational analyses. For example, a BLAST search done today will take considerably more time than exactly the same BLAST search done 2 years ago. At the cost of speed, however, more information is likely to be found from a BLAST search done today, since the databases are so much larger in size. Nevertheless, searching the exponentially growing amount of sequence information with the same traditional methods for analysis will take increasingly more time. New methods need to be developed to solve this problem and in order to fully take advantage of this rapidly increasing amount of data.
Three databases storing sequences and a lot more

Today GenBank is part of a larger consortium, the International Nucleotide Sequence Database Collaboration (INSDC). This collaborative effort connects three databases: GenBank, the DNA Database of Japan (DDBJ, started in 1986) and the European Molecular Biology Laboratories (EMBL) Nucleotide Sequence Database. These three databases are synchronized, such that they use the same identification code (called accession number, explained in more detail below) for the same DNA segment. The reason there are three of these databases is mainly historical. Nevertheless, each has its own strength. The DDBJ web portal provides tools that NCBI doesn’t offer, such as FASTA and CLUSTALW analysis, but in general it is less user-friendly than NCBI (though preferences differ between individuals). The EMBL web portal provides a list of microbial genomes with a layout superior to that of NCBI and, as NCBI does, it links smoothly into taxonomic information. The URLs of these main DNA databases are given in Box 4.1.

Box 4.1: Websites providing access to the main DNA sequence databases

<table>
<thead>
<tr>
<th>Website</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://insdc.org">http://insdc.org</a></td>
<td>INSDC</td>
</tr>
<tr>
<td><a href="http://www.ddbj.nig.ac.jp">http://www.ddbj.nig.ac.jp</a></td>
<td>DNA Database of Japan</td>
</tr>
<tr>
<td><a href="http://www.ebi.ac.uk/embl">http://www.ebi.ac.uk/embl</a></td>
<td>EMBL Database</td>
</tr>
</tbody>
</table>

As GenBank is part of the NCBI database collection, it provides cross-links to a number of other databases that are quite useful for microbiological or bioinformatic research. The best known is probably PubMed, which provides access to citations from biomedical literature. PubMed was developed by NCBI at the National Library of Medicine (NLM), located at the U.S. National Institutes of Health (NIH). Both PubMed and GenBank can be accessed online using the search and retrieval program ‘Entrez’ which is relatively user friendly. For instance, it corrects typing errors as some web search engines do. Thus, if searching PubMed with ‘pseudomonas’ or ‘psuedomonas’ Entrez will ask “Did you mean: pseudomonas” and will give the number of hits for that alternative word. (Note that Entrez is case insensitive, so ‘Pseudomonas’ is treated the same as ‘pseudomonas’). When in doubt about the spelling of a scientific term (possibly unknown to your dictionary), a quick comparison of variants in PubMed will reveal which form is most commonly used.

Each publication listed in PubMed has a unique PubMed Identification Number, or PMID. In database terminology a PMID is a primary key: a key (or code, number, identifier) that is unique and refers to a given unit of information. Other examples of a primary key are the ISBN code to identify a book, or a Social Security Number to identify a U.S. citizen. The PMID number can be used to quickly find a particular publication by entering that number in the search field. For this reason, we have added the PMID for cited references at the end of each chapter, when available.

The Taxonomy database at NCBI is also very useful for microbiologists. As PubMed, it can be searched using Entrez, with spelling corrections in place. Moreover, searching with bacterial names that are no longer in use will usually give you the modern name. Bacterial taxonomy is a moving target, as species are frequently re-named and re-ordered as a result of novel scientific insights. For instance, *Helicobacter pylori*, as we know it today, was discovered in 1982 and originally named *Campylobacter pyloridis*, which became *Campylobacter pylori*, until it was placed in a separate genus and was renamed *Helicobacter pylori*. Either entry will retrieve *H. pylori* from the Taxonomy database (and in there, you can find the earlier names as well). This feature is particularly helpful when dealing with older literature. Genome sequence entries at GenBank can be limitedly searched with outdated names. What to do if a genome of an organism has already been sequenced and published before it is moved to another genus is an issue we are likely to face in the future.

**Why three databases is still not enough**

With GenBank conveniently embedded in a number of related databases, and two other databases storing DNA and protein sequences, the demands are still not completely satisfied. There is still a need for other databases, as none of the existing ones can answer a question as simple as: “Which genomes belong to bacteria that grow at temperatures higher than 70˚C?” or “What was the geographical origin of the isolate belonging to this genome sequence?” or “How often was the isolate belonging to this pathogen cultured in the laboratory before its sequence was generated?” all questions that a microbiologist will recognize as highly relevant.

Taxonomy is not the best classification to group bacteria that share an ecological niche, such as growth at extreme temperatures. Although the use of Geographical Information Systems (GIS) is more and more frequently used in microbiology (for instance to recognize epidemiological patterns, or to map ecosystems in the ocean), there is currently no link to genomic data (or vice versa) using the public databases, although the Genomic Standards Consortium is working on this. We will further treat the background of this need in the next chapter.
The pathogenicity of a sequenced isolate belonging to a pathogenic species, another biological feature that is difficult to capture in a database, is not always obvious: in fact, multiple passage of an isolate on Petri dishes in the laboratory can affect pathogenic properties, and it is currently not known if or how such laboratory adaptation affects the genome sequence. The first bacterial genome to be sequenced, that of *Haemophilus influenzae*, was produced from a non-pathogenic strain, but this information does not show up in GenBank. In order to allow maximal use of genomic data, there are worldwide efforts amongst biologists, genome sequencing centers, and the INSDC (the consortium to which GenBank belongs) to develop a common database where additional biological and geographical information of sequenced isolates will be stored and linked to the sequences.

**How can I keep up-to-date with the newest genomes sequenced?**

There are many databases, but we find four places to be good sources of reliable up-to-date information about the microbial genomes sequenced so far. These are listed in Box 4.2; the first is the ‘GOLD’ pages (Genomes OnLine Database), in our opinion the best place to go for a list of all the bacterial genomes projects being currently sequenced. Thus, for example, if one wants to know if a particular bacterial strain has been sequenced (or is in the process of being sequenced), this is a good place to check. The second link is to a table provided by NCBI, which lists all the prokaryotic genomes currently stored in GenBank. This shows the dates of the first release and modifications, and sorting options for genome size and GC content. The third link is to our own web pages, which is not unlike the GenBank link, but provides alternative columns containing the number of rRNAs, tRNAs, genes, etc for each genome, again with the option to sort by any of these fields. The site provides additional information about each genome, as will be discussed in more detail in future chapters. The final link is to the EMBL genome web pages, which also give a good overview.

**Box 4.2: A few websites of recently sequenced microbial genomes**

http://www.genomesonline.org
http://www.cbs.dtu.dk/services/GenomeAtlas
http://www.ebi.ac.uk/genomes
What is an accession number?

It is quite important to understand what is meant with an accession number. The GenBank accession number (often simply referred to as accession number) is a primary key to uniquely identify a sequence entry in GenBank. Accession numbers are shared by EMBL and DDBJ, so that they are truly unique and can be used for information retrieval in all three databases. They usually have the format of two letters followed by six digits (AB123456), although older accession numbers can be a bit shorter: one letter, followed by 5 numbers (A12345). Although a primary key should ideally not change, an accession number is often followed by a period and a version number (e.g., AB123456.1), so if a sequence is revised by the authors who submitted the sequence, it is given a new version number but the rest of the accession number remains constant. Thus, for example, there are currently three different versions of the E. coli strain K-12, isolate MG1655, genome sequence: U00096.1 contains 4288 annotated genes, U00096.2 contains 4254 genes, and the most recent version, U00096.3 contains 4331 annotated genes.

Furthermore, the DNA sequence is also slightly different in each of these files, which may affect work done on an earlier version (such as prediction of promoters at specific locations along the chromosome). In light of this, it is important to make sure that the full accession number, including date and version number, are recorded if extensive calculations are being done on a particular GenBank file. When withdrawing information from the non-redundant database (nrDB at Entrez), the latest version of an accession number will only be included.

Some genomes in NCBI also have a special accession number called RefSeq number, and this is not the same as the GenBank accession number. To define RefSeq entries, sequences entered in GenBank are extracted, annotated and hand-curated, and put back in the database with the RefSeq number as a new identification key. Thus, every entry with a RefSeq number will have a GenBank accession number, but not the other way round. The format is two letters followed by an underscore and six digits (AB_123456); the first two letters refer to the type of sequence, using codes like NT for contigs, NM for cDNA sequences constructed from mRNA, NP for proteins, NC for chromosomes and plasmids. NZ is for shotgun unfinished sequences, which have a slightly different format: NZ followed by four letters and eight digits (i.e. NZ_ABCD12345678).

Finally, the Genome Project ID (PID) is important to mention. This is a relatively recent addition at the NCBI web pages, in which each genome has been assigned a number. The PID overcomes the problem encountered with genomes containing more than one DNA segment (e.g., a chromosome and a plasmid, each with their own unique accession number). In addition, biological information is stored here to give more background on the strain that was sequenced. Unfortu-
nately, this is not easily searchable. Currently, using ‘Entrez Genome’ to pull out the ‘Bacteria Complete Chromosome list’ does not provide the PID of the listed organisms. It is expected that this will soon be added. The links mentioned above in Box 4.2 are the easiest way to find a PID for an organism. Figure 4.2 illustrates the relation between the accession number, RefSeq number and Project ID.

![Figure 4.2](image)

**Figure 4.2.** Accession numbers are given to DNA (or protein) sequences submitted to GenBank. A subselection of these receive a RefSeq number after curation by hand. Sequences of different DNA molecules obtained from the same isolates (chromosomes, plasmids) receive a combined Project ID number.

PIDs have therefore the power to link ‘one-to-many’, in other words, to relate organism-specific data together. For instance, the individual sequence entries of chromosomes and plasmids that were generated from a particular organism are combined into one PID. Consider, giving another example, a project in which all DNA from the microbiota of the human gut were to be analyzed. These would obviously be comprised of PIDs of individual bacterial species and strains, and their PIDs could be used to obtain all the relevant information about their genomes. These examples illustrate that PIDs add flexibility to the databases, and overcome the original limitations of one-sequence-per-data-entry of GenBank and related databases.

**Data files and formats**

Computer programs access information by means of files that hold the relevant information in a proper layout. The file format defines how the information is stored

---

and organized in that file. Quite a few different formats exist in which sequences can be produced and stored or retrieved. A file format is important for particular programs, including some of the web interface tools described in this book that will work with one format but not with another. Fortunately, there are only two file formats that, in our opinion, are commonly used and central enough to mention here.

The first one is the FASTA format, introduced in Chapter 3. It represents more-or-less the naked DNA (or RNA or protein) sequence, with the possibility of one additional header line that starts with the ‘>’ sign and is separated from the sequence by a hard return. FASTA files are frequently given the suffix ‘.fsa’, and they are mostly used in those programs where the sequence is central, such as BLAST.

The other format to be discussed is the GenBank file, recognizable by the suffix ‘.gbk’ or ‘.gb’. GenBank files give more information than just the sequence, and this information is organized in a particular way (Figure 4.3). As a GenBank file of a genome sequence would easily fill this book, we’ll use the example of DNA polymerase from *Thermus aquaticus*, commonly known as ‘Taq’, the enzyme used in PCR, or polymerase chain reactions. Even the file of this single protein is still quite long so only part of it is shown. Most of the information speaks for itself, as each line or block of information is listed under an identifier (all capitals). Note that accession number and version, source and organism are specified.

The database from which this information was extracted (‘DBSOURCE’) is SwissProt (explained below). Only one publication reference is shown here (of the 10 listed in the original). Following the references are ‘FEATURES’ that seem to be duplications of the information already given: the source, gene and protein. Note, however, that if this were a complete or partial genome sequence, the location would tell you where on a long stretch of DNA this particular gene encoding the protein could be found. Please note that protein location and gene location are not necessarily the same, for instance in the case of spliced genes, a situation that less common in bacteria but very common in eukaryotes.

The Enzyme Commission (EC) number, a classification scheme for enzymatic activity, is also provided for the entry. At the bottom of the file the protein sequence is shown, grouped in 10 amino acids per block and 60 per line. The end of the sequence is indicated by ‘///’.

A click on the NCBI ‘Display FASTA’ button would give the .fsa file of this

---

5 Restriction enzymes and other enzymes used for DNA manipulation are usually named using a three-letter code, with the first letter of the genus name (thus it is upper case) and the next two for the species name. These letters are italic since the full name is also printed in italics, and can be followed by numbers or letters in roman text, e.g., *EcoRI* (derived from *E. coli*) or *HindIII* (from *H. influenzae*).
protein sequence, without empty spaces or numbers, which can be exported (saved locally on your computer) or simply copied and pasted into another application. However, all extra information listed in the GenBank file is then no longer available. This loss of information can be overcome by inserting essential data in the comment line, but this obviously has limited information capacity.

LOCUS    P19821
832 aa      linear  BCT 15-JAN-2008
DEFINITION DNA polymerase I, thermostable (Taq polymerase I).
ACCESSION P19821
VERSION    P19821.1 GI:118828
DBSOURCE  swissprot: locus DPO1_THEAQ, accession P19821;
          class: standard.

..[snip]..
KEYWORDS 3D-structure; Direct protein sequencing; DNA damage; DNA repair;
DNA replication; DNA-binding; DNA-directed DNA polymerase;
Nucleotidyltransferase; Transferase.
SOURCE  Thermus aquaticus
ORGANISM    Bacteria: Deinococcus-Thermus; Deinococci; Thermales; Thermaceae;
Thermus.
REFERENCE  1 (residues 1 to 832)
AUTHORS  Lawyer,F.C., Stoffel,S., Saiki,R.K., Myambo,K., Drummond,R. and
          Gelfand,D.H.
TITLE   Isolation, characterization, and expression in Escherichia coli of
          the DNA polymerase gene from Thermus aquaticus
JOURNAL  J. Biol. Chem. 264 (11), 6427-6437 (1989)
PUBMED  2649508
REMARK  NUCLEOTIDE SEQUENCE [GENOMIC DNA].

..[snip]..}
FEATURES    Location/Qualifiers
source  1..832
/organism="Thermus aquaticus"
/db_xref="taxon:271"
/gene=1..832
/"gene=polA"
/note="synonym: pol1"
Protein  1..832
/gene="polA"
/product="DNA polymerase I, thermostable"
/EC_number="2.7.7.7"

..[snip]..
ORIGIN
1 mrgmlplfep kgrvllvdgh hlayrtfhal kglttsrgep vqavyg faks llk
   61 avivvfdaka psfheaygg ykagraptpe dfprqalik elvdilgarr evpgeayd
   121 vlaslakae kegyevrilt addkqllylls dlhvlhphe ylltqalwe kgylrdpwa
   181 dyralgtdes dlpgykgg ekrtkllie wgsleallkn ldrkpaire kilahmdlk
   241 lsldlakvrt diplevdfak rrepderler afierlegrs lhhefgles pkaleeawp
   301 ppegaaygfy lsrkemwpdad llalaaargg rvhrpemykyvk ardikeaarg lakdelvsa
   361 lreglglpp ddpmllayll dapsnttepg arryggewe egeraale rfanlgrl
   421 egeriluksy reversplav lagreriavrfd lvdavlrls levaereal eaevefrylagh
   481 pfptnrsdq erllfedeg1 paigktektk krstsaavle aroleahpive kilqyeltk
   541 lksydpplkl dlhpntgri htrfrqdtata tgrissdop lojipnptri gyrrifrla
   601 eegwilvald ysgielrvla hlsdlrnir vfpegrdiht etasumfvgv reavpllnr
   661 aaktinfygl ygmahrlsq elaipeeqa aferiyfegf pfvkrawibekt leegrrgyy
   721 etlfgrrrvy pdlearvksv reaarermfa mprvqtaadl mklamvk1fpl rlengarml
   781 lqvdelvle apkeraeava rlakemegv yplavpleve vgigedwlisa ke

Figure 4.3. Protein sequence of Taq polymerase represented as a GenBank file. Some of the original text has been deleted, indicated by ..[snip]..
RNA databases

The URLs of most useful databases specialized on various RNA molecules are summarized in Box 4.3. There has been a major effort to collect and organize rRNA sequences, since they can be used for taxonomic classification of species. The RDP (Ribosomal Database Project) is currently run by Michigan State University and is updated on a monthly basis to keep up with newly released genome sequences. It contains over 100,000 bacterial 16S rRNA (small subunit) sequences.

Other non-translated RNAs can be found in the Rfam database, which is a joint project between Wellcome Trust Sanger Institute in the UK and Janelia Farm in the U.S. It maintains a large collection of common non-translated RNA families. Although these are collectively called ‘non-coding RNA’ in the introduction at the Rfam site, the RNA covered in this database includes 5S rRNA (but not 16S or 23S rRNA) and tRNAs. Apparently, what is meant is ‘non-protein coding’. The Rfam site allows to search for particular RNA molecules by text or accession number queries as well as by sequence alignment. It also provides a list of bacterial genomes, for each of which the annotated tRNA, 5S rRNA and other non-translated RNAs (sorted for Rfam families) can be extracted with their predicted structure and relevant references.

Box 4.3: A few websites databases dedicated to bacterial RNA

<table>
<thead>
<tr>
<th>URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://rdp.cme.msu.edu">http://rdp.cme.msu.edu</a></td>
<td>The Ribosomal Database Project</td>
</tr>
<tr>
<td><a href="http://www.sanger.ac.uk/Software/Rfam">http://www.sanger.ac.uk/Software/Rfam</a></td>
<td>The RFAM database</td>
</tr>
<tr>
<td><a href="http://lowelab.ucsc.edu/GtRNAdb">http://lowelab.ucsc.edu/GtRNAdb</a></td>
<td>The Genomic tRNA database</td>
</tr>
<tr>
<td><a href="http://www.indiana.edu/~tmrna">http://www.indiana.edu/~tmrna</a></td>
<td>The tmRNA database</td>
</tr>
</tbody>
</table>

Probably the best place to go for tRNA genes is the Genomic tRNA database (The Lowe Lab, U.S.). This site lists automated (and therefore not curated) tRNA sequences from finished and nearly finished genome sequences. In contrast to Rfam, multiple chromosomes are automatically reported as a single genome. It allows to view all predicted tRNAs per bacterial genome, and sort these by isotype (for which codon and amino acid the tRNA is used) or location. The secondary structure can be viewed and alignments can be produced. The database can be text-searched or used for alignment analysis.

For those interested in tmRNAs (which will be further introduced in Chapter 9) the Rfam database discussed above is a good start. There is also a database available exclusively dedicated to bacterial tmRNAs. At the time of writing, this website appeared to be frequently updated and even unfinished genomes were included.
Protein databases

Compared to nucleotide and protein sequences stored and retrievable at NCBI, until recently the databases specifically dedicated to sequence, structure and function of proteins were far less user-friendly. For historic reasons there are many such databases, each specialized in a particular field. The most important of these are now covered by overarching databases that (still) partly link back into the original components. An inexperienced user could easily become overwhelmed by acronyms and abbreviations. Box 4.4 gives the URLs of the databases mentioned here.

**Box 4.4: The main websites databases dedicated to protein**

<table>
<thead>
<tr>
<th>URL</th>
<th>Database Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.uniprot.org">http://www.uniprot.org</a></td>
<td>The UniProt database</td>
</tr>
<tr>
<td><a href="http://www.expasy.org/sprot">http://www.expasy.org/sprot</a></td>
<td>The SwissProt database</td>
</tr>
<tr>
<td><a href="http://www.ebi.ac.uk/trembl">http://www.ebi.ac.uk/trembl</a></td>
<td>The TrEMBL database</td>
</tr>
<tr>
<td><a href="http://pir.georgetown.edu">http://pir.georgetown.edu</a></td>
<td>The PIR database</td>
</tr>
<tr>
<td><a href="http://pedant.gsf.de">http://pedant.gsf.de</a></td>
<td>The PEDANT database</td>
</tr>
<tr>
<td><a href="http://prodom.prabi.fr/prodom/current/html/home.php">http://prodom.prabi.fr/prodom/current/html/home.php</a></td>
<td>The Protein Domain database</td>
</tr>
<tr>
<td><a href="http://expasy.org/prosite">http://expasy.org/prosite</a></td>
<td>Database of protein domains, families and functional sites</td>
</tr>
</tbody>
</table>

Starting at the top, UniProt (Universal Protein resource) is a combination of the three largest databases, Swiss-Prot, TrEMBL and PIR-PSD. Swiss-Prot is the result of a collaboration between SIB (Swiss Institute of Bioinformatics) and EBI (European Bioinformatics Institute). Proteins in Swiss-Prot have actually been isolated, and entries are carefully annotated and curated. Swiss-Prot is recognized as the ‘gold standard’ of protein annotation. Although the most reliable, Swiss-Prot is also the least complete protein database. As genome sequences produce protein information at a speed that Swiss-Prot can’t keep up with, TrEMBL (Translated EMBL) was introduced. This database is automatically generated by translating the EMBL nucleotide sequence database for all proteins not present in Swiss-Prot. Proteins predicted in TrEMBL are not annotated or reviewed, and many are hypothetical. PIR (Protein Information Resource)-PSD (Protein Sequence Database) is a maintained database of curated protein families. All of these individual components can be searched separately, but here UniProt (the overarching database) is introduced as it is the most user-friendly and has broadest applications.
UniProt has reorganized all data into three new components:

**UniProtKB:** the Knowledge database. This is the central access point for curated protein information, including function, classification and cross-reference. When searching UniProtKB, the hits are reported back as either ‘SwissProt’ or ‘TrEMBL’, depending on which database is used (the number of hits in SwissProt is generally lower than that of TrEMBL for reasons explained above).

**UniRef:** the Reference Clusters database, which provides clustered groups of UniProtKB proteins with 100%, 90% or 50% sequence identity.

**UniParc:** the Archive database. This stores the complete body of publicly available protein sequence data.

At the time of writing, the homepage of UniProt still directed searches into the ExPASy (Expert Protein Analysis System) website of SIB, but already a link to the overarching database is provided, through http://beta.uniprot.org. Compared to the old data retrieval system at ExPASy, the searches at UniProtKB have improved significantly. Improvements include that text searches now suggest spelling corrections, allow restriction of terms to organism, gene ontology (see Chapter 5), etc., and allow a choice between reviewed Swiss-Prot or unreviewed TrEMBL entries. Thus, when searching, for example, for flagellin genes of Campylobacter jejuni, one can restrict the term ‘campylobacter’ to organism, and ‘flagellin’ to protein family. That way, the reported findings exclude proteins for which the word ‘flagellin’ appears somewhere in the annotation (like a protein modifying flagellin). Searching with gene names is also a possibility. Since we have mentioned the taxonomic database at NCBI, that of UniProt should not remain unnoticed. It can be searched through ‘NEWT taxonomy’ and any findings report immediately how many Swiss-Prot and TrEMBL entries there are for the strains listed.

The choice which database to use is dictated by the information stored in there, but user-friendliness and layout are also important. As an alternative to UniProt, the PEDANT database deserves to be mentioned. PEDANT (Protein Extraction, Description and Analysis Tool) is a product of the Munich Information Center for Protein Sequences. Here, under ‘Bacteria’ you can select your genome of choice from an alphabetical name list (links to the NCBI genome project and taxonomy database are provided for each entry). One click on your species of choice will provide a list of all predicted proteins with a short description and the best BLAST hit. For incomplete genome sequences a list of contigs is available. Protein-encoded genes are separated from RNA genes (called ‘genetic elements’ at this web site); the list of RNA genes is sorted for rRNA, tRNA and ‘miscellaneous’. Stem-loop structures are also specifically listed. For each gene of interest a detailed list is produced of predicted function, localization, protein structure, and general properties. FASTA files can be exported as text files.
When you work with a novel protein gene for which you have little information, two other databases can be useful. If you want to predict a possible function of a query gene, it is worth searching ProDom (for Protein Domain database). ProDom uses a good graphical interface. The database consists of an automatic compilation of homologs domains detected in the Swiss-Prot database using a specific algorithm. It was devised to analyze specific domain arrangements within proteins. ProDom will identify similarities in domains that may or may not have conserved function. In this type of analysis, domain boundaries should always be treated with caution. For some domain families, ProDom has used the opinion of experts to correct domain boundaries on the basis of sequence and structural protein information.

Finally, ProSite should be mentioned as an alternative to ProDom. ProSite is not that different from ProDom, but it is based on conserved function. The database contains entries of biologically significant sites, patterns and profiles within well-characterized proteins that can help to reliably identify to which known family of protein (if any) a new protein sequence belongs. Further possibilities to explore when dealing with ‘unknown’ proteins are introduced in Chapter 11.

Concluding remarks and recommended reading

Our brief presentation of databases is far from complete and there are many websites specialized in listing bioinformatic resources and websites. When the mentioned databases are not satisfactory to specific needs, browsing the web may identify a database that better suits your need. A large collection of databases is presented by BioMed Central (http://databases.biomedcentral.com), which can be searched for particular subject areas or contents. Note, however, that quite a few links in this resource are inactive.

Bioinformatic databases are constantly being produced, evaluated and refined. The latest news on the front of databases is presented annually in the January issue of Nucleic Acids Research (http://nar.oxfordjournals.org) and this is a highly recommended reading.
5 The Challenges of Programming: a Brief Introduction

Outline  Similarly to the wet-lab research, bioinformatics has its own dedicated tools and techniques. While wet-lab research uses laboratory protocols, biological material and lab equipment, tools for the bioinformatician are programming languages, databases and computational techniques. Being proficient with these tools is fundamental to achieve a high degree of productivity and research insights. This chapter introduces some general concepts and vocabulary behind data manipulation. The difference between data and metadata is explained. The development of ontologies allows a more standardized and meaningful description of concepts. The process of data identification and extraction, data analysis, interpretation and representation are briefly discussed. A number of frequently used programming languages and tools are introduced.
Introduction

Computer programming is a profession of its own, and a whole book would not be enough to cover all aspects that are relevant to design applications for comparison of microbial genomes. This chapter is divided into two parts. In the first part, we will provide a general overview of a few computer science concepts that are relevant to our context, with the intent to sketch out the main ideas and principles behind the tools used in bioinformatic research. We will also introduce some of the common jargon that can then be further explored using other resources. The second part of the chapter is targeted for the more interested reader, and focuses on some more technical details.

Part 1: A brief overview of computer science concepts

What’s in a name? Data and metadata

This book deals with the analysis of DNA, RNA and protein sequences, with the goal to obtain novel insights and increase our knowledge on the properties of these sequences and of the information associated to them. A useful classification is to regard the sequence (DNA, RNA or protein) as data (sometimes called primary data), and any associated information as metadata. Metadata could be, for instance, the organism from which the sequence was obtained, the GenBank annotation, or the length and AT content of a genome. The word metadata refers to any information that describes and enhances the meaning of some other data.

The distinction between data and metadata is important, especially for deciding optimal strategies for their storage, access and manipulation, which again influences performance and ease of use. However, the division is not always clear, and depends on the context: something that is classified as data in one situation can be metadata in another case, or vice versa. To sort the two we mostly use common sense on a case-by-case basis. In general, metadata would not be meaningful without the data it describes, and is strongly related to that data. If data change, most likely the metadata will change as well, although this again is context dependent. Data can still be meaningful without the associated metadata, but its richness of contents is reduced, or in some cases nullified. Still, lacking any metadata, even those implicit by the context, can cause unnecessary confusion. Take ‘AAAA’ as an example: is this a protein sequence of alanines, or a DNA sequence of ade-
nines, or the grades of a student? Giving a bit more information, such as that this is a DNA sequence, enables to properly classify the meaning of the represented data.

In comparative genomics, the primary data are relatively simple and well defined, generally comprising of sequences of DNA or protein. The DNA alphabet consists of only 4 letters (or 5, if we allow ‘N’ for any nucleotide) and the protein alphabet has 21 letters (allowing ‘X’ for unknown). In contrast, the variability, richness and extensibility of metadata are open-ended, and is strongly context dependent. In some cases, the metadata is all we need, for instance if we want to produce a list of all extremophiles for which the genome sequence has been completed. In this case, we are not interested in the genome sequences themselves, but will have to extract specific metadata associated to them. Nevertheless, even a very detailed pool of metadata can prove worthless for a specific task: if you want to know the optimal temperature at which each of the sequenced extremophiles multiply, you are unlikely to find that in a genomes database.

We can classify the accompanying metadata in two kinds. *Primitive metadata* (also called primitive facts) are related to the sequence itself, or to the process of obtaining that sequence. An examples of a primitive fact is the organism from which the sequence was obtained, the geometry of the molecule (circular or linear), the type of molecule (a chromosome, plasmid or virus), the experimental sequencing procedure, physiological properties of the organism, the geographic location of the sample, the clinical symptoms related to the infective agent, etc. *Derived metadata* are obtained from a computational or manual annotation from either the sequence or other primitive or derived metadata. Examples are AT content, GC skew, the annotations of coding sequences produced by experimental or computer prediction methods, the gene location, etc.

Some data that we can infer from a sequence are difficult to classify as metadata and in fact become primary data. Does a Genome Atlas belong to the metadata of the original sequence, as is it comprised of derived data, or should it be regarded as independent primary data? Although the Genome Atlas plot augments the meaning of the sequence, the richness of this information gives it a life on its own, as independent data.

**The role of metadata**

When it is so difficult to differentiate between data and metadata, why bother? In fact, metadata are extremely useful and without them it would be impossible to use biological sequences the way we do today. Here are some reasons why we need metadata. Metadata are indispensable for indexing; by providing additional
information we can enable, organize and optimize access to a specific resource or group of resources. For example, by providing the AT content as metadata of sequences, we can perform an efficient retrieval of all sequences that have an AT content greater than 50%.

Metadata furthermore provide relationships between data. For example, we can specify that a particular plasmid and a particular chromosome were derived from the same organism, that particular organisms belong to a taxonomic group, that a set of sequences have all been derived form organisms that live in a particular environment at a particular time, or that an aquatic environment was sampled according to a procedure described in a given publication. Obviously, metadata allow improved or at least less ambiguous interpretation of the data. By including a description with technical information, defining measuring units, describing sampling conditions, experimental procedures and so on, we increase the ‘usefulness’ of the data. In extreme cases, the presence of the data without essential metadata (such as from what organism the sequence was obtained) makes the data more or less worthless. Metadata have a crucial role in bioinformatics, as they do in computer science in general.

**Metadata and ontologies**

Suppose a researcher is investigating the genomes of all microorganisms living in a particular marine environment: on a coral reef. A database is available that stores genomes of organisms and the environments in which they live. How would the researcher be able to retrieve only those organisms of interest? Some organisms may have been registered as living in a ‘barrier reef’, others on a ‘fringe reef’ and yet other descriptions simply mention ‘corals’. Some entries would be stored under ‘marine’ and others under ‘sea’. How to retrieve all relevant information in such a case?

This problem relates to the following complications: *(i)* Definitions should be unambiguous but unfortunately many of the metadata used in biology are not precisely defined; what pH, for example, identifies an ‘acid environment’? What is ‘room temperature’? How is a ‘virulence gene’ defined? *(ii)* Human language is not standardized. Even if we restrict ourselves to English, synonyms and variation in spelling (e.g., color vs. colour) can still cause problems in computational analysis, as a machine can’t tell what are synonyms and what not, what spelling variation is acceptable, and how text phrases should be interpreted. *(iii)* Context can influence a concept. The ‘coral reef’ of the example is an ecosystem for a (micro)biologist, but can be a geographical location for an oceanologists, and an obstacle for a marine navigator. Ideally these concepts should be connected, so that even if they appear different by virtue of the context, a machine could be able
to tell that they refer to the same concept from different points of view. (iv) Some concepts can be general, while others highly specific, and relationships can exist among them, typically of the type is-a or is-part-of. These connections must be precisely described to be properly used: a coral reef is a hydrographic feature. A river is a hydrographic feature as well. A river mouth is part of a river, but a river mouth is never part of a coral reef. Exact definitions are very important.

As the complexity and the interrelationships among metadata grow, new problems arise relative to meaning and classification of concepts. Metadata that were collected and tailor-made for one purpose can be put in a new perspective as science progresses, and its meanings or demands may shift to unforeseen perspectives. This is a general problem in computer science and in recent years the increased need for efficient information retrieval, classification and linking created a new scientific specialization: that of ontology. In computer science¹, an ontology is a formal (accepted, standardized) representation of a set of concepts, their meaning, associations, relationships, and hierarchical classification from general concepts to more specific ones. Ontologies can be used equally by humans and computer programs for searching, description, classification, and definition of relationships.

A list of ontologies relevant in the field of biology and bioinformatics can be found on the web². These can be as diverse as ‘mouse pathology’, ‘plant anatomy’ or ‘cell cycle ontology’. In each of these projects relevant terms are listed with a definition and a number. In fact, only a few of the current ontologies are relevant to microbiology, for instance pathway ontology, human diseases, separation methods, multiple alignment, proteome binders, and environmental ontology, but this is an expanding field so more relevant ontologies are likely to be added in the near future. Ontologies will enable the use of more standardized terminology in metadata.

A strong effort to standardize metadata and harmonize ontologies for sequence handling is currently carried out by the Genomic Standards Consortium³. This consortium aims at developing a community-driven standard for metadata describing a sequence. The result of this effort is the ‘Minimum Information about a Genome Sequence / Metagenomic sample’ (MIGS/MIMS) specification, a checklist document aimed at finding and describing the minimal information required to describe a sequence coming from an organism, virus, organelle or metagenomic sample. These specifications are currently in course of implementation through the file format Genomic Contextual Data Markup Language (GCDML).

¹ The term ontology is also used in metaphysics, a specific field of philosophy that deals with the nature of being.
² http://www.ebi.ac.uk/ontology-lookup/ontologyList.do
³ http://gensc.org
A look at the most common bioinformatic procedures

The most frequently performed computer tasks in bacterial genomics / bioinformatics is to extract and manipulate data, metadata and derived data that are ultimately associated to DNA or protein sequences. Such activities can lead to new hypotheses to be tested either in the ‘wet lab’ or in silico. New discoveries can increase our knowledge on the original data and will produce novel metadata or primary data. The various steps in this process can be summarized as the identification and extraction of relevant data from data sources, the analysis (modify, enrich, elaborate, extend) of the data using computer programs, the representation and interpretation of the results (for which human insight is still essential and some biological knowledge quite helpful), and eventually storage of the results for future use. This may sound familiar, as wet lab experiments follow similar patterns: the process sums up some of the core activities of any scientific investigation. Working in a computational environment does not make science more ‘precise’, less (or more) difficult, or less error prone than any other laboratory environment.

Data extraction for computational analysis

How to identify and locate the information and data you want to work with has already been described in the previous chapter, and in the following chapters more resources will be introduced. As there are few general rules that apply to data identification (it all depends on what question you want to address) we will not ponder this any further and move on to data extraction. Data are not floating in space, they are held in some kind of data container from which they have to be extracted. An example of a data container is a text file: a computer file containing written text that represents, for example, a biological sequence as a string of letters. Other information may also be present in the file as metadata, but this is normally separated in a particular way from the primary data. A text file generally uses the suffix ‘.txt’, though text files with a specific content and data organization can be given a different suffix. For instance, FASTA files are text files with suffix ‘.fsa’, and GenBank files usually have suffix ‘.gbk’ or simply ‘.gb’. A Word™ document (with ‘.doc’) is not a text file: it contains additional information that tells your computer things about fonts, page setup, formatting, etc. in a way that is unintelligible to human users, but is interpreted by the program to display a readable document. When you load such a file into a program that can only read text, this ‘hidden information’ is wrongly represented as text and the data (your sequence) gets lost in gobbledygook. In Word™ you can save a file as ‘plain text’ and the computer will warn that in that case any formatting will be lost.
Text files are used a lot in our field, as accessing information from a text file is generally easy and the human user (the computer programmer) can read the information as clearly as the machine user (the computer program). For example, we could use a text file to store a list of accession numbers, with one number per line. Next, we can feed this file into a program designed to recognize these numbers as accession numbers, retrieve the corresponding sequences from a predefined database and perform calculations on them. This enables us to work with lists of genome sequences (millions of nucleotides) on less than a page of text of real input.

A slightly more complex case is a simplified text-only version of a table. Such a text file would contain multiple information in multiple rows, each separated by some delimiter, for example a comma. Now the human user or the computer program needs to read each row and extract the fields separated by the comma. This is known as a CSV file, or Comma Separated Value file, for which a simple example is shown in Figure 5.1.

<table>
<thead>
<tr>
<th>#accession</th>
<th>ATCONTENT</th>
<th>ORGANISM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL732656</td>
<td>0.6437</td>
<td>S. agalactiae NEM316</td>
</tr>
<tr>
<td>AP006716</td>
<td>0.6721</td>
<td>S. haemolyticus JCSC1435</td>
</tr>
<tr>
<td>CP000255</td>
<td>0.6725</td>
<td>S. aureus subsp. aureus USA300_FPR3757</td>
</tr>
<tr>
<td>BA000018</td>
<td>0.6716</td>
<td>S. aureus subsp. aureus N315</td>
</tr>
<tr>
<td>BA000033</td>
<td>0.6717</td>
<td>S. aureus subsp. aureus MW2</td>
</tr>
<tr>
<td>BX571857</td>
<td>0.6715</td>
<td>S. aureus subsp. aureus MSSA476</td>
</tr>
</tbody>
</table>

Figure 5.1. An example of a comma separated value (CSV) file. The first line is a comment, as it starts with a conventional hash (#) symbol. In this case, the line is used to define the column metadata. The comma is used to separate three fields in this example, listing accession number, AT content and the organism.

The simplicity of plain text files is both a blessing and a curse. For instance, hard returns can cause problems for particular computer programs, whereas the human reader has difficulty with continuous text (try reading this chapter if there were no paragraph divisions). A computer will wrongly interpret a CSV file if a field contains an erroneous comma, which is unproblematic for the human user. Thus, the human user and the computer have different difficulties with particular file formats.

For human use the rule is the more variation allowed, the easier information can be read, but for the computer it is the other way round: the stricter the rules are for file format organization, the better. In that case it is easier to write a robust computer program that is able to interpret the file and extract data from it, an operation known as parsing. The program, or part of the program doing this operation is called the parser, reader, or importer of the file format. Of course, stricter and easier to manipulate rules normally lead to limited format capabilities.
Data extraction programs

In Chapter 4 it was mentioned that a specialized program handling access to data is sometimes called ‘database’, although a proper naming would be DataBase Management System, or DBMS. Such programs almost always use a relational model, and they are therefore called relational databases. Well-known examples are MySQL, PostgreSQL, Microsoft Access, or Oracle. A relational database basically behaves like a special data container that allows storage of multiple tables, each one organized in rows and columns.

At first glance, this is similar to CSV files, but relational databases have a vast number of advantages over simple CSV files: (i) They allow highly efficient retrieval of information through a special language called Structured Query Language, or SQL. (ii) Relational databases allow enforcement of constraints, thus enabling, for instance, to specify that a column must contain only integer values, and no duplicate values can occur in that complete column. (iii) Relational databases enable concurrent access by multiple data readers and writers as well as multiple transactions (allowing to perform multiple changes as a single operation, and not as independent steps, so to perform all of them or none at all if an error occurs). (iv) They provide relations between data, enforcing that some data cannot be present in absence of some other data, or allowing to create a new table from two other tables through some relation rule among the rows.

Relational databases are very powerful tools and their usage is sometimes mandatory when complex data manipulation and efficient retrieval has to be performed. Although the database is physically stored in files, you never read the information directly from these files as all communication is through the database management system, which interprets and performs your SQL requests.

Another popular mode to provide access to data is through a web page, to be viewed through a browser. The bioinformatic databases available on the web, and described in this book, assume a web access through a browser, like Firefox, Safari, Internet Explorer, Opera and the like. The advantages of web-based databases and analysis tools cannot be overemphasized but they have their specific problems. In fact, as will be discussed further down it is hard to extract information from a web page in an automated manner.

A recent, alternative method of data extraction that gains popularity rapidly is a so-called Web-Service. A Web-Service provides data in a similar way as a web page, but in this case the ‘page’ is structured and designed for use by a computer, and not by a human. This solves many of the issues in extracting data from a classical web page, like the need of interactive operations (clicking, filling in blank fields, etc.). A more detailed overview of Web-Services is provided in the technical part of this chapter.
**Data analysis: where the action is**

Data analysis is where the real application of a method takes place. The method of analysis can be home developed or obtained from an external source, academic or commercial. The choice of the method will largely dictate the quality of the result (both in terms of what you get, and how good the result is), and most of this book is dedicated to give examples of the sort of methods that are currently available to explore and compare microbial genomes. Here we take a look at the actual process of how an analysis is performed, while ignoring the purpose of analysis, the method itself, and the input and output of the data.

Most analyses work as an *execution pipeline*. Such a pipeline represents a network of interconnected steps, each performing a small task. In our case, each of these tasks performs a calculation on a set of given input channels, and provides a result in one or more output channels. The task depends on the availability of input channels, which are either provided by a task upstream in the network pipeline, or by the user itself. Various types of execution pipelines are drawn schematically in Figure 5.2. The scheme to the left represents a linear succession of processing steps. The pipeline is serial, and each process cannot start before the previous one is completed. The scheme in the middle allows executing two steps simultaneously by different computational units. This of course has the advantage of reducing the total execution time as perceived by the user, as long as there are no other critical bottlenecks that introduce a rate-limiting step. The latter would happen, for example, when two processors performing simultaneous calculations need to get data from a shared, highly accessed hard disk that slows down the complete process. To give another example, if communication between processes is needed during the computation (not unusual when a large problem is fragmented in smaller units that interact in some way) synchronization could slow down if one processor is slower than the others.

Fortunately, many of the bioinformatic tasks can make use of the strategy depicted to the right of Figure 5.2, which is a *trivially parallel* scheme. In this case, each task can be assigned to a different computational unit, and is computed independently from other tasks from start to finish. This scenario allows strong parallelization, because it involves no communication and no dependencies between the computational units during the calculations. An example of trivial parallelization would be a search of different proteins for BLAST matches. Each sequence can be computed independently, and assigned to a different computational unit.
Figure 5.2. Three schemes of different execution pipelines. To the left a linear succession of tasks is depicted. In the middle, processes A and B can be carried out simultaneously. The scheme on the right represents an example of a trivially parallel procedure.

Data presentation and storage

Once the data are analyzed and processed, the results will have to be presented and visualized in some way. The type of plot should be carefully chosen to represent the results in the optimal manner, dictated by the features that have to be made evident. There are many alternatives for graphical representation and the choice can depend on personal preference. However, always bear in mind whether the key findings are obvious from a figure. Compare, for instance, the two different graphical representations of the same data in Figure 5.3. Both panels show the average AT content in bacterial genomes, divided by phylum. To the left, the radius of the circle is proportional to the amount. To the right, a bar plot reports the same information using a color key to differentiate the phyla. Note that this version is easier to grip.
Figure 5.3. Two different visualization methods representing the AT content of bacterial genomes per phylum. To the left a circle radius plot makes it more difficult to spot the proportionality of the values. A better representation is to use boxes of proportional size.

