Bioinformatics Biocomputing and Perl

An Introduction to Bioinformatics Computing Skills and Practice

Michael Moorhouse
Post-Doctoral Worker from Erasmus MC, The Netherlands

Paul Barry
Department of Computing and Networking, Institute of Technology, Carlow, Ireland

John Wiley & Sons, Ltd

Bioinformatics Biocomputing and Perl

Bioinformatics Biocomputing and Perl

An Introduction to Bioinformatics Computing Skills and Practice

Michael Moorhouse
Post-Doctoral Worker from Erasmus MC, The Netherlands

Paul Barry
Department of Computing and Networking, Institute of Technology, Carlow, Ireland

John Wiley & Sons, Ltd

Copyright 2004 John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wileyeurope.com or www.wiley.com

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except under the terms of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London W1T 4LP, UK, without the permission in writing of the Publisher. Requests to the Publisher should be addressed to the Permissions Department, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, or emailed to permreq@wiley.co.uk, or faxed to (+44) 1243 770620.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-470-85331-X

Typeset in 9.5/12.5pt Lucida Bright by Laserwords Private Limited, Chennai, India
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire
This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

For my parents, who taught me the value of knowledge – MJM

For three great kids: Joseph, Aaron and Aideen – PJB

Contents

Preface

Welcome to *Bioinformatics, Biocomputing and Perl*, an introduction and guide to the computing skills and practices collectively known as *Bioinformatics*.

Bioinformatics is the application of computing techniques to the study of biology, and in particular biology research. Although the study of biology is hundreds of years old, the application of computing techniques to biology research is relatively new, with major advances occurring within the last decade. Consequently, the Bioinformatics field is evolving and maturing rapidly, and this has highlighted the need for a good, all-round introductory textbook. We believe that *Bioinformatics, Biocomputing and Perl* meets this need.

What is in this Book?

After two introductory chapters, *Bioinformatics, Biocomputing and Perl* is divided into four main parts:

1. **Working with Perl.**
2. **Working with Data.**
3. **Working with the Web.**
4. **Working with Applications.**

Part I, *Working with Perl*, introduces programming to the student of Bioinformatics. Note that the intention is not to turn Bioinformaticians into software engineers. Rather, the emphasis is on providing Bioinformaticians with programming skills sufficient to enable them to produce bespoke programs when required in the course of their research.

The programming language of choice among Bioinformaticians, Perl, is used throughout Part I. Perl is popular because of its combination of excellent file-handling capabilities, native support for POSIX regular expressions and powerful

scripting capabilities. If that sounds like *techno babble*, do not worry; the importance of these programming language features is explained in a less technical way later. Fortunately, Perl is not particularly difficult to learn. For instance, by the end of Chapter 3, the reader will know enough Perl to be able to produce simple, but useful, programs. This early material is then developed so that by the end of Part I, readers will be able to confidently create customised and customisable programs to solve diverse Bioinformatics problems.

In Part II, *Working with Data*, the emphasis shifts from creating bespoke Bioinformatics programs to exploring the tools and techniques used to organise, store, retrieve and process data. After explaining how to download datasets from the Internet, the Protein DataBank (PDB) is described in detail. A short chapter follows on the importance of non-redundant datasets, before discussion shifts to cover relational database management systems. How to create and use databases with the popular *MySQL* tool is described. In addition to using standard tools to interact with databases, the use of Perl programs to interrogate databases is also covered.

Part III, *Working with the Web*, covers a collection of web-based technologies that, once mastered, can be used to publish research -- both findings *and* data -- on the Internet. Electronic mechanisms allowing interaction with, and interrogation of, web-based data are explained. Perl again plays an important role in this part of the book, with HTML and CGI also covered.

Part IV, *Working with Applications*, describes a set of standard Bioinformatics tools and applications. Although it is often useful to be able to create a new tool from scratch, it can sometimes be more appropriate to take existing tools and control their execution and interaction. Scripting technologies, of which Perl is only one type, are particularly useful in this area. A discussion of ''The Bioperl Project'', and its importance, completes *Bioinformatics, Biocomputing and Perl*.

