Chapter 1

Finding Out What Bioinformatics Can Do for You

In This Chapter

- Defining bioinformatics
- Understanding the links between modern biology, genomics, and bioinformatics
- Determining which biological questions bioinformatics can help you answer quickly

> *Organic chemistry is the chemistry of carbon compounds. Biochemistry is the study of carbon compounds that crawl.*
>
> — Mike Adam

It looks like *bio*logists are colonizing the dictionary with all these *bio*-words: we have bio-chemistry, bio-metrics, bio-physics, bio-technology, bio-hazards, and even bio-terrorism. Now what's up with the new entry in the bio-sweepstakes, bio-informatics?

What Is Bioinformatics?

In today's world, computers are as likely to be used by biologists as by any other highly trained professionals — bankers or flight controllers, for example. Many of the tasks performed by such professionals are common to most of us: We all tend to write lots of memos and send lots of e-mails; many of us use spreadsheets, and we all store immense amounts of never-to-be-seen-again data in complicated file systems.

However, besides these general tasks, biologists also use computers to address problems that are very specific to biologists, which are of no interest to bankers or flight controllers. These specialized tasks, taken together, make up the field of *bioinformatics*. More specifically, we can define bioinformatics as the computational branch of molecular biology.

Time for a little bit of history. Before the era of bioinformatics, only two ways of performing biological experiments were available: within a living organism (so-called *in vivo*) or in an artificial environment (so-called *in vitro,* from the Latin *in glass*). Taking the analogy further, we can say that bioinformatics is in fact *in silico* biology, from the silicon chips on which microprocessors are built.

This new way of doing biology has certainly become very trendy, but don't think that "trendy" translates into "lightweight" or "flash-in-the-pan." Bioinformatics goes way beyond trendy — it's at the center of the most recent developments in biology, such as the deciphering of the human genome (another buzzword), "system biology" (trying to look at the global picture), new biotechnologies, new legal and forensic techniques, as well as the personalized medicine of the future.

Because of the centrality of bioinformatics to cutting-edge developments in molecular biology, people from many different fields have been stumbling across the term in a variety of different contexts. If you're a biology, medical, or computer science student, a professional in the pharmaceutical industry, a lawyer or a policeman worrying about DNA testing, a consumer concerned about GMOs (Genetically Modified Organisms), or even a NASDAQ investor interested in start-up companies, you'll already have come across the word *bioinformatics.* If you're good at what you do, you'll want to know what all the fuss is about. This chapter, then, is for you.

Instead of a formal definition that would take hours to cover all the ins and outs of the topic, the best way to get a quick feel for what bioinformatics — or swimming, for that matter — is all about is to jump right into the water; that's what we do next. Go ahead and get your feet wet with some basic molecular biology concepts — and the relevant questions intimately connected with such concepts — that all together define bioinformatics.

Analyzing Protein Sequences

If you eat steak, you're intimately acquainted with proteins. (Your taste buds know them intimately anyway, even if your rational mind was too busy with dinner to master the concept.) For you non-steak lovers out there, you'll be pleased to know that proteins abound in fish and vegetables, too. Moreover, all these proteins are made up of the same basic building blocks, called *amino acids.* Amino acids are already quite complex organic molecules, made of carbon, hydrogen, oxygen, nitrogen, and sulfur atoms. So the overall recipe for a protein (the one your rational mind will appreciate, even if your taste buds won't) is something like $C_{1200}H_{2400}O_{600}N_{300}S_{100}$.

The early days of biochemistry were devoted to finding out a better way to represent proteins — preferably in terms of a formula that would explain their biological (or even nutritional) properties. Biochemists realized over time that proteins were huge molecules (*macromolecules*) made up of large numbers of amino acids (typically from 100 to 500), picked out from a selection of 20 "flavors" with names such as alanine, glycine, tyrosine, glutamine, and so on. Table 1-1 gives you the list of these 20 building blocks, with their full names, three-letter codes, and one-letter codes (the *IUPAC code,* after the *International Union of Pure and Applied Chemistry* committee that designed it).

Table 1-1 The 20 Amino Acids and Their Official Codes

#	*1-Letter Code*	*3-Letter Code*	*Name*
1	A	Ala	Alanine
2	R	Arg	Arginine
3	N	Asn	Asparagine
4	D	Asp	Aspartic acid
5	C	Cys	Cysteine
6	Q	Gln	Glutamine
7	E	Glu	Glutamic acid
8	G	Gly	Glycine
9	H	His	Histidine
10	I	Ile	Isoleucine
11	L	Leu	Leucine
12	K	Lys	Lysine
13	M	Met	Methionine
14	F	Phe	Phenylalanine
15	P	Pro	Proline
16	S	Ser	Serine
17	T	Thr	Threonine
18	W	Trp	Tryptophan
19	Y	Tyr	Tyrosine
20	V	Val	Valine

Biochemists then recognized that a given type of protein (such as insulin or myoglobin) always contains precisely the same number of total amino acids (generically called *residues*) — in the same proportion. Thus, a better formula for a protein looks like this:

insulin = (30 glycines + 44 alanines + 5 tyrosines + 14 glutamines + . . .)