Sometimes, a careful layout of the represented data can provide added information. In the left of Figure 5.4, the frequency of codon usage of a bacterial genome is displayed by means of a rose plot. The panel on the right displays the same information, but codon position has been ordered in a particular way: codons sharing the third base are in the same quadrant. In this case codons ending in A or U are in the left quadrants and codons ending in C or G in the right; third-base purines (A or G) are at the top quadrants and third-base pyrimidines (C or U) at the bottom. Due to the degeneracy of the genetic code, AT-rich organisms tend to use codons with A or T (therefore U in RNA) as the third-base more frequently.

Figure 5.4. Two versions of a rose plot representation of codon usage. The left picture does not convey much information, except the actual distribution of each codon. The same data represented in a different way, to the right, reveal a preference for A or U at the third-base position.

This is now visible in the rose plot as a skew of peaks pointing to the left. A preference for purines or pyrimidines in third-base positions can also be visualized this way. Such a plot “explains itself” and tells a more complete story than the version to the left.
Colors are also useful to add meaningful information, and the choice of colors should be given careful consideration, not only for aesthetic reasons but also to increase or decrease particular emphasis. The consistent use of standard colors in particular context is also helpful. In Figure 5.4, sectors of the rose plot on the right are colored according to the standard base coloring used in sequence logos (see for instance Figure 2.7). Figure 5.5 further illustrates the benefit of introducing colors in a figure. Here, two distributions are represented in a scatter plot, showing a correlation (or lack thereof) between AT content and DNA size.

![Figure 5.5](image)

**Figure 5.5.** A scatter plot of DNA AT content versus length without and with the use of different colors. In both panels the entries for plasmids and chromosomes are distinguished by different symbols. Adding colorization, as in the right panel illustrates more clearly that the two populations are separated by size rather than AT content, and that plasmids can be as large as chromosomes.

Two populations are compared: bacterial chromosomes and plasmids. In the left panel of Figure 5.5, these two populations are indicated by the use of different symbols. Introducing different colors as well, as in the right panel, improves the figure significantly. Now it is obvious that there are a few ‘long’ plasmids mixed with ‘short’ chromosomes.

Another clear example of how important the choice of color use can be is given in Figure 5.6. The figure represents two versions of a Base Atlas (already introduced in Chapter 3) presenting the same information with different color scales. In the top panel, the wrong choice of colorization and the uniformity of the scale along all the datasets make it hard to extract any information from the figure. By using coloring scales that deviate from a neutral grey color for the most expected value and increase in intensity as values depart from that value, a bleaker version of the figure is produced but it is much easier to interpret. Adding different colors for each circle further improves the figure. The resulting atlas at the bottom is far more communicative and readable.
Figure 5.6. Two Base Atlases using the same genome, but displayed using different color scales. The difficulty of extracting meaningful information from the top atlas is evident. A proper choice of coloring produces a far more meaningful chromatic plot.
Many different programs are available to perform data visualization, either specifically dedicated for that task or embedded in the computer program used for data manipulation and computation. The statistical package R, for example, has a wide amount of plot styles, which can be extended to fulfill specific visualization needs. Matlab, Mathematica, Origin and Excel are also frequently used.

In some cases, cosmetic or graphic manipulation of the graphic is needed before publication. It is important to differentiate between vector graphics (like PostScript and PDF) and raster graphics (JPG, GIF and the like). A vector format is produced by mathematical representation of lines, points, curves and so on, while raster graphics are based on a matrix of color values best known as pixels. Some raster formats use techniques to reduce file size at the price of a slightly degraded image. This degradation sums up every time the file is converted or modified, potentially reaching unacceptable levels. Moreover, strongly enlarging a raster graphic will show the individual pixels (think of a large magnification of a newspaper photograph), whereas vector images can instead be rescaled and manipulated without loss of quality. Almost all plotting programs allow to save files in both vector and raster file formats, and both commercial and open source programs exist to edit these further, in case more manipulation is required.

Apart from proper visualization, the choice for optimal data storage should be considered. Storage is required for future reference, archiving and manipulation; data are stored as files whose format is primarily dictated by the program that generated them. Some programs produce proprietary, non-publicly available file formats, instead of standard, openly available files. Using such a proprietary format can introduce a dependency on particular software to manipulate and access the data, something making future retrieval of information difficult and complicating interoperability between different software. An advantage of application-specific file formats, though, is that they are highly optimized for the application that created them. Enhanced features or better performance could be a tradeoff that is worth the choice; the decision is best made on a case-by-case basis.

Another possibility is to store the data into a relational database, in particular when they are metadata. This is generally achieved through proper SQL statements. Storing the data requires the creation of a table layout with appropriate columns and defining the value types they are supposed to hold (integers, floating point, text, etc.). Once the table is created, the insertion or removal of data is done row by row. The database management system reports or rejects the insertion of rows that violate constraints, such as inserting text in a column supposed to accept floating points.
Achieving better automation

Let’s suppose for a particular research topic we need to collect all accession numbers of the sequences available today in GenBank. A person given this task could browse the NCBI site and extract all the accession numbers by tediously copying and pasting them into a text file. Obviously the repetitive nature of the task and the simplicity of it is an open invitation for automation. Although this appears to be simple, it is not as easy as it seems.

First we have to understand what happens when we access a web page. When we type a web address (the URL) in the browser bar and press enter, the browser issues a request for a web page to a remote computer. This request is received, processed, and if the web page is found, its content is returned to the browser. What the computer receives is basically a simple text file, and the browser converts this text into the web site that we see. Besides the content we read, this text file contains markup tags that describe the layout of the page, in the HTML language that is explained in the second part of this chapter. These markups describe, for example, that there is a list of entities that should be numbered (or not), that a particular word should be a hyperlink, that a given set of information must be laid out as a table, etc.

The web pages of NCBI give accession numbers of genes in an unnumbered list, with each line starting with the accession number followed by descriptive text. The HTML file of that web page instructs your browser to display the content this way. The user knows where to read the accession numbers even if there is no explanation given, and even if the descriptive text following each accession number contains numbers as well as text (like in strain names). For a computer, however, the numbers and descriptive texts are not meaningful. They are just symbols that are displayed in a certain way, dictated by the text file of the website that your browser has interpreted. Retrieving a web page from NCBI thus provides us (computer users) with a table containing accession numbers, but the computer received a text file, which the browser interpreted (parses) in the visual object we see. To obtain a list of accession numbers automatically, we could download the web page text file, write a program that interprets that text as the browser does, and knowing that the accession number is, for example, the first entry of each new paragraph with two letters and a 6-digit number, one can collect these. However, we have already seen in the previous chapter that the format of accession numbers can vary. If we tell the machine to only extract ‘words’ with 2 letters followed by 6 numbers, displayed at the beginning of each paragraph, we could miss some, or collect other non meaningful data that happens to look like an accession number.

Another problem arises from the fact that the HTML format is very loose, and collecting information in a reliable way is often very difficult: a small, or even
large change can have little or no effect on the visual perception of the result, but can break any program we developed to extract the information from a now changed file. The file could even contain errors or inconsistencies (browsers have huge amounts of code to deal with these errors properly) and still visually appear fine, but it could be dramatically time-consuming to deal with these errors in bioinformatic research.

As if all these complications were not enough, we can add the fact that a browser can display things that were not even in the text file of that web page. The browser can add, remove, or manipulate information with a programmed operation implemented in the JavaScript language (further explained in the second part of this chapter). JavaScript code is executed in response to user events, like clicking on a button or moving over a field with the mouse. The code can then fetch data from a remote server and display them on the fly. You can imagine how difficult, if not impossible, it is to automatically extract the data from such a web page. Web pages are designed for human use but obviously providing information that is easily digested by a computer has different requirements. The program we have hypothetically written to extract accession numbers from GenBank is fragile, meaning that its performance is highly dependent on the web page structure both as visual layout and internal HTML description. Even a small change like an additional paragraph break can confuse a fragilely programmed data extractor.

A further issue we encounter with our hypothetical accession number extraction program is that we are downloading far more data than we actually need. The web page contains huge amounts of information that concern layout, and downloading this takes time (and bandwidth on our connectivity cables), whereas all we need is the accession numbers. Sometimes automation is not the solution, and if you know the administrator of the remote site, it would be easier to ask if they can provide a file containing just the accession numbers. That means you let them do the work, and you are dependent on others, at a cost of loosing control. In case, for instance, you decided halfway the project that you really only need accession numbers of DNA from bacteria and viruses, but no others, the obtained list (without additional information!) is now useless. Obviously, a better strategy is required.

The future direction for solving this issue, among many other issues, is provided by a Service Oriented Architecture, or SOA. The idea is to rationalize and organize entities known as services, and let these services collaborate through proper orchestration in order to achieve a more complex goal. Each service provides a task-achieving unit, and it can live either on your machine or, more usually, on some provider server. Details and implementations of SOA will be explained in the technical part of the chapter.
Implementing scalable processes

In the previous chapter it was mentioned that the amount of sequences we are dealing with is rapidly increasing. This can cause scalability problems: how well or badly can a performance be scaled up to a larger dataset? As the amount of data increases, old solutions must be revised or new ones must be implemented to reduce the time, computer memory, or human intervention requirements to perform a given task, without increasing the chance of errors.

Computers do not make mistakes, but computer programs are the products of human activity, and so is their use. As in the lab, the human factor is a critical source of error. Humans are neither efficient nor accurate at performing repetitive tasks: distracting events or boredom can lead to overlooking a detail and this can introduce errors. In the lab, automation (when affordable), robotization and standardization (the use of kits) can reduce the risk of errors. Similarly, in the field of computational biology, the use of programming languages and computational tools lend themselves to a degree of automation and standardization to reduce the risk of errors. Automation is absolutely required for high throughput analyses where the vast quantity of data to manipulate would simply be too big for manual handling. A lot of research is dedicated to develop scalable, automatic solutions to manipulate sequence data.

Part 2: Some technical details of the tools and future directions

The following part of the chapter will present some of the tools available and used in comparative genomics research. The list is not intended to be exhaustive, as that would fill the rest of the book. New tools are released and developed constantly, so that any coverage cannot be complete. Here we will also go into more detail about the current trends to solve the needs for better scalability and automation: Service Oriented Architecture, Web-Services and SOAP.

Programming languages

The rules by which a computer must perform the tasks we demand are called the algorithm. An algorithm is a ‘recipe’ to produce a result, and is provided to the calculator by means of a programming language. The calculator can interpret the program written in this language to produce our scientific result. How the scientific idea is converted into an algorithm, and how this algorithm is coded into a program is a skill that requires experience, creativity and accuracy. The choice of
a particular programming language is influenced by a compromise between previous code, personal taste, performance constraints, vendor neutrality, availability of prewritten libraries, and a little dose of hype. As a result the research tools we know today are highly heterogeneous.

Computer languages come in two kinds: interpreted languages and compiled languages. Interpreted languages are suitable for applications where code is needed as glue between other programs, for web programming, and any context where performance is not a critical argument when compared to speed of development. A program written in an interpreted language is executed by means of an interpreter, which is directed by the program into executing statements. The interpreter must always be present on the machine to run these programs.

When performance is critical, a compiled language is preferred. In that case a compiler reads the code as a whole, and produces once and for all an optimized executable program that can run independently, but only on a specific computer architecture (processor and operating system). The line between interpreted and compiled languages became blurred over time: modern day interpreted languages perform a preliminary compilation step on the program to produce a more efficient representation, and then interpret this representation. Some compiled languages, such as Java and C#, compile an executable for a ‘virtual machine’, basically an interpreter that hides the different internal mechanisms of different processors and operating systems. This allows good performance and portability at the cost of depending on this virtual machine to be installed. Box 5.1 summarizes the languages most commonly used in our field.

People who speak more than one foreign language can confirm that after you’ve learned a few languages, the next one is no longer as difficult, as long as it belongs to the same family of languages. With computer languages the effect is even stronger: knowledge of several programming languages not only eases the task of learning a new one, but knowing several languages makes programming in a particular language of choice easier. In many cases, a broad portfolio of languages allows to tackle a problem with a different, more powerful style or technique. Of course, if you don’t practice a language for some time you tend to become less fluent, so a refresh is sometimes needed. It is generally advantageous to be able to program in various languages, so that it becomes easier to handle previously existing code, and an optimal choice can be made for new tasks. A more difficult step is to learn the tools, libraries, programming style and typical solutions (design patterns) of a given language. This proficiency distinguishes the occasional user from the professional.
Box 5.1: Computer languages commonly used in bioinformatics

Interpreted languages

Perl: probably the most frequently used in bioinformatics. Its points of strength are the ease of processing and extracting text from files, and the large, freely available collection of functions (libraries) able to perform and solve frequently encountered problems. Its unfavorable points are the awkward syntax and the language design that does not encourage good programming practices and code cleanliness.

Python: less frequently used as it is still relatively young compared to Perl. Its main strength is a structure that encourages (and in some cases enforces) good programming. It is very expressive and compact, meaning that fewer lines of code are needed to achieve a given result. Its lower popularity is currently a disadvantage, but its use is increasing at a rapid pace.

Shell scripts (bash, tcsh): the UNIX command interpreter can be used as a programming language, in particular when the task to achieve involves executing programs and collecting their output into files. Sometimes a simple shell script is a solution that fits many problems, but in general it should not be used for complex tasks.

PHP: mostly dedicated for web development, PHP allows to create programmatically a web page. It is an optimal choice to extract data from a data repository, generally an SQL database, and present it through a web interface.

Ruby: a very young language with a syntax style resembling a mix of perl and python. It became en vogue in web application development thanks to the very powerful ‘Ruby on Rails’ framework.

JavaScript: a language interpreted by web browsers. Normally, the code is written into the web page, although for large and complex cases the code is provided in a separate file that is referred by the page and downloaded by the browser automatically. Despite the name, JavaScript has no connections with another popular language, Java, except for a vague similarity in syntax. It is fundamental for creating interactive web pages.

R: a very powerful language based on the S language developed at Bell labs. It is tailored to scientific and statistical analysis, and can produce high quality graphs. This puts R in a separate category with respect to the previous more general-purpose languages. Many of the images presented in this book have been created with R.

Compiled languages

C and C++: quite complex languages to use and master due to their proximity to the raw details of the computer internals. They can deliver strong performance boosts when number crunching is needed.

Fortran: historically one of the first languages, it is quite simple to learn, but lacks flexibility and doesn’t have a good set of libraries for common non-numerical tasks. For brute force performance it is difficult to beat: the compilers and libraries have been strongly optimized as a consequence of its age and the very specific market.

Java: similar in syntax to C++. As stated earlier, Java behaves a little like an interpreted language. Programs are compiled into a sort of intermediate entity that is then interpreted by a ‘virtual machine’. This allows, at least ideally, running the same compiled program on different architectures by using the proper virtual machine. Java is widely used for large and complex software entities, and many enterprise level systems use it.

C# (C sharp): similar to Java, but produced by Microsoft and basically tailored to its operating system. It is still very young, and not widely used in the bioinformatic community.
Many of the programming languages summarized in Box 5.1 are ‘object oriented’ which means that data and the algorithms that work on them are contained into a single entity, called object, which can be manipulated in our program. This allows, among other things, to distinguish between the interface (external behavior) and implementation (what happens under the hood). In contrast, a procedural language (such as C and Fortran) has subroutines and data as separated entities. With an object oriented programming language, you could have for example an object representing the sequence. The advantage is that at the interface you can perform a task (ask its length, or the amino acid at position 34) without being concerned about implementation (e.g., how this sequence is stored internally: it could be in memory, on the disk, in a SQL database, or on a remote computer). If one day a better storage strategy is required, this can be done internally while all the code using the external interface will continue to work flawlessly.

**Markup languages**

**Markup languages** are a different category of languages, as they are not used to express an algorithm. Instead, they express structural or layout information relative to data. A programming language can read and interpret these directives so that data are presented in a user-friendly way, or elaborated according to their structure. The best known markup language is HTML (HyperText Markup Language), the language in which web pages are written. An example of HTML is provided in Figure 5.7, which in this case instructs the browser to draw a paragraph of text and a table made of two rows and two columns. HTML markup uses tags, small code words in between the symbols < >. Examples of tags are <BODY>, <HEAD>, <TABLE> and <TR>. Inside a tag, attributes can be specified to give contextual information about the tag, in between the symbols " ". The border="1" attribute specifies that the table must be drawn with a border with thickness 1. Some tags are closed by means of an ‘end of’ closing tag, using the symbol </>, for example </TABLE> and </BODY>.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type"
 content="text/html; charset=iso-8859-1">
<title>An example of HTML document</title>
</head>
<body>
<p>A paragraph</p>
<table border="1">
<tr><td>row 1, col 1</td><td>row 1, col 2</td></tr>
<tr><td>row 2, col 1</td><td>row 2, col 2</td></tr>
</table>
</body>
</html>
```

![Figure 5.7. An example of HTML and its representation in a web browser.](image)
As the need for clear markup languages grew, a standardized specification was developed, called XML (eXtensible Markup Language), which basically sets the rules for XML-compliant markup languages. When you apply these rules while creating a markup language, what you obtain is an XML document. Visually, it would appear similar to HTML, but it has stricter syntax requirements (for example, every opened tag must have a corresponding closed tag). The advantage of XML documents is that they are generally applicable, whereas HTML is tailor-made for web page description.⁴

XML documents also allow expressing more complex semantics. Tags and attributes can be chosen arbitrarily to give more diverse meaning to the tags and attributes themselves and their contents. A hierarchical, tree-like structure is easily representable, allowing to define containment relationships. For example, we can express a list of proteins with an XML-based document as in Figure 5.8. The flexibility of XML allows to develop a tailored format to contain any data and its meaning. XML allows us to create the so-called SVG format to describe geometrical shapes, MathML for mathematical formulas, CML for chemical processes and molecular geometries, and so on. Two more markup languages, relevant for our work, will be introduced below: SOAP and WSDL.