Maxims, Commentaries, Exercises and Appendices

All but the first two chapters contain a collection of *maxims*. These are your authors' *snippets of wisdom*. At the end of each chapter, the maxims are repeated in list form. If, having worked through a chapter, the maxims are understood, it is an indication that the associated material has been understood. If, however, a maxim is not understood, it indicates that there is a need to review the material to which the particular maxim relates.

In addition to the maxims, chapters include *technical commentaries*. Unlike maxims, it is not necessary to fully understand the commentaries on first reading. If a technical commentary is not immediately understood, it is possible to safely continue to work through the text without too much difficulty.

The majority of chapters conclude with a set of exercises that are designed to expand upon the material introduced. It is highly recommended that these

exercises are worked through, as it is only through practice and review that Bioinformatics computing skills are developed and honed.

A collection of appendices completes the book, providing information on, among other things, installing Perl on various platforms, the Perl on-line documentation and a list of Perl operators. An annotated list of references and suggestions for further reading are also presented as an appendix.

Who Should Read this Book

This book targets three distinct readerships.

The main target is the student of biology, both under- and post-graduate. *Bioinformatics, Biocomputing and Perl* is designed to be *the* must-have, introductory Bioinformatics textbook. The biology student taking a Bioinformatics module will find this book to be a useful starting point and an essential desktop reference.

Another target is the qualified, professional or academic biologist who needs to understand more about Bioinformatics. The field of Bioinformatics is still relatively new and it is only now appearing as a feature within biology course outlines and syllabi. However, there are many qualified biologists ''in the field'' requiring a good primer. This book is designed to meet that need.

The final target is the computer scientist curious to understand how computing skills might be used within this growing field.

What you Should know Already

It is assumed that some knowledge of computer use has already been acquired, including understanding the concept of a disk-file and knowing how to create one using an editor. On the Linux operating system, popular editors are `vi`, `pico` and `emacs`. On any of the Windows operating systems, `Notepad`, `WordPad` and `Word` are all editors, although the latter is a more sophisticated example. Macintosh users have `SimpleText` and `BBedit`. Any of these will suffice, so long as it allows for the creation and manipulation of plain text files. Later chapters (Parts III and IV) assume a working knowledge of HTML.

Platform Notes

All of the examples in *Bioinformatics, Biocomputing and Perl* are designed to operate on the Linux operating system, in keeping with the current trend within the Bioinformatics community. There is no attempt to explain all that the reader needs to know about Linux, as the emphasis in this book is on explaining how to exploit the growing collection of tools that run *on top of* the Linux operating

system. Two additional appendices provide a list of essential Linux commands and a quick reference to the `vi` text editor, respectively.

Accompanying Web-site

Details of the book's mailing list, its source code, any errata and other related material can be found on the book's web-site, located at:

```
http://glasnost.itcarlow.ie/~biobook/index.html
```

Your Comments are Welcome

The authors welcome all comments about *Bioinformatics, Biocomputing and Perl*. Send an e-mail to either of the following addresses:

```
m.moorhouse@erasmusmc.nl

paul.barry@itcarlow.ie
```

Acknowledgements

Michael thanks his parents for their unwavering support, be it material, practical or emotional. Their endless hours of reading and re-reading the draft chapters and manuscript produced many points of very welcome constructive criticism. Although completing a PhD., moving country and starting a new job while writing a book is not something he'd recommend, Michael thanks those around him for helping when they could and for understanding why he was so busy. Also, thanks to all in the new Department of Bioinformatics, Erasmus MC, the Netherlands, who have offered their support and understanding.