Finally, biochemists discovered that these amino acids are linked together as a chain — and that the true identity of a protein is derived not only from its composition, but also from the precise order of its constituent amino acids. The first amino-acid sequence of a protein — insulin — was determined in 1951. The actual recipe for human insulin, from which all its biological properties derive, is the following chain of 110 residues:

insulin = MALWMRLLPLLALLALWGPDPAAAFVNQHLCGSHLVEALYLVCGERG
FFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQCCTSICSLY
QLENYCN

Now, more than 50 years later, analyzing protein sequences like these remains a central topic of bioinformatics in all laboratories throughout the world. (Check out Chapters 2, 4, and 6 through 11 to quickly figure out how to analyze your protein sequence and become a member of the club!)

A brief history of sequence analysis

Besides earning Alfred Sanger his first Nobel Prize, the sequencing of insulin inaugurated the modern era of molecular and structural biology. Traditionally a *soft science* (that is, more tolerant of fuzzy reasoning and hand-waving ambiguity than chemistry or physics), biology got a taste of its first fundamental dataset: molecular sequences. In the early 1960s, known protein sequences accumulated slowly — perhaps a blessing in disguise, given that the computers capable of analyzing them hadn't been developed! In this pre-computer era (from our present perspective, anyway), sequences were assembled, analyzed, and compared by (manually) writing them on pieces of paper, taping them side by side on laboratory walls, and/or moving them around for optimal alignment (now called *pattern matching*).

As soon as the early computers became available (as big as locomotives and just as fast, and with 8K of RAM!), the first computational biologists started to enter these manual algorithms into the memory banks. This practice was brand new — nobody before them had to manipulate and analyze molecular sequences as *texts*. Most methods had to be invented from scratch, and in the process, a new area of research — the analysis of protein sequences using computers — was generated. This was the genesis of bioinformatics.

Seven additional amino acid codes

When you work with databases or analysis programs, you're likely to have some unusual letters popping up now and then in your protein sequences. These letters are either used to designate exotic amino acids, or are used to denote various levels of ambiguity — that is, a total lack of information — about certain positions in the sequence. We've listed these particular letters in the following table.

Seven Codes for Ambiguity or Exceptional Amino Acids

1-Letter Code	*3-Letter Code*	*Meaning*
B	Asn or Asp	Asparagine or aspartic acid
J	Xle	Isoleucine or leucine
O (letter)	Pyl	Pyrrolysine
U	Sec	Selenocysteine
Z	Gln or Glu	Glutamine or glutamic acid
X	Xaa	Any residue
--	-----	No corresponding residue (gap)

The B and Z codes (which are now becoming obsolete) indicated how hard it was to distinguish between Asp and Asn (or Glu and Gln) in the early days of protein sequence determination. In contrast, the J code shows how difficult it is to distinguish between Ile and Leu using mass spectrometry, the latest sequencing technique. The Pyl and Sec exotic amino acids are specified by the UAG (Pyl) and UGA (Sec) stop codons read in a specific context. The *X* code is still very much used as a placeholder letter when you don't know the amino acid at a given position in the sequence. Alignment programs use "–" to denote positions apparently missing from the sequence.

Reading protein sequences from N to C

The twenty amino-acid molecules found in proteins have different *bodies* (their characteristic residues, listed in Table 1-1) — but all have the same pair of *hooks* — NH_2 and COOH. These groups of atoms are used to form the so-called *peptidic bonds* between the successive residues in the sequence. Figure 1-1 shows free individual amino acids floating about, displaying their hooks for all to see.

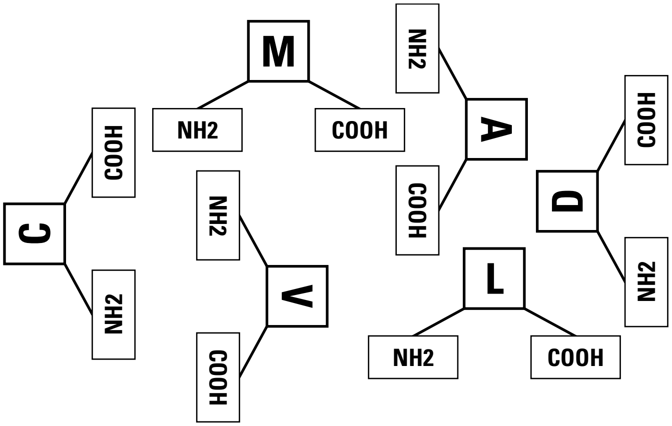


Figure 1-1: Free amino acids floating around.

The protein molecule itself is made when a free NH_2 group links chemically with a COOH group, forming the peptide bond CO-NH. Figure 1-2 shows a schematic picture of the resulting chain.

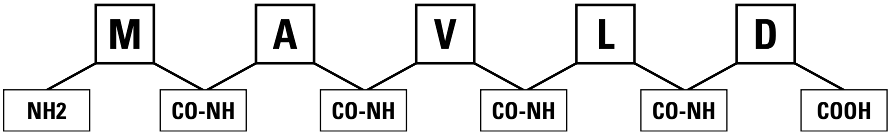
As a result of this chaining process, your protein molecule is going to be left with an unused NH_2 at one end and an unused COOH at the other end. These extremities are called (respectively) the *N-terminus* and *C-terminus* of the protein chain. This is important to know because scientific convention (in books, databases, and so on) defines the *sequence* of a protein — or of a protein fragment — as the succession of its constituent amino acids, listed in order from the N-terminus to the C-terminus. The sequence of our (short!) demo protein is then

```
MAVLD= Met-Ala-Val-Leu-Asp= Methionine-Alanine-
          Valine-Leucine-Aspartic
```