```
<?xml version="1.0" encoding="UTF-8"?>
<proteinList>
  <protein>
    <name>AA</name>
    <sequence>....</sequence>
  </protein>
  <protein>
    <name>BB</name>
    <sequence>....</sequence>
  </protein>
</proteinList>
```

Figure 5.8. A simple example of XML. As it can be appreciated, the tags are arbitrary, and reflect the concepts and relationships we expect in our data. A browser cannot represent this document as it normally does with a web page.

**Service Oriented Architecture**

The Service Oriented Architecture (SOA) provides a way to improve automation of repetitive but more complex tasks. The SOA allows implementation of important properties such as reusability, loose coupling, and separation of concerns.

⁴ Note that HTML documents are not XML documents, due to the stricter rules XML has. An XML compliant version of HTML exists, and is called XHTML. Browsers interpret and display both HTML and XHTML.
Reusability means that a particular service can be reused in a different context just by orchestrating it in a different way. Loose coupling means that the services depend on each other at the minimum level achievable, and that they have the maximum freedom for internal variation as long as their interaction behavior remains unchanged. With a loosely coupled architecture, a service can change (for example, internal code can be refined to improve performance, or rewritten completely in a different programming language) without forcing adaptation changes in another service. Separation of concerns is a consequence of loose coupling, and means that each service can be created and managed under the responsibility and the competence of any given expert in the field of the service. External users of the service are not concerned about and do not need strong competence in the service internals. They just need to provide given service regardless of the subtleties of the service itself.

A real-world example can easily clarify the concept. Suppose a service provides us with sequences of bacteria living in a particular environment. Another service allows us to perform BLAST searches on sequences. These two services have an independent identity, and they are, by default, not aware of each other. Nevertheless, they can be orchestrated into communication, so to perform a specific task like searching for proteins in soil bacteria. Note that the internal details of how these two services work are not important to get the result. Instead, interoperability is allowed by (and depends on) standard communication protocols and clearly defined behavior (interfaces). Obviously, the complexity of demanded tasks can grow by adding more services and by designing a more complex communication pattern, while the programmer designing these doesn’t have to be fully informed on the details of each individual service. Proper orchestration of loosely coupled, independent services out of a potentially unlimited palette can lead to new discoveries in a relatively short time. Each of these services can be on your personal computer, but more likely they reside on different laboratories’ hardware, and you can use their resources transparently as if they were on your computer. SOA allows different technologies, languages, and machine architectures to communicate and interact with ease.

**Web-Services and SOAP**

It is evident that SOA requires a standard way to communicate among services. The communication channel must at least allow to transport requests and responses among the services. In this scenario, Web-Services and SOAP comes into the scene. SOAP is one of many protocols designed to enable communication and data sharing between data providers and data consumers. While the idea behind Service Oriented Architecture is not novel, SOAP allows strong interoperability between different vendors, and use the web infrastructure as the main point of ref-
erence for deployment, by means of standard, open protocols. Communication with SOAP involves exchanging messages represented as XML documents, which generally travel on the same channel that web pages use, the HTTP protocol. Thus a SOAP message can be seen as a web page meant for machine consumption, instead of user consumption. It enables data transfer between a sender and a receiver.

This paradigm can be used to implement technologies in order to interact programmatically with a remote server as if it was a local entity. All the communication happens ‘under the hood’ to our advantage. The remote server exports functionalities and publishes these by means of a Web-Service interface. The description of the functions, and the data types that are accepted and returned by the Web-Service routines is described in a particular file, called a WSDL file (again, XML-based). While SOAP enables requesting a service to perform a particular task for us, WSDL informs us which tasks can be performed by the service, what information is needed to perform a correct request, and what information will be returned once the task is completed.

We can explain the meaning of these concepts by analogy to a restaurant. If the service were equivalent of the chef, SOAP+HTTP would be the waiter and WSDL the menu. To be more accurate, we could say that SOAP is the paper slip with the order, and HTTP is the waiter carrying the order to the chef.

By using services, we can achieve improved productivity, freedom for experimentation, and a distributed architecture that allows partitioning the computational cost of research onto multiple institutions. The addition of more and more services to the available palette further smoothen the way for scientific progress.

**Specific tools for bioinformatic use**

As the impact, complexity and needs of bioinformatics increased, particular tools have been developed to fulfill the needs of this field of research. Here, we will only mention a few examples.

Libraries are collections of frequently used algorithmic and data functionalities. Software development is an expensive task, and being able to reuse frequently needed functionalities is most resource efficient. The BioPerl and BioPython libraries are tailored toward common tasks in bioinformatics. They describe routine tasks such as obtaining sequences from databases, reading file formats such as FASTA, converting sequences from DNA to protein, performing BLAST searches and parsing the result, etc. For the R language, the Bioconductor package provides a powerful set of functionalities, in particular for microarray data analysis. A large collection of statistical methods, file format readers and writers, and additional
plot styles for the R language can be found by browsing the Comprehensive R Archive Network website\(^5\).

As execution pipelines are frequently used strategies to perform computations, pipeline managers are fundamental tools in the bioinformatician toolbox. A very simple utility to manage pipelines on any UNIX system is make. It is traditionally used to compile program source codes into actual programs. As the pipeline is expressed in a user-friendly, text-based format, it is easy to learn. However make has a number of disadvantages: it lacks a graphical interface, it does not interact with databases but just with files, it is difficult to administer complex pipelines with it, and sometimes it produces deceiving error messages. Attempts to solve these issues led to the creation of alternative tools, some of them specifically designed for our field. An example is BioMake, whose slightly more complex syntax is a small price to pay for its more flexible use. Amongst other advantages, BioMake allows direct interaction with databases. Another useful tool is Taverna, which provides a graphical environment to design and execute workflows. It is particularly aimed at Web-Services orchestration.

For users who prefer integrated environments, rather than a compiled collection of single-step tools, Bioclipse is a good choice. It is a very powerful integrated environment that allows performing data manipulations, computation and visualization all within one program.

**Concluding remarks**

This chapter provided a very brief description of the broad palette of tools and concepts behind bioinformatics research. We focused on tools and concepts that can resist the test of time, either as they have already been around for quite some time and are now well established, or, though still young, they are very promising technologies. The real potential of these technologies is still to be unleashed. When this will happen, a new kind of research will be possible, and new specific competence will be needed to handle these new instruments. We are witnessing the dawn of real, distributed computing facilities that span institutions, continents and research fields.

---

\(^5\)http://cran.r-project.org
6 Methods to Compare Genomes: the First Examples

Outline Information stored in genome sequences can be analyzed at different levels, as demonstrated in this chapter. Features of a genome that can be obtained as a single numerical value, such as size or base composition, can be easily compared across large numbers of genomes using statistical methods. Graphical representation then becomes crucial. One can also compare two or a few (similar) bacterial chromosomes by direct alignment to identify regions of translocations, inversions and indels. Again visual representation of the findings is essential for interpretation. Before zooming in at the genes encoded on the genome, the quality of annotation needs to be assessed. Single genes extracted from multiple genomes provide a further level at which comparisons can be carried out. Regardless of the method used, visualization of the results by graphical representations is crucial to display and interpret the findings at a genomic level.
Introduction

There are many methods to compare bacterial genomes, and the optimal method will obviously depend on the question being asked. This chapter presents some examples of genome comparison methods and introduces the importance of graphical representation of the data. The given examples, applied to prokaryotic genomes, only represent a few of the many different methods available to compare microbial genomes. More comparative methods will be introduced in subsequent chapters.

Genomic comparisons: the size of a genome

One feature that is immediately apparent from a sequenced genome is its size, and to compare relative genome sizes is the first question we will address, as an example how to compare single numerical values. One complication arises, though, from the fact that database entries for a bacterial genome do not always identify that the organism might include plasmids or multiple chromosomes. We thus have to correct for this on a case-to-case basis through the Project ID, as explained in Chapter 4.

In order to compare genome size, their values can simply be listed in a sorted table, from which the average, range, and median values can be calculated. A graphical representation of this information is much clearer than tabulated lists. A box-and-whiskers plot allows to graphically represent the range, median and distribution information of numerical values. An example is shown in Figure 6.1 where genome size is given for bacterial and archaeal phyla. The median genome length of each phylum is represented by the solid line, the 25%-75% range are covered by the box, the total range is given by the thin line and any outliers are plotted as open circles. The text box at the end of this chapter explains how to produce a box-and-whiskers plot. This is an example of a fairly simple comparison of a single numerical variable, which can be visualized to represent the degree of variation. From the figure it is obvious that genome size varies across different phyla, and that the degree of variation within those phyla is not constant. For instance, the 14 sequenced δ-Proteobacteria vary less in genome size than the two sequenced Acidobacteria Genome size is most strongly conserved within the phylum of Chlamydiae and the γ-division of Proteobacteria. (Capitalized, non-italic names are used here to indicate bacterial phyla.) Any numerical value can be compared in this way, such as nucleotide composition, size of particular genes, gene density, fraction of secreted proteins, etc. For any parameter captured in a single numerical value, a box-and-whiskers plot is a suitable representation.
Figure 6.1. Length distribution of 621 prokaryotic genomes, divided by phyla. The complete genome size, including multiple chromosomes was analyzed. The median length (the value separating the data in two halves with equal numbers) per represented phylum is shown as a black bar. The box gives the range (25%-75%) spanning the second and third quartile, so that 50% of the data are contained within the colored box. The dotted lines are the ‘whiskers’, giving the range of the highest and lowest values. However, outliers (open circles) are not included in the whiskers. The top three entries represent archaean phyla, followed by bacteria. The colors vary per phylum. Note that the proteobacterial phylum is separated in the 5 divisions.

A word of caution - which database to use?

“What is the largest bacterial genome?” seems like a really simple question. However, its answer depends not only on when you ask the question, but also which database is used, and even how that database is analyzed. For example, at the time of writing this chapter, NCBI’s list of sequenced genomes (http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi) identified the largest sequenced bacterial genome in GenBank as Solibacter usitatus strain Ellin 6076 (a soil bacterium isolated from pasture land in Australia), at just below 10 Mbp long (9,965,640 bp; accession number CP000473). However, at the same time the list was downloaded from GenBank (which ideally should provide the same data) at ftp://ftp.

\[1\] Obviously, the question depends on which available data are analyzed, and as these vary over time, a scientific publication should state when which version of a database was used.
ncbi.nih.gov/genbank/genomes/Bacteria, but there the largest bacterial genome listed was *Sorangium cellulosum* (a myxobacterium producing low-molecular weight compounds with potential pharmaceutical activities), strain So ce 56, at 13 Mbp (13,033,779 bp; AM746676). Whichever the record species is, it is bound to be overtaken soon by the next largest genome sequenced.

By sorting the list of GenBank files, multiple chromosomes and plasmids would not be taken into account. One needs to group all the chromosomes from the same organism. This wouldn’t influence the record holder listed above, but the second largest on the NCBI pages is now the soil bacterium *Burkholderia xenovorans* LB400, at 9.8 Mbp, the sum of three different chromosomes (GenBank accession numbers CP0002710, CP0002711, and CP0002712 - confusingly, in our opinion, the GenBank number for chromosome 2 only is listed on the NCBI page).

Although the PID was specifically designed to overcome this problem, a close inspection of the list of sequenced genomes currently available in GenBank revealed that there are many segments that most likely belong to the same organism but do not have a PID assigned (yet). This unfortunate problem probably reflects earlier entries to GenBank prior to the PID era; to solve it demands manual curation at a time of exponentially growing databases. With a number of colleagues active in the field we have proposed in the future all newly sequenced genomes deposited to GenBank, EMBL and DDBJ be required to have a PID assigned to each accession number deposited (Field *et al.*, 2007). Bringing the current collection of genomes up to standard and enforcing quality control in the future will remain difficult but necessary tasks.

**Record holders: the largest and the smallest bacterial genome**

The urge to break records can result in fun comparisons at meetings, where speakers may claim to have sequenced the “largest” genome (or smallest, or most AT rich or whatever record holder), when in fact their records have already been overtaken. But in all fairness, it can be difficult to keep up with a moving target, especially when even the databases are having difficulties keeping up-to-date. At the time of writing, there is much discussion amongst the people in the field who work with sequenced microbial genomes about how to best address this problem. Ideally, there would be one central repository, which would be continually maintained and updated, where any person can obtain the most up-to-date information. The interested reader is referred to our supplemental web page accompanying this book², where we will keep updated information including any future initiatives on publicly available genome listings.

² [http://comparativemicrobial.com](http://comparativemicrobial.com)
The record holder of the smallest genome discovered so far was already introduced in Chapter 3. It is *Carsonella ruddii*, an insect endosymbiont, with a genome sizing a mere 160 Kbp, which is more than 80 times smaller than the largest bacterial genome presently known. There is some debate whether this symbiont should be considered a true ‘free-living’ microbe. Indeed, its genome is rather rudimentary and it is lacking many essential genes, so it cannot survive outside the host cell it is naturally occupying. This allows the view that it is an organelle ‘in the making’ (just as in a long past endosymbionts stood at the roots of mitochondria and chloroplasts). Following this argument, though, would also exclude other endosymbionts from the list, which have reduced genomes though not as small as *C. ruddii*. Clearly, the fewer genes an organism contains, the more it is dependent on its environment to provide essential nutrients.

Another approach to define the minimum genome size for a free-living organism is to build up a genome from scratch. That approach is currently experimentally followed in a project to produce an artificial life form, using synthetic biology. The genome is not completely made up, though, as the reduced genome of an endosymbiont is taken as a starting point, to which genes are added until the organism can grow independent of a living host cell. The smallest naturally free-living bacteria currently known have a genome size around 1.3 Mbp (e.g., *Pelagibacter ubique*), which is still about ten times smaller than *S. cellulosum*.

Genome size will be further analyzed in Chapter 7, where bacterial and eukaryotic genomes will be compared. As we will see there, bacterial genome size also weakly correlates with base content and with the niche in which the organism lives.

**Pairwise alignment of genomes**

If you only want to compare two or three similar genomes, sequence alignment (introduced in Chapter 2) is a standard procedure that can be used for both long and short sequences. However, since bacterial genomes are often millions of bp long, the alignment can no longer show the DNA sequences. Instead, a graphical representation of the alignment has to zoom out, loosing resolution but producing a global overview instead. Particular software provides zoomable images, where the resolution can be adjusted at wish. An example of an alignment of three genome sequences of *Mycoplasma hyopneumoniae* (causing pneumonia in swine) is shown in Figure 6.2.
Figure 6.2. Alignment of the genome of three *Mycoplasma hyopneumoniae* strains. Sequences that find a perfect match are connected with red lines or blocks. White represent indels and blue are transitions or inversions. The figure was produced using Artemis software.

In the graphical display of the alignment, conserved sequences appear in red. The differently colored lines connect sequences that are found in different locations in the two sequences. The hourglass-shaped region in the left half of the lower comparison represents a large inversion: the same stretch of sequence is shared by strain ATCC 25934 and by strain A232 but it is present on different strands. A few indels (insertions and deletions) appear as wedge-shaped gaps separating red blocks.

Obviously, chromosomes from organisms more distantly related will show less conservation in an alignment. Pairwise and multiple alignments (up to five sequences can be tested in the used tool) are feasible, but for higher numbers of chromosomes, such alignments become tedious and the results become different to display, analyze and interpret.

A graph as in Figure 6.2 is pretty, but how would this alignment be analyzed and interpreted? Can we state the ‘degree of similarity’ of two genomes by pairwise alignment? This is not that simple, as one needs to define first what is meant under ‘similarity’. For example, one could report the fraction of identical bases for the best alignment of the whole genome (which gives the percent identity) but that doesn’t do justice to the observed variation. And how does one choose the best alignment? How many gaps should one allow? Or should the largest perfect alignment be reported, not allowing gaps at all? How the observed differences should be interpreted also leaves room for disagreement. Are indels to be inter-
preted differently from translocations and inversions? In case of translocations and inversions, the actual sequences are present in both genomes, though not in the same location. Indels, on the other hand, are present/absent in the compared genomes. And are translocations and inversions to be reported separately (as they may be the result of different genetic processes)?

The answers to these questions are not easy and in our opinion the jury is still out. The ‘best’ method is not clear, and of course it also will depend on what insights are sought. These questions are currently the subject of research, as they can provide insights into genetic and evolutionary processes. Other features to analyze based on DNA sequences, irrespective of the coded genes, will be discussed in more details in Chapters 7 and 8.

**Comparing gene content and annotation quality**

An obvious parameter to compare between genomes is the number (and nature) of protein-coding genes, tRNAs, and rRNAs. From their numbers (and the genome size) the coding density can be calculated, which is the fraction of the DNA sequence in a genome that actually codes for genes. The protein coding density of bacterial genomes usually is fairly high, varying from 85% to 95%, although for some bacterial genomes it can be less than 50%.

The general view is that bacteria are ‘highly evolved’, and that there is a strong selective pressure to delete non-essential DNA; thus most of the genome codes for genes (DNA or protein-coding) and a small portion is used for regulatory signals. In this sense, bacteria run a pretty tight ship, in that nearly all of the DNA sequence in a genome contains information. Genome density in bacterial genomes is much higher than in eukaryotic genomes, as will be seen in the next chapter.

However, there are some bacterial genomes with a lower coding density than others, and this is considered reflective of genomes in the process of genome reduction. Such genomes are considered to be eroding as they are loosing genes, decreasing their coding density in the process. *Mycobacterium leprae* (the causative agent of leprosy, Cole et al., 2001) with 3.3 Mbp and *Sodalis glossinidius* (an endosymbiont of the tsetse fly, Toh et al., 2006) with 4.1 Mbp are the only examples, to our knowledge, with coding densities below 50%.

One would expect that the coding density would be relatively conserved for similar bacteria, say, between genomes of the same species. However, comparing the coding density for two similar bacteria sometimes brings surprises. This may not always reflect ‘real’ biological variation but rather can result from differences in the quality of genome annotation amongst different research groups sequencing and annotating the genomes. The quality can unfortunately vary considerably, re-
sulting in a mixture of confidence that can be placed on the annotations in a GenBank file. We will compare, as an example, two *Leptospira interrogans* genomes to illustrate the consequences.

In the case of *L. interrogans* (the causative agent of Weil’s disease), two serovars have been sequenced. Both genomes contain two chromosomes, and are quite similar in total size (4,627,366 bp combining the two chromosomes of serovar Copenhageni (Nascimento et al., 2004), and 4,691,184 bp for the serovar Lai genome (Ren et al., 2003)). The serovar Copenhageni strain contains 3658 annotated protein-encoding genes, whilst the serovar Lai strain is predicted to contain more than a thousand additional genes (4725). Despite this difference, the coding density ‘only’ varies with 4% (78% for Copenhageni and 74% for Lai). This can only be explained if the added genes (in the case of Lai) are relatively short. Closer examination of these two genomes reveals that indeed most of the extra genes in the serovar Lai strain are indeed short open reading frames. How can we determine if too many short open reading frames were incorrectly annotated as genes in the case of Lai, or incorrectly missed in the case of Copenhageni?

One way is to compare the length distribution of predicted proteins, and to look if the proportion of short genes is exceptional. In order to do this, we have developed a method (Skovgaard et al., 2001) to compare the length distribution of all predicted proteins within a genome with those matching SwissProt, to estimate how many genes would be expected in a given genome. A length distribution plot is shown in Figure 6.3 for the two *L. interrogans* genomes.

![Figure 6.3](image.png)

**Figure 6.3.** Comparison of protein length distribution of two different *Leptospira interrogans* genomes. The grey shaded area represents the total area under the curve for all proteins annotated in the genome sequence file. Most of the short open reading frames do not have a SwissProt match. The *L. interrogans* serovar Lai genome (left) has far more of such short open reading frames annotated than the serovar Copenhageni genome (right).
By looking at these distributions, an over-abundance of short genes is apparent that do not have a match in the SwissProt database. Notice that the peak of proteins less than 100 amino acids is much more prone in the serovar Lai genome, but greatly reduced in the serovar Copenhageni genome. This peak in short genes is suspicious, and often reflects poor quality in genome annotation. Such short ORFs can occur by chance and may not function as genes in the cell. Since there is a much larger chance of finding a short ORF than a longer one, most gene-finding programs will use a cut-off. Any ORFs shorter than this are treated as suspect, and more information is needed to consider them likely gene candidates, and not an artefact. It appears the cut-off used for annotating the *L. interrogans* Lai genome was chosen lower than it should have been. Of course, there are many small genes that are real and expressed in the cell, but predictions are likely to contain many short ‘false positives’. The wisdom is trying to get all the ‘right’ genes, and exclude the shorter random artefacts.

**RNA comparisons: a look at rRNAs**

Ribosomal RNA genes are of particular interest as they are one of the core determinants for taxonomic relationship of bacteria. The three genes encoding 16S, 23S, and 5S rRNA are usually present in one rRNA operon. Bacteria can have one or more of these operons, which are usually copies of each other. Ideally, a tree based on rRNA genes should resemble a taxonomic tree.

![Figure 6.4. Phylogenetic tree of sequenced ε-Proteobacterial genomes, based on rRNA genes. The newly sequenced *Campylobacter jejuni* surprisingly didn’t match up with the other *C. jejuni*.](image)
Figure 6.4 shows an example of a tree based on rRNA genes of related organisms that was found inconsistent with our expectations. In this case, we were examining the newly sequenced genome of a *Campylobacter jejuni* isolate. We were already concerned because there were far more genes in this genome (~2800 genes predicted) than expected (~1600 genes expected). Further, many of these genes didn’t resemble *C. jejuni* genes.

The tree shown in Figure 6.4, based on a single 16S rRNA gene, separates our *C. jejuni* as a complete outlier. Closer examination of our sequence (which at this stage was still in many pieces) identified two different 16S rRNA genes. One of them was clearly a *C. jejuni*, whilst the other 16S rRNA best matched another very distantly related organism (*Thermoanaerobacter*, a thermophile). We then asked the sequencing company whether they had recently sequenced a *Thermoanaerobacter* genome, and sure enough, they had. Since the thermophile would not have grown at the 42°C we used to culture *C. jejuni*, the apparent contamination had most likely occurred in the sequencing facilities. Fortunately the rRNA genes had given an early warning.

When considering rRNA content of published sequenced genomes, again, the quality of gene annotation needs to be considered. GenBank files frequently contain rRNA genes that are annotated too long (sometimes over a thousand bp too long!) or are positioned on the wrong strand. Worse, rRNA genes can be missing from the annotations altogether. Clearly, correct identification of the rRNA genes in a genome sequence should be a quality criterion for submission of any annotated genome sequence. The reason why rRNA annotation is not always correct is because traditional methods that work quite well for finding and aligning protein genes (*e.g.*, BLAST) do not perform nearly as well to identify rRNA and other non-translated RNA sequences. The program RNAmmer was specially developed for the identification of RNA genes (Lagesen *et al.*, 2007). Chapter 9 will deal with the variation in number of rRNA loci present per genome. In addition, tRNA and other non-protein encoding RNA genes are treated there.

**Proteome comparisons: what makes a family?**

Protein genes by far outnumber non-translated genes present on a genome, and their diversity is much broader, too. Proteins are responsible for much of the activity in the cell, and since their function is dictated by their 3-dimensional structure, analyses beyond sequence comparisons are required to fully explore the proteome. Having said that, much can also be gained from a simple comparison of the same protein across many bacterial genomes.
As an example, we will consider the well-conserved protein family of sigma factors, which are part of the RNA polymerase complex (introduced in Chapter 1). A sigma factor initiates transcription by binding to the promoter regions of genes. Most cells contain various sigma factors, each coded for by its own gene. There are two main families of sigma factors that are not genetically related: the Sigma 70 family and the Sigma 54 family. Whereas members of the latter are relatively strongly conserved (though not always present in all bacteria), the Sigma 70 family is more heterogeneous. It contains three functionally different subclasses: primary sigma factors, responsible for transcription of housekeeping genes during exponential growth, secondary, and alternative sigma factors (which function under specific conditions, or regulate transcription of specific cellular processes, respectively). The latter two are not always that easy to separate, and for simplicity we combine them under one nominator here: alternative sigma factors.

This may seem confusing, but protein nomenclature is not always in line with their function and genetic relationship, due to historical discoveries. For instance, the Sigma 70 family is named after the early-discovered primary sigma factor of *E. coli*, which has a molecular weight around 70 kDa; in Firmicutes this sigma factor is known as Sigma factor A, or SigA. Sigma 54 is also named after its common molecular weight. Alternative sigma factors can be involved in stationary phase survival, flagella biosynthesis, sporulation in Gram-positive bacteria, stress response (also called heat shock sigma factors as they were discovered in reaction to heat shock), or specific responses to environmental signals (these are called ECF, for extracytoplasmic function, although they never leave the cytoplasm).

We extracted the protein sequences of all sigma factors from 353 bacterial genomes. Based on their sequence, we could separate the primary from the alternative sigma factors. This way we found that every genome contained, as expected, at least one primary sigma factor but Actinobacteria frequently contained two primary sigma factor genes.

Since the molecular weight gave the name to the two main families, we compared this parameter for all extracted proteins. The Sigma 70 proteins were divided into primary and alternative factors, and the primary sigma factor was further divided into 5 bacterial phyla. The molecular weight was plotted in a box-and-whiskers plot (Figure 6.5).

---

3 Considering its functionality, the Sigma 54 family would belong to the secondary sigma factors, but since they are genetically different we keep them in a separate family.
As can be seen, Sigma 54 proteins indeed have a relatively conserved molecular weight. However, the size of members of the Sigma 70 family varies enormously, with proteins of Proteobacteria generally larger than those of the other represented phyla. The Actinobacteria display the most variation in molecular weight.

The separation in Figure 6.5 of primary and alternative sigma factors was based on domain composition. The primary sigma factors have some additional conserved domains compared to the alternative sigma factors. To identify their structural, and thus functional relatedness, all Sigma 70 factors were aligned as protein sequences (Sigma 54 proteins were excluded from this analysis since these are genetically distinct, and ECF were also excluded as they are too diverse). The alignment was graphically represented in a phylogenetic tree (Figure 6.6). The protein names have been colored for the bacterial phyla to which the genome belongs, using the same color key as in Figure 6.5 (the same colors are being used throughout the book when comparing bacterial phyla). Although one can’t read individual protein names, it still reveals the phyla to which entries belong. In this figure more phyla are represented than in the previous figure, as this time all identified sigma factors are displayed.
Figure 6.6. Phylogenetic tree of Sigma factors from 353 bacterial genomes.
In Figure 6.6 we see several clusters, each of which is subdivided into smaller clusters. These clusters can be used to predict the function of some of the included genes. For instance, one cluster contains a number of well-recognized heat shock sigma factors, all of which are from Proteobacteria; it can be assumed that all sigma factors in this cluster function as a stress response sigma factor.

Similarly, a number of well-characterized flagellar sigma factors cluster together. The RpoS cluster is closely related to the primary sigma factors (RpoS is used during stationary phase in *E. coli*). Note that one group contains sigma factors from only the Firmicutes, Cyanobacteria and Actinobacteria and within this group we can recognize sigma factors specifically regulating spore formation; only bacteria from these phyla can form spores. This indirectly enforces the predictive value of the tree: if we had found a sigma factor from a non-sporulating species in this cluster the validity of the finding could have been questioned.

The figure exemplifies that, although the phylogenetic tree is based on homology of protein sequences only, it can be used to recognize functional similarity. Protein function is dictated by sequences, as we already stated in previous chapters, and thus sequence similarity can predict protein function, although some caution is always needed in interpretation.

Once we have an accurate list of genes from a well annotated genome, it is possible to take the set of predicted proteins and sub-divide them in various ways. Phylogenic relationships can be assessed from the primary sequence, as the example above illustrates. Intracellular location can be predicted, to indicate whether the protein is likely to end up in the cytoplasm, membrane, or secreted. It can even be predicted which secretion system is most likely active on secreted proteins. Codon usage of the proteins provides interesting insights both in molecular evolution and in gene expression. It is possible to predict which protein genes are likely to be highly expressed. All these predictions can generate novel hypothesis, that can then be tested in the laboratory, and sometimes insights are gained that would be impossible or unlikely without bioinformatic predictions.

The question how many or few proteins are required (together with the minimum of rRNA and tRNA genes) for a cell to live independently was already discussed. However, just as interesting is the question which genes define a bacterium to be what it is. What is the essential gene content that makes up an *E. coli*, or a *Mycobacterium*? We can now define a core of genes for each species, supplemented with ‘peripheral’ genes that can be present or absent, once we have sufficient numbers of sequenced genomes. These questions will be discussed more fully in Chapter 12 on bacterial pan-genomics.
Concluding remarks

There are as many methods to compare genomes as there are ways to look at a genome sequence. Some parameters are easily calculated from the genomic sequence, such as the AT content or length of the chromosome. These numbers can be considered reliable, and trusted for easy large-scale comparisons across hundreds and thousands of genomes. However, other parameters, such as the number of protein-encoded genes or rRNA genes are dependent on the quality of annotation, which can vary from one file to another, even for essentially the same organism. It is possible to get a rough idea of the expected number of proteins in a genome, based on a variety of statistical methods, but these often will completely ignore the biology of the organism, which can also play an important role, especially in terms of parasitic organisms undergoing genome reduction. In summary, the method to choose depends on several factors, including how many genomes one wants to compare, as well as what information is desired.

References:


Box 6.1: Comparison and visualization tools used in this chapter

**Genome size: box-and-whiskers plot.** Genome sizes were extracted from the GenomeAtlas (www.cbs.dtu.dk/services/GenomeAtlas) and a box-and-whiskers plot was produced using the “R” language but it can be produced by hand as follows: Sort the data numerically. Define the minimum and maximum values. Find the median as the value separating the first 50% of the data from the second 50%. Next, define the 25% quantile (containing the first 25% of the data) and the 75% quantile. Draw a box spanning these quantiles, so that the middle 50% of the data are represented in the box (this is also known as the Inter Quartile Range, or IQR). A solid black line is drawn to indicate the median value. The dotted lines, the whiskers, connect the range of the highest and lowest values that are not classified as outliers. Outliers are defined as values distant by more than 1.5 times the IQR. Outliers are represented as open circles.

**Pairwise genome alignment.** Two genomes of *M hyopneumoniae* were aligned using the Artemis Comparison tool “WebACT” available at http://www.webact.org/WebACT. The Artemis program can be downloaded from the Sanger Center web page: http://www.sanger.ac.uk/Software/ACT. We selected the genomes from the pre-computed alignments that are available for many bacterial genomes. The graphical representations were made using default settings.

**Protein coding density.** The sum of all protein-coding DNA sequences divided by the total length of the chromosome gives an estimate of protein coding density (here the effect of overlapping genes is ignored as this is only a very minor fraction of the genome). Including all rRNA and tRNA genes in the analysis gives the proximate gene coding density. The protein coding density can be found on our CBS GenomeAtlas web pages. Another way of getting the coding fraction is to use the PubMed genome website (http://www.ncbi.nlm.nih.gov/sites/entrez?db=Genome); look under ‘Microbial’ (in the left-hand blue side bar) and click on ‘Genome Projects’. Then choose the genome of choice, and click on its Project Identifier number (PID, the first column of the table). This provides a table from which the number of genes in genomes (including multiple DNA entities) can be extracted.

**Protein length distribution.** To calculate protein length distribution, follow the links described above for Protein coding density. Click on each link under ‘Proteins’ or ‘RNAs’ to obtain tables of protein and RNA genes and their sizes. The required length distribution can be extracted and computed from these tables in a semi-automated way. The protein length distribution plots of two *Leptospira interrogans* genomes were produced with the method described by Skovgaard *et al.*, 2001. Protein length distribution plots for all the prokaryotic genomes in GenBank can be found in our GenomeAtlas web pages.
7 Genome Length and DNA Base Composition

Outline  Comparison of chromosomal DNA sequences can reveal many interesting patterns even if we ignore the genes they code. For instance, one can compare genome size, global and local AT content, or base composition. Genome size of the longer eukaryotic chromosomes is not related to the biological complexity of the organism or to the number of encoded genes. In bacteria, the total length of the genome generally correlates with number of genes present. Endosymbionts generally have small genomes as they have lost genes no longer essential in the niche they inhabit. Bacteria that colonize a host (benign or pathogenic) have larger genomes, but not generally as large as free-living microbes. The global AT content of bacterial chromosomes can vary from 25% to 75% but its distribution over the bacterial kingdom is uneven. The AT content also varies along the chromosome, with higher percentages around the replication terminus a lower AT percent around the origin. A bias of G’s towards the replication leading strand produces uneven nucleotide counts for the two halves of the genome, and this can be used to predict the lagging strand and origin of replication. The Structure Atlas is introduced as an important visualization tool for comparison of bacterial chromosomes.
Introduction

This chapter, together with the next, focuses on the pure DNA sequence of a genome, irrespective of what is coded. Obvious and simple properties to consider are the length of the chromosome, and the average AT content of the chromosome. This information is simple to calculate, and completely independent of the annotations in the GenBank file. Thus, as soon as a genome has been fully sequenced and assembled, analyses like those illustrated in this chapter can be carried out, without the need of any gene finding or genome annotation. Indeed, there is much useful information to be obtained at this step, which can actually help in characterization of the genome, such as localization of the DNA replication origin and terminus regions. Surprisingly, however, this level of information is frequently ignored, although it can reveal some very interesting insights in the genetics of the organism under investigation. A bacterial genome DNA sequence does much more than merely coding for proteins.

Length of genomes: the ‘C-value paradox’

A simple property that is very easy to calculate from the DNA sequence of a genome is its length. In an organism, the total amount of haploid DNA is also easy to measure experimentally. Back in the 1960s, early experiments on the amount of DNA in eukaryotic cells revealed a surprising conundrum: the amount of DNA an organism contains bears no apparent relationship to the organism’s complexity or size. Thus, for example, a seemingly simple single-celled amoeba contains roughly a hundred times more DNA than a human cell! This is referred to as the ‘C-value paradox’, and although the reason for this variation is still not known for sure, it appears that there is a consistent rough correlation between the nuclear volume of an organism, and the amount of DNA it contains. Apparently, the concentration of DNA in the nucleus is constant. It has been postulated that perhaps DNA is playing a structural role in the cells, in addition to coding for information.

Bacteria do not have a nucleus and the total amount of DNA is not conserved between various species. As discussed in the previous chapter, bacterial genomes vary in size about a hundred-fold, from as little as about 150 kbp to around 15 Mbp long. Endosymbiotic bacteria living inside cells of another organism gener-

1 Although one could think the ‘C’ in C-value refers to ‘characteristic’, ‘content’ or even ‘chromosome’ it stands for ‘constant’, originally describing the constant number of chromosomes per cell or per individual of a given animal species. The term C-value was coined by Hewson Swift for the haploid DNA content of (cells of) individual species. (Swift, 1950).
ally have small genomes. Evolution allowed genes to be lost from their genomes that coded for proteins no longer essential in the protected and narrow niche they inhabit. Bacteria that colonize a host (benign or pathogenic) have larger genomes than endosymbionts, but generally not as large as free-living microbes.

![Figure 7.1. Variation in the length of genomes for Archaea, Bacteria and Eukaryotes (the latter grouped in 4 Kingdoms) plotted on a logarithmic scale.](image)

Eukaryotic chromosomes span an enormous range, from double the size of a large bacterial genome (for example, some ticks have a genome of only about 20 Mb long) to more than a million times as large as a bacterial chromosome. This is diagrammed in Figure 7.1. The size of chromosomes within an animal species (or between species within a genus) can vary as well. A few shaded regions in the figure indicate such genome size variation. For example, the size of Drosophila genomes varies depending on the altitude at which the fruit fly species lives. Fruit flies from Hawaii, close to sea level, tend to have larger genomes (and their cells have larger nuclei), whilst fruit flies living high in the Andes tend to have smaller genomes (and smaller nuclei).

In this case, the cell size is related to the amount of oxygen that is available to diffuse into the cells - a limiting factor at higher elevations. But what this ‘extra DNA’ does, other than to fill up the nuclei, is not known for sure. We do know, however, how it is made up: for many organisms the extra DNA consists of relatively simple short sequences repeated many times. In eukaryotes, genes are separated from each other (and split up internally) by DNA rich in long stretches of re-
peats. For example, in humans, more than two-thirds of the genome consists of repeats.\(^2\) The longest human gene, encoding dystrophin, stretches over 2.4 Mbp, of which only 14 kbp are coding sequences. There are many individual human genes (when including their introns) that are longer than a complete bacterial genome.

Note the position of the human genome in the plot shown in Figure 7.1; the human genome is at least a thousand times larger than that of bacteria, but it is also about a hundred times smaller than that of amoebas. However, although there can be nearly a million-fold difference in the amount of DNA in bacteria vs. amoeba, both encode approximately the same number of proteins. Human cells contain about 5 times more genes than \textit{Bacteroides fragilis} (a common Gram-positive bacterium colonizing the human intestine). A human being does not harbor one single \textit{B. fragilis} strain, but multiple strains, each of which sharing only part of its total gene content with the next. When we first consider the total number of different genes found in the \textit{Bacteroides} population in a human intestinal tract (consisting of multiple strains) this may already approach the number of genes present in the human genome. Taking into account all bacterial species present in humans (estimates vary from hundreds to 40,000 species present in the human gut, making up 100 trillion cells) we can conclude that less than 1% of the genes in a person are ‘human’ - most are carried by bacteria!

\textbf{Eukaryotic genomes are different}

Since many eukaryotic genomes are so large and full of repeats, sequencing an entire chromosome as one contiguous piece is considerably difficult if not impossible. Repeat sequences are more of a problem than size, especially using the strategy of current technology for sequencing, which depends on small short read lengths randomly throughout the genome, and then stitching them together into an assembled continuous piece. Some modern technologies with increased speed and reduced cost are based on extremely short reads. This makes it impossible to sequence through repeat regions, which make up more than half of many eukaryotic chromosomes. Thus, despite many claims to the contrary in the public media, very few eukaryotic genomes have been completely sequenced. In fact, as of December 2007, only one eukaryotic genome (\textit{Cyanidioschyzon merolae}, a photosynthetic red algae) has been completely sequenced (Nozaki \textit{et al.}, 2007). Even the yeast genomes still contain gaps in the telomeric regions (the telomeres are the ‘endings’ of eukaryotic chromosomes and consist almost exclusively of repeat se-

\(^2\) A particular type of these repeats can be used to generate banding patterns from the DNA by a technique called ‘DNA fingerprinting’. As the length and distance of the repeats is different per individual, such DNA fingerprints are being used in forensic medicine. Thus, your DNA fingerprint doesn’t reveal your genes.
quences). This is one reason (amongst many) why in this book we focus on the completely sequenced genomes of bacteria and archaea, ignoring microbial eukaryotes.

Nevertheless, many of the tools described here can also be used for fungal, protozoan, plant and animal genomes, with the following caveats: most eukaryotic genome sequences are not completely sequenced; there is considerably more DNA present compared to a bacterial cell; the coding density is much lower in eukaryotes; most eukaryotic genes contain introns; and finally (related to the presence of introns) eukaryotic gene prediction and annotation is fraught with many problems. With apologies to the microbiologist with an interest in eukaryotes, we will focus on bacteria in most of this book.

**Genomic average base composition: the percentage AT**

One of the most fundamental properties of a genome is its base composition, in particular its AT or GC content. The average % AT, as a measure of AT content is the fraction of the genome built up of A’s and T’s, usually expressed as a percentage; sometimes the % GC content is given instead. Knowing the one it is easy to deduce the other as % AT + % GC = 100%. (Remember that double-stranded DNA in total contains as much A as it contains T, and as much G as C, so that expressing base composition in AT content, or in GC content, is fully informative). The AT content of a genome determines some of its physical properties, as the AT base pair has different properties from the GC base pair.

A hypothetical genome containing equal distributions of all four bases would have 50% AT content. Some genomes do indeed have an AT content very close to this, for example *E. coli* genomes contain around 49.5% AT. A genome average of around 50% AT is not at all the norm, though, as some bacteria can have quite AT-rich or GC-rich genomes. It has been known for more than 50 years that bacterial genomes range in AT content from 25% to 75%. Historically, the % AT of a genome has been considered a useful guide in taxonomy; for example, organisms may be grouped (amongst other characteristics) for similarity in AT content within a narrow range, and this range is considered part of the description for these particular bacteria. Sometimes the AT content is even included in the description of a bacterial species. Indeed, the AT content is an important parameter of living organisms, both on a global scale as well as at the local level. First we’ll have a closer look at the distribution of average AT content between bacterial species.

---

3 Throughout this book we express base composition as % AT. Others refer to the % GC, and for example a particular phylum of bacteria, the Actinobacteria, is also known as ‘GC-rich Gram-positive bacteria’, whose genomes of course are low in AT.
Figure 7.2. Distribution plot of AT content for 1723 sequenced prokaryotic chromosomes and plasmids. The distribution is shown as a histogram (in grey), with the height of the bars representing the number of chromosomes within a given range of AT content, and also represented as a smooth curve (red line).

**Distribution plots of AT content in chromosomes**

If the average AT content can vary from 25% to 75%, then one might expect naively that a plot of the % AT of lots of bacterial genomes might yield a bell-shaped (‘Gaussian’ or normal distribution) curve, with most genomes containing around 50% AT, and fewer genomes having more extreme values on both ends. A distribution plot is shown in Figure 7.2 that was constructed based on more than a thousand sequenced DNA entries (sequenced chromosomes and plasmids including multiple chromosomes per species). Surprisingly, the distribution is bimodal, and it appears that there are in fact slightly fewer genomes around 50%, flanked by two peaks with high distributions, around 35% AT and 65% AT.

Why are there two peaks? One possibility is that this observation is reflective of the bias in the selected organisms: the genomes chosen to be sequenced are not random, nor necessarily reflective of the biodiversity found in the ‘real world’ environment. Most genomes were sequenced from bacteria that are easy to grow in the lab and are of particular interest for research. Thus, we know this is a biased selection. The data presented in Figure 7.2 include, for example, some bacterial
phyla with only a few genome sequences whereas other phyla are represented with several hundred entries. However, even if we remove all of the ‘redundant’ genomes (analyzing only one genome for a given genus, for example), the bimodal distribution curve remains basically unchanged.

Another possible explanation is that the obtained curve combines (at least) two different types of distributions. Since a trend can be recognized that small genomes are frequently AT-rich, we separated the chromosomes based on their size. From the original data set, we extracted all chromosomes that are 5 Mbp or longer. The subset of 101 ‘large’ chromosomes has an AT content below 50% (on average: 45% AT). A distribution plot of this subset is shown in blue in the left panel of Figure 7.3.

![Figure 7.3](image)

**Figure 7.3.** Distribution plots of percent AT distribution in sequenced chromosomes. For the ‘large’ chromosomes (larger than 5 Mbp, shown in blue on the left), a GC-rich base composition is preferred (low % AT), whilst for the ‘small’ chromosomes (between 1 and 2 Mbp, on the right, in red), the distribution is biased towards AT-rich chromosomes.

Clearly these large genomes tend to be more GC rich (or AT poor), with a preference (a major peak in the distribution) around 33% AT. However, there is also a smaller peak, around 55% AT, with a tail of large bacterial genomes that are more AT-rich. These larger genomes with a high AT content (the exceptions to the rule) are mainly Firmicutes (9 out of 12 large chromosomes that are more than 60% AT are from *Bacillus* or *Clostridium* genera), whilst the peak around 55% contains the marine bacteria *Shewanella* and the intestinal *Bacteroides* genomes. These latter chromosomes are all sized between 5 to 6 Mbp, which is towards the lower end of the (arbitrarily chosen) size range for ‘large’ chromosomes.

Similarly, ‘small chromosomes’ were selected, giving 167 entries between 1 Mbp and 2 Mbp (this removed all plasmids and only very few chromosomes). These 167 tend to be more AT-rich, as can be seen in the plot in red, to the right in
Figure 7.3, where 135 out of 167 are more than 50% AT-rich. Again a tail of ‘small’ chromosomes can be seen that are nevertheless AT-poor: the 16 small chromosomes between 30% and 40% AT are mostly soil bacteria (e.g., Burkholderia, Rhodobacter, Rhodococcus, and Sinorhizobium), or thermophiles (Methanopyrus and Thermus). The remaining 373 chromosomes of intermediate size (between 2 and 5 Mbp) were a mixed lot, both in % AT and in taxonomic or environmental groups. In summary, we can conclude that % AT is loosely associated with bacterial genome size; in that long genomes tend to be GC-rich, and small genomes tend to be AT-rich, but exceptions exist on both ends of the spectrum.

**Box-and-whiskers plot on AT content across bacterial phyla**

Base composition can vary considerably within taxonomic divisions, as shown in the box-and-whiskers plot of Figure 7.4, and the % AT is only conserved within closely related organisms. Almost all AT-rich genomes belong to eight phyla (Nanoarchaeota, Aquificae, Chlamydiae, Chloroflexi, Firmicutes, Fusobacteria, Spirochaetes, and Thermotogae); also the epsilon division of Proteobacteria tends to be mainly AT-rich. Within these phyla, very few bacteria have genomes with less than 50% AT.