Paul thanks his father, Jim Barry, for taking the time to proofread the text (multiple times). As with Paul's first book, this one is better for his father's involvement. Thanks go to Karen Mosman (formerly with Wiley's Computing Division) for suggesting Paul when the Biology Division came looking for an author with Perl experience. The Institute of Technology, Carlow, was again supportive of Paul working on a textbook, and thanks are due to Dr Dave Dowling and Joe Kehoe for enthusiastically reviewing some of the early material. Paul's wife, Deirdre, held everything else together while the production of the manuscript consumed more and more of his time, while Joseph, Aaron and Aideen kept reminding Paul that there's more to life than computers and writing.

Both authors thank the team at Wiley. Joan Marsh, this book's publishing editor, arranged for the authors to work together and never once complained when the draft manuscript went from being days late to weeks late to -- eventually -- six

months late! This book's editorial assistant was Layla Paggetti, and both authors thank Layla for her prompt and efficient responses to their many queries. Robert Hambrook acted as production editor. As with Paul's first book, this one has benefited greatly from Robert's management of the production process.

A special word of thanks to those members of the computing and biology communities who produce such wonderfully useful software technologies and tools. There are many such individuals. Specific thanks to Richard Stallman, Linus Torvalds, Larry Wall, Tom Boutell, Andy Lester and Dr Lincoln D. Stein for sharing their software with the world and for providing the authors with technologies to write about. Paul also thanks Bill Joy (for `vi`) and Leslie Lamport (for LaTeX).

1

Setting the Biological Scene

Introducing DNA, RNA, polypeptides, proteins and sequence analysis.

1.1 Introducing Biological Sequence Analysis

Among other things, this book describes a number of techniques used to analyse DNA, RNA and proteins.

To a molecular biologist, DNA is a very physical molecule: a polymer of nucleotides that are collectively called *deoxyribose nucleic acid*. It coils, bends, flexes and interacts with proteins, and is generally interesting. RNA is similar to DNA in structure, but for the fact that RNA contains the sugar *ribose* as opposed to *deoxyribose*. DNA has a hydrogen at the second carbon atom on the ring; RNA has a hydrogen linked through an oxygen atom.

In DNA and RNA, there are four *nucleotide bases*. Three of these bases are the same: *guanine* (G), *adenine* (A) and *cytosine* (C). The fourth base for DNA is *thymine* (T), whereas in RNA, the fourth base lacks a *methyl group* and is called *uracil* (U). Each base has two points at which it can join *covalently* to two other bases on either end, forming a *linear chain of monomers*. These chains can be quite long, with many millions of bases common in most organisms.

Bioinformatics, Biocomputing and Perl. Michael Moorhouse and Paul Barry
© 2004 John Wiley & Sons, Ltd ISBN 0-470-85331-X

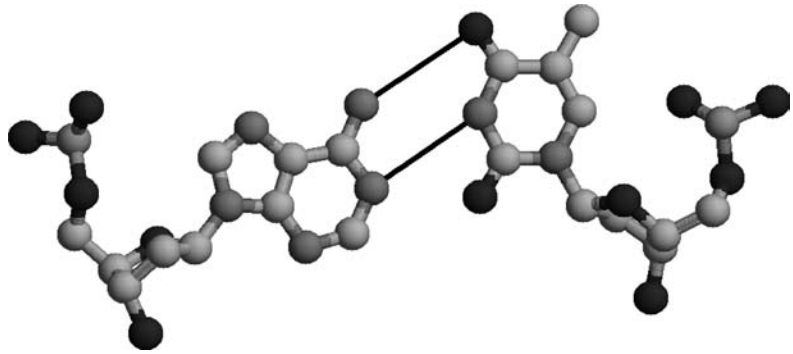

Figure 1.1 Adenine (A) and thymine (T) nucleotide bases (where the thin black lines indicate the three hydrogen bonds between the two bases).

Another interesting feature of nucleotide bases is that the four bases *hydrogen-bond* together in two exclusive pairs because of the position of the charged atoms along their *edges*, as shown in Figure 1.1 on page 2 and Figure 1.2 on page 3[1]. Three of these bonds form between C and G, whereas two form between A and T (or A and U in RNA).