Working with protein 3-D structures

The precise succession of a protein's constituent amino acids is what defines a given protein molecule. This ribbon of amino acids, however, is not what

Figure 1-2: Amino acids chained together to constitute a protein molecule.



gives the protein its biological properties (for instance, its ability to digest sugar or to become part of a muscle fiber); those come from the three-dimensional (3-D) shape that the ribbon adopts in its environment. A protein molecule, once made, is not a chainlike, highly flexible object (think like a section of chain-link fence); rather, it's more like a compact, well-bundled ball of string. The final 3-D shape of the protein molecule is uniquely dictated by its sequence because some amino-acid types (for instance, *hydrophobic* residues L, V, I) have no desire whatsoever to be at the surface interacting with the surrounding water — while others (for instance, *hydrophilic* residues D, S, K) are actively looking for such an opportunity. The protein chain is also affected by other influences, such as the electric charges carried by some of the amino acids, or their capacity to fit with their immediate neighbors.

The first 3-D structure of a protein was determined in 1958 by Drs. Kendrew and Perutz, using the complicated technique of X-ray crystallography. (Not for the faint of heart. Don't grapple with how it works unless you want to turn professional!) Besides winning one more Nobel Prize for the nascent field of molecular biology, this feat made the doctors realize that proteins have precise and specific shapes, encoded in the sequence of amino acids. Hence, they predicted that proteins with similar sequences would fold into similar shapes — and, conversely, that proteins with similar structures would be encoded by similar sequences of amino acids. The function of a protein turned out to be a direct consequence of its 3-D structure (shape). The resulting logical linkage

```
SEQUENCE⇨STRUCTURE⇨FUNCTION
```

was established, and is now a central concept of molecular biology and bioinformatics.

Playing with protein structure models on a computer screen is, of course, much easier than carrying around a thousand-piece, 3-D plastic puzzle. As a consequence, an increasing proportion of the bioinformatics pie is now devoted to the development of cyber-tools to navigate between sequences and 3-D structures. (This specialized area is called *structural bioinformatics.*) Thanks to many free resources on the Internet, it is not difficult to display some beautiful protein pictures on your own computer — and start playing with them as in video games. (We show you how to do that in Chapter 11.)

Before you get a chance to read that chapter, Figure 1-3 gives you an idea of what a 400-amino-acid typical protein 3-D (schematic) structure looks like — when you don't have a color monitor and can't make it move and turn!

Don't forget: Protein molecules, even in their wonderful complexity, are still pretty small. The one in Figure 1-3 would fit in a box whose sides measure 70/1,000,000 millimeters. There are thousands of different proteins in a single bacterium, each of them in thousands of copies — more than enough evidence that Life Is Not Simple!

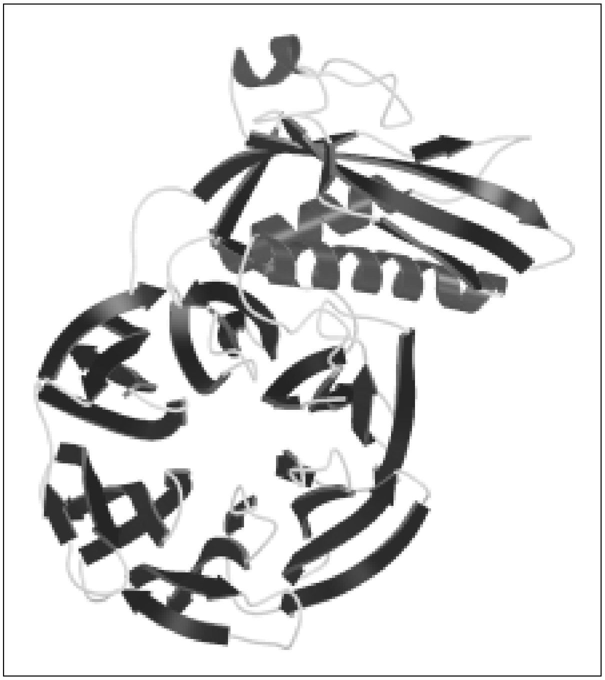


Figure 1-3: Example of protein 3-D structure (schematic).

Protein bioinformatics covered in this book

The study of protein sequences can get pretty complicated — so complicated, in fact, that it would take a pretty thick book to cover all aspects of the field. We'd like to take a more selective approach by focusing on those aspects of protein sequences where bioinformatic analyses can be most useful. The following list gives you a look at some topics where such an analysis is particularly relevant to protein sequences — and also tells which chapters of this book cover those topics in greater detail:

- Retrieving protein sequences from databases (Chapters 2, 3, and 4)
- Computing amino-acid composition, molecular weight, isoelectric point, and other parameters (Chapter 6)
- Computing how hydrophobic or hydrophilic a protein is, predicting antigenic sites, locating membrane-spanning segments (Chapter 6)
- Predicting elements of secondary structure (Chapters 6 and 11)
- Predicting the domain organization of proteins (Chapters 6, 7, 9, and 11)
- Visualizing protein structures in 3-D (Chapter 11)
- Predicting a protein's 3-D structure from its sequence (Chapter 11)

- Finding all proteins that share a similar sequence (Chapter 7)
- Classifying proteins into families (Chapters 7, 8, and 9)
- Finding the best alignment between two or more proteins (Chapters 8 and 9)
- Finding evolutionary relationships between proteins, drawing proteins' family trees (Chapters 7, 9, 11, and 13)

Analyzing DNA Sequences

During the 1950s, while scientists such as Kendrew and Perutz were still struggling to determine the first 3-D structures of proteins, other biologists had already acquired a lot of indirect evidence (via extremely clever genetics experiments) that *deoxyribonucleic acid* (DNA) — the stuff that makes up our genes — was *also* a large macromolecule. It was a long, chainlike molecule twisted into a double helix, and each link in the chain was a pairing of two out of four constituents called *nucleotides.* (A nucleotide is made up of one phosphate group linked to a pentose sugar, which is itself linked to one of 4 types of nitrogenous organic bases symbolized by the four letters A, C, G, and T.)