![Box-and-whiskers plot of base composition: % AT in 621 bacterial genomes.](image)

The most AT-rich genomes tend to be intracellular parasitic or endosymbiotic bacteria (some of which are α- and γ-Proteobacteria), including *Buchnera* and Mycoplasma. The most AT-rich completely sequenced non-endosymbiont species is a
Clostridium, and many of the genomes with an AT content above 65% AT content are environmental organisms (e.g., Methanococcus, Prochlorococcus, Methanobrevibacter, and Bacillus).

At the other extreme (the left-hand side of Figure 7.4) there are only four phyla that comprise almost exclusively of AT-poor (GC-rich) genomes (Acidobacteria, Actinobacteria, the Deinococcus / Thermus group, and Planctomycetes). Also, the delta subdivision of Proteobacteria contains mainly AT-poor species, although some outliers contain over 50% AT. If we compare the plot of Figure 7.4 with the genome size plot (Figure 6.1 in the previous chapter), again the general trend is confirmed that, on average, AT-rich genomes are smaller (Aquificae, Chlamydiae, ε-Proteobacteria, Spirochaetes, Thermotogae) and GC-rich Actinobacteria have large genomes.

The ‘top 10’ most AT-rich genomes, for which a sequence was available at the time of writing, have an average length of 680 kbp, whilst the average length of the most GC-rich genomes is more than ten times larger, 7.0 Mbp. Many of the extremely GC-rich genomes are from organisms commonly found in the soil, such as Burkholderia, Streptomyces, and Frankia. It is interesting along these lines to note that DNA sequences obtained directly from environmental samples (metagenomics, to be covered in more detail in Chapter 13), tend to be more GC-rich in DNA from soil (terrestrial) samples, and more AT-rich in marine environments.

As will be discussed in more detail in the next chapter, codon usage plays a major role in shaping the AT content of a genome. Unlike eukaryotic genomes, which have low coding densities, in most bacterial genomes the majority of DNA codes for proteins, and thus codon usage can drive the % AT (or vice versa, as cause and effect cannot be determined). From the weak correlation between AT content and genome size, it follows that codon usage also weakly correlates with the size of the genome, however, the biological mechanisms behind such observations are currently obscure.4

4 The questions why or how small bacterial genomes are AT-rich remain speculative. The divergence has possibly evolved through mutational preferences. There is a tendency for GC base pairs to mutate to AT base pairs over time, due to deamination of Cytidine, amongst other processes. This drift will accumulate over time in the absence of DNA repair mechanisms. Since smaller genomes may not contain (as many) repair enzymes, the argument goes, they will drift towards an increased AT composition. This is one of several explanations that have been postulated to explain the observations.
GC skew- bias towards the replication leading strand

So far, we have dealt with average base composition in terms of % AT, but the A’s and T’s are not always equally distributed over the two DNA strands. Consider investigating base composition in a window of, say, 200 nucleotides sliding along the genome, and you will observe that the A’s and T’s differ from global values as one scans along the length of a chromosome. This can be illustrated on a Base Atlas (which, together with the sliding window analysis, was already introduced in Chapter 3). For some genomes there is a bias of A’s towards the leading strand, and a bias for T’s the other strand (depending on the organism it can also be the other way round). As a more general rule, for nearly all bacterial genomes there is a tendency for G’s to be biased towards the replication leading strand. Remember that replication starts at the origin, and progresses in both directions, in clockwise and counter-clockwise direction away from the origin. Thus, for each direction the leading strand is different, because the leading strand is synthesized from the 5’ to 3’ direction. Replication finishes when both ends meet, at 6 o’clock from the replication start.

When following one strand of the genome in clockwise direction all along a circular chromosome (which is how we ‘read’ a genome sequence, represented by one strand of sequence only) the leading strand goes from about 12 o’clock (start of replication) to roughly 6 o’clock, after which the sequence becomes lagging strand, as we now enter the half of the genome where replication uses the other strand as the lead. The result is a shift from G-rich sequences in the first half to C-rich in the opposite half.

Such deviations in global base composition for G’s and C’s are known as the GC skew (already introduced in Chapter 3). To give an example, a GC skew is clearly seen in the chromosome of Clostridium tetani (a soil bacterium and the cause of tetanus) for which a Base Atlas is shown in Figure 7.5. Note the striking difference in color between the right and left halves of the chromosome. From the figure it is clear that not only G’s (blue) but also A’s (green) are biased towards the replication leading strand, clearly visible for the first (right-hand) half of the genome. T’s (red) and C’s (magenta) are biased towards the replication lagging strand (visualized for the second half of the genome). A bias in A’s and T’s is called an AT skew, so C. tetani displays both a GC and an AT skew.

The GC and AT skew are represented in the inner lanes. As an aside, it is worth mentioning that AT and GC skew are irrespective of the % AT content. The latter doesn’t show a significant difference between the left and right-hand halves of the genome, as shown by the innermost circle in the figure.
Figure 7.5. A Base Atlas showing bias of G’s and A’s towards the leading replication strand in \textit{Clostridium tetani}. The orientation of most genes favors the leading strand in this organism.

For most bacterial genomes, the GC skew can be used to identify the DNA replication origin: it is the position where the GC skew lane changes from magenta to blue, indicated in Figure 7.5 by an arrow. For the \textit{C. tetani} chromosome, the replication origin is not exactly positioned at the first nucleotide or the 12 o’clock position, but a bit further along the sequence (around 52,000 bp). Although the numbering of a published genome would ideally start at the origin of replication, in more than two-thirds of the sequenced bacterial chromosomes in GenBank, the replication origin is not located close (less than 100 kbp) to the first numbered base. Apparently, GC skew is not always tested or interpreted to identify the origin of replication and a circular genome is artificially ‘cut open’ in another position to start the numbering. Nevertheless, though not always as pronounced as for the given example, GC skew is observed in most bacterial chromosomes so far sequenced, with G’s almost always biased towards the replication leading strand.

From Figure 7.5 it is also obvious that most genes of \textit{C. tetani} are coded on the leading strand. Thus, most genes are located on the positive strand for the right-hand half of the chromosome (given as blue blocks on the 5th ring), whereas the second half has most genes present on the negative strand (red in the figure). Such observations are organism-dependent and do not follow general rules, other than that genes are not generally highly overrepresented on the lagging strand.
Figure 7.6. Bias of G’s towards the leading replication strand in the *Desulfotalea psychrophila* chromosome. Note that the A’s are biased towards the replication lagging strand in this organism, which is the opposite from that in Figure 7.5. Gene orientation is evenly distributed between leading and lagging strand in this organism.

Compare the Base Atlas of the Gram-positive *C. tetani* of Figure 7.5 with the Base Atlas for the Gram-negative *Desulfotalea psychrophila* in Figure 7.6. *D. psychrophila* lives in arctic waters, and likes cold water, as its name implies, and can even multiply at temperatures below freezing (Rabus et al., 2004). Note that for the *D. psychrophila* chromosome, the replication origin is located around 710,000 bp into the sequence. In *D. psychrophila*, the A’s are biased towards the replication lagging strand, which is the opposite that found for *C. tetani*. (That is, the green and turquoise are together in Figure 7.5, but opposite in Figure 7.6). For the *D. psychrophila* chromosome, the genes are more or less randomly distributed between the leading and lagging strands. Strand bias, or *base skew*, towards the leading and lagging strand influences the structure of the DNA, as we will see below.
Bias of oligomers towards the replication leading and lagging strands

Although GC skew can easily be used to find the replication origin in chromosomes such as those of *C. tetani* and *D. psychrophila*, for some bacterial genomes the bias is too weak to pinpoint the origin. Fortunately, in addition to the global bias in G’s and A’s towards one strand, there is also a bias of base composition in short sequences, such as dimers, tetramers or even octamers and this provides a strong signal. We have developed an automated method to plot the bias in all oligomers up to octamers, and this can assist to identify the replication origin (Worning *et al.*, 2006). Figure 7.7 illustrates the strand bias of these oligomers, by analyzing one strand for all short DNA ‘words’ of up to 8 nucleotides in length, along each position in the chromosome. The X-axis represents the position and the Y-axis the total strand bias for all the oligomers (single bases, dimers, trimers, etc. up to octamers), shown by the red curve for the *C. tetani* and *D. psychrophila* chromosomes.

![Figure 7.7. Strand difference of oligomers in two bacterial chromosomes. The *C. tetani* E88 chromosome on the left is the same as used in Figure 7.5, and contains a bias of both A’s and G’s towards the same strand (the green and blue lines follow each other). The *D. psychrophila* genome at the right (see also Figure 7.6) displays a clear bias of G’s towards the leading replication strand, and the A’s towards the lagging strand.](image)

In both panels, there are two peaks visible. Note that the bias is quite strong in *C. tetani*, with a maximal peak intensity of 4 million bits, whilst for *D. psychrophila* the peak is approximately half as prominent. Which of the two peaks corresponds to the location of the DNA replication origin, and which to the replication terminus region? That can be read from the green line, which plots the weighted averages of GC bias present in oligomers with a positive score for G and a negative value for C. Since G’s favor the replication leading strand, a positive peak in the green line corresponds to the origin. Even for genomes with a GC skew too weak
to identify the origin of replication by visualization in a Base Atlas, a peak in GC bias of all the oligomers can still identify the origin of replication.

When we were developing this method for finding replication origins, we were puzzled why the A’s are sometimes biased towards the leading strand, and sometimes towards the lagging strand, as in the two plots of Figure 7.7, but as more genomes were analyzed, a pattern began to emerge. First, the Gram-positive Firmicutes (Clostridium, Bacillus), tend to have the G’s and A’s biased on the same strand, whereas many of the Gram-negative bacteria, tend to have the G’s and A’s biased towards opposite strands. Further, this bias is strongly correlated with the presence of the PolC proofreading enzyme for the lagging strand. Its presence in the Gram-positive bacteria correlates with the bias of G’s and A’s towards the same strand, and its absence correlates with the opposite trend. Although this can provide a reasonable mechanistic explanation, at this stage the observation is only a correlation and not proof (Worning et al., 2006). The correlation illustrates, however, that bioinformatic analysis can lead to hypotheses that can be tested by wet laboratory approaches.

**Global chromosomal bias of AT content**

So far we’ve compared average % AT between chromosomes, and base composition along a chromosome. A third parameter to investigate is how the % AT of a chromosome varies along the sequence, as there are regions that are relatively AT-rich, and regions that are more GC-rich. Such variation can occur on a local scale (within 100 bp) or on a global scale (larger than 100 bp; the distinction for local and global is chosen arbitrarily). There are also differences in AT content between coding and non-coding sequences, and even between particular classes of genes.

In contrast to the GC skew, which can be visualized on the right and left half of the atlases shown above, the global region around the origin of replication, represented at the top of the atlas tends to be more GC-rich, whilst the region around the terminus is more AT-rich\(^5\). Comparing hundreds of genomes, this trend is observed independent of the average AT content of the chromosome. Figure 7.8 shows a box-and-whiskers plot for 150 bacterial chromosomes, comparing AT content around the origin and terminus of replication.

The replication process starts by separating the two strands at the origin, and one might think this region would be more AT-rich to enable such local melting

---

\(^5\) The reason we don’t see this on the Base Atlases of Figure 7.5 and 7.6 is that the default setting of a Base Atlas is not sensitive enough to visualize these trends. With different settings, though, the variation in base composition around origin and terminus of replication can be shown.
(indeed it is on a very local level of the few nucleotides where the DNA starts to melt). Counter-intuitively, the replication origin sits in a region that is more GC rich than the rest of the genome. One possible biological explanation is that this way ‘false starts’ would be prevented. The reason why the replication terminus region is more AT-rich could be that it contains more curved DNA. Curved DNA is a particular structure of the DNA helix that will be further discussed below.

**Figure 7.8.** Average AT content for 80% of the genome (designated ‘middle section’) and the 10% around the replication origin (given by the 36 degree segment of the arc) and the 10% flanking the replication terminus. The schematic to the left shows the three segments (the leading strand for each is depicted by colored arrows). To the right the deviation in AT content is shown for these three segments of 150 chromosomes in a box-and-whiskers plot.

**Deviations of average AT content in coding and non-coding regions**

Sometimes, a single gene or a multi-gene locus is identified in a bacterial genome with an AT content deviating from the global average, and this is taken as evidence for horizontal acquisition. From what has been presented in this chapter it will be clear that such a conclusion can only be drawn with care. More correctly, the AT content of any gene should be compared to a local average, to correct for differences around the origin and terminus of replication. Accepting that the AT content of a chromosome deviates from the mean around these regions, how con-
stant is the base composition when we consider genes distributed all along the genome?

In fact, the AT content can locally vary quite strongly between genes, especially for particular gene groups. The ribosomal RNA genes, for instance, are so strongly conserved between species that only limited variation in base composition is allowed. Therefore they appear as relatively AT-poor in AT-rich genomes, and stick out as AT-rich in genomes that on average are AT-poor. Moreover, in some genomes genes encoding proteins expressed on the outside of the cell are richer in AT content than other gene families.

![Distribution of AT content in genes in the B. cenocepacia chromosomes and plasmid](image.png)

**Figure 7.9.** Distribution of AT content in genes in the *B. cenocepacia* chromosomes and plasmid. The global genome % AT is indicated by a hatched line.

In Figure 7.9 the distribution is plotted for AT content of individual genes in the genome of *Burkholderia cenocepacia*, strain AU 1054 (a marine organism). As this organism contains three chromosomes and a plasmid, all four of these DNA segments were separately analyzed. As can be seen, the base composition of the genes found on each of these chromosomes varies, producing a distribution approximately around the global genome average AT content (shown by a dashed line). The distribution of two of the three chromosomes is similar, but chromosome 1 has a slightly different distribution, and genes in the plasmid (shown in red) tend to be more AT-rich, with a broad range of % AT. Note that the most frequently encountered % AT in coding genes does not equal the global genome content for any of the DNA molecules. The genes with base composition most different from the average could be acquired horizontally, but without additional evidence that conclusion may be premature.
In addition to the variation between genes, AT content can locally vary quite strongly between genes and intergenic regions. The intergenic regions are typically about 5% more AT-rich than the gene-coding regions - this is true for nearly all bacterial chromosomes studied, whether they are AT-poor or AT-rich (Bohlin et al., 2008). The slightly higher AT content of intergenic regions could explain why the genome % AT average in Figure 7.9 is slightly higher than the most frequently encountered % AT in its coding genes.

**DNA structures**

What are the implications of variation in AT content and other base compositional changes along the chromosome? DNA sequence dictates DNA structure, which can hint at function. It is an under-appreciated fact that there are different structures that DNA can adopt locally and globally, depending on the sequence and environmental conditions.

Which of many types of DNA structures is formed will depend on the sequence, the local environmental conditions (e.g., salt, ions, etc.) as well as the superhelical density (see below). Even though the environmental conditions can change, the likelihood of particular structures can be predicted from a DNA sequence. Since most of these structures are sequence dependent, they can specifically be searched for from a DNA sequence. Physical parameters can be calculated from a DNA sequence that affect the formation of particular structures. Three examples are intrinsic curvature, stacking energy and position preference (introduced in Chapter 3). All three are typically visualized on both our standard ‘Genome Atlas’, as well as on an atlas specifically dedicated to structures. A more detailed technical explanation for these three DNA structural parameters with references is given in Box 7.1.

The physical parameters that we plot on a *Structure Atlas* predict the general feature of local DNA structures, such as whether the DNA is expected to be rigid and inflexible, or very flexible and easy to melt. In addition, DNA can form a number of structures with different energetic states, some of which are produced with the help of proteins and others are the result of the DNA sequence itself. The following DNA structures are worth mentioning here (more structures will be introduced in the next chapter):
• **Relaxed double-helical DNA**, also called B-DNA, is the DNA helix that is normally depicted. Though it is not common in a living cell, it is the average configuration of DNA in solution, for example DNA produced in *vitro* by DNA polymerase in a PCR reaction.

• **Superhelical DNA**, or supercoiled DNA, is the default structure in the cell. (The term ‘supercoiled’ probably doesn’t need an explanation when thinking of telephone cords). The double-strand right-handed helix is supercoiled with the help of enzymes, for which local strand breaks in the phosphodiester backbone are introduced and, after coiling, sealed again to keep the DNA in a permanent supercoiled state. DNA can be positively supercoiled (when extra helices are introduced) or negatively supercoiled (when helices are removed). In a circular molecule, supercoiling changes locally when another part of the DNA strand unwinds, as during transcription or replication. As the superhelical density of DNA is the product of enzyme activity, it cannot be predicted from its sequence.

• **Curved DNA** refers to a three-dimensional structure where the DNA forms a curve or a spiral in space, rather than a flexible rod. AT-rich DNA sequences tend to be more curved than GC-rich DNA. Curved DNA can bring separated sequences into each other’s vicinity, and affect the affinity of DNA-binding proteins. It plays a role in gene expression, amongst other things.

• **Melted DNA** is where the DNA helix is separated into two strands, for instance for opening up the two strands to allow replication or transcription. DNA needs to be locally unwound (inducing supercoiling in other parts) before it can melt. AT-rich DNA melts more easily than AT-poor DNA.

• **A-DNA** is another right-handed helical form of DNA with a shorter and fatter helix than found in a B-DNA helix. A-DNA is favored in double-stranded RNA, as well as RNA/DNA hybrids. Most bacterial chromosomal DNA exists as a mixture of A-type and B-type helices. Purine stretches (G’s and A’s on the same strand) tend to favor an A-type of helix.

• **Z-DNA** is a left-handed helix and is uncommon in bacterial DNA, but is associated with alternating G’s and C’s, which are quite common in eukaryotes, and rare in most bacteria (a notable exception are the *Burkholderia* genomes). In general, such stretches tend to form less stable helical regions, and can melt more readily.

---

6 An easy way to understand how local winding or unwinding affects supercoiling in other parts is to envisage holding a rubber band in both hands; twisting a fraction of the band with one hand while fixing another part with the other hand will rapidly produce supercoils in opposite directions.

7 A-DNA was the first double helical form of DNA characterized by Rosalind Franklin. Hence it’s name ‘A-DNA’, as opposed to the second type of structure, which is now more commonly known, the ‘B-DNA’ helix made famous by Watson and Crick.
Certain DNA sequences are known to be much more readily compacted than others. Here, we are interested in DNA structural elements that can play a role in chromatin organization, as these may affect mutational stability of a region, or gene expression for example. These DNA helix architectural parameters can be calculated, based on dinucleotide or trinucleotide models. DNA curvature and base-stacking measure structural properties of DNA alone, whereas position preference measures the ability of the DNA helix to be distorted by proteins.

**Intrinsic DNA curvature** is calculated based on dinucleotide models using the CURVATURE program (Bolshoy et al., 1991, Shpigelman et al., 1993). The term ‘curved DNA’ here refers to DNA that is intrinsically curved in solution and can be readily characterized by anomalous migration in acrylamide gels. The scale used in a Genome Atlas is in arbitrary units ranging from 0 (no curvature) to 1.0, the curvature of DNA when wrapped around nucleosomes (a tight structure). Using a 10,000 bp smoothing window, curvature values are distributed so that the progressively colored region (-3 std to +3 std) lies between dark orange (uncurved regions) and dark blue (strongly curved regions).

**Stacking Energy.** Base-stacking energies are calculated from dinucleotide values according to Ornstein et al. (1978). The scale is in kcal/mol on a negative scale, with a range from -3.82 kcal/mol (will melt easily) up to -14.59 kcal/mol (which would require the most energy to destack or melt the helix). A positive peak in base-stacking (i.e., numbers approaching -3.82) reflects a region that would de-stack or melt more readily. Conversely, minima (more negative values) would represent more stable regions of the chromosome. Using a 10,000 bp smoothing window, curvature values are distributed so that the colored region in a genome atlas (-3 std to +3 std) lies between dark green (more stable) and dark red (will melt easier).

**Position Preference** is a measure of anisotropic DNA flexibility. It is a trinucleotide model based on the preferential location of sequences within nucleosomal core sequences, as described by Satchwell et al. (1986). We use the absolute magnitude value as a measure of DNA flexibility (Baldi et al., 1996). The trinucleotide values range from close to zero (0.003, presumably maximum flexibility) to 0.28 (considered rigid). Since very few of the trinucleotides have values close to zero (e.g., little preference for nucleosome positioning), this measure is considered most sensitive towards the low (‘flexible’) end of the scale. Using a 10,000 bp smoothing window, curvature values are distributed on an atlas so that the colored region (-3 std to +3 std) lies between dark green (flexible) and dark purple (rigid).
The Structure Atlas

The most informative structural parameters that can be predicted from a genome sequence are plotted together on a Structure Atlas. The predicted structural features visualized on a Structure Atlas can add information to gene annotation and expression predictions. Structure Atlases can facilitate such predictions, for instance it can assist in identifying rRNA gene loci.

Figure 7.10. DNA Structure Atlas for the two chromosomes of *Agrobacterium tumefaciens*. The circular chromosome 1 is given at the top and the linear chromosome 2 at the bottom. The circled regions numbered 1 to 5 are discussed in the text.
A Structure Atlas should not be interpreted on its own, without the context of other features. Three of the features of a structure atlas therefore appear on a Genome Atlas, where the findings can easily be related to the presence of repeats, base skew and other observations.

Figure 7.10 shows the Structure Atlas of the two chromosomes of Agrobacterium tumefaciens, one of which is circular and the other linear. The outermost 3 circles (in the linear chromosome the top three lanes) show the three structural parameters discussed in Box 7.1, given as 3 or more standard deviations variation. The other structural parameters all deal with how easily the DNA helix is deformed, and are described in more detail elsewhere (Jensen et al., 1999). These parameters are all correlated with the % AT, shown in the bottom lane; note that the % AT shown here is also scaled showing extreme values 3 standard deviations from the average, unlike in the Genome Atlas, where the scale usually goes from 20% to 80%, with 50% AT in the middle. Five regions have been circled in Figure 7.10, and we will have a closer look at each of these particular regions.

Region 1 is a predicted strongly curved region (dark blue in the first lane) that will melt easy (red in the second lane) but without any remarkable flexibility features. This region might be somehow involved in chromosome structures (perhaps a region where the chromosome attaches to the membrane, for example). Further, because it is AT-rich, this region might be expected to mutate at a higher rate than the chromosomal average. Region 2, however, is neither strongly curved nor melting easily, but it is very flexible (dark green in the third circle). This indicates that this region of the chromosome will not fold well around chromatin proteins, and hence genes located here might have a tendency to be highly expressed, under the right conditions. Region 3, on the other hand, is both flexible and melts easily, though it is not very strongly curved. This region contains an rRNA operon, and rRNAs are known to be highly expressed. Region 4 is the opposite of region 2, in that it is predicted as rigid with a tendency to melt less easily. The origin of replication is located near to this region. Finally, region 5 (on the linear chromosome) is like region 1 in that it is strongly curved and will easily melt, but it also is predicted as quite flexible. Here, genes could be located that are potentially highly expressed but might be repressed by histone-like proteins that recognize curved DNA. These regions are just a few of the striking features of these two chromosomes that are visible in a Structure Atlas.

Bias in purines - A-DNA Atlases

As discussed above, for the C. tetani chromosome, both purines (the G’s and A’s) are biased towards the same strand, which means that the chances of finding a purine stretch in this chromosome will be quite high. In contrast, for the D. psy-
chromophila chromosome, with G’s on one strand and A’s on the other, the chance of finding a purine stretch is considerably less. The frequency of purine stretches is significant, as short runs of purines can stabilize A-DNA. An A-type helix is favored at higher salt concentrations, and also in RNA / DNA hybrids, as well as in double stranded RNA (such as the stems in rRNA genes). Finding purine stretches along a chromosome can be used as an aid to predict the presence of potential non-coding RNA regions. In general, purine stretches are overrepresented in many chromosomes compared to their expected occurrence by chance (Ussery et al., 2002).

Tracts of repeated homonucleotides (AAAA) and of purine or pyrimidine stretches (e.g., AGAAGG and CTCCCT) are plotted in an A-DNA Atlas. Figure 7.11 shows the A-DNA Atlas for the Chlamydia muridarum chromosome (an intracellular lung pathogen of mice). Note that the G stretches (outermost G circle) and A stretches (2nd circle from outside) seem to be biased on opposite strands (the GC and AT skew, not shown here, is similar to that of D. psychrophila). Nonetheless, a different distribution is seen for pyrimidine stretches compared to purine stretches, analyzed here for homo-tetranucleotides (circle following the annotations) and for hetero-decamers (second circle from the inside). We will see in later chapters that the lanes depicting structural features can be of assistance for making predictions on the presence of genes, but mostly in combination with other features; structural features do not allow clear-cut predictions by themselves, but are used to enforce other observations.

Figure 7.11. An A-DNA Atlas for the Chlamydia muridarum main chromosome.
For completeness, we mention here that a Z-DNA Atlas can also be constructed for bacterial chromosomes, by analysis of alternating purine / pyrimidine stretches (TATA, CGCG, CACA or TGTG, or (RY), for short). Since such structures are uncommon in bacteria, an example is not presented here but the tool is available from the Genome Atlas website.

Concluding remarks

This chapter has described features of bacterial genomes that can be deduced from DNA sequences irrespective of the genes they encode. Much can be learned from a genome sequence even before the presence of genes is assessed. The Base Atlas, Structure Atlas and A-DNA Atlas have been introduced. Different atlases provide different information. In the next chapter we will introduce the Repeat Atlas, the last specialized atlas from which we use a section in our standard Genome Atlas.

References:


---

8 http://www.cbs.dtu.dk/services/GenomeAtlas


Swift H. “The constancy of desoxyribose nucleic acid in plant nuclei”, Proc Natl Acad Sci USA 36:643-654. [PMID 14808154]


**Box 7.2: Comparison and visualization tools used in this chapter**

*Distribution plots of genome AT content.* After extracting genome and plasmid sequences from GenBank, their AT content was calculated from the sequence. The obtained values were plotted in a histogram. The displayed figure was made using standard statistical methods: we used a gaussian kernel density estimation. The width of the kernel was estimated from the data as described in Silverman, 1986.

*Strand difference plots.* The strand difference plots were produced as described by Worning et al., 2006. Pre-calculated plots for published genomes are available from http://www.cbs.dtu.dk/services/GenomeAtlas. Select the organism and genome of choice, then select ‘Predicted Origin’.
8 Word Frequencies, Repeats, and Repeat-related Structures

Outline DNA sequences contain various types of repeats, from simple short ‘words’ that can occur frequently or rarely, to longer repeats such as complete rRNA operons. Several methods exist to search for the frequency of particular words. Compared to eukaryotes, bacterial genomes are streamlined, and contain few repeats. There are four different types of repeats: direct, inverted, mirror and everted. For the first two the distance of the repeat units is important, and can be investigated on a local or on a global scale. Mirror and everted repeats are only of importance on a local scale. Local repeats correlate with AT content, in that genomes with high or low AT content have more local repeats. The repeats that are found can form various structures of biological significance. The structure atlas is further extended to fully explore the structural features of DNA, which can help to predict exceptional properties of the genes encoded in particular regions, such as the likelihood of high expression.
Introduction

In the previous chapter the sliding window analysis was mentioned and this type of analysis will be further introduced here. The method can be used to analyze the frequency of particular nucleotide combinations, or ‘words’, in order to compare this with expected frequencies to a random DNA sequence. Obviously, DNA sequences in a genome are not random, and how exactly biological genomes differ from randomly generated genomes can give important insights. The sliding window analysis is also suitable to identify sequences that are repeated within a genome.

Analyzing word frequencies in a genome

For a given genome, the frequency of short combinations of nucleotides that may occur at a certain frequency can easily be determined with a computer program. In particular, dinucleotides and trinucleotide ‘words’ are frequently being investigated. Given the base frequency of a particular genome, one can calculate the expected frequency of each possible base combination (word), thus giving expected frequencies to occur in the genome by chance, using a mathematical approach known as Markov Chain Method (MCM). Thus, based on the known amounts of A’s, T’s, C’s, and G’s in a genome, it is possible to calculate how frequent all the dinucleotides (such as ‘AA’ or ‘GC’) would be. Next, the observed frequency of all dinucleotides is determined, and compared to the expected frequency.

An analysis of the frequency of dinucleotides can provide some interesting observations. For example, the chance of finding two purines (either A or G) adjacent to each other is easy to calculate as 0.5 times 0.5, or 0.25. This is because, of the four bases, two are purines (A and G), and two pyrimidines (C and T), so the chance of any given base being a purine is simply two out of four, or 0.5. Based on this, a string of 8 purines (or 8 pyrimidines) should be found about once in 256 nucleotides by chance. This is regardless of the AT content, because the chance of finding a purine is still one in two, independent of the AT fraction.

Turning now to real observations rather than expectations, for some bacterial chromosomes (especially thermophiles), there are many more purine stretches than would be expected. This also applies to the Firmicutes, such as Clostridium tetani for which the Base Atlas was introduced in the previous chapter (Figure 7.5). In these organisms there is a bias towards A’s and G’s being present on the same strand - hence the chance of finding a purine stretch is greater than would be the case for a random distribution. Similarly, genomes with a strong bias of G’s towards one strand, and A’s towards the other (for which we have also seen an ex-
ample in Chapter 7) would be more likely to have fewer purine stretches than expected. Correcting for composition base bias, the observed frequency of purine stretches can be satisfyingly predicted in bacterial chromosomes. For observations in eukaryotic chromosomes, however, the bias towards purine stretches is much larger, and a Markov Chain Method fails to predict accurately the observed frequencies in those organisms (Ussery et al., 2002).

There are several different statistical methods available to calculate deviations from expected averages, and the optimal method depends on which question is being asked. These include Relative Oligonucleotide Frequencies (ROFs), di- to hexanucleotide Zero Order Markov methods (ZOMs) and second order Markov Chain Methods (MCMs). We have recently compared these methods on all sequenced bacterial genomes then available, and found that ROFs are best suited for distant homology searches, whilst the hexanucleotide ZOM and MCM measures tend to be more reliable for phylogenetic analysis (Bohlin et al., 2008). Without going into the rather technical details, the conclusion is that the tetranucleotide ZOM method is a good measure to detect horizontally transferred regions or to compare phylogenetic relationships between plasmids and hosts.

The frequency of trinucleotides is special as they represent the genetic code in coding sequences, and thus are under particular selection pressure. Their frequency will be treated in Chapter 9. As dinucleotides are parts of trinucleotides they also are affected by codon usage. That does not apply as strongly to tetranucleotides, which give ‘only’ 256 (4^4) possibilities, a number that can still be analyzed without too much complexity. As an example of applying an ROF model, the occurrence of tetramers within a selection of sequenced bacterial chromosomes has been examined. Some tetramers are overrepresented and others are underrepresented. When analyzing such deviations from expectations it was found that each microbial species has a unique profile, though closely related organisms are similar, even when they differ considerably in global AT content (Pride et al., 2003). From such analysis one can identify those short oligomers that give the highest overrepresentation and lowest underrepresentation.

Table 8.1 summarizes some of these ‘top scorers’ for a number of organisms. Note that global genome AT content doesn’t dictate the nucleotide composition of the most extreme over- or underscorers: *Deinococcus radiodurans* (a bacterium that is surprisingly resistant to radiation due to highly efficient DNA repair systems) with only 33% AT has a most overrepresented word consisting of four A’s (the most overrepresented word means that it deviates most extreme from expectations given the base composition of the complete genome). The same tetranucleotide AAAAA is also most overrepresented in *S. typhimurium*, which has a global AT content close to 50%. A second observation is that particular words are frequently shared as most underrepresented in unrelated organisms, such as CTAG.
Table 8.1. Tetranucleotides that are most strongly underrepresented and overrepresented in 8 bacterial genomes. Tetranucleotides that were also observed as strongly under- or overrepresented in other organisms (18 bacterial genomes were analyzed) are shown in bold. Blue italic sequences are palindromes.

<table>
<thead>
<tr>
<th>Organism</th>
<th>% AT</th>
<th>Most underrepresented</th>
<th>Most overrepresented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deinococcus radians R1</td>
<td>33.0</td>
<td>CTAG</td>
<td>AAAA</td>
</tr>
<tr>
<td>Mycobacterium tuberculosis H37Rv</td>
<td>34.4</td>
<td>CTAG</td>
<td>TCGA</td>
</tr>
<tr>
<td>Mycobacterium leprae N</td>
<td>42.2</td>
<td>CCCC</td>
<td>TCGA</td>
</tr>
<tr>
<td>Salmonella typhimurium LT2</td>
<td>47.2</td>
<td>CTAG</td>
<td>AAAA</td>
</tr>
<tr>
<td>Escherichia coli K12 MG1655</td>
<td>49.2</td>
<td>CTAG</td>
<td>TTTT</td>
</tr>
<tr>
<td>Haemophilus influenzae KW20</td>
<td>61.1</td>
<td>CTAG</td>
<td>CGCC</td>
</tr>
<tr>
<td>Helicobacter pylori 26695</td>
<td>61.1</td>
<td>CTAC</td>
<td>AGCG</td>
</tr>
<tr>
<td>Campylobacter jejuni NCTC 11168</td>
<td>69.4</td>
<td>ACGT</td>
<td>GCTT</td>
</tr>
</tbody>
</table>

Finally, we observe that the most underrepresented word is a palindromic sequence (explained below) in most given examples, in contrast to the overrepresented words. In fact, the second most underrepresented sequence (not listed in the table) is also commonly a palindrome: of the 3 most underrepresented words analyzed in 18 genomes, 31 out of 54 were found as palindromic words. There is a biological explanation for this observation; palindromic tetranucleotides are potential targets for restriction / modification enzymes, and are thus avoided by selective pressure of mobile DNA containing restriction enzymes.

DNA repeats within a chromosome

We have introduced short DNA sequences as words, and in analogy a word (or sequence) that is repeated somewhere else on the chromosome is called a repeat. The simplest form is a repeat of dinucleotides arranged in tandem (such at GAGAGA or CTCTCTCTCT). Such tandem repeats of short oligomers are called simple repeats. Longer and more complex words that are repeated in tandem are called direct repeats (a simple repeat is thus a special form of a direct repeat: shorter words that are repeated more often). Figure 8.1 shows sequences with simple and direct repeats, and illustrates that direct repeats can sometimes have spacers between the repeat unit. All these are classified as local direct repeats, meaning the repeat units are closely together (‘local’) and on the same strand, in the same direction (‘direct’).

1 The presence or absence of spacers is ignored in the rest of the chapter.
Figure 8.1. Different kinds of repeats are drawn for double-strand DNA. To the left, from top to bottom are two types of direct repeats: simple repeats, and local direct repeats without and with a spacer. At the bottom, a schematic chromosome is depicted with two global direct and one global inverted repeat. To the right are (top to bottom) local inverted repeats, palindromes (a special class of inverted repeats), mirror repeats and everted repeats. In this figure the repeat unit is shaded in light blue when read in the biologically correct direction (from 5' to 3') and its complementary sequences are pink. Note that for palindromes the repeat unit is its own complement, here given in blue only. The dark blue arrows indicate a repeat unit read in the wrong direction (from 3' to 5') for mirror and everted repeats.

Repeats can also be present on a global scale, where the repeat units are separated by a longer stretch of DNA (up to many kbp apart). There is no agreed definition separating local from global repeats and the distinction is arbitrary. Figure 8.1 also illustrates what inverted repeats are: here, the word is repeated on the complementary DNA strand. Again, inverted repeats can be present with or without a spacer, and can be found on a local or global level.

In texts, a palindromic word can be read both backwards and forwards (such as ‘level’) and the same is true for a palindromic DNA sequence; however, we have to read backwards on the complementary strand in order to read the same word. As a result, a DNA palindrome is an inverted repeat where the repeat units are on one and the same location (see Figure 8.1). Four-letter and six-letter palindromic
sequences are often recognition sites for restriction enzymes\(^2\), and are particularly common. Longer and more complex palindromes can also occur in a bacterial genome, for instance as part of the excision machinery of a mobile element.

Two more classes of repeats should be mentioned, although they are less common: mirror repeats and everted repeats. Note that in both the repeated sequence is read in the wrong direction of DNA: the non-biological direction from 3’ to 5’.

The reason why such biological ‘nonsense’ repeats are nevertheless important, is that they can form particular structures, but these are outside the scope of this book.

---

**Figure 8.2.** Schematic of how to find and label repeats in a chromosome. To the left is depicted how global repeats are identified and ‘scored’ on the genome, to the right the procedure for local repeats is displayed. Searching on the same strand would identify direct repeats, whereas the analysis of the complementary strand would provide inverted repeats.

How we identify repeats and report their frequency in a genome sequence is schematically drawn in Figure 8.2. It is another example of a sliding window analysis that is normally performed within a chromosome.\(^3\) In the depicted examples, we search for in the frequency of global and local repeats. Global repeats are searched with a window of 100 nt, for which the best match somewhere along the chromosome is identified and a repeat score is placed at position 51 in the window. Thus, not only a perfect repeat with 100% identity for the complete repeat unit is identified, but also imperfect repeats, with decreasing scores for ‘repeats’ that are less than perfect.

---

\(^2\) Although most restriction enzymes recognize palindromic sequences, some use non-palindromic recognition sites. For some palindromic tetra- and hexamers, on the other hand, no restriction enzymes have been identified yet.

\(^3\) In theory, a DNA sequence could be repeated on two chromosomes of a multi-chromosome organism, but in practice this is uncommon and we don’t routinely search for this.
In the next step, the window is shifted by 1 nucleotide, after which the analysis is repeated. The score is again entered at position 51 of the window, which will be next to the nucleotide that already received a score. With this method, the complete chromosome sequence will be scored, with the exception of 50 nucleotides at the begin and at the end of the artificially opened chromosome. Analyzing a window of 100 bp on the same strand over the complete genome would give you all global direct repeats. Searching for matches on the complementary strand would identify global inverted repeats.

A similar approach is followed for local repeats but here the window size determines how close the local repeats are allowed to be 100 bp. As a consequence of this choice repeat units separated by more than 100 bp are screened as ‘global’ in our analysis. The searched repeat unit is chosen shorter as well (up to 15 bp in the shown example). Note that in some cases one can have multiple types of repeats simultaneously (e.g., TATATATATA is a simple dinucleotide, but it is also a tetranucleotide repeat, as well as a direct repeat and an inverted repeat). If this is the case, all repeats are scored. We will see examples where the same location is reported for local direct and local inverted repeats, which is usually due to simple repeats.

In general, bacterial genomes have few global repeats, especially compared to eukaryotes. An example of a perfect global direct repeat present on a bacterial chromosome would be multiple copies of the same gene, as for multiplied rRNA operons (see the next chapter). Imperfect direct local repeats, on the other hand, are frequently gene orthologs, resulting from gene duplication and evolutionary forces (more on this will be covered in Chapter 14). Repeats can also occur in non-coding sequences, such as the repeats known as ‘CRISPRs’ (Clustered, Regularly Interspersed Short Palindromic Repeats). These repeats might provide an important defense against phages (Sorek et al., 2008). These examples show that repeat sequences play an important role in a number of biological processes, including those that drive evolution in a significant way.

We screened the presence of global and local direct and inverted repeats in a number of bacterial phyla and plotted their frequency (Figure 8.3). For each given bacterial chromosome, we calculated the fraction containing local (15 bp region, within a 100 bp window) and global repeats (the best match of a 100 bp region found anywhere else in the chromosome) of at least 80% or more similarity. These scores were averaged per phylum.

---

4 As the sliding window will eventually reach the repeat units identified earlier, and here will find its partner sequence again, every repeat unit is marked. Thus, every repeat marked has a counterpart somewhere else on the chromosome that is also marked.
As can be seen, the frequencies of the various classes of repeats vary both between them, and amongst bacterial phyla. For the analyzed bacterial genomes, on average less than 3% of the DNA comprises of global repeats and approximately twice as much form local repeats. The observed fraction representing global direct and global inverted repeats are similar for a particular bacterial phylum. Local direct and local inverted repeats are particularly common in Actinobacteria, Firmicutes and Proteobacteria; in fact, these three phyla also have the highest fraction of global repeat DNA. In contrast, Bacteroides contain relatively few local repeats compared to their global repeat fraction. Global repeats are uncommon in Spirochaetes and few global inverted repeats are found in Chlamydiae. From the finding we can conclude that, biologically speaking, the distance between repeat units is significant, thus giving different frequencies for global versus local repeats. There are at least two reasons for this. First, the local repeats are strongly correlated with AT content as we will see (and global repeats are not), and second, local repeats can be involved in DNA structures as will be discussed below.

In general, organisms in which global direct repeats are common also have high numbers of global inverted repeats. Exceptions to this rule are *Photorhabdus luminescens* and *Mycobacteria*, both having relatively many global direct but few global inverted repeats. At the other end of the spectrum the correlation holds as well: organisms that have few global direct repeats, such as Chlamydiae, also have few global inverted repeats. In Figure 8.4 a scatter plot is shown (top left) of global direct and global inverted repeats of 330 bacterial genomes, and a weak correlation is visible. From the top right panel of the figure it can be seen that, in-

**Figure 8.3.** Average relative frequencies of direct and inverted repeats in bacterial genomes.
stead, the frequencies of global and local direct repeats do not correlate. The lower two panels show that local direct and local inverted repeats do correlate strongly, as do mirror and everted repeats. A scatter plot of local direct versus local everted or local mirror repeats would also show a strong correlation. In conclusion, presence of various kinds of local repeats is strongly correlating, but the presence of global repeats does not predict the presence of local repeats.

**Figure 8.4.** Scatter plots for the frequency of repeat in 330 bacterial genomes: global direct repeats vs. global inverted repeats (top left), global direct vs. local direct repeats (top right), local direct vs. local inverted repeats (bottom left) and local mirror vs. local everted repeats (bottom right). Organisms with extreme high values are not shown. The color codes of the dots is similar to Figure 8.3.
Introduction to the DNA Repeat Atlas

Plots such as those shown in Figures 8.3 and 8.4 are suitable to compare general trends. However, for a single genome an atlas is more informative. Figure 8.5 shows a standard Repeat Atlas for the *Pelotomaculum thermopropionicum* chromosome. *P. thermopropionicum* is a member of the Clostridium group of bacteria, and is involved in anaerobic methanogenic biodegradation of organic matter. Figure 8.5 shows both local and global repeats within the *P. thermopropionicum* chromosome. The GC skew lane has been added to allow comparison of this atlas with Base Atlases we’ve seen in the previous chapter.

The outermost circle of the Repeat Atlas shows the presence of global direct repeats which are relatively common for this organism: 6.1% of the chromosome contains global direct repeats of at least 80% identity or more. The colors in the atlas reflect the scores as described for the window analysis above: a score of ‘9’ represents 90% or greater match, a score of ‘5’ is more than 50% identity but less than 60%, etc. The colors are scaled such that they vary from white, for less than 50% to darkly colored for more than 75% identity.

Figure 8.5. Repeat Atlas for the *Pelotomaculum thermopropionicum* chromosome. Global repeats are plotted on a fixed average scale, whereas simple repeats (local direct repeats up to 15-mers) and local direct and inverted repeats are plotted as 3 standard deviations from average.
The next circle shows the global inverted repeats, on the same scale. Note that many of the inverted repeats match the direct repeats, that is, the sequence is repeated both in the forward and reverse directions: these are (imperfect) palindromes. Repeats present in only one of the first two circles identifies the non-palindromic global direct and global inverted repeats. The next circle represents simple repeats (green in the figure); for these, all oligomers up to 15-mers are examined for repeats within a window of 100 bp. The simple repeats appear to occur more frequently in the top half of the chromosome (between 9 o’clock and 3 o’clock). Following the annotation circles are the other local repeats.

In contrast to many bacteria, for this genome there are fewer local than global direct repeats (4% of the chromosome). Note the interesting long stretch of local direct repeats around 2 Mbp. As these are flanked on both sides by palindromes and appear more AT-rich than the rest of the genome, this is a likely candidate for some mobile DNA element such as a transposon. The local inverted repeats seem to be localized in the top half of the chromosome as well.

The innermost two lanes are base composition properties (discussed in the previous chapter): the GC skew, which clearly shows the DNA replication origin near the 12 o’clock position, and the % AT. The latter shows dark red regions that have more than 60% AT, and turquoise for less than 40% AT; the chromosomal average is 47% AT, which is indicated by the turquoise colour for most of the inner circle. Most of the coloring for % AT is at the top of the atlas, mainly for AT-poor sequences. This fits with the observation that repeats are more common in that half as well: the occurrence of repeat sequences and extremes in % AT are correlated, as we will see below.

**Local DNA repeats are related to chromosomal AT content**

At the beginning of this chapter we have already seen that the frequency of purine or pyrimidine stretches correlates with base strand preference. It also interesting to relate the presence of local repeats to the global base composition, such as the % AT. It can be expected that local repeats increase in frequency when the base composition diverges from 50% (for example, the more A’s and T’s in a genome, the bigger the chance that these two will produce repeats). Figure 8.6 shows the fraction of the genome containing local direct repeats, plotted against the % AT. The explanation for the ‘smiley face’ shown in Figure 8.6 is simple. For a chromosome of 50% AT content and no strand preference, there is an equal chance (one in four) of finding any given base randomly at a position within a sequence. The chance of finding GATC (or any other tetranucleotide) would be 1 in 256.
However, if a genome’s composition would approach 100% AT, then the chances would be very low to find GATC, since there are so few G’s or C’s in the genome. In fact, though, the chances of finding a tetramer composed of only A’s and T’s are actually greatly increased, because the genome consists of mainly those two bases. Similarly, for a GC-rich genome the chances of finding a repeat are also greater than in the 50% AT genome. Thus, as one moves away from 50% AT content, either more AT-rich or more GC-rich, the chances of finding multiple copies of short oligomers (which is what local repeats are) will increase. This explains the strong correlations we saw for local repeats within a genome in Figure 8.4. Since global repeats are not or far less affected by base composition, no correlation was observed between local and global repeats.

**DNA structures related to repeats in sequences**

Some structures can only form when a particular type of local repeat is present in the sequence. This is true for *cruciforms*, which require a local inverted repeat (including palindromes), as shown the right in Figure 8.7. *Insertion sequences* (responsible for insertion and excision of mobile elements) are specifically inserted in palindromic repeats in the bacterial chromosome as they depend on formation of the cruciform structures (Tobes and Pareja, 2006). *Hairpins* (half a cruciform formed in single-stranded molecules) are especially common in RNA, and as such are structural requirements for functional rRNAs and tRNAs. Indeed, mobile ele-
ments that insert themselves with the help of Insertion Sequences are frequently found inserted in a tRNA gene. Slipped strand structures require a local direct repeat (left part of Figure 8.7) and from the schematic drawing it is obvious how slipped strand structures can be involved in gene excision or duplication. Slipped strands are responsible for phase variation in *Neisseria* spp. and other organisms.

![Figure 8.7. Frequently occurring structures that are dependent on local repeats.](image)