These bonds, while considerably weaker than the covalent bonds between atoms, are enough to stabilise structures such as the famous *double helix*, in which the bases line up nearly perpendicular to the axis of the helix, as shown in Figure 1.3 on page 4. There are several important consequences of the *double helix*:

- Where there is a G in one chain, there is a C in the corresponding location in the other, and the two chains are said to be *complementary* to each other. The chains are often referred to as *strands*.
- This complementarity means that there is 50% redundancy in the information stored in both chains; consequently, only one chain is needed to store all the information for both (as one can be deduced from the other)[2].
- Because of the structure of the nucleotide bases, DNA molecules have *direction*. This is a subtle, but important, point. The phosphate backbones *attach* to the sugar rings at different locations: the 3' and 5' hydroxyl groups.

[1] These diagrams were produced with *Open Rasmol* on the basis of protein structure `1D66`.

[2] Of course, in an evolutionary world, where DNA can be damaged, keeping a spare copy is an evolutionary advantage as an organism can often reconstruct the damaged regions from any intact parts.

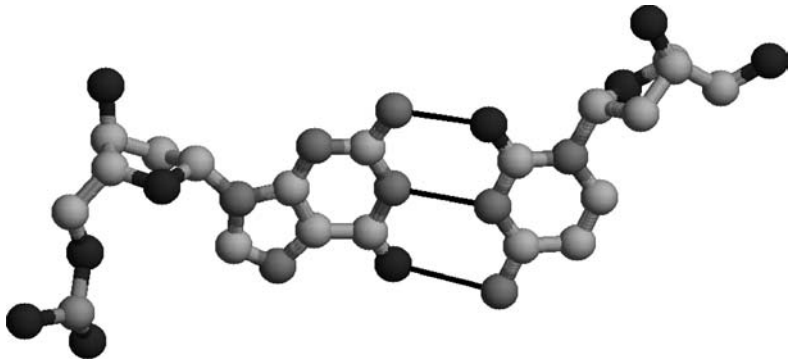

Figure 1.2 Guanine (G) and cytosine (C) nucleotide bases (where the thin black lines indicate the three hydrogen bonds between the two bases).

> When DNA is run in opposite directions, one end of the helix is the 3' end of one chain and the 5' end of the other. When the order of the nucleotide bases is written down, it is conventional to start at the nucleotides at the 5' (the 'left-most' nucleotide) end of the DNA molecule and work towards the 3' end at the right (the 'right-most' base). The importance of this directional feature will become clear later in this chapter, when *open reading frames* are described.

In general, RNA copies of DNA are made by a process known as *transcription*. For most purposes, RNA can be regarded as a *working copy* of the DNA *master template*. There is usually one or a very small number of examples of DNA in the cell, whereas there are multiple copies of the transcribed RNA.

A common term related to the number of nucleotide bases in a particular sequence is a reference to *base pairs*[3], for example "400 base pairs". This term is a generic term that can literally mean "400 paired bases". More often, though, it is used to acknowledge that while there are 400 nucleotides in a particular sequence being actively considered, there are another 400 nucleotides on the complementary strand running in the other direction. In this context, the use of base pairs is a tacit acknowledgement of their existence that may be of great importance, as the *feature* under investigation may be on *the other strand*. In nearly all cases, both strands should be considered.

There are many interesting features of DNA. As this discussion is an overview, a description of some of these features (such as *promoters*, *splice sites*, *intron/exon boundaries* and *genes*) is deferred until later chapters.

[3] Or "bp", for short.

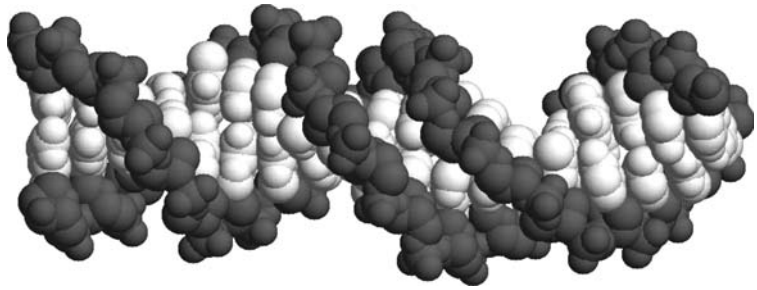

Figure 1.3 The DNA "double helix" (where the backbones, in black, run in opposite directions).