However, molecular biologists had to wait until much later — the 1970s, to be more precise — before they could determine the sequence of DNA molecules and get direct access to the sequences of gene nucleotides.

This was a revolution (earning A. Sanger his second Nobel Prize!) because the small DNA sequence alphabet (4 nucleotides, as compared to 20 amino acids) allowed a much simpler and faster reading — and quickly lent itself to complete automation. Currently, the worldwide rate of determining DNA sequences is faster (by orders of magnitude) than the rate of protein sequencing.

Reading DNA sequences the right way

As was the case for the 20 amino acids found in proteins, the 4 nucleotides making DNA have different bodies but all have the same pair of hooks: 5' phosphoryl and 3' hydroxyl (pronounced *five prime* and *three prime*) by reference to their positions in the deoxyribose sugar molecule, which is part of the nucleotide chaining device. Figure 1-4 shows what free individual nucleotides look like.

Forming a bond between the 5' and 3' positions of the constituent nucleotides then makes the DNA molecule. Figure 1-5 shows a schematic representation of the resulting DNA strand.

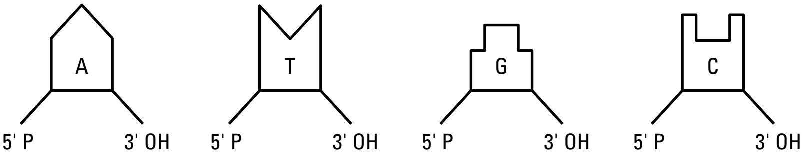


Figure 1-4: The four nucleotides making DNA.

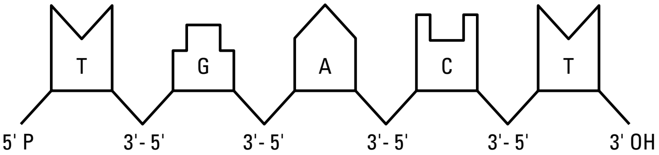


Figure 1-5: Chained nucleotides constituting a DNA strand.

After the nucleotides are linked, the resulting DNA strand exhibits an unused phosphoryl group (PO_4) at the 5' end, and an unused hydroxyl group (OH) at the 3' end. These extremities are respectively called the *5'-terminus* and the *3'-terminus* of the DNA strand.

REMEMBER

A DNA sequence is always defined (in books, databases, articles, and programs) as the succession of its constituent nucleotides *listed from the 5'- to 3'- terminus* (that is, end). The sequence of the (short!) DNA strand shown in Figure 1-5 is then

```
TGACT = Thymine-Guanine-Adenine-Cytosine-Thymine
```

The two sides of a DNA sequence

In the same laboratory where Kendrew and Perutz were trying to figure out the first 3-D structure of a protein, Watson and Crick elucidated — in 1953 — the famous double-helical structure of the DNA molecule. These days everybody has a mental picture of this famous spiral-staircase molecule; the elegance of the DNA *double helix* probably helped make it the most popular notion to come out of molecular biology. But what made this discovery so important — earning one more Nobel Prize for molecular biology — was not the helical shape, but the discovery that the DNA molecule consists of two complementary strands, shown in Figure 1-6.

The IUPAC code for DNA sequences

The following table lists the one-letter codes (IUPAC codes) used to work with DNA sequences. Official IUPAC codes, from the International Union of Pure and Applied Chemistry, are defined for all possible two- and three-way ambiguities. The table shows only the ones most frequently used.

Most Common Letters Used for DNA Nucleotide Sequences

1-Letter Code	*Nucleotide Name*	*Category*
A	Adenine	Purine
C	Cytosine	Pyrimidine
G	Guanine	Purine
T	Thymine	Pyrimidine
N	Any nucleotide (any base)	(n/a)
R	A or G	Purine
Y	C or T	Pyrimidine
--	-----	None (gap)

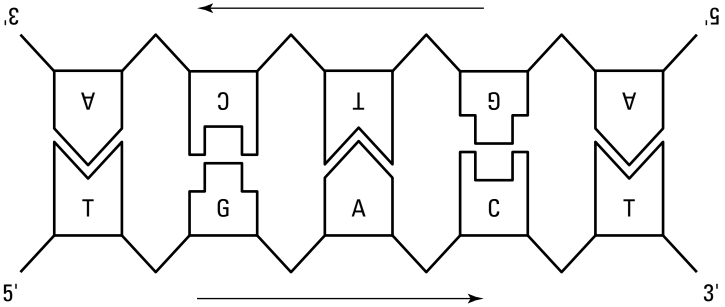


Figure 1-6: The two complementary strands of a complete DNA molecule.