### More on Structure Atlases

The Structure Atlas has already been introduced in the previous chapter. Here, three new features are explained: DNAse I sensitivity, propeller twist and protein formability. DNAse I sensitivity is a reflection of the anisotropic flexibility or bendability of a particular DNA sequence. It is based on a trinucleotide model, empirically determined from experimental data. The trinucleotide values range from -0.280 (rigid) to +0.194 (very ‘bendable’ towards the major groove). Smoothing over a large region, which is necessary for viewing entire genomes, tends to smooth out local differences in bendability. Another measure of the helix rigidity is the average propeller twist of a region of DNA, since the propeller twist angles are inversely related to rigidity of the DNA helix in crystals (el Hassan *et al.*, 1996). Thus, a region with high propeller twist would give a quite rigid local helix, and conversely regions that are quite flexible would have a low propeller twist. Propeller twist values were originally obtained from crystallographic data (el Hassan *et al.*, 1996), with the exception of the dinucleotide step from T to A, which is a theoretical estimate (Gorin *et al.*, 1995). The final measure is called protein induced deformability, and is based on dinucleotide values from protein induced deformation of DNA helices as determined by examination of over a hundred crystal structures of DNA / protein complexes (Olson *et al.* 1998). The dinucleotide values range from 2.1 (the least deformable dinucleotide) to 12.1 (i.e., the dinucleotide step from C to G that is most readily deformed by proteins). Thus, on this scale, a larger value reflects a more deformable sequence whilst a smaller value indicates a region where the DNA helix is less likely to be changed dramatically by proteins.
Figure 8.8. Structure Atlas of the ‘hairy root’-inducing plasmid of *Agrobacterium rhizogenes*. The grey blocks indicate the T-DNA segment and a type 4 secretion system.

Figure 8.8 shows a Structure Atlas of the plasmid of *Agrobacterium rhizogenes*. The bacterium lives in root nodules of plants and uses the root-inducing (Ri) plasmid to induce hairy root syndrome in the plant. The syndrome is caused by integration and expression of the plasmid’s ‘Transferred DNA’ or T-DNA in the plant genome. This T-DNA carries genes that induce uncontrolled plant tissue proliferation (oncogenesis) and genes coding for synthesis of unusual amino acids, that are then used by the bacteria as a nitrogen source. The T-DNA locus has some striking properties (Figure 8.8). It is predicted to consist of strongly curved DNA (blue in the first outmost circle) that would easily melt (red in the second circle).

The position preference lane tells a mixed story of local DNA likely to be tightly wrapped (magenta) and DNA more likely to be highly expressed (green regions). The propeller twist lane is deep red, indicating the DNA helix in this area is stiff or rigid, and the same is indicated by the brown color for protein deformability. The relative AT content is extremely high, which correlates with the high likelihood that this region would easily melt. It turns out that T-DNA is first transferred to the host-cell cytoplasm, where it is made single-stranded, wound around various bacterial and host proteins, and imported into the plant cell nucleus. Knowing this, it is obvious why the DNA has the physical properties predicted in the Structure Atlas of Figure 8.8.
Have a look at the Structure Atlas of Figure 8.9 now. This represents the large plasmid of *Desulfotalea psychrophila*, for which we have seen the chromosome Base Atlas in the previous chapter (Figure 7.6). This strain contains a small plasmid as well, which won’t be considered here. The large plasmid contains a section between 90 and 95 kbp (around 9 o’clock) with physical properties similar to those of the T-DNA locus of the *A. rhizogenes* plasmid. In this region, genes for LPS biosynthesis are present (glycosyl transferases, capsular polysaccharide biosynthesis proteins), which can be predicted to be highly expressed. In contrast, an area around 40 kbp shows the complementary colors for all displayed structural parameters. In that segment, genes encoding the enzymes acetyl-CoA synthetase, and pyruvate dehydrogenase subunits are present. These enzymes are involved in the tricarboxylic cycle and, predicted on the observed local structures, their genes seem to be expressed at a lower level than other genes.

---

5 Although *Desulfotalea psychrophila* is commonly described as an ‘extremophile’ and can grow at temperatures below freezing, its optimal growth temperature is 10°C. It should be noted that such temperatures are rather common on Earth and not at all extreme.
The Genome Atlas: our standard method for visualization

There are many features one can plot on a Genome Atlas (and there are many ways in which Genome Atlases can be visualized). So far, we have introduced the Base Atlas, Structure Atlas and Repeat Atlas. By experience we have learned that a lot of information is contained in a Genome Atlas that combines the most informative lanes of these three.

A typical Genome Atlas, shown in Figure 8.10, starts with the three DNA structural properties that were introduced in the previous chapter; the combination of these various parameters is actually very informative. The annotation circles for protein-coding genes (separated for the two strands both strands) and RNA follow next; annotation of RNA and protein genes will be further treated in the next two chapters. Next, repeat circles are added, usually either global or local repeats but always both direct and inverted (as for the global direct and inverted repeats depicted here), as this provides the most information. For instance, the two rRNA loci are easily identified by very strong global direct repeats; their annotation color is light blue. The GC skew lane (plotted as fixed average) and AT skew (as deviation from average) that we know from the Base Atlas follow next.

Figure 8.10. Genome Atlas for Pelotomaculum thermopropionicum.
Striking in the *P. thermopropionicum* Genome Atlas (for which the Repeat Atlas was earlier shown, in Figure 8.5), is the region around 2 Mbp that contains strong colors for each circle represented but for the GC skew (which is unusually low here). It is an AT-rich segment that is highly bent (blue on the intrinsic curvature scale), will melt easily (red for stacking energy) and tightly wrapped (position preference magenta). The gene coding density is lower than for most of the genome. It is flanked by global direct and inverted repeats - these are strong indications for the presence of a mobile element. Indeed, the GenBank file identifies a number of integrase and transposase genes, although a clear mobile element is not annotated.

Compare this to the area around 2.5 Mbp, again with high scores for intrinsic curvature and stacking energy but where a strong position preference is lacking, and the GC skew and gene density are normal. The mostly hypothetical proteins encoded here can be expected to be highly expressed. Two very AT-rich regions (around 1.6 Mbp and 2.65 Mbp) are similar in their strong intrinsic curvature and stacking energy properties but they don’t contain global repeats (so they are not copies). In fact, all these three regions encode genes involved in modifying structures exposed to the outside world. Note also the dark green regions in the position preference circle, which correspond to genes that can be highly expressed. As we will discuss in the next chapters, where you are along the chromosome strongly influences expression.

**Concluding remarks**

In this chapter we finally introduced all parameters that we consider informative to represent on a Genome Atlas. Other features exist that can be visualized in a similar manner, but essentially the Genome Atlas is now complete. This means it is time to turn to the genes that are actually encoded in the DNA sequence.

**References:**


Box 8.1: Comparison tools and visualization tools used in this chapter

Scatter plots of repeat frequencies. Pre-calculated repeat frequencies for sequenced bacterial genomes are available on http://www.cbs.dtu.de/services/GenomeAtlas. Select one or more organisms of interest and then choose ‘Repeats’ from the buttons of available tables. The resulting list can be sorted at wish. Using the link to ‘Compare Within Search’ one can select from a number of parameters which one to plot on the X-axis and Y-axis after which a scatter plot is generated automatically.
9 Transcriptomics: Translated and Untranslated RNA

Outline  Genes can code for either RNA that is translated (to produce proteins) or that remains untranslated, such as ribosomal RNA (rRNA) genes, of which a genome must contain at least one set. Many bacteria have multiple rRNA copies, up to ten or more, and the number of rRNA copies roughly correlates to growth rate. Transfer RNA (tRNA) genes remain also untranslated, and although in principle only 20 tRNAs are needed to encode the 20 amino acids, most bacteria have more copies. The redundancy in the genetic code predicts 61 possible different tRNA genes but no organism is known to contain them all. Instead, particular tRNA genes are duplicated so that more than 100 tRNA genes can be present. The presence of multiple copies of a particular tRNA gene can be related to the frequency of use of the corresponding amino acid in protein genes, or to its presence in highly expressed genes. The frequency of codon usage in all predicted mRNAs can be plotted in a rose plot. Some genomes exhibit strong bias in which codons are used. Usually the bias is in the third position, which generally relates to the total AT content of the genome. Finally, there are many other genes that do not code for proteins, rRNA or tRNA, but are transcribed and have a biological function. As an example, tmRNA is introduced.
Introduction

The Central Dogma of molecular biology dictates that DNA is the template for RNA from which protein is made, or, in terms related to the genomics era, the genome defines the transcriptome, which defines the proteome. This chapter marks the middle part of this chain of events, the production of the transcriptome. The previous two chapters discussed the DNA sequence in genomes, and proteins will be the focus of attention in the next chapter. For most bacteria, nearly all of the genome is transcribed, and most of that is translated into proteins. However, there are some RNAs that remain untranslated. This chapter will review three general types of untranslated RNA genes: the ribosomal RNA (rRNA) genes, which make up the RNA part of ribosomes, the transfer RNA (tRNAs) genes, and the transfer-messenger RNA (tmRNA) gene, which provides a fail-safe for erroneous translation. Apart from these three types other ‘small RNAs’, (sRNAs) exist, which play a role in regulation of gene expression. Since relatively little is known about sRNAs beyond a few well-studied organisms, they are not treated here.

Counting rRNA and tRNA genes

The three rRNA genes required for a functional bacterial ribosome are usually coded side-by-side on an rRNA operon. In contrast, the ribosomal proteins, necessary to make a functional ribosome, can be found in clusters or spread around the genome. All cells require at least one rRNA operon but frequently bacterial genomes contain multiple copies. The number of rRNA operon copies recognized in bacterial genomes is given to the left of Figure 9.1. As can be seen the current maximum is 15 rRNA operons present in Photobacterium profundum (Photobacteria are bioluminescent bacteria living in association marine crustaceans; P. profundum can live under high pressure in deep and cold waters).

Cells also require at least 20 tRNA genes, one for each of the 20 amino acids. Since the genetic code is redundant, in theory 20 tRNAs would be able to cover all 61 possible protein-encoding codons, as a wobbling third base pairing allows a single tRNA to serve various codons through an imperfect match. However, Ser and Leu are coded by different 6 codons, both requiring at least two tRNAs, so

---

1 Mitochondria are considered ‘organelles’ and not cells. They use a slightly different genetic code than the eukaryotes in which they reside, and they perform their own translation with the help of many proteins that are encoded on nuclear DNA. For translation, mitochondrial DNA contains two specialized rRNA genes, 12S and 16S rRNA; a 5S rRNA is usually missing. Mitochondria contain between 0 to 27 tRNAs. Any missing tRNAs are imported from the nucleus.

2 We will ignore the existence of selenocysteine, the 21st amino acid that is present in a limited number of bacteria and in many eukaryotes, using the UGA codon.
that the minimally required number of tRNAs for organisms using the standard genetic code is 22. In addition, individual tRNA genes can be present as multiple copies. To the right of Figure 9.1 it can be observed that bacterial genomes contain more tRNAs than the minimally required 22, with some genomes containing more than a hundred tRNA genes.

Figure 9.1. The number of rRNAs (to the left) and tRNAs (to the right) in 595 bacterial genomes. Note that 110 genomes have the minimally required single rRNA operon (darker shade). None of the genomes have the minimum of 22 tRNA genes and few organisms have fewer than 30 tRNA genes.

In general, organisms with many rRNA operons also have many tRNA genes, and these can have fast doubling times under optimal conditions. For instance, *Bacillus cereus* (a soil-dwelling opportunistic pathogen that can cause food poisoning) with 108 tRNA and 13 rRNA genes, is capable of cell division in only a few minutes. Note, that we don’t know the optimal growth conditions for all bacteria sequenced to date, so that their maximum rate of division cannot always be observed. In contrast to fast growers, organisms with a single rRNA operon and a low number of tRNA genes can take days to divide, such as *Mycobacterium tuberculosis*. Thus, just by counting the number of rRNA and tRNA genes present in a genome, it is possible to make an educated guess about the likely lifestyle (or at least the comparative growth rate) of an organism. Possibly, though, high numbers of tRNA and mRNA copies do not always correlate with rapid growth rate, as there may be other biological principles dictating copy numbers as well.

A closer look at ribosomal RNA

Bacterial ribosomes are made up of approximately 50 proteins and 3 RNAs, as shown schematically in Figure 9.2. These rRNA constituents are known as 23S,
16S, and 5S\(^3\). Together with ribosomal proteins they build the large subunit (LSU or 50S subunit, which harbor the 5S and 23S rRNA molecules) and the small subunit (SSU or 30S subunit in which the 16S rRNA is included).

![Diagram of ribosome composition](image)

**Figure 9.2.** The composition of bacterial ribosomes. One ribosome combines one large subunit (at the top, made up of 23S rRNA, 5S rRNA and 31 proteins) and one small subunit (constituting 16S rRNA and 21 proteins).

The 16S rRNA molecule is about 1540 nucleotides long, and forms many stem-loop structures, as shown in Figure 9.3. The 3-dimensional structure is essential for its function, and there has been a strong selection for preservation of its sequence within bacteria. This is probably because the RNA molecule interacts with so many proteins at the same time, which, together with its essential role in translation, would conserve its sequence. Some changes in a stem region are acceptable as long as there is a compensating mutation on the complimentary part of the stem, thus preserving the overall structure. On the other hand, loop regions have less freedom to mutate as these interact with ribosomal proteins and hence there is evolutionary selection for conservation. (As will be discussed in the next chapter, proteins have different strategies in terms of sequence conservation.) Finally, it should be noted that in some organisms rRNA genes come in pieces, with either introns interrupting the rRNA sequences (which are spliced out and the rRNA is ‘glued’ into one piece) or with intervening sequences that are removed by processing after which the fragmented rRNA is incorporated as such into the ribosome.

Many commonly used methods for finding sequence similarity, such as BLAST, will perform poorly when aligning rRNA and other non-coding RNA sequences, as they search specifically for identity in letters, not in structures. Maybe because of this, and also due to an emphasis on finding exclusively protein encoding genes, the rRNAs are not always annotated in bacterial genomes. Occasionally

\(^3\) The ‘S’ in 5S, 16S and 50S \textit{etc.} stands for ‘Svedberg unit’, a unit for the sedimentation coefficient determined by ultra-centrifugation, which increases in a non-linear way with molecular weight.
they are annotated on the wrong strand, or the genes are annotated to be much larger than they should be. It is therefore necessary to test the reliability of a 16S rRNA gene sequence extracted from a GenBank file of a bacterial genome.

Figure 9.3. The small (16S) ribosomal RNA gene forms multiple hairpin structures. In the 2-dimensional projection to the left, four domains can be recognized. To the right a three-dimensional model is shown of the 30S ribosomal subunit consisting of 16S rRNA (in light blue and magenta), 5S rRNA (dark blue and brown) and proteins (mixed colors).

Transcription of the three rRNA genes is usually initiated in front of the 16S rRNA, and continues all the way to produce an RNA molecule with all three units terminating after the 5S rRNA. This polycistronic RNA is then cleaved by a specific endo-RNase and further trimmed at the 3’-end by other enzymes. The location of the rRNA operon (or operons) on a microbial genome can vary, but it should always be present on the chromosome, as the presence of an rRNA operon is the criterion by which a DNA molecule is called a chromosome. Nevertheless, there are currently 9 bacterial ‘plasmids’ described that bear an rRNA operon, so the definition is not always strictly applied. Some organisms have an extra copy of 5S rRNA; extra copies of 16S or 23S are less common but do exist. The 16S and 23S genes are frequently separated by one or more tRNA genes. There is a strong preference for the rRNA locus to be located on the leading strand (which corresponds with the positive strand for only half of the chromosome, as explained in Chapter 7), especially in organisms with multiple copies. Multiple rRNA operon copies can occur in tandem or on single loci, but they are usually located close to the origin of replication. The multiple operons are kept identical by concerted evolutionary processes.
Comparison of rRNA sequences

Comparison of rRNA sequences within a bacterial species is hardly informative, as the rRNA sequence is usually one of the key parameters by which bacteria are speciated. Indeed, rRNA genes within the species are generally conserved although the *Internally Transcribed Spacer* (ITS) that separates the 16S and 23S genes shows sequence variation in many α-Proteobacteria. This ITS is sometimes used for genotyping. Moreover, particular single-nucleotide mutations are known to infer resistance to certain classes of antibiotics, such as macrolides or aminoglycosides.

Comparison of rRNA sequences between species, on the other hand, is a useful exercise. 16S rRNA trees are used to investigate long-distance evolutionary relationships. In fact, one of the key differences between eubacteria and archaea is their difference in rRNA (other important differences are the way their membrane is built and details in their replication and transcription machinery). Figure 9.4 shows a phylogenetic tree, based on the CLUSTAL alignment of 10 different 16S rRNA genes. These come from 10 different organisms (archaea and bacteria) whose genus name starts with the letter ‘A’. Although 16S rRNA, is used as one of the key markers for taxonomy, from the obtained tree one can see that the Proteobacteria phylum is split up, with the δ-Proteobacterium *Anaeromyxobacter* removed from the other members of this phylum.

As a rule of thumb, a 5% difference in 16S rRNA is considered sufficient to call two organisms different species. As pointed out by Broughton (2003), using sequence variation of any marker gene for taxonomic purposes presupposes that
evolution of the genome progresses at a constant rate and that genes are inherited from generation to generation and not shared between existing cells via horizontal gene transfer. We know that this is not always the case, and even 16S rRNA genes are not always conserved within the species.

The most striking example is the actinomycete *Thermobispora bispora*, which has two functional pairs of 16S rRNA genes that between them differ at 98 nucleotides, producing a difference in 6.4% of the complete gene. It combines this with three near-identical 23S rRNA copies. Other examples exist where lateral gene transfer of 16S rRNA genes has likely occurred, which weakens the power of 16S phylogeny as a determinant for taxonomy (van Berkum et al., 2003). Whether 16S sequences should be put in the foreground of taxonomic divisions is still debated amongst bacteriologists. It should be noted in this context that taxonomy aims to give a meaningful, descriptive and unique ‘name tag’ to an organism, whereas phylogeny describes the evolutionary history of that organism (or of genes therein). Clearly the two serve different purposes and their essentials do not always overlap. Chapter 14 will further deal with the differences between phylogenetic and evolutionary signal in genomes and their individual genes.

Figure 9.5. GenomeAtlas of chromosome 1 of *Photobacterium profundum*. The positions of rRNA loci are indicated. Multiple rRNA operons can be present per locus. All 14 rRNA operons are located in the top half of the genome, near the origin of replication.
**rRNA and AT content**

The genes coding for rRNA are visualized in the annotation lane of our Genome Atlas using a separate color. Figure 9.5 shows a Genome Atlas for chromosome 1 of the *Photobacterium profundum* genome, which contains 14 rRNA operons. These rRNA operons are all positioned at the top of the atlas where there are 4 locations of 3 operons each. They are localized in areas with low position preference (bright green in the third lane from outside), which indicates that they are likely highly expressed. Their location close to the origin of replication is related to high expression, too, as will be explained in the next chapter. This marine organism apparently has many copies of rRNA that can undergo high expression - signs of a high growth rate under optimal conditions. In this organism, the regions around the rRNA operons are more GC-rich than the rest of this chromosome (with a total AT content of 58%), as can be seen from the local coloring in the innermost circle. Since rRNA genes are amongst the most strongly conserved genes between organisms, this raises the question if their AT content is maybe less variable than that of complete bacterial genomes.

We compared the global AT content of 567 bacterial genomes (46 archaea and 521 bacteria) with that of their coding sequences, their rRNA genes and their tRNA genes. As can be seen in the box-and-whiskers plots in Figure 9.6, clearly the rRNA operons are lower in AT content than the complete genome and the tRNAs are lower still.

![Figure 9.6. Comparison of AT content in bacteria (left) and archaea (right) for complete prokaryotic genomes, coding sequences, tRNAs and rRNAs. The latter are also analysed for 16S rRNA, 23S rRNA and 5S rRNA.](image)

The variance in AT content is smaller for RNA genes of archaea than those of bacteria, but this could be caused by the smaller sample size. Within the three RNA genes 5S rRNA has generally the lowest AT content but, despite its smallest size, also contains most variance. If the current AT content of RNA genes would reflect the AT content of the ancestors in which they originated, the conclusion
would be that these ancestors generally had lower AT contents (around 40%) than bacteria on average have nowadays. More likely, though, the base content of the RNA genes is a reflection of functional constraints. The differences in AT content between classes of genes most probably reflect the fact that evolutionary forces are not the same on every gene of a genome.

**Genes encoding transfer RNA**

The second important class of non-translated RNA molecules are tRNAs, which are essential components for the translational machinery. Without tRNAs (that have to be loaded with a specific amino acid by amino acid-tRNA acyltransferases) the genetic code represented by mRNA could not be translated into amino acids. As for rRNAs, tRNA molecules contain 3 important hairpin structures. Figure 9.7 shows a 2-dimensional and 3-dimensional model of a typical tRNA.

![Figure 9.7](image)

**Figure 9.7.** Two-dimensional (left) and three-dimensional model (right) of a tRNA.

As stated previously, tRNA genes can be present with variable degrees of redundancy. Which amino acids are represented by more than one tRNA copy can be related to the frequency of their use in protein genes, or in their presence in highly expressed genes. As with rRNAs, the location of tRNA genes near the origin of replication allows higher levels of transcription.

The tRNA genes of bacteria are frequent ‘docking’ sites for mobile elements. Notably, particular transposons preferentially integrate themselves in tRNA genes. Genome islands are also frequently inserted in a tRNA gene, as will be seen in Chapter 14. tRNAs are frequently found in a tandem arrangement on the chromosome, although they can also be dispersed. Some bacteria carry tRNA genes on plasmids. In *E. coli*, tRNA genes that are arranged in tandem are produced from
one promoter, and a polycistronic messenger is cleaved by a specific endo-RNase to separate the single tRNA molecules (tRNAs can also be combined with rRNA or mRNA on polycistronic messengers). Extra nucleotides are subsequently removed from the maturating tRNA by exo-nucleases. Most of the work on tRNA maturation has been done in *E. coli* (the same applies for rRNA maturation) but observations from other organisms show that co-evolution has resulted in similar processes (using different RNAses) in, e.g., *Bacillus subtilis*. Removal of leader sequences at the 5’-end is simpler than maturation at the 3’-end; the latter requires an AU-rich element (AUE for short) that is conserved in nearly all organisms (Li *et al.*, 2005).

**Genes coding mRNA: comparing codon usage between bacteria**

As explained in Chapter 1, mRNA contains both coding and non-coding sequences. The non-coding parts are mostly at the 5’- and 3’-end (and in between genes in case of polycistronic messengers) and the AT content in these non-coding regions is usually higher than in the coding part. For most of their part mRNAs code for proteins, and we will take a closer look at their base composition.

*Figure 9.8.* Codon usage rose plots for three organisms with different AT content: *Thermus thermophilus* on the left, *Bdellovibrio bacteriovorus* in the middle and *Buchnera aphidicola* at right. The frequency scale used in the rose plots is represented at the top. Below each rose plot is the sequence logo plot of the base frequency in the first, second and third position of the codons. Bases represented at the top of the sequence logo plots are most frequent at that position.
Codon usage is one of the determinants for AT content, as most of the bacterial DNA codes for proteins, and the third base of each codon therein allows the most variability without changing the amino acid. In Figure 9.8 the relative codon usage is shown in a rose plot for three different organisms, AT-rich *Buchnera aphidicola* (with 74.7% AT), *Bdellovibrio bacteriovorus* with 49.4% (a small species that feeds on other bacteria) and *Thermus thermophilus* (obviously a thermophile), with 30.5% AT. The way in which the data are ordered around the rose plots was already discussed in Chapter 5. From the plots it is clearly visible that *T. thermophilus* prefers G or C at the third base position, whereas *B. aphidicola* mostly uses A and T. From the sequence logo plots it appears that there is far less difference in use of bases in the first and second codon position. *B. bacteriovorus* displays no preference at all for any of the bases in its codons.

Figure 9.9 shows a different way of looking at codon usage. This time the relative codon frequencies in all predicted genes are represented per amino acid, again in three organisms with variable AT content: this time *Streptomyces coelicolor* (an soil Actinobacterium that produces more than half of the known natural antibiotics, with a genome AT content of 27.9%), *Lactobacillus delbrueckii* strain Bulgaricus, the starter culture organism for making Bulgarian yoghurt (with 50.3% AT) and *Wigglesworthia glossinidia* (a Proteobacterium living as endosymbiont in the tsetse fly, with 77.5 % AT). Blue bars represent codons for which the third base is a G or C, red are codons with an A or U as the last base. The amino acids are grouped in functional groups as indicated.

From Figure 9.9 it is clear that the most frequently used amino acid is not the same for each of these organisms: whereas Ala is most common for *S. coelicolor*, it is Leu for *L. bulgaricus* and Ile for *W. glossinidia*. This is not surprising: if *W. glossinidia* were to use Ala frequently, it would have to have a less AT-rich genome, as the codons for Ala require at least G and C as the first and second base. Conversely, *S. coelicolor* would have difficulty with the frequent use of Ile. Note, however, that these three amino acids all belong to the functional group of aliphatic amino acids. The same replacement pattern is seen for the positively charged amino acid. An alternative ‘strategy’ is observed for the two negatively charged amino acids; here, one of the two codons is differently preferred between the organisms. Thus, wherever possible an organism with an extreme AT-content will suit its needs of amino acids and codons according to the preferred codons. Interestingly, the three amino acids most frequently used by *S. coelicolor* are also those that are represented 6 times in the genetic code but this could be a coincidence.

The relative frequency of codons (and amino acids) present in mRNAs also determines which tRNAs are mostly needed for translation. For many proteins translation is the rate-limiting step, provided transcription is sufficient (regulation of transcription is discussed in the next chapter), and within the translation process
the amount of available tRNA can be rate limiting. Thus, a correlation exists between relative codon and amino acid usage in different mRNAs and the efficiency of their translation.

**Figure 9.9.** Relative frequencies of codon usage *S. coelicolor*, *L. bulgaricus* and *W. glossinidia*. Red represents codons with A or U at the third position, blue has G or C. Since there is only one codon for Met and for Trp these is not shown.
Other non-coding RNA: tmRNA

When bacterial translation was introduced in Chapter 1, some details were excluded for simplicity. There are far more factors involved in correct translation than the ones schematically represented in Figure 1.4, of which tmRNA, short for transfer-messenger RNA, is the most intriguing. Both the start and termination of translation are carefully regulated, the latter by release factors that recognize stop codons on the mRNA, bind to them and by doing so force the ribosome to quit the messenger. If faulty mRNA is produced that isn’t recognized by release factors (for instance, due to a transcription error, premature termination or a break in the molecule), release factors can’t bind and ribosomes pile up on the unfortunate messenger. As such stalled ribosomes can no longer be recycled, such a situation should be avoided.

Bacteria have a quality control system to prevent such stalled ribosomes, employing tmRNA, a non-coding RNA of only about 200 nucleotides in length. tmRNA has affinity to stalled ribosomes and corrects the process by what is called trans-translation. The tmRNA gene (also called ssrA) is conserved in all bacteria but frequently missed in the annotation of older genomes, as historically it has not specifically been searched for, despite the fact that it is strongly conserved and thus easy to identify. However, most of the more recently sequenced genomes now include tmRNAs in their annotation. The tmRNA gene is present in all eubacteria and in some phages and mitochondria, but it is absent in archael genomes.

Figure 9.10. 2-dimensional structure of tmRNA. The dark-shaded area at the top is the m-RNA-like domain, loaded with Ala. The grey-shaded structures are pseudoknots (a topological structure) The purple-shaded domain is coding for a degradation tag, ending in a stop codon. To the right a is spliced tmRNA represented that is found in some bacteria, in this case Prochlorococcus marinus.
Figure 9.10 shows a 2-dimensional structure of tmRNA, from which it is obvious that it has both tRNA and mRNA features (hence its name). Some organisms (e.g., Prochlorococcus marinus or Caulobacter crescentus) have a split tmRNA whereby base pairing keeps the two parts together, as is depicted at the right. Like a tRNA, tmRNA is loaded with an amino acid, Ala in this case, but it doesn’t contain an anti-codon. Instead, it contains a short reading frame ending with a stop codon in its mRNA-like domain, which works as a decoy for release factors to unblock the stalled ribosomes. The Ala-loaded tmRNA binds to the stalled ribosome as if it were a tRNA (stalled ribosomes are still translating, adding identical copies of one amino acid to the unfortunate protein-in-the-make) and thus replaces the defective mRNA. The ribosome will continue to translate the mRNA domain of tmRNA that encodes a 10 amino acid long degradation tag: by adding this to the defective protein, it is marked for degradation and can no longer do harm.

![Figure 9.11. Phylogenetic tree of tm RNA from 12 bacterial species.](image)

Similarly to the 16S rRNA analysis, we selected tmRNA genes from bacterial species whose name begins with ‘A’, this time using 12 species (these are not necessarily the same as were used for the 16S rRNA analysis). The phylogenetic tree in Figure 9.11 shows that this well-conserved gene contains a much weaker taxonomic signal than 16S rRNA: various bacterial phyla are split up.

**Concluding remarks**

The RNA-coding genes within a genome can reveal important biological information, even when they do not code for proteins. This chapter provided some examples of the kind of analyses that can be done on RNA sequences, but many methods developed for DNA analysis can be applied to RNA as well. The next chapter will address regulation of expression, including regulation of transcription.
References:


Box 9.1: Comparison and visualization tools used in this chapter

Phylogenetic tree of 16S rRNA. The evolutionary history of 16S rRNA sequences in the tree of Figure 9.4 was inferred using the Neighbor-Joining method (Saitou and Nei, 1987). Bootstrap values were calculated according to Felsenstein (1985). The evolutionary distances were computed using the Kimura 2-parameter method (Kimura 1980) and are expressed in units representing the number of base substitutions per site. The rate variation among sites was modeled with a gamma distribution (shape parameter = 1). All positions containing alignment gaps and missing data were eliminated only in pairwise sequence comparisons (pairwise deletion option). There were a total of 2310 positions in the final dataset. Phylogenetic analyses were conducted in MEGA (Tamura et al., 2007).
10 Expression of Genes and Proteins

Outline  Protein expression is the combined process of gene transcription, translation of the resulting mRNA, post-translational processing of the protein, and possibly secretion. All of these steps can be regulated, some even at various levels. Transcription is regulated at multiple levels, where downstream effects and feedback mechanisms can form intricate regulatory networks. This makes the prediction of protein expression complex, but for nearly each step prediction tools are available. Once produced, a bacterial protein can be located in the cytosol, be trapped in the membrane of the cell, or be secreted into the medium. The cellular location of a protein can be predicted with acceptable accuracy. Secreted proteins are of particular interest in medical microbiology, since for instance toxins must be secreted in order to kill cells. Based on localization of proteins and on predicted properties of antigen binding, it is possible to identify putative candidate peptides for vaccine development.
Introduction

All genetic information is present in every cell, but not all this information is used all the time. Some genes code for proteins that need to be constantly replenished in a living cell, whereas other proteins are only needed under particular conditions. Proteins may be needed in high amounts at one time, but lower amounts may suffice at other times. When functional protein is actually produced from a gene, we say that the protein is being expressed. The production resulting in the final, functional protein is the combined effect of transcription, translation, protein folding, and any necessary protein modification or secretion. The term gene expression is often used to describe transcription exclusively, for instance when referring to ‘regulation of gene expression’ which more precisely describes regulation of transcription, or ‘expression microarrays’ when the presence and concentration of particular mRNA is measured. For clarity, it is important to distinguish gene expression (transcription) from protein expression, as the latter includes transcription but involves further steps. Bacteria have multiple strategies to regulate protein expression and this chapter will concentrate on those regulatory strategies that can be predicted by bioinformatical analysis. Prediction tools are based on experimental observations, and since a tremendous amount of work has been done on E. coli, this species will serve in most of our examples in this chapter.

Comparing gene expression and protein expression

For the production of proteins mRNA is required, but not every mRNA produces protein, and when it does, not always in equal amounts. Furthermore, mRNA usually has a shorter lifetime than the protein it codes for, so that a particular protein can still be detectable in the cell when its corresponding mRNA is no longer present. Figure 10.1 shows an Expression Atlas comparing the amount of detectable mRNA with that of detectable proteins from an E. coli culture. The colors represent relative concentrations that were experimentally determined: the stronger the color, the more abundant the product. The concentration of specific mRNAs was measured by expression microarray analysis and the protein data came from 2D-gel electrophoresis combined with N-terminal protein sequencing (Ussery et al., 2001). Maybe surprisingly, only few genes give an expression signal at all (on mRNA or protein level). The mRNA signal was deliberately tuned down in order to envisage only highly expressed genes. Moreover, only abundant protein spots were quantified and thus shown.
Figure 10.1. An Expression Atlas for *E. coli* K-12. The outermost circle represents protein concentrations with color intensity as a measure for relative concentration, based on 2D-gel electrophoresis. The next lane depicts the relative amount of mRNA, based on microarray experiments. Around the atlas the mRNA gene names (in blue), and protein names (in black) are indicated. The position of rRNA genes is given in green. However, rRNA was removed from the RNA sample before microarray analysis so that it was not detected.

As would be expected, for quite a few genes a strong expression signal for both mRNA and protein is visible; there are however also regions with abundant protein, but little visible mRNA. This could be explained by a stable protein but an unstable mRNA; conversely there are regions with lots of mRNA but little protein, where perhaps translation of the mRNA is not taking place. We observed significant differences in relative concentrations of mRNA and protein in roughly a third of all genes. Further, even if we detect a given amount of a particular protein in a cell, not all of this might be in the biologically active conformation.

Transcription is the first step in protein expression and as such it is often tightly regulated, although there are genes that are constitutively transcribed, apparently producing mRNA all the time at a more or less constant rate. As protein expression begins with transcription, and transcriptional regulation is very important in bacteria, the first and major part of this chapter deals with this subject, after which we briefly turn to regulation of translation and then on to protein secretion.
Part 1: Regulation of transcription

Regulation of transcription can occur at various ‘levels’, varying from global, rather non-specific regulation to very specific mechanisms that act on one or a few genes only. Some aspects involved in transcription regulation can be predicted quite accurately from genome sequences but others are more subtle and difficult to predict, as depicted in Figure 10.2. In the figure the various mechanisms of regulation are sorted in seven levels, from global and relatively nonspecific - with high numbers of key players involved - to local, highly specific, where only a few molecules may be needed for a regulatory effect. The relative position of some of these mechanisms in the scheme is debatable, and the division in these ‘levels’ is not that strict. Nevertheless, stratifying the multiple strategies of transcriptional regulation in this way is helpful to recognize some of the underlying principles of regulation of cellular processes. We will therefore look at each level in a bit more detail, giving examples of prediction tools as we go along.

**Figure 10.2.** Transcription of genes can be regulated at various levels (numbered 1 to 7), from global, non-specific regulatory levels to local, very specific levels, listed in the grey box. The blue box gives rough estimates of how many copies of a particular factor would be present in the cell. Some aspects involved in regulation can be accurately predicted from genome sequences, here listed to the left, whereas others, listed to the right, are not easily predicted. The various regulation mechanisms are further explained in the text.
**Seven levels of transcriptional regulation**

The first and most global level how transcription can be regulated is dictated by gene location: DNA near the origin of replication will be present in higher amounts in a replicating cell, so that more copies of genes located here are available for RNA production. The picture in Figure 10.2 shows how replication introduces multiple copies of genes located near the origin (like the gene represented in blue in the picture) compared to genes closer to the origin (such as the red gene). In the previous chapter it was introduced that rRNA genes, which are relatively highly transcribed, are preferentially positioned near the origin of replication.

The second level of transcription regulation depends on variation in the local DNA structure, regulated by histone-like proteins. Some DNA stretches are more tightly packed around histone-like proteins than others. The tighter the package, the less easy transcription can be started. One histone-like protein, IHF, was already introduced in Chapter 2 and its binding site was shown in a sequence logo in Figure 2.7. The information of that sequence logo was used to search through the genome of *E. coli* for regions that match the consensus. For searches like these, weight can be given to specific conserved sequence positions. The analysis included a search for binding sites for FIS, another histone-like protein.

![Figure 10.3. Chromatin Atlas for the *E. coli* K-12 genome. The first lane shows the same mRNA expression data as in Figure 10.1, now smoothed over a window of 100,000 bp. The second and third lanes predict IHF and FIS binding sites, respectively, also smoothed. Note the opposed correlation between the presence of histone-like protein binding sites and high gene expression.](image)
A search for IHF sites in the whole *E. coli* genome will give thousands of hits scattered along the chromosome. However, by smoothing the data with a large window one mimics the concerted activity of multiple protein copies binding to the DNA. In Figure 10.3 the mRNA expression data from in Figure 10.1 are smoothed using such a large window in what we call a Chromatin Atlas. Now it is visible that the half of the genome where the origin of replication resides contains more highly transcribed genes, probably due to the gene copy effect discussed above.

Compare this smoothed RNA expression lane with the next two lanes that identify (smoothed) DNA density regions based on predicted histone-like protein binding (IHF and FIS). Now it can be seen that regions that contain highly expressed genes are predicted as less dense, due to lack of binding of IHF and FIS, whilst regions with lots of binding sites have fewer transcriptional activity (Ussery *et al.*, 2001). This chromatin silencing is well known in eukaryotes, but is often ignored in bacteria. The example shows that location of a gene is important: transcription can be generally higher or lower as dictated by the location of genes, and regions in the genome can be recognized that generally undergo higher or lower expression. Such global regulation is super-imposed on more local regulation.

When transcription is regulated more specifically, levels 3 to 7 of Figure 10.2 can apply in various combinations. The level of transcription depends highly on the efficiency with which transcription is initiated, as the machinery functions with constant speed once started (with a few exceptions). Before we discuss the next level, we have to introduce level 7, which defines the promoter and other regulatory sequences in front of (‘upstream of’) genes, acting in cis to the gene in question. Transcription is started by the binding of a sigma factor to a specific site, which is traditionally called the promoter, although the latter term sometimes includes all binding sites that regulate transcription. The strength of a promoter determines how much mRNA is produced, and this is mostly dictated by the efficiency by which the sigma factor binds, which again depends on the promoter sequence and on how easily the local DNA will melt.

![Figure 10.4](image)

Figure 10.4. Three different situations of transcribed genes. The two green genes are divergently transcribed, each from its own promoter. The two orange genes are transcribed in the same direction, but again both have their own promoter, whereas the two red genes are co-transcribed. The latter two situations can of course apply to genes present on either the positive or negative strand.

Not every gene is transcribed from a promoter located directly upstream of its coding sequence, so when and where to search for promoters? Figure 10.4 shows three possible situations of promoter location relative to the genes they regulate.
Co-transcribed genes present on polycistronic messengers share a common promoter. The term *operon* is mostly reserved for polycistronic genes that code for proteins involved in a single biological process. The *lac* operon of *E. coli*, containing genes involved in lactose transport and metabolism, is a classical example. However, some bacteria produce polycistronic mRNA coding for products that are not functionally closely related. Gene direction is of course the first indicator whether to search for a promoter: two genes that are divergently transcribed need the presence of two promoter sequences, as a promoter only works in one direction.

The sigma factors (of which cells have various kinds) that bind to promoters are highly conserved between species, as was introduced in Chapter 6. Sigma 70 binds to DNA at two locations: once about one helical turn upstream of the start of transcription\(^1\), which is a distance of about 10 nucleotides, thus the term ‘-10 position’, and again about 2.5 helical turns further upstream (-35 position). The two binding sites together form the -10, -35 promoter (other values apply to other sigma factors). This ‘two-grips’ binding allows the protein to grab hold of the DNA, inducing a torque to twist open the double helix, as depicted in Figure 10.5.

\[\text{Figure 10.5. Initiation of transcription in bacteria. In the first step, sigma factor binds to the DNA on two locations (in case of Sigma 70 the -35 and -10 sites). RNA polymerase (a complex of two \(\alpha\), one \(\beta\) and one \(\beta'\) subunit) binds next, after which the DNA wraps around the protein. Sigma induces local strand separation so that RNA polymerase starts producing RNA (in green), The sigma factor is then released and RNA polymerase proceeds along the DNA, moving with a local bubble of melted DNA (indicated by the arrows).}\]

\(^1\) This numbering does not use the translation start of the gene as 1, since the messenger starts upstream of the translation start (see Chapter 1, Figure 1.5). Any signal acting in trans that is found in the 5'-direction of a reference point is referred to as ‘upstream’. Thus, a sigma binding site is positioned upstream of the transcription start (reference point).
Obviously, the availability of the required sigma factor is a way to regulate transcription, which we call the third level of regulation. In particular, transcription of genes dependent on Sigma 54, stress-response sigma factors and sporulation sigma factors, is regulated by the abundance of that particular sigma factor. Sigma factors themselves are of course produced from their own genes, and they also need a sigma factor for their own production, which means there are complete interconnected networks of regulation, with positive and negative feedback mechanisms, to regulate the amount of mRNA for most genes in a cell. At the bottom (or top, depending on the view point) of these networks are signaling proteins that ‘feel’ extra-cellular and intra-cellular conditions, so that the cell is optimally equipped to respond to any changes.

The promoter upstream of a particular gene or operon will determine which sigma factor binds, and the choice of sigma factor we call level 4. Sigma factor binding sites can be predicted with varying accuracy. As the binding site of Sigma 54 is strongly conserved, its prediction is relatively easy. A sequence logo of the Sigma 54 promoter consensus of *E. coli* is shown in Figure 10.6. The consensus binding site for Sigma 70 is also shown, and as can be seen this is less conserved, even within this single species.

![Figure 10.6. Sequence logo plots of the consensus binding sequence for Sigma 54 (top), and of Sigma 70 (bottom), based on 37 (for Sigma 54) and about 500 (for Sigma 70) *E. coli* binding sites. The binding site of Sigma 54 is more strongly conserved than that of Sigma 70.](image-url)
For other sigma factors the recognition sequence may be conserved between closely related species, but a ‘general’ promoter sequence that will work in every bacterial cell unfortunately doesn't exist. The nature and number of sigma factors present in a bacterial species can vary considerably, though within genera there is more conservation. For this reason, genes that have to be artificially expressed in a foreign bacterial species (a frequent application in biotechnology) are usually constructed behind an endogenous promoter, to ensure transcription is correctly initiated in the hosting cell.

The Sigma 70 consensus binding site of *E. coli* shows a strong TA conservation at the beginning of the -10 box, which is also known as the ‘TATAAT’ or ‘Pri-bnow box’. It has been known for a long time that a pyrimidine followed by a purine (a purine/pyrimidine step) destabilizes the helix, so that it is easier to open up, and of the possible purine/pyrimidine steps 'TA' take the least amount of energy to pop open.

The binding of sigma factor to a promoter can be affected by other proteins that either stimulate initiation of transcription, or block it. Such regulation is usually quite specific, which is why we call this level 6. In Figure 10.2 level 5 is reserved for mRNA stability but which of these two levels is more global or more specific is hard to say, as their effects vary depending on the genes in question. Positive regulators are called transcription factors, and their genes can be present on a different genome location to the gene they actually regulate: they act *in trans*. Inhibitors of transcription are also trans-acting proteins. There are many (hundreds) of different types of bacterial transcription factors. One example of an important bacterial mechanism to regulate gene expression via transcription factors in an environmentally responsive manner is the ‘two-component signal transduction’ system, where one protein (usually embedded in the membrane) responds to some change in the environment, and upon stimuli modifies (by phosphorylation) another protein, which then changes shape and now activates expression of certain genes through binding to DNA. Protein prediction of transcriptional regulators and two-component regulators is relatively straightforward, as these proteins have a conserved DNA-binding domain that can easily be recognized. However, it is more difficult to predict to which sequence the regulator will bind, and thus expression of which gene is affected is not always clear.

Level 5 of transcription regulation, acting once transcription has begun, is given by the stability of mRNA. Messengers with a short half-life will produce less protein than mRNA with a longer half-life. Once an mRNA is produced, it will sooner or later be actively degraded, since otherwise there would be unlimited and unwanted production of protein. The half-life of mRNA, generally in the order of seconds or minutes (the average half-life of mRNA in *E. coli* is 2 minutes) can be extended by structures that protect the molecule from degradation. Moreover, non-coding RNAs (ncRNA) can bind to mRNA in a specific way and regulate
transcription by hampering or even stimulating translation. When several genes are coded by one mRNA molecule (as in polycistronic RNA), the spacer sequences separating these genes can form structures that prevent RNA polymerase to continue, thus lowering the relative mRNA amount of downstream genes in favor of upstream genes. Degradation of polycistronic mRNA would also affect downstream genes more than the genes located towards the 5'-end. Again, ncRNA can change the relative translation efficiency of genes within a polycistronic mRNA.

Stability of mRNA, affecting its half-life, is mostly dependent on stem loop structures at the 3'-end of the molecule (RNAse, the enzyme degrading RNA, nibbles nucleotides off from the 3'-end). Prediction of stem-loop structures are mainly based on the presence of local inverted repeats in close proximity to the stop codon of a gene, and these can be visualized in a zoom of a Repeat Atlas.

Finally, at level 7 we turn to the relative strength of the promoter itself, as even promoters using the same sigma factor can vary with effects on transcription initiation. The trans-acting factors that stimulate or inhibit transcription of specific genes, mentioned as level 6, will also bind to specific cis-acting elements in the proximity of a promoter. Moreover, upstream of the -10 and -35 sites one can frequently find an UP element, which helps to wrap the DNA around RNA polymerase as it induces a natural curve in the DNA. RNA polymerase is a large molecule, about the size of a eukaryotic nucleosome; more than 100 bp of DNA needs to wrap around the RNA polymerase, and in bacterial cells this is a major source of protein-constrained supercoiling, being responsible for about half of the restrained supercoils.

Sometimes, binding sites for histone-like proteins can be found upstream of a promoter. These proteins will bind and bend the DNA helix, again facilitating the wrapping of the RNA polymerase around the DNA. In some experiments, a piece of naturally curved DNA was introduced to replace the IHF binding site, with the result that the promoter was always active. But if there is an IHF binding site present, the binding of IHF can act as a transcriptional positive regulator. However, we started this section describing how multiple binding sites for IHF and FIS can induce chromatin silencing; IHF binding sites must be carefully dosed in order to stimulate transcription.

As the presence and distance of all these cis elements have very local effects we also include these at level 7. Obviously, levels 5 and 7 are part of the same mechanism, as a trans-acting factor cannot act without the cis-acting element to which it must bind. All of these regulatory mechanisms have been described for bacterial genes, alone or in combinations. Publications describing laboratory observations on (regulation of) transcription are mostly based on experiments where specific mRNA molecules are made visible, and the effect of mutations introduced