1.2 Protein and Polypeptides

DNA is the *nobility* of the cellular world. Proteins are the *worker-serfs*.

To a biochemist, proteins are the *functioning units of cellular life*. Proteins do physically useful things such as catalysing reactions, processing energy rich molecules, pumping other molecules across cellular barriers and forming connective and motility structures. Proteins do just about anything else in the cell that can be considered "real work".

In molecular terms, proteins are chains technically termed *polypeptides* and formed from 20 different types of *amino acids*. These may be modified in different ways to alter their properties, the structure that is formed and the final function of the molecule. For example, certain amino acids can be *glycosylated*[4], which can be used as recognition *tags*, while other proteins associate with small molecules called *ligands* that have special properties useful in the catalysis of reactions.

The structure of a protein is generally more variable than DNA. It is at the level of proteins that the variety of the information contained in the order of DNA bases is used. The result is that the amino acid chain produced *fold* into structures that are closely linked to that particular protein's functional role within the cell (and these can vary enormously). This folding has another important consequence in that parts of a protein (i.e. its amino acids) can be physically close together in space, but distant in terms of their location in the sequence of the amino acids.

Consider, as an example, the well-studied *catalytic triad of chymotrypsin*. The critical parts of the protein for its function (which is to degrade other proteins) are the amino acids *asparate* at position 102 in the polypeptide chain, *histidine* at 57 and *serine* at 195. The triad is presented in Figure 1.4 on page 5. The right-hand side of the image shows the catalytic site in close-up, with the three critical amino acids located closely in physical space, but distant in sequence. The inset (left-hand image) shows the general structure of the protein demonstrating how the complex folding of the chain brings these residues together.

[4] Have sugars added.

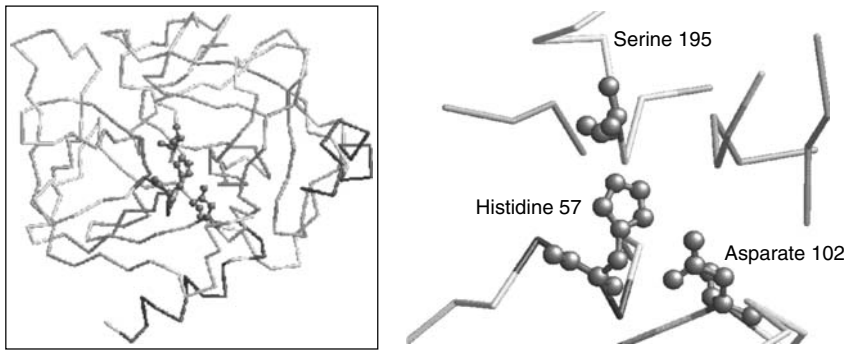

Figure 1.4 The catalytic triad of chymotrypsin (PDB ID: 1AFQ).

1.3 Generalised Models and their Use

The relationships between DNA, RNA, protein, structure and function follow a generalised model. Unfortunately, like most generalisations, it is oversimplistic for many situations. If this is the case, why use it? There are two good reasons:

1. The model is a "good enough" description of what happens *most of the time*. Certainly, there are important exceptions. There are non-standard amino acids included in proteins via some other mechanism (which are ignored in this book). Possibilities such as the section of DNA coding for single protein being *discontinuous* are additional complexities that are considered later. However, overall, the model is a valuable approximation to reality that has useful predictive power when working with new systems.
2. The model is a "lie-to-children"[5]: it allows the basic features to be understood without confusing things by considering exceptions and enhancements. Once such a simple system is understood, it can be extended to cover more complex aspects and specific examples. In short, a start has to be made somewhere, and the generalised model is as good a place to start as anywhere.