By *complementarity,* we mean that a thymine (T) on one strand is always facing an adenine (A) (and vice versa) — and guanine (G) is always facing a cytosine (C). These couples, A-T and G-C, although not linked by a chemical bond, have a strict one-to-one reciprocal relationship. When you know the sequence of nucleotides along one strand, you can automatically deduce the sequence on the other one. This amazing property — and not the stylish helical structure — is the Rosetta Stone that explains everything about DNA sequences. For instance, when living organisms reproduce, each of their genes must be duplicated. In order to do this, nature doesn't go about it the way a photocopier would — by making an exact copy. Rather, nature separates the DNA strands and makes *two complementary* ones, thanks to the magical two-sided structure of DNA molecules.

This double strand structure of DNA makes the definition of a DNA sequence ambiguous: Even with our convention of reading the nucleotides from the 5' end toward the 3' end, you may decide to write down the bottom or the top sequence. Convince yourself that they're both equally valid sequences by turning this book upside down! Thus, at each location, a DNA molecule corresponds to two — totally different — sequences, related by this reverse-and-complement operation. This isn't complicated; simply keep it in mind every time you work with DNA sequences.

Fortunately, most database mining programs, such as BLAST, know about this property, and take both strands into account when reporting their results. But some programs don't bother — and only analyze the sequence you gave them. In cases where both strands matter, always make sure that a complete analysis has been performed. (We discuss these details further in Chapters 3, 5, and 7.)

Palindromes in DNA sequences

Newcomers to DNA sequence analysis are usually confused by the notion of reverse complementary sequences. However, in due time you'll be able to recognize right away that the two sequences

```
ATGCTGATCTTGGCCATCAATG and CATTGATGGCCAAGATCAGCAT
```

correspond to facing strands of the same DNA molecule.

One fascinating property of DNA complementarity is the fact that regions of DNA may correspond to sequences that are identical when read from the two complementary strands. Figure 1-7 helps illustrate this magic trick.

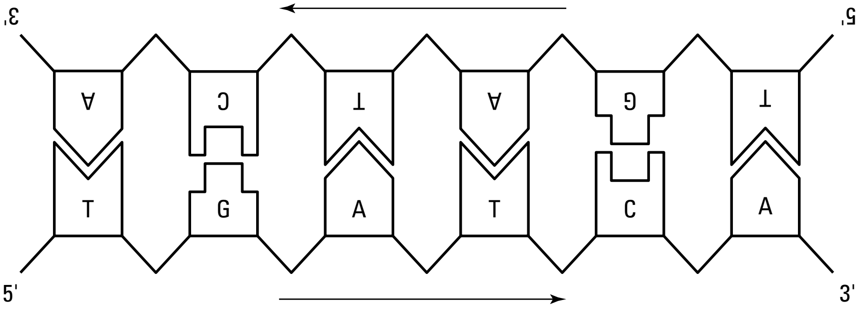


Figure 1-7: How two complementary strands can be read the same way.

Such sequences are called *palindromes,* after the term for a phrase or sentence that reads the same in both directions (such as "Madam, I'm Adam" or "A man, a plan, a canal: Panama.") *Palindromic* sequences aren't merely a curiosity; they play important biological roles. For instance, most DNA cutting enzymes (so-called *restriction enzymes*) have palindromic target sequences. Other palindromic sequences serve as *binding sites,* where regulatory proteins stick so they can turn genes on and off. Palindromic sequences also have a strong influence on the 3-D structure of DNA molecules. (And not just DNA. See the next section for more on palindromic sequences in RNA.) Looking for exact or approximate palindromes in DNA sequences is a classic bioinformatic exercise.

Analyzing RNA Sequences

DNA (deoxyribonucleic acid) is the most dignified member of the nucleic acid family of macromolecules. Its sole and only task is to ensure — forever — the conservation of the genetic information for its organism. It is thus very stable and resistant, and lies well-protected in the nucleus of each cell. *Ribonucleic acid* (RNA) is a much more active member of the nucleic acid family; it's synthesized and degraded constantly as it makes copies of genes available to the cell factory.

In the context of bioinformatics, there are only two important differences between RNA and DNA:

- RNA differs from DNA by one nucleotide.
- RNA comes as a single strand, not a helix.

The one-letter IUPAC codes for RNA sequences are shown in Table 1-2.

Table 1-2 Most Common Letters Used for RNA Nucleotide Sequences

1-Letter Code	*Nucleotide Base Name*	*Category*
A	Adenine	Purine
C	Cytosine	Pyrimidine
G	Guanine	Purine
U	Uracil	Pyrimidine
N	Any nucleotide	Purine or Pyrimidine

(continued)

Table 1-2 *(continued)*

1-Letter Code	*Nucleotide Base Name*	*Category*
R	A or G	Purine
Y	C or U	Pyrimidine
--	-------	None (gap)

Some programs automatically handle the U-instead-of-T conversion — and many don't even distinguish between the two classes of nucleic acids. So don't be surprised if a database entry displays RNA sequences (such as messenger RNA) with a T instead of a U. In fact, like proteins, RNA sequences are encoded in the DNA. For this reason, people have adopted the habit of working with the sequences of the RNA *genes* (written in DNA) rather than with RNA sequences.

RNA structures: Playing with sticky strands

Even though RNA molecules consist of single strands of nucleotides, their natural urge for pairing with complementary sequences is still there. Think of each such single strand as a free-floating piece of Scotch tape: You know that it won't take long for that tape to become a messy ball, until no sticky part remains exposed. This is exactly what happens to the single-stranded RNA molecule — more or less (for the sake of poetic license) — although Figure 1-8 shows more precisely how the stickiness works.