in promoters, trans- or cis-acting factors, or the mRNA itself, can be quantitatively determined. When enough data are available, general patterns can be recognized, which can serve as the basis to build prediction models. Applying such prediction models to sequenced bacterial genomes can sometimes lead to surprising observations that can weaken or even contradict experimental evidence. Laboratory experiments are frequently conducted with a small selection of strains only, and these have often been cultured in the laboratory for many years. Such strains may have adapted to growth on culture media, and may not at all be true representatives of the bacterial populations living in the real world. An example of conflicting results obtained by wet-lab work and computer predictions was found when we analyzed the ribosomal gene promoters of *E. coli*, as discussed below.

**Promoter structures: the rRNA promoter**

Each *E. coli* rRNA operon is transcribed from a strong Sigma 70 promoter, which has been studied in extensive detail. In fact, each operon has two promoters, known as P1 and P2, labeled in the schematic drawing of Figure 10.7. Different growth conditions determine which of the two is active; in exponentially growing cells P1 is used almost exclusively, and cells in stationary phase use P2. In addition to the -10 and -35 sites, an UP element is present in front of each promoter, and a number of fis elements (the binding sites for FIS protein) are present upstream of P1.

![Figure 10.7. The two Sigma 70-dependent promoters of the ribosomal RNA gene operons in *E. coli*. The drawing at the top shows the position of two promoters in front of an rRNA operon (not on scale). The enlarged view in the middle shows that three to five fis sites, followed by an UP element, are present in front of promoter P1; an UP element also exists for P2. At the bottom the consensus for fis element, UP element, -35 and -10 boxes are shown, based on experimental evidence described in the literature.](image-url)
As literature data were available on the consensus sequences of the fis, UP, -35 and -10 regions (Hengen et al., 1997; Estrem et al., 1998; Huerta et al., 2003), shown in Figure 10.7, we used this information to specifically search for all of these elements in front of all rRNA loci in 28 E. coli and Shigella genomes.

The resulting sequence logos for these binding sites are shown in Figure 10.8. The sequence logos we obtained for the -10 box and the -35 region are different, and more strongly conserved, than the general Sigma 70 logo of Figure 10.6; this can be explained as fewer sequences were compared this time, all of which represent strong promoters. However, the sequence logos are also quite different from the experimentally acquired data summarized in Figure 10.7. The genome sequence data tell us that the consensus sequence of P1 and P2 are similar but not identical. The UP element is essentially made up of two A-tracts, spaced about a helix turn. Finally, the fis binding site is strongly anchored by a G and C with an AT-rich region in the middle.

If we now compare the obtained sequence logos between closely related organisms, as in Figure 10.9, it becomes apparent that even the promoters of the rRNA gene loci are not completely conserved within the gamma division of the Proteobacteria, as the examples for E. coli, Salmonella and Yersinia species illustrate.
Figure 10.9. Sequence logos for consensus sequences of elements of P1 and P2 promoters for the 65 genomes belonging to the *E. coli* / *Shigella* group at the top (also shown in Figure 10.8), based on 23 genomes, for 18 *Salmonella* genomes (middle) and for 21 *Yersinia* genomes (bottom).

**DNA Structural properties of promoters**

Knowing the general structural characteristics of promoters, as depicted in Figure 10.5, can we find and predict promoters based on local DNA structural properties? In general, a strong promoter can be recognized by a curved region of DNA, followed by a region that is rigid (the spacing between -35 and -10), and then a region that can easily melt (the -10 region). Based on these characteristics, the presence of a strong promoter can be recognized from sequences with some accuracy. However, the relative position to an open reading frame is essential, and in AT-rich genomes many sequences are ‘promoter’-like by chance.

Figure 10.10 shows a Structure Atlas of the region around one of the seven rRNA operons in an *E. coli* K-12 genome. The three lanes shown are the DNA structural properties normally present in our standard Genome Atlas: DNA curvature, helix stability, and position preference (see Chapter 7 where these parameters were introduced). Based on the general pattern a curve is expected upstream of a promoter (a blue region in lane A of Figure 10.10), closely followed by a region that is rigid (purple in lane C), and then a region which will melt easily (red in lane B). These features should be in the vicinity of a gene that of course should be transcribed in the correct direction. We have added two additional lanes to this plot. Lane E shows where the DNA helix is expected to open up when it is supercoiled; for promoter sequences in general we expect a high probability (blue color) of opening. Lane F shows the destabilizing energy, visualizing how much energy...
it would take to open the helix (also called SIDD values, for Superhelical Induced Duplex Destabilization). This lane shows up red when little energy is required for melting. The two bright red bands in the bottom lane, coinciding with blue in lane E, point to the position of P1 and P2. Working up from this, the parameters fit for lanes A to C as well, whereby P1 fits all expectations of a strong promoter. These are the only likely promoters in the shown 10,000 bp region. Thus, combining all measures of the plot, these structural features can help locate potential promoters in regions upstream of genes.

**Part 2: Regulation of translation**

As we concluded from the first figure in this chapter, there are some proteins that are very stable, even if the mRNA only lasts for a few minutes (for example the outer membrane protein OmpA of *E. coli* is stable for days). In bacteria, translation starts even when transcription has not been fully completed, as the two processes occur in the same location (in eukaryotes, mRNA has to be translocated out of the nucleus before translation can occur). Thus, while the mRNA chain is growing, ribosomes already bind to it and translation begins. The rate-limiting steps in translation are probably the availability of tRNAs, and the loading of these with the correct amino acids, for which specialized enzymes are needed, one for each amino acid. How efficient the loading of tRNAs will occur cannot easily be predicted, as availability of amino acids depends on the metabolic status of the cell. But the relative abundance of tRNA is largely dictated by the redundancy in their copies, and these can be deduced from a genome sequence. In addition, particular
tRNA genes can be located near the origin to increase their relative gene copy numbers during replication. Location within the rRNA gene operon also indicates a high need of that tRNA. These parameters can be predicted, and from this it can be deduced which proteins preferentially use those tRNAs (as dictated by their codon use and amino acids), indicative of highly expressed proteins.

The regulation works at both extremes: highly-expressed proteins will preferentially use amino acids whose codons are covered by abundant tRNAs, whereas proteins using amino acids for which the tRNA is ‘uncommon’ are slowed down during translation. Thus, codon usage will influence translation, affecting protein expression. This is nicely demonstrated when a gene is artificially introduced to be expressed in a foreign cell. For optimal expression (and sometimes, to obtain any produced protein at all), its codon usage needs to be adapted to the hosting cell, in order to allow efficient translation. Codon adaptation tables are available for efficient translation in *E. coli*, for instance. A specialized database is dedicated to such ‘synthetic’ genes\(^2\). The natural equivalent of ‘artificially’ introduced genes would be genes that have undergone horizontal gene transfer. The common perception is that such genes can be recognized by their sub-optimal codon usage (and, related to this, their aberrant AT content). However, codon usage is most likely a strong success factor for any horizontally acquired gene: those genes that already have suitable codon usage (and that are beneficial, or at least not disadvantageous, to the organism) are most likely to be maintained in the population (Medrano-Soto *et al*., 2004).

**Part 3: Protein modification and cellular localization**

Some proteins are active immediately after they are produced, others may instead require final steps such as proper protein configuration, particular modification or trafficking through the membrane in order to be biologically active. Some of these steps can be predicted with good accuracy, whereas other steps are still largely a black box.

**Protein folding affecting gene expression**

Correct protein folding is essential for protein function, and when its configuration is not correct a protein may partially or completely loose its function; it has changed into a so-called ‘denatured’ state. Protein denaturation can be caused by extreme temperatures (heating, freezing), pH or ion concentrations. Some proteins

\(^2\)http://www.evolvingcode.net/codon/sgdb/index.php
can fold back into their native, fully functional configuration when the extreme conditions are restored to normal, but more frequently denaturation is permanent. In a living cell this would be detrimental, and a protective stress-response is in place to repair the damage. Specialized proteins called chaperones (also called chaperonins, heat-shock proteins or stress proteins) can recognize and bind to denatured protein, and by means of protein-protein interaction the denaturation can be reversed. When the damage is beyond repair the denatured protein is degraded.

The expression of chaperone genes is regulated so that they are over-expressed under stress conditions, mostly by stress-response sigma factors. However, they are not only needed to repair damaged proteins; many ‘heat-shock’ proteins are produced at low amounts in the unstressed cell, to bind to newly-produced proteins and thus enable or accelerate their correct folding. Chaperones are particularly important for proteins that, in their native state, are embedded in the membrane, and thus are not soluble in the cytosol. Chaperones hold these in a state that allows their transportation to the right cellular location (see below), before they are released and spring into their correct, native form.

Post-translational modification

Bacterial proteins can be refolded, glycosylated, phosphorylated, acylated, or circularized, to name a few of the more common types of modification that take place after production of the ‘raw’ version of the protein. Although the enzymes responsible for such activities are easy enough to predict, there are currently no standard tools available to accurately predict their substrates. In this respect, eukaryotic research is more advanced, so most, if not all tools available on the web have been specifically designed for eukaryotic sequences. Unfortunately, these do not predict reliably on bacterial proteins. This specialization of microbial bioinformatics is still very much under development.

Extra-cellular location of proteins

A bacterial protein can be located in the cytosol, be trapped in the membrane of the cell, or be secreted into the medium. Protein location is not at all trivial. For instance, most antigens (proteins against which the immune system of a given host will raise a response) are on the surface of a bacterium. These are possible candidates for vaccine development. Membrane-embedded proteins, on the other hand, are frequently involved in adaptation to environmental conditions (such as the membrane components of two-component regulators). Secreted proteins, those that leave the cell completely, are frequently toxic to other bacteria or to host cells,
and are sometimes injected into host cells to rapidly kill them. So prediction of protein location adds important information to protein annotation.

Gram-positive bacteria produce proteins that can be cytosolic, embedded in the membrane (facing the cytosol or the outside of the cell, or spanning the membrane and sticking out on both ends), or secreted into the medium. Gram-positive bacteria have only one membrane barrier to be crossed (their cell-wall is a porous structure that doesn’t provide a secretion barrier) Since Gram-negative bacteria have two membranes, proteins can either span the inner membrane, be localized in the periplasm (the space between inner and outer membrane), or be embedded in the outer membrane. Which of these locations apply can be predicted to some degree based on their amino acid sequences. Absence of hydrophobic regions and transmembrane domains will result in a predicted soluble, cytosolic protein unless there are secretion signals present to steer the protein across the membrane.

Secreted proteins, sometimes called effectors, need the necessary machinery to leave the cell: the secretion components that build the secretion systems. Gram-negative bacteria have a number of different secretion systems in place. Secretion thus depends on two things: the candidate protein that needs to be secreted (which needs to be identified as such by the cell), and an active secretion machinery. Already in 1971 it was suggested that short dispensable peptides facilitate the trafficking of proteins to specific secretion machinery (Blobel and Sabatini, 1971). These peptides are commonly known as signal peptides (or signal sequences) and are usually located at the N-terminus of the protein; they are nearly always cleaved from the preprotein during protein translocation across the cytoplasmic membrane, resulting in a secreted ‘mature’ protein. Several distinctive bacterial N-terminal signal peptides have been identified for targeting the preprotein to the different secretion pathways.

The most abundant secretion signal targets the protein into the Sec-dependent secretion system by means of a Sec-signal peptide. Genome comparison has confirmed that the Sec-dependent secretion system is conserved in all bacteria. Preproteins carrying a Sec signal peptide are recognized by specific chaperones that carry them to dedicated translocation pores. This pore translocates the proteins in an unfolded (denatured) conformation. Although the Sec signal peptide shares little sequence identity across bacterial phyla, their structural features are conserved. Sec signal peptides are made up of a short (7-15 amino acids) alpha helix domain flanked on the N-terminal end by a short track of preferably positively charged amino acids (Arg and Lys), whereas a short track of small amino acids (preferably Ala and Ser) separate the alpha helix from the cleavage site for signal peptidase I. Slight modifications of this general pattern apply to lipoproteins that are secreted by Sec (von Heijne, 1989).
An alternative translocation system can be found in some Gram-positive and Gram-negative bacteria, known as ‘twin Arginine’, or Tat secretion. It depends on a different secretion machinery but the secretion signal is similar to that of Sec though generally shorter in length, and contains two adjacent arginines (Berks, 1996). In contrast to Sec, the Tat secretion system translocates proteins that have already entered their fully folded configuration in the cytoplasm.

A minority of the secreted proteins use alternative secretion systems. Notably, the ABC transport system (also known as type I secretion) is worth mentioning. ABC transporters are found mainly in Gram-negative organisms and depends on three components: a transporter containing an ATP-binding cassette (hence the name ABC transporter), a membrane fusion protein anchored in the membrane to create a connection between the inner and outer membrane (in case of Gram-negative bacteria), and an outer membrane trimeric protein, which provides the connection to the outside of the cell (Delepelaire, 2004). Particular bacterial toxins, enzymes such as lipases or proteases, and some bacteriocins leave the cell by Type I secretion. Usually, the ABC transporters are specific for one effector, and their genes are co-located on the genome. Secretion is not determined by an N-terminal signal. Instead, the C-terminal end of the protein determines T1SS, and this is not cleaved during secretion.

Some Gram-negative bacteria can possess one of two highly specialized specialized secretion systems. A Type IV secretion system (T4SS) forms a hair-like structure protruding from the cell, resembling fimbriae (pili). Proteins are secreted through these into the medium or into target cells, though particular T4SS secrete DNA instead of protein (and some can do both). Type III secretion, which was discovered earlier than T4SS, is only found in some Proteobacteria. It consists of a needle-like structure crossing both membranes and protruding on the outside of the bacterial surface. This structure allows translocation of specific proteins from the bacterial cytoplasm directly into the environment. More importantly, the ‘needles’ can penetrate host cell membranes, so that the secreted effectors are released in the host cellular cytoplasm. Type III secretion systems (T3SS) are mostly found in pathogenic members of the Enterobacteriaceae including some plant pathogens. The secretion system is composed of approximately 20 structural components, and requires additional regulatory proteins and chaperones to be functional. Some of its structural components bear resemblance to flagellar components, and a common genetic ancestor is probable. The signal directing a protein to T3SS is not yet resolved. However, all T3SS structural genes discovered so far have been found to reside on genome islands (further discussed in Chapter 14) and effector proteins are encoded within those islands. Thus, co-localization on the genome is the strongest determinant for predicting effector genes whose product is secreted by a T3SS.

3 Sec-dependent secretion is sometimes called type II secretion
**Prediction of the secretome**

Identifying the *secretome*, that is all the surface-expressed and secreted proteins in a given organism, is of high importance to pharmaceutical and biotechnological industry. Many genome sequence studies provide an estimated number of secreted proteins. These estimates vary largely between bacteria but with the current knowledge on secretion machineries and their signals, it is quite possible to accurately predict which proteins make up the secretome of Gram-positive bacteria, and by which mechanism each protein is secreted. The secretome of Gram-negatives is a bit harder to predict. Box 10.1 lists some web based tools to predict the presence of secretion signals, both for individual proteins and in predicted proteomes.

**Box 10.1: A few websites of tools to predict secreted bacterial proteins**

<table>
<thead>
<tr>
<th>Website</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.cbs.dtu.dk/services/SecretomeP-2.0">http://www.cbs.dtu.dk/services/SecretomeP-2.0</a></td>
<td>Predicts signal-independent protein secretion using SecretomeP.</td>
</tr>
<tr>
<td><a href="http://www.membranetransport.org">http://www.membranetransport.org</a></td>
<td>TransportDB is a relational database describing the complete predicted membrane complement of a sequenced organism.</td>
</tr>
<tr>
<td><a href="http://www.tcdb.org">http://www.tcdb.org</a></td>
<td>A database detailing the Transporter Classification system, analogous to the EC system for classification of enzymes.</td>
</tr>
</tbody>
</table>
Using the first three prediction tools listed in Box 10.1 we predicted the secretome of bacteria from their genome sequence. Figure 10.11 shows an estimate of fractions of the proteome expected to be secreted as they carry a Sec, lipoprotein or Tat signal, respectively, for a number of bacterial phyla. The results are represented in a violin plot. This is a type of box-and-whiskers plot where the shape of the box is adjusted for local data density. Outliers are included as elongations (very pronounced for the Proteobacterial data).

As can be seen from the figure, the Proteobacteria and Bacteroides secrete the largest fraction of the total proteome, but there is a lot of variation between species within these phyla. Sec-dependent secretion is more frequently used than the other secretion signals, although again Bacterioides uses the lipoprotein secretion signal quite frequently.

Antigen and epitope prediction

Another group of proteins that receive a lot of attention are bacterial antigens and their epitopes. Antigens are proteins that elicit an immune response in an animal host. Nearly all proteins would produce a humoral immune response (that is, antibodies would be produced that would specifically bind to them) upon injection in an animal host, especially if the protein were denatured. One can boost such an immune response by the addition of an adjuvant. However, during infection, the host’s immune response is most effective in recognizing bacterial surface struc-
tures. Thus, antigen prediction largely concentrates on proteins present on the bacterial surface. Proteins that are secreted but remain attached to the membrane may be likely vaccine candidates. The part of a protein to which antibodies or immune cells bind is called an epitope. Again, a completely denatured protein will expose many epitopes to which antibodies can be produced, but the epitopes that are exposed when the protein is in its native configuration may be more relevant (for instance, to produce an effective vaccine). As antigen - antibody interaction depends on both components, prediction of epitopes needs to take antibody properties into account as well. A number of web interfaces are available for epitope prediction, of which a limited list is presented in Box 10.2.

**Box 10.2: A few web interfaces to predict epitopes**

- [http://www.cbs.dtu.dk/services/NetMHCpan](http://www.cbs.dtu.dk/services/NetMHCpan)
  Predicts interactions between proteins or peptides and MHC class I molecules
- [http://www.cbs.dtu.dk/services/NetMHCIIpan](http://www.cbs.dtu.dk/services/NetMHCIIpan)
  Predicts interactions with MHC class II molecules
- [http://immunax.dfci.harvard.edu/PEPVAC/](http://immunax.dfci.harvard.edu/PEPVAC/)
  PEPVAC, suitable for the prediction of viral multi-epitope vaccine candidates
- [http://bioinfo.ernet.in/cep.htm](http://bioinfo.ernet.in/cep.htm)
  CEP aims to predict conformational epitopes for single proteins

**Concluding remarks**

We have covered a lot of territory in this chapter, from promoters and gene expression to protein modification and localization to prediction of epitopes. In a sense, gene expression can be likened to the value of a house - the value of both strongly depends on location. Where the gene is located along a chromosome is important. There are a few pockets of highly expressed genes, surrounded by vast suburbs of genes that are rarely transcribed. Once the gene is transcribed, protein-coding genes are translated by the ribosomes, and then folded and maybe transported to their correct position in the cell (or outside in the case of secreted proteins). Based on the location, it is possible to predict which proteins are likely to be antigenic, since a cytoplasmic protein will have a slim chance of being exposed to the host immune system. Typically, less than 10% of a bacteria’s proteins are likely to be surface-exposed. Such a sub-selection can be run through an epitope predictor to find regions within the protein that are good regions to consider for use in vaccine development.
References:


Box 10.3: Comparison and visualization tools used in this chapter

Violin plot of secreted proteins. Prediction of the secreted fraction of proteins for bacteria grouped per phylum was done using SignalP, LipoP and SecretomeP tools listed in Box 10.2. How violin plots are produced is very clearly explained Hintze and Nelson, 1998.
Outline The ‘proteome’ is the sum of all the proteins described by an organism’s genome. Of course, most of the information in proteins comes from the amino acid sequence, which determines the structure; the structure is what determines function. In general, although there are many methods that attempt to predict protein structure, it is still often difficult to accurately predict the structure of novel sequences. Currently homology searches can be based on primary sequence and on structural features that go beyond the sequence itself. If one can find a good sequence match to a protein with known structure and function, it is inferred that the unknown (query) protein has that or a very similar function. Three typical research questions will be treated: what can we learn from analyzing a single protein-coding gene? How can we annotate all genes in a bacterial sequence? And finally, what can learn from comparing a single gene, a group of genes or complete genomes between organisms? To address the latter question, the BLAST Atlas is introduced.
Introduction

The majority of DNA in a bacterial genome consists of protein-coding genes. Nevertheless, so far relatively little has been said about these genes, although a complete book could easily be filled with this subject. This chapter is divided in three parts, in which we will treat three questions that a microbiologist frequently encounters during research, with increasing level of complexity: (i) What can we learn from analyzing a single protein-coding gene? (ii) How do we annotate all genes in a bacterial sequence? (iii) What can we learn from comparing a single gene, a group of genes or complete genomes between organisms? First we will explore what a single gene can tell about the protein it will encode, and which tools are available for analysis. Then we will describe the relatively easy gene finding and automated (but far trickier) genome annotation. Finally, examples of comparison of proteomes from several different organisms will be presented.

Part 1: Analysis of individual protein-coding genes

The scenario is common: during a particular microbiological investigation the researcher stumbles upon a gene for which a sequence is known (extracted from a genome sequence, or obtained by direct sequencing). Can this sequence predict what the protein does in the cell? It is now routine to compare the gene sequence with entries in the public databases as a first step; usually this is done by BLAST, as described in Chapters 2 and 4. If the analysis produces a strong identity to a gene with a clear and proven function, it is mostly assumed that the query gene has a similar or related function in the investigated organism. This is actually not always the case. Predictions from sequence matches ideally should (must) be supported by wet lab evidence of the function, but this is just not practical in view of the numbers of proteins coded by a bacterial genome.

Note the caveats: a ‘strong’ identity and a ‘clear and proven’ function. Identity at the protein level of more than 50% is usually considered highly significant, but as the saying goes, the devil is in the details. Also, it is not so simple to give a percent threshold: in some cases, proteins with 30% (or less) identity can have the same function. A critical look at the annotated function of the hits can reveal if their function was again deduced from homology only, or whether direct experimental evidence was available. Besides identifying similarity for the complete length of the protein, BLAST also identifies local similarities, which can help to predict functional domains. Thus, even when a novel protein has only limited similarity to known proteins, the presence of conserved domains can indicate the possible function of the protein.
A word of caution is needed here. The question “What is the function of this protein?” assumes that each individual protein has its own unique role in the cell, and that we can, and should strive towards, assigning a single function for each gene. However, as is often the case, things are more complicated upon closer inspection. The very same protein can have quite different functions under various circumstances. A classic example is the crystallin protein that makes the lens in eyes, which also functions as an enzyme. Another example is the ‘DEAD box’ protein, which is part of a complex that degrades RNA in *E. coli*; this protein is also more commonly known as enolase, playing a central role in metabolism (Py *et al.*, 1996). Thus it is often the case that one protein can have more than one function, so caution should be taken when assigning functional categories to proteins, based on their sequence homology (Piatigorsky, 2007).

**Similarity and homology**

The difference between similarity and identity is already explained in chapter 2 and the two should not be confused. A third term that is frequently misused is *homology*. This term should be exclusively reserved for similarity between entities that are genetically derived from each other. Thus, ‘homology’ implies there is a common genetic lineage. When gene A and gene B share significant sequence identity and also a common lineage, they are homologs that can be related in various ways: A can be an evolutionary offspring of B (with or without various steps in between the two forms), or B was derived from A, or A and B were both derived from the earlier ancestor C.

As a special case of homologs, the term *ortholog* is reserved for homologous genes that have diverged in different species (or in different strains within a species); they share a genetic origin in the last common ancestor. The term *paralog* is typically used for homologous genes within a genome that have originated from gene duplication, after which each duplicon evolved into a (slightly) different function. Finally, there are genes that share a function but not a genetic lineage (these could be the result of concerted or parallel evolution), which of course are not homologs but can nevertheless display considerable similarity at the sequence level.

**Functional categories**

Obviously two very similar proteins are likely to have similar functions, and the less similar their sequences, the more likely it becomes that their functions differ. In more general terms, it is possible to divide proteins into broad sets of functional
categories. There are various schemes, but one commonly used grouping is known as the Cluster of Orthologous Genes (COGs). A database at NCBI\(^1\) is dedicated to COG grouping, where proteins from a bacterial proteome are classified in 23 specific functional categories (sorted in three major functional groups), as shown in Figure 11.1. These categories are now frequently used to sort all annotated genes in a genome for functionality.

<table>
<thead>
<tr>
<th>Code</th>
<th>COGs</th>
<th>Domains</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Information storage and processing</strong></td>
</tr>
<tr>
<td>J</td>
<td>245</td>
<td>10,572</td>
<td>Translation, ribosomal structure and biogenesis</td>
</tr>
<tr>
<td>A</td>
<td>25</td>
<td>137</td>
<td>RNA processing and modification</td>
</tr>
<tr>
<td>K</td>
<td>231</td>
<td>11,271</td>
<td>Transcription</td>
</tr>
<tr>
<td>L</td>
<td>238</td>
<td>10,338</td>
<td>Replication, recombination and repair</td>
</tr>
<tr>
<td>B</td>
<td>19</td>
<td>228</td>
<td>Chromatin structure and dynamics</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Cellular processes and signaling</strong></td>
</tr>
<tr>
<td>D</td>
<td>72</td>
<td>1,678</td>
<td>Cell cycle control, cell division, chromosome partitioning</td>
</tr>
<tr>
<td>Y</td>
<td>-</td>
<td></td>
<td>Nuclear structure</td>
</tr>
<tr>
<td>V</td>
<td>46</td>
<td>2,380</td>
<td>Defense mechanisms</td>
</tr>
<tr>
<td>T</td>
<td>152</td>
<td>7,683</td>
<td>Signal transduction mechanisms</td>
</tr>
<tr>
<td>M</td>
<td>188</td>
<td>7,853</td>
<td>Cell wall/membrane/envelope biogenesis</td>
</tr>
<tr>
<td>N</td>
<td>96</td>
<td>2,747</td>
<td>Cell motility</td>
</tr>
<tr>
<td>Z</td>
<td>12</td>
<td>128</td>
<td>Cytoskeleton</td>
</tr>
<tr>
<td>W</td>
<td>1</td>
<td>25</td>
<td>Extra-cellular structures</td>
</tr>
<tr>
<td>U</td>
<td>159</td>
<td>3,743</td>
<td>Intracellular trafficking, secretion, vesicular transport</td>
</tr>
<tr>
<td>O</td>
<td>203</td>
<td>6,206</td>
<td>Post-translational modification, protein turnover, chaperones</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Metabolism</strong></td>
</tr>
<tr>
<td>C</td>
<td>258</td>
<td>9,830</td>
<td>Energy production and conversion</td>
</tr>
<tr>
<td>G</td>
<td>230</td>
<td>10,816</td>
<td>Carbohydrate transport and metabolism</td>
</tr>
<tr>
<td>E</td>
<td>270</td>
<td>14,939</td>
<td>Amino acid transport and metabolism</td>
</tr>
<tr>
<td>F</td>
<td>95</td>
<td>3,922</td>
<td>Nucleotide transport and metabolism</td>
</tr>
<tr>
<td>H</td>
<td>179</td>
<td>6,582</td>
<td>Coenzyme transport and metabolism</td>
</tr>
<tr>
<td>I</td>
<td>94</td>
<td>5,201</td>
<td>Lipid transport and metabolism</td>
</tr>
<tr>
<td>P</td>
<td>212</td>
<td>9,232</td>
<td>Inorganic ion transport and metabolism</td>
</tr>
<tr>
<td>Q</td>
<td>88</td>
<td>4,055</td>
<td>Secondary metabolites biosynth., transport and catabolism</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Poorly characterized</strong></td>
</tr>
<tr>
<td>R</td>
<td>702</td>
<td>22,721</td>
<td>General function prediction only</td>
</tr>
<tr>
<td>S</td>
<td>1346</td>
<td>13,883</td>
<td>Function unknown</td>
</tr>
</tbody>
</table>

Figure 11.1. Cluster of Orthologous Genes codes for functionally related gene categories. The table was reproduced from the NCBI website as it appeared in March 2008. 

\(^1\) http://www.ncbi.nlm.nih.gov/COG
The most difficult situation arises when for a query protein no obvious similarity is identified in the database, or if all hits belong to the ‘poorly characterized’ classes R and S of Figure 11.1. Here a gene may only be annotated as ‘conserved in Enterobacteria’ or ‘hypothetical protein’, or a match is only found against proteins in metagenomic data sets, discussed in the next chapter. Can we still obtain a hint on the function or properties of that protein, by analyzing its sequence?

Fortunately, there are a number of standard analyses that are available on the web, to provide additional information about a protein sequence. The microbiologist working with eukaryotic or archaeal genes has the advantage of ProtFun\(^2\), a tool that can be used to make a prediction of the function even if there is no good hit of sequence similarity in any of the databases. A bacterial version of this service is under construction. In some cases, other properties of the protein can lead to an educated guess for its function. For example knowing the location of a protein and some of its possible post-translational modifications might help in determining the broad functional category, even though there is not a solid sequence-based match in any of the current databases.

At the risk of seeming narcissistic, we point out that a good variety of prediction tools have been developed over the years at CBS\(^3\), and we consider this a good place to start browsing for tools that are available (although PubMed is of course the best place to search for literature on this subject.) Some of the general approaches that are particularly helpful, common or informative are listed below, but any omissions should not be seen as a lack of quality or usefulness; we simply can’t cover all tools currently available.

**Prediction of protein structure: secondary and tertiary structure**

As we have mentioned since the very first chapter, protein structure determines function, and hence prediction of structural features is the first logical step in predicting function. Proteins have a primary, secondary and tertiary structure. The amino acid sequence represents its *primary structure*. The *secondary structure* describes the presence of particular structures within the protein, notably alpha helix and beta sheet (also called beta strand) structures. The *tertiary structure* describes how these particular structures are combined in a three-dimensional configuration (at a fourth level, the *quaternary structure* describes how various subunits are combined in a multi-subunit protein). A protein can only function when it is properly folded.

---

\(^2\) [http://www.cbs.dtu.dk/services/ProtFun](http://www.cbs.dtu.dk/services/ProtFun)

\(^3\) [http://www.cbs.dtu.dk/services](http://www.cbs.dtu.dk/services)
While the protein molecule is growing during translation, secondary structures will spontaneously form, as these are energetically favored states of a protein. The presence of alpha helices and beta sheets is governed by the physical properties of the amino acids, their order in the sequence, and the properties of the surrounding environment. Some amino acids, such as Glu, Leu and Ala, are frequently present in alpha helices, whereas Val, Ile and Phe are common in beta sheets. In contrast, Pro cannot be present in an alpha helix (it is a ‘helix breaker’) and the charged amino acids are unlikely to be present in beta sheets. Beta sheets can run either parallel or anti-parallel, depending on the relative direction of the peptide bonds connecting the amino acids. The presence of alpha helix and beta sheets can be predicted with relative accuracy.

These relatively rigid parts of the proteins may take on a particular configuration by the bending of more flexible loops (frequently with the aid of chaperones), and can be held into position by hydrogen bonds, ionic forces between charged amino acids, or even ‘glued’ together by covalent disulphide bonds. The tertiary structure of a protein describes how the helices, sheets and loops are configured in space, and prediction tools for this level of geometrical organization have improved over the past several years. At the early stage, computer predictions are compared with crystallographic data, to evaluate how reliable the prediction is. Crystallography is only possible for completely pure, water-soluble proteins. For this reason, the structural prediction of proteins that cannot be crystallized, for instance because their correct configuration is water-insoluble, or their configuration is changed by the removal of lipids, is notoriously difficult.

Protein structures are available from a specialized database, the Protein Data Bank (PDB)\(^4\), an initiative of the Research Collaboratory for Structural Bioinformatics (RCSB). This database provides 3-dimensional models of well-characterized proteins, which can be downloaded and rotated at wish with relatively easy to use software (free to download from the PDB website). An example of a PDB figure was included in Chapter 1 (Figure 1.7). Such visualization tools are extremely informative to get a feeling for the structure of proteins. Nonetheless, the PDB database cannot keep up with the pace of newly sequenced proteins, and so a protein with unknown function is unlikely to be represented in the PDB.

As an example how to deduce a possible function from a protein starting with a sequence only, we will take the amino acid sequence shown in Figure 11.2, representing a translated gene from *Bacillus weihenstephanensis*, a food spoilage bacterium and a close relative to *B. cereus*. If this would be the amino acid sequence of an unknown protein, the first sensible thing to do would be to BLAST this protein against the NCBI database, and see which proteins are similar. In this example,

---

\(^4\) http://www.rcsb.org/pdb
there are several close homologs in *Bacillus* genomes, but pretend for the sake of the argument that no good matches were found. What to do in that case?

Bacillus weihenstephanensis KBAB4 unknown gene

MKFPFKVITLATVITATNGSTIHALDAQETAQEQKIENYALGPEGLKVLAKTG5HILVMMLY
AKTMKQPNVNLSDNLSSGGEELIKNIHLNQELSRINANYWLDATAKPIQIRN1VNYDEQFQN
YDYRTLVDTKDKAGLKEGIDLIGTIHTNSNEVTEIIKMLEAFIKLVTNTDFKNVNGDPQ
GGLTALAGKQALVPQLAQAEIENLRSTQKAFDNVLAWSGGLGAAALVIGTIAAGAVVIVTGT
ATPAVVGLTALGAAGIGLGETAGVEASNHMSYNEISNKIGELSMKADLANQAVISLTNTKDTLT
YLQTVEQIAIMLSITSIQQWNTMGANYKDLID1QMEHKLSSLIPDDLKAAQ5WNDIHKADEFI
SKOIAFKQKETN

Figure 11.2. Protein FASTA file of an ‘unknown’ gene believed to be involved in toxin production, isolated from *Bacillus weihenstephanensis* strain KBAB4.

The protein structure of ‘unknown’ proteins can be predicted using a tool called *Copenhagen Models*, available from the CBS website\(^5\). Figure 12.3 shows the structure prediction of the *B. weihenstephanensis* protein. This prediction is based on a protein with similar structure (which may not always be detected by BLAST searches), whose structure is given to the right. The two proteins are similar but not identical, as can be seen from the lacking beta sheets in protein KBAB4. Regions for which the model may be inaccurate are red.

![Protein KBAB4 of Bacillus weihenstephanensis](image1)

![Protein 03B108 of Bacillus cereus](image2)

Figure 11.3. Predicted structure of the *Bacillus weihenstephanensis* protein using Copenhagen Models. The model shown to the left is based on the crystal structure of a similar protein from *B. cereus* (shown to the right), with alpha helices colored light blue, beta sheets orange and loop regions green. The overall sequence identity between the template structure and the *B. weihenstephanensis* protein is 25%. Regions corresponding to insertions in the model relative to the template are highlighted in red. Such regions need extra attention, as they are more prone to modeling errors. Notice the absence of beta sheets in the predicted model to the left.

---

\(^5\)http://www.cbs.dtu.dk/services/CPHmodels
Prediction of trans-membrane helices

One of the oldest prediction models applied to protein sequences was based on predicting hydrophobicity, which is a measure of how likely a protein is to be membrane-embedded. These models rely on the hydrophobicity characteristics of individual amino acids: those having a polar, charged or hydrophilic side chain would produce a water-soluble (hydrophilic) protein, whereas amino acids with an apolar, aromatic or hydrophobic side chain would produce lipophilic, membrane-embedded proteins. In practice, proteins are made up of a mixture of amino acids with either characteristics, but their order is random.

Once hydrophilic and hydrophobic domains have been predicted, the transition from clearly hydrophilic to clearly hydrophobic (or the other way round) can be interpreted to represent a trans-membrane domain of a protein that sits embedded in the membrane, with parts sticking out of the membrane on one or both sides. Such trans-membrane domains are typically folded into alpha helixes and beta sheets, which can also be predicted with relative accuracy. Prediction models assume that a protein is produced within the cell, and then predict which sections of the protein will be ‘in’ and which will be sticking ‘out’. Figure 11.4 shows an example of our *B. weihenstephanensis* protein, using one of many tools for prediction of trans-membrane domains, in this case the Phobius website6 (Käll et al., 2004, 2007).

![Figure 11.4. Prediction of the membrane topology for the Bacillus weihenstephanensis protein, produced on the Phobius website. Two trans-membrane domains are indicated by strong peaks (in grey). A signal peptide is recognized for the first 35 amino acids (red curve). The relative positions of cytoplasmic (green) and non-cytoplasmic (blue) topology are predicted for the protein as it is being produced inside a cell.](http://phobius.sbc.su.se)

---

6 http://phobius.sbc.su.se
Guilt by association: prediction of interacting proteins

Sometimes the function of a protein can be deduced from predictions of protein-protein interaction. This information can be specifically searched within the STRING database⁷ (developed by EMBL in Heidelberg), which holds data derived from several hundred bacterial genomes (von Mering et al., 2007). We used the FASTA sequence of our example protein to search the STRING database, and a screenshot of the result is shown in Figure 11.5. In this case, the program predicts strong interactions with two other proteins, the constituents of the tripartite bacterial toxin. A view of the co-occurrence in other bacterial genomes reveals that these three proteins are found only in Bacillus cereus. In this case that information is already known, but one could easily imagine a case of using the same tools for truly unknown proteins.

Figure 11.5. Predicted protein interactions of the Bacillus weihenstephanensis protein, using the STRING database. The figure shows the interactive view of the results with AAT602141 representing our protein. Other views (including co-occurrence in other bacterial genomes) can be chosen by clicking on the buttons at the bottom.

In addition to the STRING database, EMBL provides another web tool where one can search for COG categories, as discussed at the beginning of this chapter, only this time the clustering is done automatically, in a non-supervised manner (Jensen

⁷ http://string.embl.de
This database is called eggNOG\(^8\) (for evolutionary genealogy of
genomes: Non-supervised Orthologous Groups), and the output for the closest cluster
to our ‘unknown protein’ is shown in Figure 11.6. This search is based on homology
to the closest category. The cluster groups are based on a Smith-Waterman
alignment using reciprocal best hits. These examples illustrate some of the many
comparison tools that are available. The reader is encouraged to explore other
tools available on the web.

Figure 11.6. Predicted protein interactions of the *Bacillus weihenstephanensis*
protein, using the eggNOG database.

Part 2: How to annotate a complete genome

A digression: about current sequencing technologies

The technological developments to further improve sequencing throughput are
rapid and almost certain to change and further improve continually for the next
several years. Figure 11.7 depicts what is currently possible in some laboratories:
starting with a purified genomic DNA sample, a complete sequence coverage can
be reached within 24 hours, then automated bioinformatic processing, including
genome assembly, gene finding, functional prediction and annotation can be com-
pleted in a few more hours. The price of a complete bacterial genome sequenced
this way is in the range of a few thousand dollars per genome, but costs are stea-
dily decreasing.

As the figure shows, preparation of the library from genomic DNA takes
approximately 6 to 7 hours, and is mostly automated. During this step, the genomic

\(^8\) [http://eggnog.embl.de](http://eggnog.embl.de)
DNA is broken up into small fragments and mixed with very small beads. These are coated with a primer, and diluted so that, on average, there is about one DNA fragment per bead. These beads form bubbles in an oil phase, and by means of PCR each bead is coated with clonal copies of the fragment. The beads are next immobilized for subsequent sequencing. The sequencing machines perform in a fully automated way after manual set up, producing banding patterns representing the DNA sequence of variable lengths per read (see below), which are put together by assembling programs.

Figure 11.7. The time required to generate a complete bacterial genome sequence is now approximately 24 hours. Bioinformatic processing, including assembly, gene finding and gene annotation adds a few more hours, but in less than 36 hours a Genome Atlas or annotated GenBank file can be produced, starting with a tube of purified genomic DNA.

When sufficient overlapping fragments are available, the genome can be pieced together into one or a few large fragments, called contigs (for contiguous fragments). These can be fed into a gene finding program (details about automated gene finding were discussed in Chapter 6) after which gene annotation is the next step.

Technical developments in genome sequencing

Before we move on to how to annotate a genome sequence, it is important to take a moment and put things in perspective. One word of caution on technological developments is needed. With some novel techniques to identify and read the DNA sequence, the length of individual reads can be quite small. The current read length of some newly developed sequencing techniques is around 25 to 30 nucleo-
tides, rapidly producing highly accurate and longer sequence reads. The high accuracy is extremely important, as wrongly assigned bases can result in frame shifts and apparently truncated proteins. Compared to the length of an average bacterial gene of about a thousand bp, these short reads are only a tiny fraction (Figure 11.8). With prokaryotic genomes, which vary in size from a few hundred kbp to 15 Mbp, it is obvious that assembling such short reads into a complete genome is hard, if not impossible. Even accurately assembling the reads to individual bacterial genes can be difficult.

In our opinion, these solutions are wonderful for looking at small nucleotide changes (SNPs or Single Nucleotide Polymorphisms), as the reads are highly accurate and of very good quality. For re-sequencing genomes in order to identify particular mutations the current short reads may also be acceptable. However, until these technologies are further improved to produce longer reads, their applications to direct sequencing of novel genomes is limited. The expectations are that the technology will develop further, so that in the near future longer reads are expected. It is indeed already possible to sequence bacterial genomes quite inexpensively, obtaining full closure, that is, in one single contiguous piece, in single run (Tauch et al., 2008). If current trends continue, within the next few years a decent cup of coffee may cost more than to sequence a bacterial genome.

![Figure 11.8. The length of sequence reads produced using various methods, compared to the size of a eukaryotic and a bacterial gene.](image)

The current technologies already allow high-throughput generation of an enormous amount of sequencing data, usually producing over 10-fold coverage of a bacterial genome. However, one should remember that with shorter read lengths it becomes more difficult to correctly sort repeats and sometimes to assemble the genome into reasonable pieces that contain multiple genes, rather than just fragments of genes. Particularly troublesome is that some of the current ‘Whole Genome Shotgun’ datasets generated with early versions of some of the high-throughput technologies suffer from unacceptably high error rates, which results in the appearance of truncated genes. The bottom line is that one has to be careful about the quality of the genome sequence: this can have a major impact on the re-
liability of the predicted proteome. As a final note, it should be stressed once again that the presented arguments are based on a ‘snapshot’ of the current sequencing technology, which will very likely improve considerably over the next several years.

**Going from DNA sequence to genes to a GenBank file**

The question that some microbiologists might now be facing may be a bit embarrassing: once a genome sequence is available, what to do next? We will briefly outline the pipeline that we have developed in our research group to analyze the bacterial genomes we have sequenced. Most currently used technologies will yield the sequence as large contigs, that have been assembled from individual reads. Typically, depending on the amount of repetitive DNA and other factors, one can expect less than 100 contigs for a bacterial chromosome. The first decision thus is what to do with the multiple contigs.

One can try to close the gaps. In case the genome sequence of a closely related bacterial genome is available, this can be used as a reference to orientate the pieces and predict which gaps need to be filled by PCR or other experimental techniques. Alternatively, one can accept the gaps for the time being, and artificially combine all the pieces together, introducing stop codons (in all 6 possible open reading frames) at the artificial junctions. Such a sequence can then serve as the basis for gene finding to identify the protein-coding and non-translated genes, or for any other analyses. Note, however, that any parameter that depends on location or on global trends, such as the identification of the origin of replication, must be interpreted with care, as the ‘genome’ is still an artificial product of non-finished segments.

**Gene finding is relatively easy...**

An obvious next step would be to identify the genes on our genome sequence. It is important to emphasize that gene finding, *i.e.*, the prediction of the location of genes in a sequenced bacterial genome is not the same as gene annotation, the prediction of the function of those genes. The former is relatively straightforward, whilst the latter can take more time and resources.

Gene finding in bacterial genomes is now highly automated. A computer program can automatically detect all ORFs in a given sequence and filter out ORFs that are too short to be likely gene candidates. There are a handful of gene finders, (such as Glimmer, GeneMark, and EasyGene) and all generally agree on most of the genes. Usually for each gene there is a score of how ‘believable’ the prediction
is; an ORF with a very biased amino acid composition, or an extreme proportion of repeats, would receive a low score, for instance. The importance on a rightly chosen cut-off to remove short ORFs has already been discussed in Chapter 6. In addition, false negatives can occur: real genes that are not predicted and are thus missed. From personal observations we know that roughly 10% of the proteins detected in proteomic experiments with E. coli were not predicted to be encoded in the genome, although they could be identified by reverse engineering, once amino acid sequences are known from proteomics experiments.

There are a variety of possible reasons why genes might have been missed. The earliest genes that were sequenced (in times when sequencing complete genomes was still science fiction) were backed up with experimental evidence, but sometimes, if an ORF existed on both strands, the wrong strand was annotated, if the sequencing window was too small to visualize the better open reading frame going the other direction. If genes are automatically annotated by sequence homology, the error will be maintained. So how do we know which strand is the correct one? One could specifically detect mRNA, as in transcriptome analysis, but that may not always be practical. Checking the length and codon usage of ORFs overlapping on complementary strands is a good alternative. As a rule of thumb, the longer the gene, and the closer codon frequencies are to the genome codon usage, the more likely an ORF is a gene. In the case of E. coli, proteome analysis revealed that of the most highly expressed proteins, again approximately 10% were not found in the annotated genes - but in a number of cases a gene was found on the other strand, and codon usage data suggested the wrong strand was annotated. Although experimental evidence is mostly seen as the ultimate ‘truth’ in gene annotation, against which computer predictions are usually tested, some (published) ‘experimentally verified’ data may not really have verified what is claimed.

**Annotation of protein function in genomes is more difficult**

After finding all ORFs, removing those that are too short, and checking for the correct strand in case of overlap, one then is left with the huge task of gene annotation. Gene annotation for a complete genome is not different from finding the possible function of a single query gene - but now there can be thousands of query genes. Fortunately, many of these query genes have conserved functions in a cell, and are thus likely to have already been annotated correctly in existing genomes. The next chapter will deal with the subset of genes conserved within and between species.

There are a number of developed pipelines that can automatically predict proteins against large databases by BLAST and, based on the best match, extract a likely function for each protein. However, sometimes the matches might be quite
good in most of the structural part of the protein, but perhaps different in the crucial 'active site' region. Thus, just looking at percent identity might not be enough - there remains a need to check which regions within a protein are matching, and more importantly, which regions do not align. Of great assistance here is the option to search for protein domains of known function, using tools such as the Pfam database (Finn et al., 2008).

**Compare a single gene between organisms**

Once the identified genes have been annotated, the genome sequence will serve as a reference in GenBank that can be used as a basis for any further research. As scientific demands will broaden out rapidly at this stage, few general concepts apply, so that we will not go into more details. One aspect of comparative genomics, however, deserves to be mentioned.