Before considering the mechanisms by which information is conserved and converted *along the pathway*, let's consider another important point about the abstract nature of the data to be used.

Bioinformaticians are generally concerned with information at an abstract level: DNA, RNA and amino acid sequences are "just" strings of letters. It is sometimes easy to forget that these are actual representations of molecules that exist in the cellular world and, consequently, must interact with the physical

[5] *Jack Cohen*, *Ian Stewart* and *Terry Pratchett* discuss this concept and some general theories of science in their *Science of the Discworld* books. These are well worth a read if you fancy a laugh while pretending to work.

universe *in general*, let alone existing within a cellular environment. How much a Bioinformatician needs to know about the real-world context of the data being analysed depends on the analysis that is performed[6]. In some cases, quite superficial knowledge suffices, while others require a deeper understanding of the fundamental physical and biological processes at work.

Only through experience can the Bioinformatician hone the skill and professional judgement necessary to decide how much understanding of the underlying biological system is needed for any particular analysis. The idealistic response is "the more the better", which is like all ideals: something to aim at but rarely achieved in practice. Time is often a factor for the Bioinformatician. If too long is spent becoming versed in the biological background, the risk of not completing an analysis within a useful timescale will increase. Conversely, there is also the risk of an analysis being compromised because too little is known about the system under study. This is where the balance between the two extremes comes in. This book attempts to guide the reader in this regard through the examples presented and provide useful pointers beyond. However, in the end, it all comes down to *experience* and *professional judgement*.

1.4 The Central Dogma of Molecular Biology

The DNA to Functional Protein Structure Model discussed above is often referred to as the "Central Dogma of Molecular Biology". It is summarised in a slightly extended form in Figure 1.5 on page 6. The arrows represent *information flow* from that stored in the order of the DNA bases through the folding of the polypeptide chain to a fully functional protein.

1.4.1 Transcription

Transcription is the conversion of information from DNA to RNA, and is straightforward because of the direct correspondence between the four nucleotide bases of DNA and those of RNA.

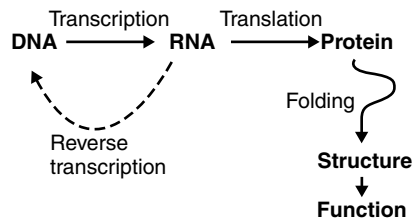

Figure 1.5 The central dogma of molecular biology.

[6] This is so obvious that it is often forgotten.

There is an interesting exception in *RNA Retroviruses*, the most famous example being HIV (the Human Immunodeficiency Virus) that causes AIDS. In retroviruses, RNA is used as the information storage material. This is then copied (badly in the case of HIV) into DNA, which then integrates into the nucleic acid material of the cell under attack. This "trick" allows the virus (and its information) to lie dormant for long periods *in relative safety*, whereas the original RNA material is more likely to be actively degraded by cellular enzymes.

This RNA to DNA conversion ability is also useful for molecular biologists, as DNA can be more easily stored or manipulated using standard techniques. This has important implications, which are discussed later.

1.4.2 Translation

In a protein-coding region of DNA, three successive nucleotide bases, called *triplets* or *codons*, are used to code for each individual amino acid. Three bases are needed because there are 20 amino acids but only four nucleotide bases: with one base there are four possible combinations; with two bases, 16 (4^2); with three, 64 (4^3), which is more than the number of amino acids.

The *RNA transcript* is used by a complex molecular machine called the *ribosome* to translate the order of successive codons into the corresponding order of amino acids. Special *stop codons*, such as UAA, UAG and UGA, induce the *ribosome* to terminate the elongation of the polypeptide chain at a particular point. Similarly, the codon for the amino acid *methionine* (AUG in RNA) is often used as the *start signal* for translation.