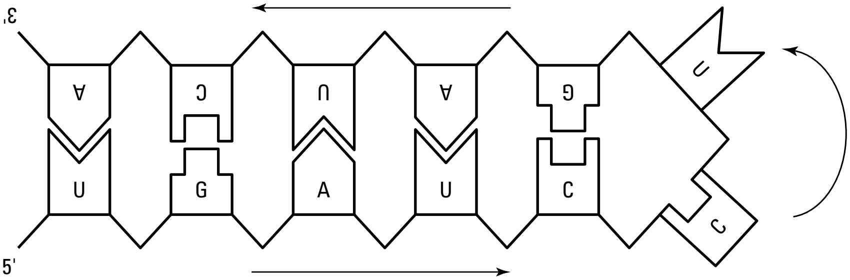


Figure 1-8: How RNA turns itself into a double-stranded structure.

Now you understand why we insisted on the notion of strand complementarity (refer to Figure 1-6). Single-stranded RNA molecules pair different regions of their sequences to form stable double-helical structures — admittedly less regular than (but quite similar to) the double-helical structure of DNA. Once synthesized, each RNA molecule quickly adopts a compact fold — trying to pair as many nucleotides as possible, while keeping the chain not only flexible but true to its own geometry. Hairpin shapes, as shown in Figure 1-8, are the basic elements of RNA secondary structure; they're made up of *loops* (the unpaired C-U in Figure 1-8) and *stems* (the paired regions).

Just for fun, verify for yourself that a palindromic RNA sequence results in a perfect hairpin, with no loop. While attempting to pair as many nucleotides as possible, the RNA chain folds in space, resulting in a specific 3-D structure that's dictated by its sequences. As with proteins, the linear sequence of the building blocks dictates the final 3-D shape. The biological function of RNA molecules derives from their 3-D shapes or from their sequence complementarity with specific genes.

Computing (predicting) the final fold of an RNA molecule from its sequence is a challenging problem that drove many historical developments in bioinformatics. The recent discovery that small RNA molecules can switch off the activity of a number of genes is what triggered a renewed interest in these sticky sequences. (Go directly to Chapter 12 if your main interest is in RNA bioinformatics.)

More on nucleic acid nomenclature

Don't panic if you get the impression that books, courses, and the technical literature all use many different words and abbreviations to designate the building blocks of nucleic acids: That's actually true — for example, you'll find "base," "base pair," "nucleoside," and "nucleotide" — but note: These different names designate slightly different chemical entities, and those differences are irrelevant for us just now. So far we've used the term *nucleotide* — abbreviated *nt* (as in "a 400-nt-long sequence"). This way of labeling a sequence refers to the length of the DNA (or RNA) molecules in terms of the number of positions they have available for nucleotides. For instance, the sequence in Figure 1-5 is 5 nt long.

Notice that we say *number of positions* rather than *number of nucleotides.* A 400-nt long DNA molecule has 400 positions for nucleotides, but it actually contains twice that many (800) because every position contains a pair of nucleotides. To make this clearer, DNA sequence sizes are often given in *base pairs,* abbreviated *bp.* Thus the DNA sequence in Figure 1-5 is 5 bp long. Larger units, such as *kb* (1000 bp) or *Mb* (mega-bp) are also used.

DNA Coding Regions: Pretending to Work with Protein Sequences

Of the hundreds of thousands of protein sequences found in current databases, only a small percentage correspond to molecules that have actually been isolated by somebody or experimented upon. That's because determining the sequence of a protein is much more difficult than sequencing DNA — but all the proteins that a given organism (whether microbe or human being) can synthesize are encoded in the DNA sequence of its genome. Thus, the smart shortcut that molecular biologists have been using is to read protein sequences directly at the information source: in the DNA sequence! This way, we can pretend to know the amino-acid sequence of a protein that has never been isolated in a test tube.

Turning DNA into proteins: The genetic code

When you know a DNA sequence, you can translate it into the corresponding protein sequence by using the genetic code, the very same way the cell itself generates a protein sequence. The genetic code is universal (with some exceptions — otherwise life would be too simple!), and it is nature's solution to the problem of how one uniquely relates a 4-nucleotide sequence (A, T, G, C) to a suite of 20 amino acids; we're using symbols (rather than actual chemicals) to do the same. Understanding how the cell does this was one of the most brilliant achievements of the biologists of the 1960s. Yet the final answer can be contained in a (miraculously small) table — as shown in Figure 1-9. Have a look, but feel free to indulge in awed silence as you enter the most sacred monument of modern biology.

Here's how to use the table shown in Figure 1-9: From a given starting point in your DNA sequence, start reading the sequence 3 nucleotides (one *triplet*) at a time. Then consult the genetic code table to read which amino acid corresponds to the current triplet (technically referred to as *codons*). For instance, the following DNA (or messenger RNA) sequence is decoded as follows:

1. **Read the DNA sequence:**

   ATGGAAGTATTTAAAGCGCCACCTATTGGGATATAAG

2. **Decompose it into successive triplets:**

   ATG GAA GTA TTT AAA GCG CCA CCT ATT GGG ATA TAA G . . .

3. **Translate each triplet into the corresponding amino acid:**

   M E V F K A P P I G I STOP

If your DNA sequence is correctly listed in the 5' to 3' orientation, you generate the protein sequence in the conventional N- to C-terminus as well. This approach has an advantage: You don't have to think about these orientation details ever again.