A comparison between gene homologs is a frequent routine: to compare alleles of single genes (from various isolates within a species, for instance) that have the same function in the cell. Some variation can reside in the amino acid sequence, but more variation is usually found in the nucleotide sequence, as the redundant genetic code allows for synonymous mutations, where a nucleotide difference does not alter the amino acid. Consider a hypothetical surface-exposed protein that has a C-terminal domain that is highly antigenic and an N-terminal domain that is highly conserved, where some kind of active site resides. The two domains are separated by a membrane-embedded middle part. Mutations would presumably occur by chance anywhere in the protein, but if they would affect the amino acid sequence of the N-terminal domain, the resulting protein is likely to be less active due to changes in the active site. Thus, there is selection for conservation of amino acids (also called 'purifying' or 'negative' selection) in this part of the protein, though synonymous mutations can occur at no cost (other than codon usage preference). The rate of non-synonymous mutations, $K_a$, is kept low whereas the rate of synonymous mutations, $K_s$, is not affected by this negative selection.

In contrast, there would be far more liberty for mutations to occur in the C-terminal half of the protein, and it could even be advantageous to the organism to vary this domain, for instance when there would be selection by antibodies produced by the host. In this domain $K_a$ can be expected to be high. The middle part of the protein allows non-synonymous mutations as long as the resulting amino acids are hydrophobic which places particular constraints to $K_a$. In general, in a protein the ratio of $K_a$ over $K_s$ will be low in functionally conserved domains that are under strong negative selection (with a pressure for conservation), and high for domains that are under positive selection for change. Thus, analyzing the $K_a/K_s$
ratio of protein domains can give insights in evolutionary processes⁹ (which are further explored in Chapter 14). The $K_a/K_s$ ratio is also related to population size: a large population is thought to be better equipped to survive strong negative selection pressures, hence a low ratio is thought to correlate to a large population size.

**Part 3: Proteome comparisons**

The next step up, from comparing multiple variants of a gene between genomes, would be to do a comparison of all genes between one or several genomes. What can we learn from comparing a group of genes or complete genomes between organisms? And how do we visualize the results of such analyses? In general, we favor two methods for visualizing BLAST comparisons of bacterial proteomes. An overview can be obtained with a **BLAST Matrix**, which plots the number of hits of a given set of proteomes against each other. Another visualization tool we find very powerful is the **BLAST Atlas**, which displays protein conservation on a reference genome chromosomal map.

**Introducing the BLAST Matrix**

A **BLAST Matrix** is a table where the results of pairwise BLAST comparisons between genomes are summarized. Such a table can be made for 2 genomes up to as many as is practical. For comparison of hundreds of genomes, colored heat maps can replace numerical values. Specific subsets of proteins (e.g., the secretome or metabolome) can be also used to compare across genomes.

As an example, consider the *E. coli* proteome. Currently, there are three different genome sequences available from various isolates belonging to the K-12 serotype; isolate MG1655 was published first, isolate W3110 was sequenced around the same time, and more recently isolate DH10B has been deposited to GenBank. How do these compare with each other, and how do they compare with a pathogenic *E. coli*? As an example for the latter, we will use uropathogenic *E. coli* strain CFT073, whose genome is about a million bp longer than K-12 and contains a thousand extra genes. Starting with one genome, every gene identified in that

⁹ Note that $K_s$ can become ‘saturated’ due to multiple mutations at one site; hence it is only meaningful to assess the $K_a/K_s$ ratio in closely related protein sequences. Moreover, synonymous mutations are not completely ‘neutral’ as they are affected by codon usage but the purifying selection on $K_s$ is considered far lower than that of $K_a$. Finally, recombinations whereby partial protein fragments are exchanged between alleles obscure the effect of evolution by mutation, and the relative importance of recombinations over mutations can differ in different organisms.
genome is compared by BLAST with each of the other genomes, and in case a significant hit is found, this is scored. Every gene is also used to search within its own genome for similarity (for instance to detect paralogs). In the latter case, self-to-self hits are not scored. The procedure is repeated for every genome, so that each genome is compared (by BLAST) to all the others and itself.

This all seems simple enough, but when is a hit significant to be scored? In other words, how does one define two proteins as being homologous? Where should the cut-off value be set? To find an answer, we allow two score parameters to be chosen for automated BLAST scores: the fraction of the gene that should at least align, and the E-value. Figure 11.9 shows a BLAST Matrix for the four *E. coli* genomes analyzed for this example. The chosen score values given as ALR (alignment length region) are the default settings, which mean that the alignment region must be at least 75% of the length of the query protein, and have an E-value of less than one in 10 billions (10^{-10}).

In each cell of the matrix the numbers of genes are given for which a match was found. For example, the top left cell shows that when the CFT073 genome was BLASTed against MG1655, of the 5,379 genes present in CFT073 3,997 found a match above the cut-off. Note that when the BLAST search was performed the other way round, of 4,132 MG1655 genes only 3,291 matched somewhere on the CFT073 genome (bottom right cell).

![Proteome comparison of Escherichia coli](image)

**Figure 11.9.** BLAST Matrix comparing 4 different *E. coli* genomes. Intra-genomic BLAST results are given in the shaded cells. The three different sequenced K-12 isolates are similar, whilst the CFT073 strain is quite different, and only about two thirds of the CFT073 genes are found in the K-12 genomes.
The diagonal shows the fraction of genes that found a paralog within its own genome (the shaded cells), which varied from 7.5% (for MG1655) to 12.9% (for DH108). Inter-genomic comparisons within the K-12 strains find 89% to 97% of the genes the same, but the fraction is significantly lower for the comparisons with CFT073 genome (left-hand column and bottom row). From this we can easily see that the three K-12 genomes are more similar to each other than any of them is to the uropathogenic CFT073. Reading these values from cells in a matrix table becomes harder as more cells are added. Adding color as a measure for the fraction of matched genes can help visualization. We report intra-genomic searches for paralogs (self-to-self BLAST) as shades of red, and inter-genomic searches as shades of green. Figure 11.10 shows a BLAST Matrix of 41 sequenced genomes belonging to the family of Enterobacteriaceae. Now the value of adding colors becomes apparent: even when the print is too small to read individual numbers, the color shading easily identifies those genome combinations for which high or low numbers of matches were found.

The figure represents an enormous amount of data, considering that 41 times 41 already gives 1681 combinations, and every cell of the matrix represents approximately 5000 BLAST searches (one for each gene in the query genome), giving 8.4 billion searches in total! This BLAST matrix took about 22 hours to compute, on a Silicon Graphics ALTIX machine, using 50 processors in parallel. Thus, if one were to do this on a PC at home, doing the BLAST calculations could well take more than 2 months. If one were to analyze the results in detail, looking at one BLAST report for two seconds only, this would already take an additional 6 months! Despite this avalanche of data, an overview can still be perceived quite easily by proper visualization.

The Yersinia species (columns and rows 1 to 9) form a rather homogeneous group that is distinct from E. coli / Shigella. This is concluded from the green box containing the 9 times 9 matrix of Yersinia genomes compared against each other which is darker green, indicative of a higher degree of similarity within Yersinia genomes than between Yersinia and the other represented genomes. Column 10 (and row 10) correspond to the one genome of Wigglesworthia glossinidia (a small endosymbiont), and it is the most different within the analyzed group, visible by the pale color. The rest of the matrix (mainly Shigella, Salmonella, and E. coli) contains genomes quite similar to each other. The distinction of Shigella and E. coli in different genera and species is mostly historical; based on their observed genetic similarity it could be argued they all belong to the same species. Two isolates of E. coli O157:H7 (another pathogenic strain, columns 29 and 30) are highly similar, and share more genes with Shigella than with other E. coli strains.
Figure 11.10. BLAST Matrix of 41 genomes from Enterobacteriaceae. The species included were: Y. pseudotuberculosis (1-3), Y. pestis (4-8), Y. enterocolitica (9), Wigglesworthia glossinidia (10), Sodalis glossinidius (11), Shigella sonnei (12), S. flexneri (13-15), S. dysenteriae (16), S. boydii (17), Salmonella enterica (18-23), Photobacterium luminescens (24), Pectobacterium atrosepticum (24), Klebsiella pneumoniae (26), Escherichia coli (27-39), Enterobacter spp. (40) and Enterobacter sakazakii (41). The squares point out similarities between the Shigella species (12-17), between the two E. coli O157:H7 (29 and 30), and between the three E. coli K-12 genomes (31-33). W. glossinidia (10) is the most distinct genome included, as indicated by the arrow.
Introducing the BLAST Atlas

The BLAST Matrix only shows number for genes that match in a given other genome, but it won’t tell you which those genes are. This can be visualized in a BLAST Atlas, which allows viewing conservation of protein-coding genes at the level of chromosomal region. For visualization, we use one reference genome, specified in the center of the atlas, to show gene location. Around this genome, the BLAST scores obtained in other genomes are plotted, so that conserved regions (and regions which are variable) are visualized. Location of the found match is not given, so all scores in the lanes are located relative to the reference genome.

Figure 11.11 shows two BLAST Atlases that were constructed from the four *E. coli* strain comparisons of which the BLAST Matrix was shown in Figure 11.9. One atlas is plotted with reference to K-12 strain W3110, and the other has CFT073 as the reference. Since CFT073 has approximately a thousand genes more than the K-12 strains, it should not be surprising that large white areas are found where CFT073 genes don’t find a match in the K-12 genomes. These regions represent genes that are present in the CFT073 strain but missing in all three of the *E. coli* K-12 strains. That information is not visible from the BLAST Atlas to the left. However, there are some regions near the top of the *E. coli* K-12 reference genome that appear to be missing in one of the other K-12 strains. In particular, there seems to be a region present in the MG1655 and W3110 isolates, which is missing in the DH19B isolate. In order to see such different conservation trends, it is best to use several different available genomes as the reference in a BLAST Atlas.

How exactly are these searches done? Let’s start with the first gene (gene A), located at position 1 to 1000 in the reference genome (which we call Genome 1). With this translated gene a BLASP is performed against all the proteins in Genome 2. A good match is identified to gene A’ in Genome 2, with 100% identity over a large region of gene A (e.g., for BLAST Atlases an alignment of 50% of the gene length is normally required). This result is scored by the mapping of value ‘1’ for each triplet of nucleotides (of gene A) finding perfect matches to gene A’, and value ‘0’ for triplets that don’t match. The only information retained from the searches is the quality of the protein match, not its location. More about BLAST Atlases can be found in Hallin *et al.*, 2004 and Binnewies *et al.*, 2005. More recently, some web services have become available for this tool (Hallin *et al.*, 2008).
Figure 11.11. BLAST Atlas of the four *E. coli* strains of Figure 11.9, with K-12 W3110 as a reference strain to the left, and CFT-73 to the right.

The BLAST Atlas is particularly informative when comparing many different genomes from different isolates of the same species, or closely related species, as we will see in the next few chapters. Figure 11.12 shows two BLAST Atlases, with the same two reference genomes as in Figure 11.11, now showing 20 different *E. coli* (including *Shigella*) strains, a subset of the species included in the BLAST Matrix shown earlier. The genes present in the reference genome were also BLASTed against the UniProt database (outermost lane). Genes of CFT073 absent in the UniProt database (producing a gap in that lane) mostly correspond to non-translated genes.

It should be mentioned that these analyses were done on chromosomal DNA only, thus ignoring the plasmid DNA that gives some of these organisms their particular virulence properties. Notice that even in the smaller *E. coli* K-12 genomes, there are islands of genes that are present in the K-12 isolates, but not found in other *E. coli* strains. Thus, even smaller genomes can contain genes unique to that set of strains. Again, there are many gaps found in other genomes, when these are compared with the CFT073 reference genome. Where this diversity is coming from will be the subject of the next few chapters.
Figure 11.12. BLAST Atlas of 20 different \textit{E. coli} strains, with the same two reference strains as shown in Figure 11.9. The simplified legend only shows the color codes of the BLAST lanes.
Concluding remarks

In this chapter we have addressed a few common questions encountered in microbiological research. The complex process of sequencing and annotation of a genome can always be improved. Nevertheless, with the tools described in this and previous chapters, a genome sequence can already reveal quite a lot about the organism. The BLAST Matrix and BLAST Atlas have been introduced to compare multiple genomes, comprising an enormous amount of information into a single figure.

References:


12 Microbial Communities: Core and Pan-genomics

**Outline** By comparing multiple genome sequences from the same species we can define the ‘pan-genome’ as all genes that can potentially be present in that species. The species pan-genome can be about twice the size of an individual genome. Within a species pan-genome there is a conserved core set of genes present in every member of that species, which we define as the core genome. Those genes not belonging to the core genome are responsible for the sometimes surprisingly large variance in size of individual genomes within a species. Pan- and core genomes can also be defined for genera. Once again, the range of biological diversity is much larger than previously recognized. Moving one level up, one can even conduct ‘comparative pan-genomics’ by comparing the pan-genomes of different species. Comparative pan-genomics is a very new field of research, and promises to yield much information about genome organization and evolution.
Introduction

The previous chapters mostly concentrated on individual genomes, and how to compare particular observations based on these genomes. The next few chapters will look at bacterial populations and communities. In this chapter we compare multiple genomes that are relatively similar to each other, derived from the isolates belonging to the same species or genus. By comparing gene content of genomes, one can recognize common trends within such clearly defined taxonomic groups. The next chapter will discuss analysis of the entire DNA collection from environmental samples without the need to culture individual species. The final chapter of the text will discuss how evolution leaves its marks on a bacterial genome, and how bacterial species are changing with time, becoming better adapted and optimized to their changing environments or dealing with novel selection pressures.

Defining pan-genomes and core genomes

When the second genome of the same bacterial species was published (Helicobacter pylori, published by Alm et al. in 1999) some people considered it a waste of resources. Why bother sequencing a second ‘copy’ if so many species were still completely unexplored? Naturally, the publication presenting the work spent a good deal on comparing the two available genome sequences, both highlighting the commonalities and the differences. Few people envisaged that within 10 years such comparisons would become a novel scientific field on its own.

This new branch of comparative genomics became ‘official’ in the year 2005, when Tettelin et al. compared 8 different Streptococcus agalactiae genomes, of which they had sequenced 6 themselves. S. agalactiae is a leading cause of meningitis and sepsis in newborns. The authors coined the word pan-genome to describe the collection of all the S. agalactiae genes that were found amongst these 8 genomes. Most of the S. agalactiae genomes contain around 2100 genes; however, genes that are present in one genome can be missing in another. By adding all these genes up, Tettelin et al. found that there can be roughly 3000 different genes present in the 8 investigated S. agalactiae isolates to make up their pan-genome. Some of these genes are only found in one or a few genomes, whereas others are present in most or all of the sequenced isolates. Further analysis revealed that only over half of these (~1800 genes) are conserved in all 8 genomes. These are called the ‘core genes’ and their combination would produce a hypothetical core genome. Thus, whilst a typical S. agalactiae genome contains ap-
proximately 2100 genes, the pan-genome of the investigated genomes comprised around 3000 genes and the core genome consisted of about 1800 genes.

It should be noted that neither a core genome nor a pan-genome exist in nature. These are hypothetical combinations of genes that describe the full genetic repertoire of an investigated population (as in the pan-genome, in this case covering 8 strains of one species), or the hypothetical set of genes that will always be present in the investigated population (the core genome). Nonetheless, these concepts aid us conceptually in understanding the relative relationships between bacterial isolate, compared to its neighbors.

Since the core genome covers all genes conserved between all (sequenced) members of a species, it would bear some essential ‘signals of identity’: here we expect to find those genes that make a species what it is. However, the core genome will also contain all genes that are essential for all life forms, such as genes coding for transcription, translation, replication and essential metabolism proteins. The latter, universally conserved gene set, is also referred to as the minimal gene set. The difference between core genome and pan-genome gives the number of genes that are not essential for survival, at least not in all conditions that the bacteria encounter. These are sometimes referred to as dispensable genes. The number of dispensable genes is not constant between species and can vary between 4% to more than 50% of the genes present an individual species’ genome. Some of these dispensable genes may form intricate networks of redundancy, meaning that if a particular gene (or set of genes) is present, others can be absent. We are only scratching the surface to unravel such inter-dependencies and this is an exciting field of comparative genomics that will almost certainly make much progress in the next few years.

In the previous chapter the BLAST Atlas was introduced. Here, we emphasis that for ‘true’ comparisons, one needs a standardized way of gene finding. Naturally, analyses like these, where different genomes are compared for gene content, would be heavily polluted if one genome were annotated with stricter rules on gene finding than another. Therefore, all examples treated in this chapter are performed with genomes that have been run through the same gene finder program, with the same cut-off threshold. We do not claim to have the ‘best’ gene finder and annotation pipeline, but at least it provides a standard within the analysis. If our gene finder makes mistakes, it will be a consistent mistake throughout, and thus will not affect the outcome at large. For the definition of a pan- and core genome, we have to define a cut-off value beyond which we consider two genes no longer as part of the same gene family. We use a cut-off of 50% identity over 50% of the protein length (comparing amino acid sequences), for short the ‘50-50 rule’. Choosing another cut-off value will affect the resulting core or pan-genome only moderately, as our analyses have shown.
**Hypothetical pan- and core genomes**

Before we move on to real observations, imagine two different scenarios. In the first scenario, we sequence two genomes of a species that consist of completely identical isolates (as if the whole population existed as one clone). In this case, the two genomes would contain the same genes, and pan- and core genome would overlap. In real cases this is unlikely, as point mutations do occur in any natural population, but a single point mutation is not sufficient to screen two genes as different. Nevertheless, even genomes that were sequenced from very closely related isolates can differ, as we saw with the three different genomes of *E. coli* K-12 presented in the last chapter.

In a different scenario a species may be made up of isolates (strains, clones, serovars, pathotypes or whatever bacteriologists would call them) that differ very extensively in gene content. In this case, the pan-genome of the two strains will be bigger than their core genome. That would apply, for instance, comparing the relatively large genome of *E. coli* CFT073 with the smaller genome of its commensal K12 sibling. But what happens if we would add, as an additional genome, another large *E. coli* genome? This newly added genome is likely to be more closely related to the large genome already present in our comparison than to the smaller K12 genome.

![Figure 12.1. Venn diagram of the number of genes shared by three strains (A, B and C) of *S. agalactiae*. Strains A and B are most similar. To the right the pan-genome of the three strains is compiled, either by starting with A and adding B first and then C (top right) or adding C first and then B (bottom right).](image-url)
A similar situation is depicted in Figure 12.1, representing a Venn diagram for three \textit{S. agalactiae} strains, this time annotated by our standard gene-finding program. The figure to the left shows diagrams for each pairwise comparison, and for the triple comparison, giving the number for shared and unique genes for each of them. The interceptions show that strain A and B are more similar to each other than either is to C (they share a few more genes). The core genome, comprising genes present in all three strains, is given in the middle as 1658 genes (this lower number compared to the estimate by Tettelin \textit{et al.}, may be due to the chosen cut-off value for gene-finding). Note that the number of genes shared by any two strains is higher than the number of strains shared by all three. If we look at the pan-genome of the three pairs from Figure 12.1, the pan-genome of A and B (2317) is smaller than that of the two other combinations. This can be explained by the fact that the total number of genes present in each genome is not constant. To the right of the figure it is shown how the number of genes identified in the pan-genome will increase with each novel genome added but the shape of this graph is influenced by the order of addition, although the final point remains the same. Tettelin \textit{et al.} explored this further, by analyzing all the possible combinations of the eight genomes. When analyzing larger numbers of genomes, however, the problem quickly scales out of hand, as there are \textit{n}! possible orders. As an approximation for doing large comparisons, we sort genomes of the same species by the number of proteins present per genome. This might not be the most satisfying method for mathematicians, but this practical solution suffices to see general trends, allowing relatively fast computer calculations.

At the end of this chapter we will show the computations for comparison of 54 \textit{Burkholderia} genomes, which took two days to compute using 50 high speed processors in parallel. Analyzing 54 genomes by looking at all possible orders is simply impossible (if analyzing 54 genomes in one order would take 24 hours, the full combination would take about 54! days, which is 2.3x10^{73} days or 6.3x10^{68} years, compared to the ‘big bang’ being about 14x10^9 years ago). Even if it were possible to calculate, we would not know a whole lot more from all possible combinations than we do from our approximation. Continuing on our theoretical example, adding a novel genome that is significantly different to those already analyzed will cause a ‘jump’ in the pan-genome graph, as illustrated in Figure 12.2, panel A, showing a pan-genome of two species belonging to the same genus. Panel B illustrates how the number of core genes goes down as more genomes are added. The solid dark bars in Panel B represent the number of new genes discovered; the first bar gives the total number of genes in the first genome, and then for each additional genome the number of novel genes this genome adds to the pan-genome is given. Panel D gives the number of gene families. For example, the \textit{E. coli} K-12 MG1655 genome has 4302 genes annotated belonging to 4054 gene families (ABC transporters are an example of one gene family), so the gene family bar in Panel D would be a bit lower than the corresponding gene bar in Panel C.
Figure 12.2. Theoretical pan-genome and core genome plots of 10 genomes representing 2 species belonging to one genus. The pan-genome (panel A) increases as more genomes are added, with a ‘jump’ when a second species is introduced. The core genome (B) is reduced with each added genome, when genes are lacking that were conserved in the previous genomes. Panel C shows the number of new genes added to the pan-genome for each new genome. Panel D gives the number of new gene families. The genomes are ordered for increasing size.

When we add more genomes, it makes sense to focus on newly added gene families, as these have a stronger biological signature when compared to new genes. In our standard pan- and core genome analyses, we represent the blue pan-genome and red core genome using gene family data, but we provide the bars for both individual genes and gene families. The hypothetical pan-genome and core genomes of Figure 12.2 level off as more genomes are added within one species. This is what we observe with real genomes, too, as will be shown below. However, it depends on the species how many genomes are needed before a plateau is reached, and for some organisms it is predicted that a plateau will never be observed. That means that every novel genome sequence of an independent isolate would add novel information to the core and pan-genomes. Such a species has a so-called open pan-genome.
Current data available for pan- and core genome analysis

As of March 2008, there are more than 30 different bacterial genera for which multiple genomes have been sequenced. Taking ongoing sequencing projects into account, there will be a large number of species available soon for analysis. The developments are illustrated in Table 12.1, where the total number of genome sequences are listed for the bacterial genera of which the most sequenced genomes are expected to be soon publicly available to date, based on ongoing sequencing projects. Of these, we will pick three from the top of the list, to show some general trends.

<table>
<thead>
<tr>
<th>Genus</th>
<th>Sequencing projects</th>
<th>Nr. of species</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nr. of finished</td>
<td>With projects</td>
<td>Nr. of finished</td>
</tr>
<tr>
<td></td>
<td>projects</td>
<td>in progress</td>
<td>genomes</td>
</tr>
<tr>
<td>Streptococcus</td>
<td>27</td>
<td>76</td>
<td>8</td>
</tr>
<tr>
<td>Clostridium</td>
<td>15</td>
<td>59</td>
<td>10</td>
</tr>
<tr>
<td>Burkholderia</td>
<td>16</td>
<td>59</td>
<td>9</td>
</tr>
<tr>
<td>Bacillus</td>
<td>17</td>
<td>55</td>
<td>10</td>
</tr>
<tr>
<td>Salmonella</td>
<td>7</td>
<td>43</td>
<td>2</td>
</tr>
<tr>
<td>Escherichia</td>
<td>11</td>
<td>43</td>
<td>1</td>
</tr>
<tr>
<td>Vibrio</td>
<td>7</td>
<td>35</td>
<td>5</td>
</tr>
<tr>
<td>Mycobacterium</td>
<td>17</td>
<td>32</td>
<td>10</td>
</tr>
<tr>
<td>Listeria</td>
<td>4</td>
<td>29</td>
<td>3</td>
</tr>
<tr>
<td>Yersinia</td>
<td>11</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>Mycoplasma</td>
<td>13</td>
<td>24</td>
<td>11</td>
</tr>
<tr>
<td>Shewanella</td>
<td>16</td>
<td>24</td>
<td>11</td>
</tr>
<tr>
<td>Pseudomonas</td>
<td>14</td>
<td>24</td>
<td>7</td>
</tr>
<tr>
<td>Borrelia</td>
<td>3</td>
<td>23</td>
<td>3</td>
</tr>
<tr>
<td>Haemophilus</td>
<td>6</td>
<td>23</td>
<td>3</td>
</tr>
<tr>
<td>Staphylococcus</td>
<td>18</td>
<td>22</td>
<td>4</td>
</tr>
<tr>
<td>Campylobacter</td>
<td>9</td>
<td>22</td>
<td>2</td>
</tr>
<tr>
<td>Synechococcus</td>
<td>11</td>
<td>21</td>
<td>5</td>
</tr>
<tr>
<td>Francisella</td>
<td>7</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>Lactobacillus</td>
<td>11</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>Rickettsia</td>
<td>10</td>
<td>15</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 12.1. The number of sequenced genomes and ongoing projects of a various bacterial genera and the current numbers of available multiple genomes per species). The top-scoring genera are listed only.
The pan- and core genome of *Streptococcus*

The genus *Streptococcus*, a member of the Firmicutes, is currently represented by 27 genome sequences in GenBank, with almost 50 other genomes in progress. Figure 12.3 combines the four kinds of plots introduced in Figure 12.2. In their original *S. agalactiae* pan-genome paper, Tettelin et al. looked at 8 sequenced *S. agalactiae* genomes, although only three of them were finished to one contiguous piece. Three years later, the other five are still very fragmented (ranging from 155 to 553 contigs). As the publication shows (and as was introduced in the previous chapter) one can do this kind of analysis on incomplete genome sequences, as long as the number of contigs is not too big (otherwise genes are missed as too many gaps are not yet closed) and the sequence is of sufficient quality. In our experience, incomplete genome sequences of more than 1000 contigs must be regarded with suspect and interpreted with caution, as the sequence could be of poor quality, with lots of sequencing errors or unknown nucleotides, which interfere with correct gene-finding and identification. Therefore, in Figure 12.3 only 26 completed *Streptococcus* genomes (available in few contigs or of acceptable quality) are used.

![Figure 12.3. The pan-genome (blue line) and core genome (red line) for *Streptococcus*. The number of discovered novel genes (dark bars) and novel gene families (light-grey bars) are also shown for each added genome.](image-url)
Since the majority of finished genomes are from *S. pyogenes* (12), this species is shown first. *S. pyogenes* is a normal inhabitant of the human nasopharyngeal cavity, but it commonly causes uncomplicated pharyngitis and tonsillitis. In rare occasions *S. pyogenes* can cause infections of the skin that can grow out of control, giving it the scary nickname ‘flesh-eating bacterium’. The genome of *S. pyogenes* (as other species of the genus) contains approximately 2000 genes. The way the pan- and core genome curves level out after the addition of the 6th genome of *S. pyogenes* suggests that the true variation of this species is more or less covered.

As expected, there is a jump when adding a new species, in our case *S. pneumoniae*. This bacterium lives in the same niche as *S. pyogenes*, and causes more frequently sinusitis, otitis media and conjunctivitis than the pneumonia that it is named for. Introducing *S. thermophilus* only has a small impact on core and pan-genome, although it is known for a different ‘lifestyle’: it belongs to the lactic acid bacteria that are frequently used as starter cultures for fermented dairy products such as yoghurt and cheese. Imagine that such a harmless organism is so closely related to the ‘flesh-eating bacterium’. Following are the three *S. agalacticae*, and two more species, each currently represented by single genomes.

The pan-genome of the currently sequenced *Streptococcus* genus contains approximately 10,000 genes, roughly 5 times more than are present in a typical *Streptococcus* species. However, if we look at the pan-genome of an individual species, for instance *S. pyogenes* or *S. pneumoniae*, their species pan-genome would contain approximately 25% more genes than their typical genomes. The core genome of a single *Streptococcus* species contains approximately 75% of their individual genomes. Thus, approximately a quarter of genes are dispensable genes.

One thing to note from Figure 12.3 is that the core genome can be hardly affected when moving from one species to the next (with the exception of the shift from *S. pyogenes* to *S. pneumoniae*), whereas the pan-genome is significantly increased, using the same scale. This is particularly obvious for the shift from *S. pneumoniae* to *S. thermophilus*, and from that to *S. suis*. This adds credibility to the concept of bacteria species as confined groups, all of which belong to a genus, and apparently share a significant core genome. Some taxonomists currently argue that there is no such thing as a bacterial species. Yet, we see clear grouping of organisms based on these plots, with the only data being used a sequence file of proteins from each bacterial genome. The jumps seen in the pan-genome plot were not predefined or manipulated in any way, but is dictated by the genes represented in the sequences. In our opinion it truly is an independent reflection of a set of conserved genes complemented with variable but still species-specific genes within a species.
The current *Bacillus* pan- and core genome

The genus *Bacillus* will serve as a second example. The 24 *Bacillus* genomes of sufficient quality that were publicly available were used to construct a pan-genome and core genome plot of the *Bacillus* genus, shown in Figure 12.4. Each *Bacillus* genome contains approximately 5000 genes. The first 9 genomes represented are all from *B. anthracis* (the cause of anthrax). In this part of the graph the core and pan-genome curves are relatively shallow, separating only slowly as one moves along the added genomes. Hardly any novel genes or gene families are identified in each additional *B. anthracis* genome; this is an example of a species that behaves as if it were one clone. With genomes from isolates originated in North America, Europe, South Africa, and Asia this is pretty amazing. The core genome of *B. anthracis* is estimated around 4800 genes (96% of the genome) and the pan-genome is only marginally larger than individual genomes.

![Figure 12.4 The pan-genome and core genome for 24 sequenced *Bacillus* genomes.](image)

The next, most closely related species is *B. cereus* (an opportunistic pathogen that can cause food poisoning), but the species jump is not dramatic. In fact, if this analysis had included plasmids, rather than being based on chromosomes only, there might have been a slight change in pattern, as *B. anthracis* and *B. cereus* have very similar chromosomes and most of the extra genetic material for patho-
genesis is in the plasmids. *B. anthracis* could be regarded as a *B. cereus* with the addition of two virulence plasmids that makes the bacteria so pathogenic (some *B. cereus* strains even contain these plasmids but without the toxins that are responsible for the disease anthrax). The fact that the core and pan-genome curves depart more rapidly for *B. cereus* shows that there is more variation in gene presence or absence. Here, more novel genes and gene families are detected for each newly added genome.

A significant jump is introduced when moving to *B. subtilis*, a soil bacterium and the next member of the genus included for which we currently only have one genome sequence available. The two *B. thuringiensis* genomes appear very similar again (*B. thuringiensis* is a soil bacterium and insect pathogen that produces Bt toxin, an insecticidal toxin used in genetically modified crop plants). The last 5 genomes are again single representatives for their species. The currently known pan-genome of the *Bacillus* genus is about 20,000 genes, and since the curves of most of its species have not leveled off yet, it is likely to grow as more genomes are being sequenced. Compared to the genome size of a typical *Bacillus* species, the pan-genome of the genus contains 4 times more genes.

Comparing figures 12.3 and 12.4, it seems that the *S. pyogenes* core genome and the pan-genome diverge more than for *B. anthracis*, but correcting for the difference in scale, in fact the two species have similar levels of diversity.

**An overview of some Proteobacterial pan- and core genomes**

With the explosion in genome sequencing, there are now many organisms with multiple genomes sequenced. By far, the Proteobacterial group has been over-represented, with about half of all bacterial genomes sequenced belonging to this group of Gram-negative bacteria, although in the ‘real world’ environment, it is likely that this phylum comprises only a tiny fraction of the total diversity. Nonetheless, it is worth having a look at some of the different organisms in this well-sampled group. Figure 12.5 shows pan-genome plots for five different Proteobacterial genera. As in the previous examples, the part of the curve connecting genomes belonging to the same species tends to be more flat, and then a jump is seen when a genome from a different species are added. The *Salmonella* figure is based on genome sequences of *Salmonella enterica* only, and although the *E. coli* figure combines genomes of several *Shigella* species, it can be argued that these could all be treated as one species (see the previous chapter). Compared to the other graphs, all representing multiple species per genus, the *E. coli* pan- and core genome curves are strikingly divergent.
A study based on 7 sequenced E. coli genomes predicted that the E. coli pan-genome was open, and that each new E. coli genome sequenced would have, on average, about 441 genes (Chen et al., 2006). This predicted number of new genes seems rather high, compared to the 30 or so genes predicted to be added with each new Streptococcus genome (Tettelin et al., 2005). We have estimated the number of new genes as approximately 80 per E. coli genome, based on 32 sequenced E. coli genomes; we estimate the E. coli species pan-genome to contain about 9433 genes, whilst the core genome contains 2241 genes (Willenbrock et al., 2007).

The **Burkholderia** pan- and core genome

As a final example of what insights can be obtained from multiple genome sequences, we will consider the Burkholderia genus (another Proteobacteria, in this case of the β-division). There are 54 Burkholderia genomes currently available, and this genus represents quite a broad spectrum of organisms, including one of the largest genomes sequenced so far (B. xenovorans is 9.8 Mbp). We included all 54 Burkholderia genomes in our analysis shown in Figure 12.6. This shows an estimate of the full extent of diversity one can observe in bacterial genera. Note that
in this figure, the number of genes in the human genome is indicated by an arrow, and it is about half the total diversity found within the Burkholderia gene families! However, at this level, it is now difficult to see individual species, although the Burkholderia pseudomallei genomes (from position 13 to 23) form somewhat of a plateau.

![Figure 12.6](image)

Figure 12.6 The pan-genome and core genome for Burkholderia. The arrow indicates the number of genes identified in the human genome.

There are 10 B. pseudomallei and 9 B. mallei genomes included in Figure 12.6. B. mallei is the cause of glanders (an infectious disease in animals, common in domesticated animals such as horses that was already described by Aristotle) and its taxonomic position in Burkholderia is relatively recent\(^1\). B. pseudomallei is more common in tropical areas and is an opportunistic animal pathogen (it lives in the soil); it causes melioidosis which is endemic in Asia and Australia. These species have two chromosomes, which of course have both been included in the analysis. Their typical genome contains approximately 5000 (B. mallei) to 6000 (B. pseudomallei) genes.

\(^1\) Previously, Burkholderia species have been classified as being members of Bacillus, Acinetobacter, Pseudomonas and other genera.
A zoom of the pan- and core genome graphs is shown for these two species in Figure 12.7. Now it is visible that many of the *B. mallei* strains are quite similar, and there is no clear jump to the *B. pseudomallei* genomes. This is the same if the order is switched, and the *B. pseudomallei* genomes are given first (right panel). This trend makes sense in light of evidence that *B. mallei* arose as a clonal deletion from a *B. pseudomallei* strain. Nevertheless, the diversity within each of these species is greater than that of the Gram-positives we have compared before. For instance, the pan- and core genome of these two *Burkholderia* species differ with 4500 genes, compared to 1500 genes difference for *S. pyogenes* and 500 for *Bacillus anthracis*. The number of dispensable genes for these species is approximately 30% (for *B. mallei*) and 50% (for *B. pseudomallei*) of their genome. This high degree of variation in gene content is also apparent from the pan-genome of the genus. Based on 54 *Burkholderia* genomes, the genus pan-genome would contain more than 50,000 different gene families, whereas its core genome consists of only 1339 protein genes families.

![Figure 12.7 Pan-genome plots for *B. mallei* and *B. pseudomallei* genomes. The order of the two species is reversed in the two panels.](image)

**Where do all these genes in the Burkholderia pan-genome come from?**

An obvious question is what are all of these dispensable genes, and where do they come from? Some of the genes are only found in one (or a few) *Burkholderia* isolates. Part of this diversity comes from *genomic islands*, which are regions of a chromosome with an AT content different from the rest of the genome and which have most likely been acquired by horizontal gene transfer. They are typically found only in a subset of strains within a species.
Genome islands can be visualized on a Genome Atlas. An example of this is shown for the *B. pseudomallei* strain K96243, in Figure 12.8. There are several regions that are quite a bit more AT-rich, and can be readily visualized for their structural properties (blue for curvature, red for stacking energy, green for position preference). Their deviant base content is better visible on the GC skew than on the percent AT lane, due to the used scale. These regions contain genes coding for proteins which cannot be found in the proteome of the *B. pseudomallei* strain 1106a genome, as shown by the outer circle which contains a BLAST lane for the strain 1106a proteome.

**Figure 12.8.** Genome Atlases of *Burkholderia pseudomallei* strain K96243, with an extra outer lane added from a BLAST Atlas, comparing *B. pseudomallei* strain 11106a. The position of some of the genome islands is indicated by arrows where genes unique to strain K96234 reside. These genome islands have strong structural properties.

Do genomic islands such as those identified Figure 12.8 correspond to regions that are missing in other *B. pseudomallei* genomes as well? This can be easily visualized in a more complete BLAST Atlas, shown in Figure 12.9, which now includes 10 sequenced *B. pseudomallei* genomes, along with 9 *B. mallei* and 17 other related genomes. Note that *B. mallei* contains several deletions from the larger *B. pseudomallei* genomes; this can be seen more clearly in the BLAST Atlas of chromosome 2, shown at the bottom of Figure 12.9. Genome islands will be further discussed in Chapter 14.
Figure 12.9. BLAST Atlases comparing 36 different *Burkholderia* genomes. The *B. mallei* genomes contain large deletions compared to the *B. pseudomallei* genomes, especially in chromosome 2 (bottom).
Concluding remarks

The pan- and core genome analyses introduced in this chapter are extremely useful to investigate the true diversity within and between bacterial species. We have only commented on the fractions of genes comprising the pan- and core genome, compared to individual genomes, but once the genes are identified that are part of the core genome or that make up the variable part, interesting questions can be asked. It should always be stated on which genome collection the predicted pan- or core genomes are based, as the selected genomes will have an effect on this outcome.

References:


13 Metagenomics of Microbial Communities

Outline  Metagenomics is the study of the DNA from all the genomes in an environment. The term meta implies that this transcends traditional genomics. Most of the bacteria living in an environment will not grow in standard laboratory media. This is true for more than 99% of the species present in a typical soil sample, and similar numbers are likely to hold for bacteria growing in different environmental niches. Thus, by sampling all of the DNA from a given environment, it is possible to gain much additional information that would not be available from traditional methods that depend on single, pure monocultures of a well-characterized bacterium. The area of metagenomics is relatively new and rapidly changes as technology allows more and better sampling of the environmental DNA. The consequences of current improvements in speed and output of genome technology to future research are discussed.
Introduction

The term ‘metagenome’ was first introduced in the scientific literature more than 10 years ago (Handelsman et al., 1998) to describe “the cloning and functional analysis of the soil microflora, called the soil metagenome”. Metagenomics has since been defined in various ways, but in general it describes the characterization of all DNA present in a particular environment. More specifically, in our field of research the interest would be limited to all microbial, or even all bacterial DNA (excluding any viral or eukaryotic samples, through filter procedures, for example). Such an approach enables us to study the presence of bacterial ecosystems, independent of the ability to culture their components in the laboratory. Metagenomic analysis follows up on the simpler approach to characterize all the 16S rRNA genes present in a sample (without culturing), as this gene is often used for taxonomic classification (see Chapter 9).

Only in very recent years has it become feasible to sequence all given DNA of a particular environment, as sequencing costs came down and the computer power needed to put all pieces of the DNA puzzle together has increased. Nevertheless, complete genomes are hardly ever produced in metagenomic analysis. In most cases, gaps of missing DNA or repeat sequences hamper assembling complete genomes. Not that this matters, though. In a more holistic approach, the emphasis is on the genetic potential of an ecosystem, without too much concern which organism actually carries which genes. Think of a biofilm in which many species live together as a super-organism: biological compounds made by one species will be used by the next so that the metabolism of the biofilm can be more than the sum of its individual components. In a book cited at the end of this chapter that is dedicated to the subject, the term metagenomics is said to transcend individual genes and genomes, meaning that it goes beyond the limits of traditional genomics. In contrast to classical genomics as the study of DNA of single organisms, metagenomics looks at populations and communities.

The metagenomic approach is necessary to put our knowledge on the bacterial kingdom into perspective. If one were to take a spoonful of soil, it would likely contain billions of bacteria, representing many thousands of different species. But try to grow bacterial cultures from this, and far less than 1% of the bacterial species present in the soil will yield cultures in the lab (reviewed by Streit and Schmitz, 2004). It is sobering news that the proud history of more than a century of bacterial methodology has been dependent on a limiting step that removes many (most) bacteria: the prerequisite of growing a monoculture in standard laboratory media. There is a lot more out there that until recently we have not been aware of, or at any rate has been difficult to study.
Metagenomics based on 16S rRNA analysis

Direct amplification of 16S rRNA sequences has shown many times that bacteriological culture only detects a fraction of the real bacterial diversity of a given environment. The amplification of 16S rRNA, and investigation of the resulting phylogenetic relationship, has been performed on soil, ocean surface water and deep sea vents, hot springs, the animal and human gut, the human oral cavity and other likely and less likely sites. Such investigations tell us what organisms are out there, but not much more. The bioinformatic analysis of the resulting DNA sequences would involve a neighbor joining algorithm with maximum parsimony, the technical details of which can be the sole subject of textbooks, such as Felsenstein (2004).

Figure 13.1. Phylogenetic tree based on 16S rRNA of 49 bacterial species representing 31 different phyla, indicated by colors. Bold names represent sequences derived from uncultured organisms. Grey blocks identify phyla for which a genome sequence is currently not available. Phyla names appear in red if species are not grouped together in the tree.
Figure 13.1 shows a 16S rRNA tree, for bacterial species representing over 30 known phyla. For each entry at least one sequenced genome was used when available, but for 7 phyla this wasn’t possible, for instance for the phyla mainly representing thermophiles at the bottom of the tree. In fact, available genome sequences are heavily biased towards only a few phyla, as indicated in Figure 13.2. More than half of all bacterial genomes sequenced so far come from the Proteobacterial phylum (333 out of 626); if one adds Firmicutes and Actinobacteria as the next two most abundant phyla more than 80% of all sequenced genomes is covered although they represent less than 10% of all bacterial phyla. As a consequence, the databases are heavily biased towards the overrepresented species. If one were to randomly select a gene from a current genome database, the chance of it belonging a Proteobacterium, Firmicute or Actinobacterium is closer to one out of two than to one out of 30.

For an environmental sample, 16S rRNA metagenomics is certainly a good place to start, in terms of determining “who is out there”, and perhaps even obtaining information on their relative abundance. However, constructing a “proper” 16S rRNA tree from the obtained sequences is not without problems. Even with high-quality, full-length 16S rRNA sequences, there are problems with spacer regions and other differences that make alignment more difficult. Further, as discussed in...
Chapter 9, some of the annotated rRNA genes in the GenBank files for genomes can be incorrectly labeled or are occasionally missing altogether. Finally, often only part of the gene is amplified and sequenced, thus further complicating the issue. Despite these practical difficulties, in general 16S rRNA trees are considered the ‘gold standard’ of phylogenetic, and sometimes taxonomic, relationships. For genomes that are fairly closely related to each other, a simple alignment will usually be good enough for construction of a reasonable phylogenetic tree.

Metagenomics based on complete DNA sequencing

The question ‘who is out there’ of course is not quite the same as ‘what are they doing’, which became the next challenge to investigate. After the first paper describing the possibilities of metagenomics, it took a few years for other people to begin to design experiments and get results from sequencing multiple genomes from the same environment. A major breakthrough came from a ‘pilot study’ in 2004, with the impressive publication of an environmental sample of the Sargossa Sea (Venter et al., 2004). The authors of this publication proudly announced that they had deposited more than a billion bp of sequence to GenBank, including over a million new genes, from about 1800 different species, including 142 new bacterial phylotypes.

Soon, other metagenomic projects followed. Three early studies published data from a Minnesota farm soil sample (Tringe et al., 2005), from a biofilm growing in the unlikely place of a mine acid waste drainage (Tyson et al., 2004), and from a whale carcass as the marine equivalent of a species-rich biological hotspot (Tringe et al., 2005). Since then, the number of publications has steadily increased, as shown in Figure 13.3.

![Figure 13.3. Number of papers available in PubMed on metagenomics of bacterial genomes.](image)
The sequencing data that these projects have generated are not yet completely publicly available for analysis, in part because of their overwhelming quantity. It has taken many by surprise how much and how soon metagenomic projects are being done. At the time of writing, the Genomes Online Database lists 123 genome projects\(^1\), but not all of these present data yet. NCBI currently lists 160 projects (finished or ongoing), but again, not always linking into the sequences\(^2\). The original pilot study of the Sargossa Sea now has its own set of more detailed web pages, with additional samples, integrated in the CAMERA project, which hopefully will provide more and better links to many projects in the near future\(^3\).

**Environmental influences on base composition**

With the restricted amount of information publicly available, the analyses that can be carried out on metagenomic data are currently limited. But from what data we have, one observation is striking: bacteria sharing a particular ecological niche are also likely to share a common ‘genetic dialect’, as they more commonly contain DNA with similar AT content and codon usage than would be expected by chance. This largely enhances the potential gene pool for an individual organism: when DNA is taken up by horizontal gene transfer, the chance that such DNA will be fixed in the offspring of the recipient will depend on any positive effect; which again is more likely to occur if the AT content and codon usage of the new DNA isn’t too different from the recipient’s genome, otherwise there are translational constraints on expression (as discussed in Chapter 10).

We illustrate this observation with three metagenomic samples: the human oral cavity microflora, the human gut microflora, and from uranium-contaminated water, for which codon usage plots are shown in Figure 13.4. The figure for the human oral cavity microflora shows three graphs superimposed, to illustrate that some individual variation exists, but in general there is an overlap in common codon usage for all bacteria living in this environment. These data were based on an unknown number of species representing a gene pool with a total AT content of 63%. Compare this plot with the one in the middle, showing a sample from human intestinal microflora with a shared AT content of around 50%. Especially the distribution of third-base pyrimidines (bottom half of the circle) shows remarkable differences. The acid mine drainage biofilm (of which the data are not shown) has a shared AT content of 54%, and its codon usage plot resembles that of the human gut. A very different picture is visible for the uranium-contaminated water sample (to the right of the figure), which contains organisms with a low AT content (total

---

AT content about 37%) that have a very different codon usage to the other two metagenomic samples.

**Figure 13.4.** Codon usage in metagenomic sequences derived from three different environmental samples. At the left, three plots are superimposed, obtained from three human oral cavity samples. This plot is based on 3,908 protein-encoding genes. In the middle is a human intestinal microbiota sample (20,523 genes), and to the right is a uranium-contaminated water sample (12,335 genes).

**Visualization of environmental metagenomic data**

One of the problems with metagenomic data is that the results cannot be represented in a Genome Atlas as long as the genome sequences are incomplete. The BLAST Atlas as introduced in Chapter 10 requires a reference genome, but without a genome to start with, it is difficult to use this kind of reference plots. Imagine a particular protein of interest were searched by BLAST against a metagenomic database, such as the Sargasso Sea environmental sample. A good hit to this database would mean that a similar protein was present somewhere in the bucket of water that was taken from the sea. But this is pretty much all that is known from this result. There would be no evidence to suggest which organism produced this protein. This is one of many challenges in dealing with vast amounts of data generated from random shotgun reads of an environmental sample.