The section of DNA between the *start* and *stop* codons is called an *open reading frame*. There is a complication in that the codons found depend on how the sequence of nucleotide bases is divided. This is dependent on where the count starts. There is no biological reason why the first nucleotide base reported in a DNA sequence should be related to the protein coding regions.

A common solution is to calculate the codons produced from all possible open reading frames and select the most plausible on the basis of the results. The *correct* open reading frame for a particular region of DNA is generally that which has the longest distance between any start and stop codons. Though there are exceptions, especially in some viruses and bacteria, each nucleotide is involved in coding for only one amino acid and, hence, only one open reading frame is correct. The incorrect reading frames are generally short and as a consequence, do not resemble recognisable proteins.

With three nucleotide bases in each codon, it is reasonable to assume that there are reading frames starting at the first, second and third nucleotide bases relative to a particular nucleotide. This is due to the fact that all subsequent reading frames are repeated and could start to occur anywhere else in the sequence. Consequently, it is easiest to start at the beginning. It is also important to consider the other DNA chain that *base-pairs* with the one that you have as an example, as this has *another* three reading frames. By convention, the reading on

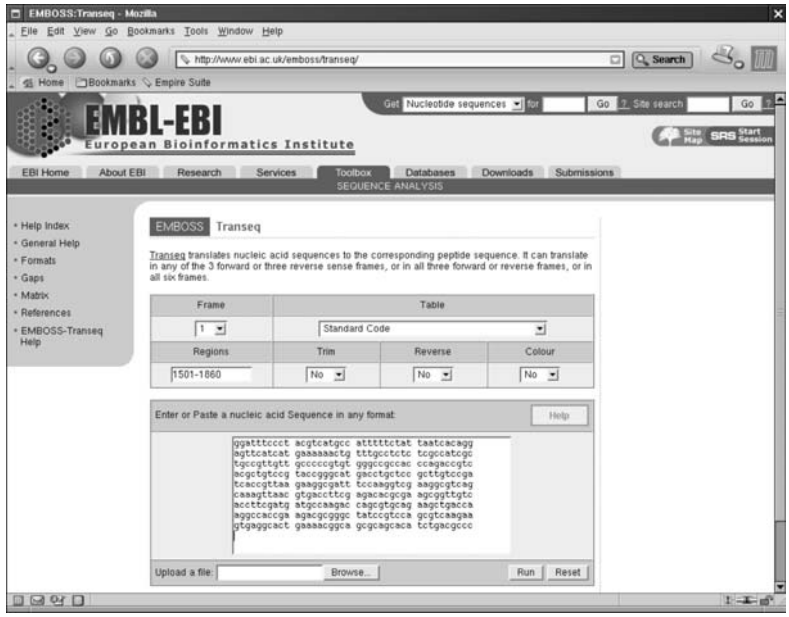

Figure 1.6 The EMBOSS/Transeq page at the EBI.

the sequence under study are referred to as +1, +2 and +3, while those on the complement strand are −1, −2 and −3.

The effects of choosing the correct and incorrect reading frames can be investigated using the *Transeq* tool contained in the *EMBOSS* suite of programs. As these tools are discussed later in this book, a number of the details are glossed over here in favour of illustrating the point at hand. Figure 1.6 on page 8 shows the *Transeq* interface provided by the EBI at the following Internet address:

```
http://www.ebi.ac.uk/emboss/transeq/
```

For this example, consider bases Bases 1501 through 1800 from EMBL entry `M245940`. This sequence is chosen because it contains the MerP protein. These particular bases are easy to extract from a disk-file using any text editor. From the entry, the six lines of DNA bases (near the end of the EMBL data-file) can be copied. The line numbers at the end of each line can be removed and then the resulting data can be pasted into the box on *EMBOSS/Transeq* WWW form (refer to Figure 1.6). Here's what the data looks like *before* the editing takes place:

```
ggatttccct acgtcatgcc atttttctat taatcacagg agttcatcat gaaaaaactg      1560
tttgcctctc tcgccatcgc tgccgttgtt gcccccgtgt gggccgccac ccagaccgtc      1620
```