Thus, if you know where a protein-coding region starts in a DNA sequence, your computer can pretend to be a cell and generate the corresponding amino-acid sequence! This simple computer translation exercise is at the origin of most of the so-called *protein sequences* that you can find in databases. Many sequence analysis programs acknowledge this fact by offering on-the-fly translation, so you can process DNA sequences as virtual protein sequences with a simple mouse click.

More with coding DNA sequences

Using the example in the first paragraphs of the section "DNA Coding Regions: Pretending to Work with Protein Sequences," you can see that the resulting protein sequence depends entirely on the way you converted your DNA sequence into triplets before using the genetic code. For instance, using the second position as starting point leads to

```
1- ATGGAAGTATTTAAAGCGCCACCTATTGGGATATAAG

2- A TGG AAG TAT TTA AAG CGC CAC CTA TTG GGA TAT AAG

3-  W K Y L K R H L L G Y K
```

Beginning with the third position (GGA-AGT- . . .) again leads to an entirely different translation.

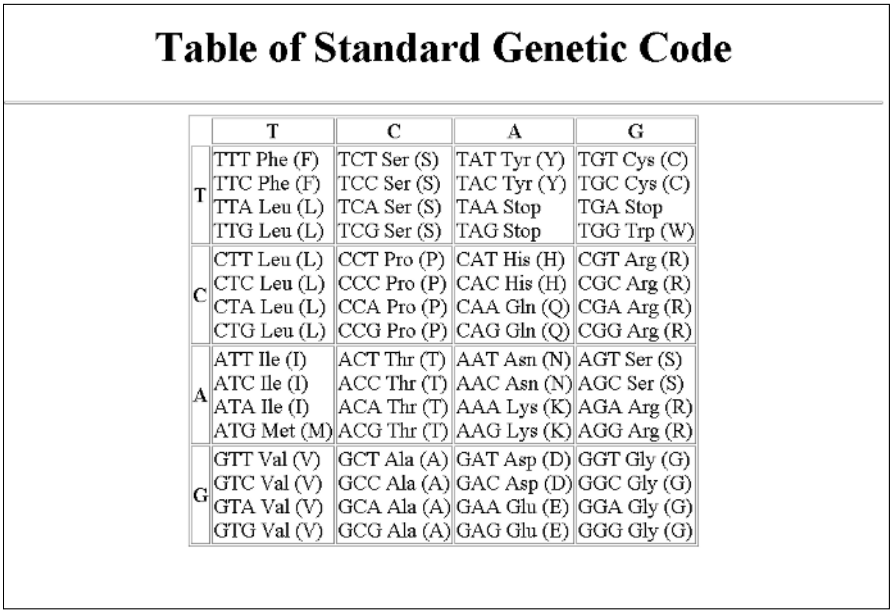


Table of Standard Genetic Code

	T	C	A	G
T	TTT Phe (F)	TCT Ser (S)	TAT Tyr (Y)	TGT Cys (C)
	TTC Phe (F)	TCC Ser (S)	TAC Tyr (Y)	TGC Cys (C)
	TTA Leu (L)	TCA Ser (S)	TAA Stop	TGA Stop
	TTG Leu (L)	TCG Ser (S)	TAG Stop	TGG Trp (W)
C	CTT Leu (L)	CCT Pro (P)	CAT His (H)	CGT Arg (R)
	CTC Leu (L)	CCC Pro (P)	CAC His (H)	CGC Arg (R)
	CTA Leu (L)	CCA Pro (P)	CAA Gln (Q)	CGA Arg (R)
	CTG Leu (L)	CCG Pro (P)	CAG Gln (Q)	CGG Arg (R)
A	ATT Ile (I)	ACT Thr (T)	AAT Asn (N)	AGT Ser (S)
	ATC Ile (I)	ACC Thr (T)	AAC Asn (N)	AGC Ser (S)
	ATA Ile (I)	ACA Thr (T)	AAA Lys (K)	AGA Arg (R)
	ATG Met (M)	ACG Thr (T)	AAG Lys (K)	AGG Arg (R)
G	GTT Val (V)	GCT Ala (A)	GAT Asp (D)	GGT Gly (G)
	GTC Val (V)	GCC Ala (A)	GAC Asp (D)	GGC Gly (G)
	GTA Val (V)	GCA Ala (A)	GAA Glu (E)	GGA Gly (G)
	GTG Val (V)	GCG Ala (A)	GAG Glu (E)	GGG Gly (G)

Figure 1-9: The universal genetic code.

Because of the triplet-based genetic code, a given DNA interval, *on a given strand,* can theoretically be translated in three different ways — basically three perspectives that are known in the field as *reading frames.* Because the DNA can be used from both strands, a total of six possible reading frames are possible for translating a DNA sequence into proteins. With very few exceptions (found in exotic viruses), only one of these six frames is used for any given DNA coding region. An interval of DNA sequence that remains free of STOP (the translation of TAA, TGA, or TAG) is called an *open reading frame* (ORF).

Additional complications arise from the fact that some DNA sequences are not encoding proteins at all — and that higher organisms have large pieces of noncoding DNA inserted within their genes. A large part of bioinformatics is devoted to the development of methods to locate protein-coding regions in DNA sequences, to delineate precisely where genes start and end, or where they are interrupted by the noncoding intervals (called *introns*).