There would be far more information available if complete genome sequences could be produced from metagenomic samples. Although technology continues to improve, as discussed in the previous chapter the current read lengths of current technology are still relatively short. All these short pieces are difficult enough to put together for DNA derived from only one clonal bacterial genome with few repeats. However, with data from literally thousands of different organisms in an environment, it becomes almost impossible to realistically assemble these small pieces into meaningful larger fragments.
There is a way of visualizing metagenomic data, of course, which is to use an existing, completely sequenced genome as a reference. Ideally, one would choose a reference organism that is known to be abundant in the same environment and that is part of the same phylum. For instance, when dealing with metagenomic samples originating from soil, one may choose a reference genome of a soil bacterium, such as a *Clostridium* species. In Figure 13.5, *Clostridium botulinum* is used as a reference strain in a BLAST Atlas comparing it to other *C. botulinum* strains and to closely related *Clostridium* species. Clostridia belong to the major players in anaerobic environments and contain some common soil species with streamlined genomes that allow them to divide quite quickly, in less than 10 minutes under favorable conditions. The reference strain *Clostridium botulinum* A3 strain Loch Maree was isolated from duck liver paste during a botulism outbreak at a hotel in the Scottish highlands in 1922. When compared to 7 other *C. botulinum* genomes, strong sequence similarity is detected, but in contrast *C. perfringens* genomes or other *Clostridium* species show little similarity.

![Figure 13.5. BLAST Atlas of Clostridium botulinum compared to 12 other Clostridium genomes, including 7 C. botulinum genomes.](image-url)
When a BLAST comparison is made between genomes of organisms sharing a particular ecological niche, this could be regarded as a Metagenome Atlas. Examples for terrestrial bacteria are given in Figures 13.6 and 13.7 using four different bacterial reference genomes. The first reference genome is that of *Heliobacterium modesticaldum*, at the top of Figure 13.6. It is compared to the same set of *Clostridium* genomes of the previous figure, and to a number of other soil bacteria, all belonging to the Clostridia class. Note, that by selecting organisms that share an ecological niche, one may add alternative genera, classes or even phyla, which may differ considerably to the reference genomes. Hence, a Metagenome Atlas can be less colorful than classical BLAST Atlases. In the lower Metagenome Atlas of Figure 13.6, another member of the Clostridia class is used as a reference genome, that of *Pelotomaculum thermopropionicum*, a syntrophic bacterium living off products from other organisms that can perform anaerobic biodegradation of organic matter while producing methane (Koska et al., 2008). This reference strain is compared to the same set as used in the upper panel, but in addition 3 metagenomic sample sets are added: two sludge samples and a metagenomic gut sample. These lanes are as good as empty, since hardly any genes were detected with significant similarity. Metagenomic samples contain many, many genes that don’t find a BLAST hit in the classically sequenced bacterial genomes. In fact, this again illustrates how biased our sequenced genomes from cultured organisms are, as the environmental samples detect very few genes with similarity to those genomes.

The two reference genomes in Figure 13.6 were obtained from environmental bacteria that were classically sequenced from a pure culture. The other two reference genomes for which a Metagenome Atlas is shown, in Figure 13.7, were truly sequenced from metagenomic data, this time allowing to assemble a complete genome. The top atlas of Figure 13.7 shows *Cloacamonas acidaminovorans*, an unculturable bacterium found growing anaerobically in a wastewater treatment plant that probably represents a new bacterium phylum. In this atlas a metagenomics lane is added, but even though the reference genome was also obtained from metagenomics, few BLAST hits are identified. Two explanations can be given for this absence of similar genes: either the metagenomic DNA contained more genes than those sequenced (which is quite likely), so that genes may have been present in the sample but are not represented in the sequence, or the genes were truly absent from the metagenomic sample. In that case the genomic diversity in environmental bacterial DNA is even more immense than we think.

Figure 13.6 (next page). Metagenome Atlas of *Heliobacterium modesticaldum* (top), and *Pelotomaculum thermopropionicum* (bottom) against 30 sequenced members belonging to the Clostridia class. The atlas at the bottom also has lanes included for intestinal *E. coli* and *Bacillus fragilis* (in blue) and for three metagenomic samples (green).
The bottom Metagenome Atlas of Figure 13.7 contains a genome of Termite Group 1 (TG1) bacterium as a reference, which supposedly also belongs to a new phylum. This unculturable organism, whose sequence was obtained from a metagenomics project, grows anaerobically in the intestine of termites. In the Metagenome Atlases of Figures 13.6 and 13.7 we added two other bacterial species commonly found in intestinal flora (E. coli and Bacteroides, shown in blue). Reference genomes H. modesticaldum and Termite Group 1 bacterium identify more similarity to these gut organisms than the other two reference genomes. As TG1 bacteria live in the intestine of an animal, it is of no surprise that it detects more genes in the E. coli genome than the terrestrial metagenomic C. acidaminovorans. After all, the difference between the mammalian and an insect host isn’t that significant.

When we compare the four Metagenome Atlases it can be seen that that the least similarity is detected with Cloacamonas acidaminovorans as a reference genome, as this panel contains the least color. As C. acidaminovorans was sequenced from a metagenomic sample, it may contain a relatively large proportion of ‘novel’ genes. C. acidaminovorans is a candidate for a novel phyla, so it is also possible that the genome contains many ‘unique’ genes for this reason. However, the TG1 bacterium is also considered to represent a new phylum but it recognizes more genes in the Clostridia genomes. Possibly, its phylum is more closely related to Clostridia than C. acidaminovorans is. The other striking observation is that the two Metagenome Atlases based on a reference genome belonging to a different phylum than the query genomes (in Figure 13.7) do not detect fewer genes than the intra-phylum Clostridia comparisons of Figure 13.6. This may either mean that the diversity within Clostridia is already so huge that the detected genes approach the minimum gene set, or it may be that bacterial classes and phyla have less influence on gene conservation than the lifestyle of the organisms that are compared.

**Marine metagenomics**

A final example of a Metagenomic Atlas is given for marine bacteria. This time we choose the Prochlorococcus marinus (a Cyanobacterium) genome as a reference, and compare this with marine metagenomic samples derived from various depths. The record holder is a metagenomic sample from 10 km deep. A number
of classically sequenced genomes of *Prochlorococcus* and other marine Cyanobacteria is also included, ordered for depth of the isolate (Figure 13.8). The amount of blue color shows that this reference genome detects more BLAST hits in these marine samples than was the case for our terrestrial examples. Genes are relatively well conserved within all *P. marinus* genomes included. However, for both the metagenomic samples (in blue) and deep sea organisms other than *Prochlorococcus* (red lanes) similarity is detected to the area around 1250 kbp that appears absent in *Prochlorococcus* lanes. The biggest surprise maybe is that the *Geobacillus kaustophilus* isolated from over 10,000 m depth still shows extensive similarity to the reference genome originated from the surface of the ocean.

![Figure 13.8. Metagenome Atlas of *Prochlorococcus marinus* compared to other *Prochlorococcus* genomes (green), marine organisms other than *Prochlorococcus* (red) and marine metagenomic DNA sequences (blue). The lanes are sorted for the depth of the isolates, ranging from close to the surface of the ocean (outermost lanes) to deeper water, with a depth of over 10 km for the innermost red lane. Two archaeal species are included as indicated.](image)
Other metagenomic applications

Metagenomic sequence analysis has been carried out on various mammalian niches, notably the human intestinal tract, the oral metagenome, skin and the urinary tract during infection. A recent study has found that roughly a quarter of the bacterial genomes involved in urinary tract infections are not readily detected using conventional methods of growing cultures in the laboratory [Imirzalioglu et al., 2008]. The intestinal metagenome of life stock animals is also under investigation.

One application that deserves to be mentioned separately, as it follows a rather unconventional strategy is to use discarded non-eukaryotic sequences from projects dedicated to sequence a eukaryotic genome. The first example of this approach produced a genome sequence of a new Wolbachia species that lives as an endosymbiont in Drosophila. The genome project of Drosophila had produced bacterial sequences as a result of ‘contamination’. The raw sequence data that were publicly available were used by the TIGR team to extract these bacterial sequences. Using this ‘waste bin’ approach, a near-to complete Wolbachia genome sequence could be pulled out that represented a new species. An estimated 98% of the genome is sequenced (Salzberg et al., 2005). Although this is not classical metagenomics, it is worth to keep in mind that discarded sequences can still be quite useful.

Concluding remarks

Environmental genomics can be thought of as a good strong dose of the reality of what we do not know. There is an enormous amount of information contained in the world in which we live, with robust ecosystems containing many important bacteria unknown to most people. Some of these bacteria play key roles, such that their disappearance from the environment could have catastrophic effects on our lives. Metagenomics is an attempt to at least begin to address what is out there, and possibly what is going on, in terms of complex biochemistry in many environments.

Book on Metagenomics

References


Outline  Evolution can be thought of as the adaptation or optimization of species to their environment. Since, at the level of microorganisms, there can be considerable differences in microenvironments, it is not hard to imagine that many bacteria have a constant need to be adaptable and ready to change to new surroundings. In this final chapter, we will take a look at the processes that drive evolution, and at the evolutionary traces that are visible in the DNA sequences of genomes. Mobile DNA elements play an important role in evolution and an example is given for insertion sequences in *Shigella flexneri*. Genome islands can be considered genetic ‘building blocks’ that can be added to, or removed from a genome core. Finally we will take a closer look at *Vibrio cholerae*, how this species differs from other *Vibrio* species, and how a relatively small set of genes can be responsible for niche adaptation (and sometimes speciation). The amount of genomic diversity within closely related bacterial populations is far greater than anyone had imagined, and the raw material for evolution is abundant in the microbial world.
Introduction

As mentioned in the first chapter, cells obey the laws of chemistry and physics, and there is no need to invoke supernatural forces to explain the physical mechanical events happening inside of bacterial cells. One of the undercurrent themes of this book has been to build up a firm ‘post-genomic’ foundation from which to view the bacterial community in the world we see around us. We’ve now come full circle, and in this last chapter, we will have a look at the evidence for evolution within individual genomes, and how we can extrapolate such observations to bacterial populations.

In order for evolution to happen, three components are necessary: (1) a number of organisms must have a diverse set of traits that have different advantages under different conditions, (2) these traits must have the ability to change, and finally (3) selection must take place by some particular condition so that (some of) these traits become dominant in the offspring population. We can add the factor time to this as an essential component, as evolution is rarely instantaneous, and will take time. Before turning to biological examples, we will first take a closer look at evolution in general.

Where does diversity come from?

The first requirement of evolution is a diverse set of organisms from which to select a subpopulation. Some creationists who oppose evolution claim that there is not enough diversity in bacterial populations for evolution to occur. In contrast, from a genomics perspective, it should be obvious from all previous chapters that the amount of diversity we observe in sequenced bacterial genomes is very large indeed, far larger than many (most) people ever expected. As we’ve seen in Chapter 12, the diversity of bacterial species can be many times the size of any individual genome. In Chapter 13 we’ve illustrated that we only know a tiny bit of the variation that is out there in the real world.

So where does this diversity come from in the first place? Again, in contrast to what creationists may believe, the presence of diversity is not the result of a ‘vitalist’ divine force: genetic diversity will develop from a homogeneous population as a result of mistakes made during DNA replication (as well as from viral, IS element and transposon DNA inserting and being deleted from the chromosomes,

---

1 See for example “The Edge of Evolution: The Search for the Limits of Darwinism” by Michael J. Behe.
amongst other mechanisms), a slow but steady process that produces genetic diversity over time. On many occasions in this book the Structure / Function Relationship was investigated, which states that DNA sequence determines structure, and structure determines function. Changes in the DNA sequence can sooner or later result in changes in function of the encoded protein, which can then produce a minor or major advantage to the cell during a particular condition. Some of these changes can become dominant in the population when conditions apply that give the carriers of that ‘novel’ trait an advantage. DNA can also form structures (discussed in Chapter 8) that can result in duplications (as well as deletions) during replication. Further, some regions of the chromosome are ‘hot spots’ for insertion of mobile DNA. This way, one or several genes can be duplicated (or even occasionally a complete genome) and since this extra copy is now free from selective constraints that work on the original, novel mutations can accumulate. Deletions that remove genes or parts of genes will also have functional implications.

In addition to these scenarios, DNA can sometimes be exchanged between cells, so that an advantageous trait can spread more rapidly and reach a larger population. Bacteria can exchange DNA by a number of ways, some of which are highly sophisticated and others can be as simple as ‘eat what is there’. Bacteriophages, plasmids and other mobile elements also propagate DNA transfer. There are restrictions that limit DNA transfer, though, and the most obvious is that cells must physically be in each other’s vicinity; many more factors determine the possibility and efficiency of DNA transfer but we will ignore these in this context.

Figure 14.1. Genomes get bigger and smaller, due to exchange of plasmids, IS elements, and of course insertions and deletions. The numbers refer to genetic events, explained in the text.
As shown in Figure 14.1, a genome can undergo enrichment by gene acquisition, or reduction by gene loss, via the mechanisms listed at the top and bottom of the figure. The numbers indicated next to the expanding and shrinking genome represent individual genetic events, where the genome expands (left) or shrinks (right) from the light grey form to the dark grey form. In events 1, 2, and 3 (left), a piece of DNA inserts itself, and at a later stage in events 4 and 5 a new insertion takes place. In the case of event 5 this happens at the same site as insertion 3 did. During shrinking, in event 6 the inserted region from event 1 is removed but not completely, leaving a remnant visible on the genome. In event 7, more DNA is excised than was originally inserted, resulting in a deletion compared to the original situation. In event 8, two parts of the individual insertion events 3 and 5 are deleted, producing a novel junction. The DNA that had been inserted during event 4 is permanently fixed in the genome. Such events take place all the time, leaving evolutionary imprints from the past on any genome that we can recognize today.

**Evolution takes time**

A lot of time is needed for mutations to accumulate in progeny due to replication mistakes. Compared to this, the transfer of genetic information from one existing organism to another is extremely rapid (taking place in seconds) but the chance that this happens, and that the DNA is maintained in the receiving cell can be quite low; hence time is needed before its consequences can be detected. If there is only a minor advantage for having some new trait, selection can take a long time before the complete (or a detectable) population contains this trait. However, a strong selection pressure and a huge advantage can result in rapid spread and fixation of a trait, resulting in a rapid shift of the population. Think of the application of antimicrobials (antibiotics) to bacterial cultures (or to treat a bacterial infection): these work as the ‘grim reaper’ nearly wiping out complete populations, with only few lucky cells that survive. Their offspring are the only to propagate and as a result the population has ‘become resistant’. But of course what happened is that resistant bacteria were positively selected for; these could have been present in the population a priori in below-detection levels, or they could have mutated by chance on the spot, or even they could have received a resistance gene from some other bacteria that happened to be present. This is evolution in a nutshell. Although some microbiologists would argue that this is ‘adaptation’ and not evolution, there is no fundamental difference between the two. Many adaptive steps accumulate to larger evolutionary processes.

Although evolution of eukaryotes follows the same principles, the process is more difficult to observe in real time than for bacteria. Eukaryotes have generally longer generation times, reproduce mainly sexually, and phenotypic differences or
similarities can obscure underlying evolutionary processes. Evolution in bacteria, however, can be observed on the spot and in viruses it can be even faster.

The time factor is needed to accumulate incremental evolutionary steps, each of which may only have minute effects, but all of which can sum up to a significant difference, say, of a gene having a new function, or a bird’s population becoming different enough to be called a new species. Some religious people are opposed to the idea of ‘random chance’ playing such an important role in life; the term ‘evolution’ is currently quite controversial in the U.S. and in some countries in the Middle East, though evolution is scientifically accepted by the vast majority of biologists all over the world. From our perspective, instead of fretting about evolution, we should instead perhaps marvel at the miracle of evolution and the wonderful diversity of life we see all around us.

Evolution is a continuous process, but its time scale is not always easy to work with. When you grow a bacterial culture from a single cell (which would produce a colony on an agar plate), all those cells are identical, or nearly identical, to the original predecessor cell. So why isn’t this bacterial population evolving? It is, but time is too short, mutations are too few, and selection is too weak, to show the result of ongoing evolution in this case. However, if you were to sequence the genome of a bacterial colony, culture some of its cells for, say, 10,000 generations, and sequence a colony again, you would detect differences. If the bacterial population would have had a chance to exchange DNA with other organisms, compete with other bacteria and fight off infectious bacteriophages (all of which is not the case in a pure, axenic culture), and if rigid selection pressure had been applied, the detected differences may not at all be minor or trivial. So, although evolution seems slow in our day-to-day perspective, it nevertheless is a continuous process.

Another way to look at evolution is to search for its evidence in an existing genome. In that case it can appear as if evolution is extremely common: you see evidence of past evolutionary processes all over the genome. How can that be explained in view of the ‘slowness’ just mentioned? Again, time plays its tricks. We can’t tell how much time was needed for all that evidence to accumulate. Genomes are an evolutionary snapshot in time, but the exposure time isn’t given.

---

The species concept is not as strictly defined for bacteria as it is for organisms that reproduce exclusively sexually, and as we will see, considering the genetic diversity between and within bacterial species, the species division becomes more or less useless to describe evolutionary processes in bacteria.
Evidence of evolution in a single genome

A nice example of how some of the principles discussed above can be observed in a DNA sequence is presented in Figure 14.2. It shows the virulence plasmid of *Shigella flexneri*, a gamma Proteobacterium that causes bacterial dysentery. *Shigella* are closely related to *Escherichia coli* and if the species were to be defined anew, they would be classified as members the same species. *S. flexneri* contains a virulence plasmid whose Genome Atlas is represented in the figure. The most striking feature is the large number of repeat sequences. These are complete and partial copies of a wide variety of insertion sequences, ISs for short. *S. flexneri* seems to be collecting ISs; as many as 314 different ISs were detected in the genome of this strain. These ISs are repeated all over the chromosome and plasmid, with many incomplete copies present as well.

The propagation of these ISs is the result of evolution (an IS is a typical example of ‘selfish DNA’). Incomplete repeat copies were most probably still complete when they were inserted, to become degraded (and fixed) with time. In addition to

---

3 As we saw in Chapter 12, enteroinvasive *E. coli* strains (also causing diarrhea) share most biochemical and virulence factors with *Shigella*. The sequence divergence of *Shigella* and *E. coli* is about 1.5%, less than the variation within the *Escherichia coli* species. The distinction of the *Shigella* genus is purely historic, at that time introduced to differentiate the organisms causing such severe disease from less severe *E. coli* infections.
the presence of ISs as direct evidence of evolutionary processes, they also can drive evolution. As we have discussed in Chapter 8, repeat sequences can form structures that can induce translocations, deletions and transversions. Since the \textit{S. flexneri} chromosome is so closely related to that of \textit{E. coli}, it is interesting to compare the chromosome of \textit{S. flexneri} 301 with that of \textit{E. coli} K-12 (Jin \textit{et al.}, 2002). Figure 14.3 shows this comparison, produced with the Artemis software. There are many segments of the chromosomes that are translocated or transversed, and nearly each of these segments is bordered by a complete or incomplete IS on the \textit{S. flexneri} chromosome. This illustrates the genetic changes that have occurred in this organism, driven by the presence of ISs, and such changes again drive evolution as they create diversity.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{chromosome_alignment.png}
\caption{Chromosome alignment of \textit{S. flexneri} strain 301 (top), \textit{E. coli} strain MG1665 (middle) and \textit{E. coli} strain O157:H7 Sakai (bottom).}
\end{figure}

To see evidence of selection in this genome, we return to the plasmid shown in Figure 14.2. The striking area where ISs are completely absent, between 95 and 130 kb, happens to encode a type III secretion system (T3SS) and the effectors it secretes. \textit{S. flexneri} lives most of its time inside the cells of its involuntary host, and while it is intracellular it needs a T3SS to ‘inject’ effector proteins into the cy-
toplasm of the host cells, where they wreak havoc. Without a functional T3SS, *S. flexneri* would lose its pathogenicity. Thus, if an IS would insert itself within this region of the plasmid, the bacteria carrying that unfortunate plasmid are no longer fit to survive in their niche, and they will not be fit enough to be maintained in the population. We can be certain that ISs have inserted themselves in the T3SS locus (and they may still do so), but such cells will not be detectable in the total population. The plasmid sequence bears witness of the selection pressures that work on *S. flexneri*.

**Genome islands**

We will one more time return to the plasmid of *S. flexneri*, in Figure 14.2. The innermost lane shows the AT content of this plasmid, and as can be seen the T3SS locus is locally much more AT-rich than the rest of the plasmid. This is an important characteristic for a DNA segment that was likely acquired by horizontal gene transfer; in this case it increases the virulence potential of the recipient organism. Such segments became known as *Pathogenicity Islands* (PAIs for short). The term PAI was introduced for uropathogenic *E. coli* (Blum *et al.*, 1994) to describe an observed cluster of virulence-associated genes that could be spontaneously excised from the genome (it was called ‘instable’), dramatically reducing the pathogen’s virulence potential. A typical PAI will contain virulence genes, have direct repeats at both flanks that serve as mobility elements, and a transposase or integrase responsible for excision (or such a gene is found in its vicinity). A PAI is typically inserted in the chromosome in a tRNA, but as in the case of *S. flexneri*, it can also be present on a plasmid. As more PAIs were discovered in other organisms, a bigger and more general picture emerged. Not all PAIs contain a gene locus for a type III secretion system, though all T3SS loci known today are part of a PAI. T3SS-PAIs have been described for a number of Proteobacterial enteric and plant pathogens, notably *Pseudomonas syringae*. In one occasion a T3SS PAI is found in an apathogenic organism: *Sodalis glossinidius*, a symbiont of the tsetse fly. The similarities between various PAIs strengthened the idea that these islands have been spread through bacterial populations by horizontal gene transfer.

The observed differences between PAIs soon loosened the original definition, so that integration in a tRNA, or presence of active mobility elements, are no longer necessary to define a PAI. Any clustering of virulence-associated genes that have an aberrant base composition can be called a PAI (though remnants of the mobility repeats should ideally still be visible). The definition was further broadened to GEnome Islands (GEIs) in general, covering any a stretch of DNA with a base composition different from the total genome (allowing for the local variation we’ve discussed in Chapter 7), bearing a cluster of functionally related genes, flanked by (imperfect) direct repeats, with an integrase or transposase close
to it. Thus, there are antibiotic resistance islands (frequently residing on plasmids), colonization islands, symbiosis islands, metabolic islands, etc. Genome islands can be regarded as variable ‘genetic building blocks’ of genomes. By acquisition of a GEI, a whole new phenotypic property can be obtained, as all necessary genes are acquired in one instance. Viewed this way, a genome can be considered as built up of an essential core, to which variable regions are added that form a flexible gene pool. How to identify the genes that build the core has been described in Chapter 12, but this ‘core genome’ should not be seen as a fixed skeleton to which any variable genes are added in an organized way: the gene order of core genomes may not remain constant during the evolutionary storms that populations encounter.

Considering that genomes are in constant flux, it can be easily envisaged how GEIs evolve. When a number of different genes are all needed for a particular phenotypic property, and when genes are reshuffled by genomic rearrangements at a low but steady rate, sooner or later clusters of functionally related genes will emerge as ‘lumps’ in a genome, and once all these genes cluster together there is a huge advantage in that now a whole set can travel together, making their spread much easier. This principle can be observed with antibiotic resistance genes in particular, because they are under a strong selection pressure, and are frequently found on mobile elements, which speeds up evolution. Antibiotic resistance genes tend to cluster in blocks that then can be transferred in one step to a new recipient strain. Selection for one antibiotic will give resistance to a number of other compounds for free, as a result of co-selection.

**Example of genome islands in Burkholderia pseudomallei**

We have already discussed the comparison of several different Burkholderia genomes in chapter 12, where figure 12.8 identified a number of GEIs, as marked by arrows, on a Genome Atlas. By definition, one would expect GEIs to be present in only a subset of strains of a given species. Figure 12.9 indeed identified these as large gaps on a BLAST Atlas with *B. pseudomallei* as the reference genome. However, even a single genome can reveal GEIs.

Figure 14.4 shows the Genome Atlas of four different sequenced *B. pseudomallei* strains, and one can readily see the GEIs lighting up in the three structural parameter lanes of the atlases. The K96243 strain that was used as a reference genome in Figure 12.9 is presented in the upper left panel of Figure 14.4. Note the two large GEIs indicated by arrows. It is easy to see that these darkly colored regions in the structural lanes correspond with GEIs, by simple comparison with the two added BLAST lanes in the figure. The outermost lane represents the 1339 proteins found to be conserved across all *Burkholderia* species (the genus
core genome, identified in Chapter 12), whilst the penultimate red circle reports the BLAST hits with *B. pseudomallei* strain DM98, which has the largest number of proteins of all the sequenced *B. pseudomallei* genomes currently in our database\(^4\). To the right is the Genome Atlas shown of strain 1710b, which also has two GEIs but in different positions, again indicated by arrows. Note also that for this genome, the replication origin is a bit off-centered. The other two strains (1106a, lower left, and strain 668, lower right) both have multiple, smaller GEIs, on the left side of the chromosome in the figure.

![Genome Atlases](image)

**Figure 14.4.** Genome Atlases of four *B. pseudomallei* strains, where genome islands are identified by arrows. The outer two lanes are BLAST Atlas lanes, as explained in the text.

\(^4\) *B. pseudomallei* strain DM98 contains 8559 protein gene families, compared to 5316 for *B. pseudomallei* strain 1106a; these are preliminary values as some of these genomes are not (yet) sequenced to one contiguous piece. The average for *B. pseudomallei* is about 6900 proteins per genome, with a standard deviation of 1300 proteins.
Evolution on a chip

Another of the common criticisms leveled against evolution by some creationists is that cells are not able to generate any proteins with new function, and that all mutations are deleterious. If this argument were true, all living organisms would be perfect as they are, since any genetic change (any mutation) would have negative effects. Indeed, living organisms are quite well adapted to the environment where they live in, but if the environment changes, they’ll have to change as well, as then they are no longer so perfect. Since environments are never constant (think of changes in temperature, moisture, and nutrients, to name a few of the changing conditions bacteria will encounter), a ‘perfect for all’ set of genes doesn’t exist. And it doesn’t have to, as we have already seen how genes can change over time. As a result, the population may contain some less-than-perfect cells for a given set of conditions, and as long as the conditions remain constant these will be selected out (removed from the population over time). In case they don’t do any harm, they won’t be selected out but will remain in the population. When conditions change, however, selection will let those organisms that best fit the new requirements survive and multiply.

We can observe this process in real time. There is an entire area of research, known as directed evolution, which uses on a daily basis the enormous power of evolution to create new proteins which are vastly more optimized for a given selective criterion (Parales and Ditty, 2005). For example, suppose one wanted to use an enzyme to manufacture some product at room temperature (~20°C), starting from an E. coli enzyme, which works optimally at 37°C, but gives low yields at room temperature. Using directed evolution, many different versions of the protein are produced by introducing mutations in the cellular DNA, and the one that gives optimal performance is selected. This method can result in selection of proteins that are more than a million-fold more efficient than uncatalysed reactions (Seeleg and Szostak, 2007). The E. coli cells that live in nature don’t produce these variants, as they don’t need this enzyme at 20°C with optimal efficiency: they need it at 37°C.

Recently, advances in microfluidics have been applied to make an ‘evolution chip’, which contains billions of RNA molecules that can evolve to different phenotypes (and genotypes) depending on the selection conditions chosen (Paegel and Joyce, 2008). This allows observing evolution in real time, driving the process in particular directions by applying desired selection pressures. Thus it is possible to design novel RNA genes for new functions, using directed evolution on a chip. In addition, novel functions for proteins have been created in the laboratory, using directed evolution (Rarales et al., 2005; Yuan et al., 2005).

\textsuperscript{5} See for example “Genetic Entropy & the Mystery of the Genome” by John C. Sanford.
Species and speciation: *Vibrio cholerae*

We will use the genus *Vibrio* as an example of a marine species to produce a BLAST Atlas comparing genomes from a single species (*V. cholerae*), as well as comparing genomes from related species. Figure 14.5 shows the two chromosomes of *V. cholerae* (strain N16961), compared to several other *Vibrio* genomes. Both chromosomes are highly conserved within the *V. cholerae* species (dark read circles), although chromosome II is not that well conserved outside the *V. cholerae* species. Also note that there is a large region in the lower right hand side of chromosome II, known as the ‘superintegron’, which has unusual base composition properties; this region is also poorly conserved, even within *V. cholerae* genomes. The superintegron is a special kind of genome island where multiple toxin-antitoxin genes and the typical VCR (for *V. cholerae* repeats) are found. In *V. parahaemolyticus* the superintegron is present on the large chromosome, which would not show up in our BLAST Atlas as these are chromosome-specific.

In Figure 14.5, one can see several ‘gaps’ along the chromosomes, which contain regions that are conserved within *V. cholerae*, but are missing in other *Vibrio* genomes. Some of these gaps are labeled around chromosome I in the figure; notice that some of these gaps (e.g., gaps 5 and 6) are from regions which are very strongly conserved (dark red bands) in *V. cholerae* chromosomes, but completely missing outside the *V. cholerae* species. Thus, there are *V. cholerae*-specific regions; what could be the function of these genes? Many of the genes in gap 5 code for surface proteins. This is not too surprising, since surface proteins evolve at a higher rate, and in general are more variable, due to their strong effect on selective processes (think of immune host responses reacting to surface antigens, for instance, or surface adhe- rance factors). Remember that *V. cholerae* spends most of its time in the ocean where it forms biofilms on various surfaces (including on zooplankton, crustaceans, insects and plants). Surface proteins are very important in the lifestyle of this organism.

However, of more interest are the genes in gap 6; this region contains genes encoding regulatory proteins, including a set of two different 2-component signal transduction systems, with two histidine kinases and a set of response regulators. One could easily imagine how these environmental sensors and regulatory responses could (partly) be responsible for niche adaptation of *V. cholerae* to a particular environment. Thus, acquisition of a small set of genes can in a sense help a bacterium to live in a new place, and perhaps eventually become a new species. The example may represent a transition state to eventually produce a new bacterial species.

Figure 14.5 (opposite). BLAST Atlases of the two *Vibrio* chromosomes, compared to reference genome *V. cholerae* strain N16961. The two lanes following the UniProt BLAST scores are measures of gene synteny (a parameter indicating how well gene order is conserved). Gaps of interest in chromosome I, and the large superintegron region in chromosome II, are indicated.
Can we predict evolution? *Escherichia coli* genome reduction

There are trends in science, just like there are trends in design, tourism, arts, or any other human activity. One of the current fashions in science is ‘synthetic biology’. This describes attempts to build a genome from scratch or, before this ultimate goal is reached, to peel down existing genomes to the bare minimum. For instance, Pósfai et al. (2006) used synthetic biology to reduce the *E. coli* K-12 genome by about 10%. But since evolution has had ample chance to work under the right conditions (in this case, selection for a smaller genome) why not look for the evidence in nature?

![Figure 14.6. Size of sequenced genomes, number of genes and AT content for a selection of Enterobacteriaceae members](image)

Figure 14.6 displays a number of sequenced genomes from the Enterobacteriaceae (a subgroup within the γ-Proteobacteria) to which *E. coli* belongs, sorted for the number of coded genes. Their genome size is also plotted. The largest *E. coli* genome (that of strain CFT073) has 5379 predicted proteins. Currently the smallest *E. coli* genome has 4331 proteins. There are two sequenced *Salmonella* genomes with even fewer proteins, and *Yersinia* generally have even a smaller number of proteins. *Sodalis glossinidius* has an interesting genome because it is roughly the same size as *E. coli* and *Salmonella* (about 4 Mbp), but only codes for 2400 genes; this is a genome caught in the act of reducing its size (Toh et al., 2006).
From there it is a fairly steep drop to the genomes of endosymbionts, most of which have around 600 or fewer genes, and also have a considerably higher AT content. The smallest genome included, that of *Buchnera aphidicola*, has as few as 357 protein-coding genes. From a taxonomic point of view, and by 16S rRNA gene phylogeny, *B. aphidicola* is a close relative of *E. coli* even though it has less than 7% of the number of genes found in the largest *E. coli* genome.

Thus, even just looking at *E. coli* and its close relatives, it is clear that the diversity in number of genes can be very large. One might ask whether it is possible to model genome reduction. That is, could one model evolution on a computer, starting with an existing genome (say that of *E. coli* K-12, the common, well-characterized laboratory strain), and predict which genes are essential for life in a reduced genome? Using metabolic networks and the known selective conditions in two different environments, researchers could predict the likely final set of genes in *Wigglesworthia* and *Buchnera* genomes, with more than 80% accuracy (Pal *et al.*, 2006). With these reduced genomes containing 10% or less of the genes present in a full-fledged *E. coli* genome, the trick is to accurately predict which 90% will be not essential under specified growth conditions.

**Concluding remarks**

Evolution means change, and bacterial genomes are certainly capable of changing with time. Some genomes are getting larger, through the addition of phage insertions, IS elements, plasmids, and other mobile elements. As a result of evolutions, genomes can increase as well as decrease in size, as in the case of reduced endosymbiont genomes. Other genomes do not vary significantly in size, but nonetheless contain an enormous amount of diversity. Evolution and speciation are continuous processes that leave their marks in bacterial genomes all the time.

**References**


Index

16S rRNA, 63, 100, 151, 163
metagenomics, 231
structure, 153
ABC transport system, 182
accession number, 36, 59, 60, 61, 63, 73,
81, 82, 93, 94
Acidobacteria, 115
Acinetobacter baumannii, 28
Actinobacteria, 115, 138
adaptation, 248
Adenine, 6
A-DNA, 124, 128
A-DNA Atlas, 127, 128
Aeropyrum pernix, 31
Agrobacterium rhizogenes, 144
Agrobacterium tumefaciens, 126, 127
algorithm, 25, 66, 83
alignment
multiple alignment, 25
pairwise alignment, 95, 106
alpha helix, 14, 181, 191, 192
alphabet
DNA alphabet, 26
protein alphabet, 19
Anaeromyxobacter, 154
annotation
over-annotation, 31
strand direction, 100, 200
under-annotation, 31
annotation quality, 97
anticodon, 9, 157
antigen prediction, 184
Aquificae, 114
archaea, 2, 10, 48, 109, 154
Artemis, 96, 106, 251
AT content, 40, 49, 76, 77, 78, 107, 111,
112, 113, 115, 116, 120, 122, 179
bias, 121
deviation, 121
distribution of, 122
of intergenic regions, 123
of metagenomic DNA, 234
AT skew, 39, 40, 42, 116, 128, 146
Atlas
A-DNA Atlas, 127
Base Atlas, 39, 116
Blast Atlas, 206
Chromatin Atlas, 169
Expression Atlas, 167
Genome Atlas, 41, 146
Metagenome Atlas, 237
Repeat Atlas, 140
Structure Atlas, 123, 126, 143
Z-DNA Atlas, 129
ATP-binding cassette, 182
Bacillus anthracis, 220, 224
Bacillus cereus, 151, 195, 220
Bacillus subtilis, 44, 158, 221
Bacillus thuringiensis, 221
Bacillus weihenstephanensis, 192, 193,
194, 195
bacterial species concept, 219
bacteriophage φX174, 36
Bacteroides, 113, 138, 184
Bacteroides fragilis, 110
Base Atlas, 39, 40, 42, 78, 79, 116, 117,
118
absolute Base Atlas, 39, 41
relative Base Atlas, 40
base composition, 114
base composition bias, 133
base pairs, 7, 8, 12
base skew, 118
Bdellovibrio bacteriovorus, 158, 159
B-DNA, 124
bendability, 143
beta sheet, 191, 192, 193, 194
Bioclipe, 90
Bioconductor, 89
biofilm, 230, 256
BioMake, 90
BioMed Central, 66
BioPerl, 89
BioPython, 89
BLAST, 22, 31, 55, 152, 188, 200
bit score, 22
E-value, 22, 23, 24, 203
score value, 203
BLAST Atlas, 202, 206, 207, 208, 225,
226, 236
BLAST Matrix, 202, 203, 204, 205, 206
BLASTN, 22
BLASTP, 22
BLASTX, 22, 24
Borrelia burgdorferi, 49
box-and-whiskers plot, 92, 101, 106, 114,
120, 121, 156
browser, 74
Buchnera, 114
Buchnera aphidicola, 158, 159, 259
Burkholderia, 115, 222, 223, 253
Burkholderia cenocepacia, 122
Burkholderia mallei, 224
Burkholderia pseudomallei, 223, 253
Burkholderia xenovorans, 94, 222
C and C++, 85
C#, 85
Campylobacter jejuni, 23, 99, 100
Carsonella ruddii, 47, 95
Caulobacter crescentus, 162
cellular localization, 179
chaperone, 14, 180, 181, 192
chimera, 24
chimeric sequence, 23, 30
Chlamydia muridarum, 128
Chlamydiaceae, 92, 114, 115, 138
Chloroflexi, 114
Chromatin Atlas, 169, 170
chromatin silencing, 170, 174
chromosome, 4
chromosome numbering, 117
cis-acting element, 174
Cloacamonas acidaminovorans, 237
cloning insert, 44
Clostridia, 237
Clostridium, 115
Clostridium butylicum, 236
Clostridium tetani, 116, 117, 118, 119
CLUSTAL alignment, 28, 154
CLUSTALW, 25
Cluster of Orthologous Genes, 190
coding density, 97, 98, 106
coding strand, 8, 12
codon, 9, 10, 77, 150
codon usage, 77, 133, 158, 160, 179, 234
codon-anticodon recognition, 10
COG, 190
comparative genomics, 47, 212
compiler, 84, 85
complementary strand, 12, 19, 135, 136, 137, 200
computing time, 204, 215
contamination database, 30
metagenomic application, 242
sequence, 100
contig, 59, 197, 198, 199
Copenhagen Models, 193
core genome, 212, 213, 214, 216, 219, 220, 221
creationism, 246, 255

Cruiform, 142
CSV file, 73
curved DNA, 121, 124, 125, 127, 144, 174
Cyanidioschyzon merolae, 110
Cyanobacteria, 241
Cytosine, 6
data
data extraction, 72
extraction, 72
metadata, 68
primary data, 68
data visualization, 76
data base, 52, 53
non-redundant, 23, 59
data base contamination, 30
Database Management System, 74
DBMS, 74
DDBJ, 56
degradation tag, 161, 162
Deinococcus / Thermus, 115
Deinococcus radiodurans, 133
deletion, 19, 225, 226
deoxyribose, 6
destabilizing energy, 177
Desulfotalea psychrophila, 118, 119, 145
dideoxy sequencing, 36
dideoxynucleotide, 36
dinucleotide, 132
direct repeat, 134, 141
distribution plot, 98, 106, 112, 113, 130
diversity, 246
DNA
A-DNA, 124
alphabet of, 26
B-DNA, 124
bendability, 143
curved DNA, 124
double helix, 7
double-strand, 12
melted DNA, 124
nucleotides, 6
strand direction, 12
structure, 123, 142
supercoiled, 124
Z-DNA, 124
DNA fingerprint, 110
DNA melting, 8
DNA nucleotides, 5, 6
DNA polymerase, 13, 36, 61
DNA repeat, 134
dNA transfer, 247
double strand DNA, 7, 12
downstream, 10
Drosophila, 4, 109, 242
dsDNA, 12
E. coli, 101, 111, 176, 177, 202, 203
E. coli K-12, 167, 169, 206
E. coli O157:H7, 42, 204
EC number, 61
effector, 181, 182
EMBL, 56
Enterobacter sakazakii, 205
environmental change, 255
epitope, 185
prediction, 184
eubacteria, 10, 154, 161
Eucarya, 2
eukaryotes, 3, 5, 10, 61, 109
eukaryotic genome, 110
E-value, 22
everted repeat, 135, 136, 139
evolution, 246
directed, 255
prediction of, 258
time scale, 249
execution pipeline, 75, 76, 90
ExPASy, 65
expectation value
see E-value, 22
expression
gene, 166
protein, 13, 166
Expression Atlas, 167
extremophile, 145
FASTA, 25
FASTA file format, 36, 61, 62, 72
file format, 60, 73
FASTA, 36, 61, 62, 72
GenBank, 61, 62, 72
plain text, 72
Firmicutes, 114, 120, 132, 138
FIS, 170
binding site, 169, 174
Fortran, 85
fragile program, 82
Frankia, 115
Fusobacteria, 114
gap cost, 19
GC skew, 39, 40, 116, 117, 119, 120, 140, 225
GEI, 252
GenBank, 22, 52, 54, 56, 57
GenBank file format, 61, 62, 72
gene
clustering, 253
content, 97, 213, 214, 224
duplication, 137, 189
location, 61, 169, 179
orientation, 117, 118
gene acquisition, 248
gene expression, 125, 150, 166, 169
gene family, 100, 213, 215
gene finding, 31, 32, 196, 197, 199
gene synteny, 256
 genetic code, 9, 150
redundance, 10, 150
genome
reduction, 248
genome alignment, 90, 106
genome annotation, 30, 32, 97, 108, 188, 196
Genome Atlas, 41, 42, 43, 44, 46, 47, 123, 146, 147, 156, 177, 225, 253
genome island, 157, 182, 225, 252
genome length, 108
Genome Project 1D, 59
genome reduction, 97, 258
genome size, 92, 106, 258
Geobacillus kaustophilus, 241
global direct repeat, 135
global inverted repeat, 135
global repeat, 136, 140
GOLD, 58
Guanine, 6
Haemophilus influenzae, 45, 46, 47, 58
hairpin, 142, 157
Helicobacter pylori, 30, 48, 57, 212
Helicobacter modestaludum, 237
helix breaker, 192
histone-like protein, 25, 127, 169, 170
binding site, 169, 174
H-NS, 25
homolog, 189, 201
HTML, 81, 86
hydrophobicity, 194
hypothetical protein, 191
identity score, 21
IHF, 15, 26, 29, 169
binding site, 26, 169, 170, 174
imperfect repeat, 136
in silico, 32, 72
indel, 19, 91, 96, 97
initiation of transcription, 171
insertion, 18
insertion sequence, 30, 142, 250
Integration Host Factor, 14, 15, 26
internally transcribed spacer, 154
interpreter, 84
intrinsic DNA curvature, 42, 125, 147
inverted repeat, 42, 135, 141
Java, 85
JavaScript, 85
key
  identification key, 59
  primary key, 57
Kingdom, 2
Klebsiella pneumoniae, 205
Lactobacillus delbrueckii, 159
lagging strand, 13, 40, 107, 116, 117, 118, 119, 120
leading strand, 13, 40, 107, 116, 117, 119, 120, 121, 153
Leptospira interregens, 98
lipoprotein, 181
local direct repeat, 134
local repeat, 137, 138, 141
make, 90
Markov Chain Method, 132, 133
Markup language, 86, 87
markup tag, 81, 86
attribute, 86
melted DNA, 124
membrane topology, 194
messenger RNA, 7
metadata, 68, 69, 70
derived metadata, 69
primitive metadata, 69
metagenome, 230
Metagenome Atlas, 237, 240, 241
metagenomics, 230
  of marine bacteria, 240
  of soil bacteria, 237
  visualization of, 235
Methanobrevibacter, 115
Methanocaldococcus jannischii, 48
Methanococcus, 115
microbial community, 211
minimal gene set, 213
mirror repeat, 135, 136, 139
mitochondrial DNA, 10, 150
mobile element, 136, 142, 147, 157
miRNA, 7, 10, 158
  concentration, 166
  half-life, 173
  stability, 174
multiple alignment, 28
Mycobacteria, 138
Mycobacterium leprae, 97
Mycobacterium tuberculosis, 151
Mycoplasma, 10, 114
Mycoplasma genitalium, 47
Mycoplasma hyopneumoniae, 95, 96
Nanoarchaeota, 114
Nanoarchaeum equitans, 47
NCBI, 53, 54
ncRNA, 11, 174
non-coding RNA, 11, 161, 173
non-synonymous mutation, 201
nucleotides, 5, 6
oligomer
  bias, 119
  strand difference, 119
ontology, 70, 71
open pan-genome, 216
open reading frame, 9
operon, 171
ORF, 9, 31, 99, 199, 200
Ori, 13
origin of replication, 13, 107, 117, 119, 120, 127, 153, 157, 169, 170, 199
ortholog, 137, 189
over-annotation, 31
PAI, 252
pairwise alignment, 22
palindrome, 42, 134, 135, 141
palindromic repeat, 142
pan-genome, 212, 213, 214, 215, 216, 218, 219, 220, 221, 223, 224
paralog, 189, 203, 204
parser, 73
pathogenesis, 221
pathogenicity, 58
pathogenicity island, 252
PDB, 14, 192
Pectobacterium atrosepticum, 205
PEDANT, 65
Pelagibacter ubique, 95
Pelotomaculum thermopropionicum, 140, 146, 237
periplasm, 181
Perl, 85
Photobacterium profundum, 150, 155, 156
Photorhabdus luminescens, 138, 205
PHP, 85
phylogenetic tree, 2, 3, 29, 99, 104, 154, 162, 163, 231
  rooted tree, 29
  unrooted tree, 29
phyllum, 92
  sequenced genomes, 232
PID, 59, 60, 94, 106
pipeline, 75, 76, 90, 199, 200, 213
PISR-PDS, 64
Planctomycetes, 115
PMID, 57
polycistronic messenger, 158, 171
dpolyriboplastic mRNA, 153, 174
position preference, 42, 123, 125, 144, 147
post-translational modification, 15, 180
preprotein, 181
primary data, 68, 69
primary key, 59
primary structure, 191
Prochlorococcus, 115
Prochlorococcus marinus, 161, 162, 240
ProDom, 66
programming, 67
programming language, 83
compiled, 84
interpreted, 84, 85
object oriented, 86
prokaryotes, 2
promoter, 10, 170, 175
consensus sequence, 176
location, 11, 177
structural property, 177
structure, 175
propeller twist, 143
ProSite, 66
protein
alphabet of, 19
category, 190
denaturation, 179
family, 101
functional category, 189
homology, 189
membrane-embedded protein, 194
modification, 179
similarity, 189
structure prediction, 191
water-soluble, 192
protein database, 64
search, 22
protein expression, 13, 166
protein folding, 14, 179
protein function, 189, 200
protein induced deformability, 143
protein length distribution, 95, 106
protein secretion, 15, 180
protein structure, 14, 192
protein-coding genes, 188
protein-protein interaction, 195
Proteobacteria, 92, 115, 138, 154, 221
proteome, 100, 202
Pseudomonas syringae, 252

PubMed, 56
purine, 6
purine stretch, 128, 133, 141
purine/pyrimidine step, 173
pyrimidine, 6
pyrimidine stretch, 128, 141
pyrosequencing, 196
Python, 85
quaternary structure, 191
query sequence, 18
R, 85
raster graphic, 80
reading frame, 20, 21
recombination, 202
RefSeq, 59
regulation
of gene expression, 173
of transcription, 11, 166, 168, 169
of translation, 178
regulatory protein, 11
relational database, 74
relative oligonucleotide frequency, 133
relaxed DNA, 124
release factor, 161
repeat
global direct, 135
mirror, 135
repeat, 134
direct, 134
everted, 135, 136, 139
frequency, 138
global, 136
global direct, 139
global inverted, 135, 139
imperfect, 136
inverted, 135
local, 137
local direct, 134, 139
local everted, 139
local inverted, 139
mirror, 136, 139
simple, 134
Repeat Atlas, 140
repeat score, 137
repeat spacer, 134, 135
replication, 12, 13
restriction enzyme, 61, 134, 136
ribose, 6
ribose-phosphate backbone, 7
ribosomal RNA, 11, 151
ribosome, 9
large subunit, 152
small subunit, 152
stalled, 161
ribosome binding site, 10
RNA
mRNA, 7, 10
ncRNA, 11
polycistronic, 153, 174
rRNA, 11
tmRNA, 150, 161
tRNA, 9, 10, 11, 150
RNA database, 63
RNA nucleotides, 6
RNA polymerase, 8
rose plot, 158
rRNA, 11
gene count, 151
rRNA gene, 150
rRNA operon, 47, 99, 150, 175, 177
AT content, 156
location, 156
rRNA promoter, 175
Ruby, 85
Salmonella, 25, 176, 177, 204
Salmonella enterica, 221
Salmonella typhimurium, 133
Sanger sequencing method, 36
Sargossa Sea, 233
scalability, 83
scatter plot, 78, 139, 148
Sec-dependent secretion system, 181
secondary structure, 191
secretion
sec-dependent secretion, 181
Tat-dependent secretion, 182
type I secretion, 182
type II secretion, 182
type III secretion, 182, 251
type IV secretion, 182
secretion signal prediction, 183
secretome, 183, 184
Sec-signal peptide, 181
selection, 246
selection pressure, 201
selenocysteine, 150
selfish DNA, 250
sequence, 5
alignment, 18, 19, 20, 21, 26
assembly, 197
conservation, 18, 201
similarity, 21
sequence logo plot, 27, 158, 172, 176
sequence read, 198
sequencing technology, 196, 198
dideoxy sequencing, 36
pyrosequencing, 196
Sanger method, 36
Service Oriented Architecture, 82, 87
Shell scripts, 85
Shewanella, 113
Shigella, 176, 177, 204, 221
Shigella flexneri, 250
Shigella sonnei, 205
shotgun cloning, 44
shotgun DNA sequencing, 44
SIDD value, 178
Sigma 54, 101, 102
binding site, 172
Sigma 70, 101, 102, 171, 175
binding site, 172, 173
sigma factor, 10, 101, 102, 103, 170, 171, 172
alternative sigma factor, 101, 102
binding, 171, 173
binding site, 172
primary sigma factor, 101, 102
stress-response, 180
signal peptide, 181
similarity chain, 30
similarity score, 21
simple repeat, 134, 140, 141
Single Nucleotide Polymorphism, 198
sliding window analysis, 42
SNP, 198
SOAP, 88
Sodalis glossinidius, 97, 252, 258
soil bacteria, 114, 233
Solibacter usitatus, 93
Sorangium cellulosum, 94
Spirochaetes, 114, 115, 138
Spiroplasma, 10
SQL, 74, 80, 86
stacking energy, 125, 147
stalled ribosome, 161
start codon, 9, 10
stop codon, 9, 10
strand
lagging, 116, 117, 118, 119, 120
leading, 13, 40, 107, 116, 117, 119, 120, 121, 153
strand difference plot, 130
strand direction, 12
annotation, 100, 200
Streptococcus, 218
Streptococcus agalactiae, 212
Streptococcus pneumoniae, 219
Streptococcus pyogenes, 219, 224
Streptococcus thermophilus, 219
Streptomyces, 115
Streptomyces coelicolor, 159
Structure Atlas, 123, 126, 127, 143, 144, 145, 177
Structured Query Language, 74
subpopulation, 246
supercoiled DNA, 124
supercoiling, 174
superhelical DNA, 124
superintegron, 256
surface protein, 256
SwissProt, 99
synonymous mutation, 201
synthetic biology, 95, 258
synthetic gene, 179
Tat secretion, 182
Tat secretion signal, 182
TATAAT box, 173
Taverna, 90
taxonomy, 57, 65, 111, 154, 155
telomere, 110
termite Group 1 bacterium, 240
terrestrial bacteria, 237
tertiary structure, 191
tetrancleotide, 133, 134
text file, 72, 73
Thalassiosira pseudonana, 2
Thermoanaerobacter, 100
Thermobispora bisexual, 155
thermophile, 114, 232
Thermotoga, 114, 115
Thermus aquaticus, 61
Thermus thermophilus, 158, 159
Thymine, 6
tRNA, 150, 161
database, 63
structure, 161
trans-acting factor, 174
transcription, 8
global regulation, 168
local regulation, 168
termination, 10
transcription start, 10, 171
transcriptional regulation, 169
transcription, 150, 200
transfer RNA, 9
transfer RNA gene, 157
transfer-messenger RNA, 161
translation, 8, 9
translation efficiency, 160
translocation pore, 181
trans-membrane helix, 194
transposon, 141
trans-translation, 161
TrEMBL, 64, 65
trinucleotide, 132
trivially parallel task, 75
tRNA, 9, 10, 11, 150
database, 63
gene, 11, 63
gene count, 151
structure, 157
tRNA gene, 150
twin Arginine secretion, 182
two-component signal transduction, 173, 256
type I secretion, 182
type II secretion, 182
type III secretion system, 182, 251
type IV secretion system, 182
under-annotation, 31
UniParc, 65
UniProt, 64
UniProtKB, 65
UniRef, 65
untranslated RNA, 150
UP element, 174
upstream, 10, 170
Uracil, 6
vector graphic, 80
Vibrio cholerae, 49, 256
Vibrio parahaemolyticus, 256
violin plot, 184, 186
visualization methods, 77
visualization tool, 130, 148, 163, 186
Web-Service, 74, 88
Wigglesworthia glossinidia, 159, 204
window analysis, 136, 170
Wolbachia, 242
word frequency, 132
WSDL, 89
XML, 87
XML document, 87
Yersinia, 176, 177, 204
Z:DNA, 124
Z:DNA Atlas, 129
zero order Markov method, 133