DNA/RNA bioinformatics covered in this book

Need a road map to the bioinformatic analyses that are relevant to DNA/RNA sequences covered in this book? Here it is:

- Retrieving DNA sequences from databases (Chapters 2 and 3)
- Computing nucleotide compositions (Chapter 5)
- Identifying restriction sites (Chapter 5)
- Designing polymerase chain-reaction (PCR) primers (Chapter 5)
- Identifying open reading frames (ORFs) (Chapter 5)
- Predicting elements of DNA/RNA secondary structure (Chapter 12)
- Finding repeats (Chapter 5)
- Computing the optimal alignment between two or more DNA sequences (Chapters 7, 8, and 9)
- Finding polymorphic sites in genes (single nucleotide polymorphisms, SNPs) (Chapter 3)
- Assembling sequence fragments (Chapter 5)

Working with Entire Genomes

The first truly efficient technique to sequence DNA was discovered in 1977. In 1995, the first sequence of an entire genome (from the microbe *Hemophilus influenzae*) was determined. Between these two dates, DNA-sequencing technologies improved steadily, but such technologies still tended to concentrate on mining individual genes for information. During this period, biologists were mostly sequencing DNA fragments that were a few thousand nucleotides in length, simply because they were interested in specific genes that they had started working on years before. Most of the bioinformatics tools available today were created during that period. They include

- All basic sequence-alignment programs
- Phylogenetic and classification methods
- Various display tools adapted to relatively small-sequence objects (such as protein sequences no more than a few thousand characters long)

Genomics: Getting all the genes at once

The determination of the first complete genome sequence terminated the gene-by-gene routine and initiated the era of *genomics,* the genetic mapping, physical mapping, and sequencing of entire genomes. As a consequence, the DNA sequences we have to work with now are much longer — close to a million-bp in length for microbes and up to several billion-bp in length for animals and humans. This revolution called for the design of new bioinformatic tools and databases capable to store, query, analyze, and display these huge objects in a user-friendly manner. Chapters 3, 5, and 7 present some of the questions that biologists address at the genome scale, and show the relevant bioinformatic tools in action.

In contrast to the early days of the gene-by-gene approach, DNA sequences are now often obtained (along with the presumed protein sequences derived from those DNA sequences) without any prior knowledge of what is actually there. In essence, genes are both sequenced *and discovered* at the same time. This development prompted the emergence of an entirely new branch of bioinformatics devoted to the *parsing* of large DNA sequences into their components (genes, transcription units, protein-coding regions, regulatory elements, and so forth). This first pass is then followed by a longer phase of genome *annotation,* where the biological functions of these various elements are (more or less tentatively) predicted. Part IV of this book presents you with some of these most advanced techniques.

Figure 1-10, representing the whole genome of the bacterium *Rickettsia conorii,* illustrates this new level of complexity. This circular DNA molecule is 1.3 million bp long, on the small side for a bacterium. Each little rectangle in the two most external circles of features (one circle per strand) corresponds to a protein-coding gene in the circular genome. Each rectangle corresponds to approximately 1000 bp. Nobody knew which genes — or which proteins — were in that bacterium before the sequencing started. Almost everything we know now about this bacterium (and many others we can describe as fairly inaccessible, such as those thriving on the ocean floor near volcanic vents at 100°C) has been derived from bioinformatic analyses.

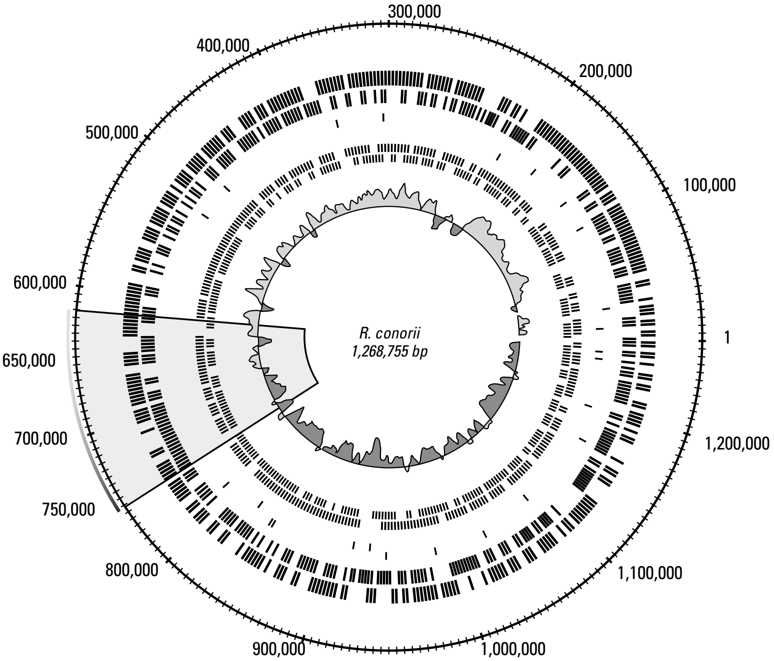


Figure 1-10: Representation of a bacterial genome.

Genome bioinformatics covered in this book

The following list lets you know where in this book you'll find more in-depth coverage of specific topics (some of them bristling with scary, mouth-filling terms) related to genome bioinformatics:

- Finding which genomes are available (Chapter 3)
- Analyzing sequences in relation to specific genomes (Chapters 3 and 7)
- Displaying genomes (Chapter 3)
- Parsing a microbial genome sequence: ORFing (Chapter 5)
- Parsing a eukaryotic genome sequence: GenScan (Chapter 5)
- Finding orthologous and paralogous genes (Chapter 3)
- Finding repeats (Chapter 5)