

## Chapter 2

# Bioinformatics for Microbiologists: An Introduction

**Outline** Bioinformatics is the study of biological information using computational approaches. It depends on knowledge of both the underlying biology and 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 results from the lab. Of these tools, sequence alignment methods, such as BLAST, are of crucial importance. This chapter will focus mainly 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*.

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.

Query. The complete sequence of the genome of ..... has been determined by  
 Subject. The complete sequence of the genome of ..... has been determined by

Query. the whole genome shotgun method .... The entire length of the genome was  
 Subject. assembling the sequences ..... The entire length of the genome was

Query. 1,669,695 bp. The authenticity of the entire \_\_\_\_\_ sequence was supported by  
 Subject. 1,738,505 bp. The authenticity of the entire genome sequence was supported by

Query. restriction analysis of long PCR products, which were directly amplified from the  
 Subject. restriction analysis of long PCR products, which were directly amplified from the

Query. genomic DNA. As the potential protein-coding regions, a total of 2,694 open  
 Subject. genomic DNA. As the potential protein-coding regions, a total of 2 061 open

Query. reading frames (ORFs) were assigned. DNA Res., 6:83-101. (1999).  
 Subject. reading frames (ORFs) were assigned. DNA Res., 5:55-76. (1998).

**Fig. 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

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 missing in the query due to a *deletion*. Not knowing which text came first,<sup>1</sup> 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 they were by the same author, or perhaps the later author was aware of the earlier version. 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 contains only four 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

<sup>1</sup> In this case we know the date of the publications, but often a query finds similarities against database entries without knowledge of 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.

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, because 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 increases 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 Fig. 2.1, it could be guessed that ‘assembling the sequences’ was somewhat similar. Thus, alignment can identify the 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 top appear unrelated. 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 in fact the same sequence is being compared, noting 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.

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 less than that of their translated protein sequence, as illustrated in Fig. 2.3.

```

Query.   5' GGCCTAGTAGCCCATAGACTATACACCCGGATA 3'
          : : : : :
Subject. 5' TAACCGGGTTTATAGGCTATGGGGTAGTAGGCC 3'

Query.   5' GGCCTAGTAGCCCATAGACTATACACCCGGATA 3'
          : : : : : : : : : : : : : : : : :
Subject. 5' GGCCTACTACCCCATAGCCTATAAACCCGGTTA 3'

Subject (ds) 5' TAACCGGGTTTATAGGCTATGGGGTAGTAGGCC 3'
              3' ATTGCCCAATATCCGATACCCCATCATCCGG 5'

```

**Fig. 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

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

Query	GlyLysValAlaMetLeuProSerVal	(Frame 2)
	:	
Subject	ArgGluSerArgHisTyrAlaIleSerMet	(Frame 1)
	Identity: 1 in 10 positions (10%)	

Query	AlaLysSerProCysCysHisLeuTyr	(Frame 3)
	:	
Subject	ArgGluSerArgHisTyrAlaIleSer	(Frame 1)
	Identity: 1 in 10 positions (10%)	

**Fig. 2.3** Optimal alignment of a DNA sequence (*top*) followed by the corresponding amino acid sequence (represented in one-letter code). This illustrates that the similarity is generally greater at the amino acid level than at the DNA levels. Identity is indicated by ‘:’. However, if this piece of DNA is part of a protein in a different reading frame (next two alignments), similarity at the amino acid level is much less than that of the DNA level

The reading frame of a DNA sequence may not always be known, and shifting it by one position has dramatic effects on the translated amino acid sequence. The similarity at the protein level can be completely destroyed, as Fig. 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 enzyme 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 (Fig. 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,

```

Query  DADTGTGKAAAGVELTTKVVGAPVSALVGAKVLGNVGKISQYIIAQRAAQGQTLTGQPNTQ
      : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : :
Subj   DAPAGQKAAAGAEPALQLLGGGIAAAVGARLTGGTELASSIALALAAARGKTAALLGGLT
Alignment length: 50 aa      Query length: 60 aa
Identity over alignment length: 40%  Identity over query length: 33%
Similarity over alignment length: 66%  Similarity over query length: 55%

```

**Fig. 2.4** Alignment of two protein sequences, indicating amino acids that are identical with ‘:’ and similar to ‘.’. The scores for identity and similarity are given, both for the alignment length (to the left) and for the complete query length (to the right)

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 matters only that an amino acid is charged, and 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 to 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 one to specifically search all or particular selections of microbial genomes.<sup>2</sup> 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 Fig. 2.3 will not be missed. BLASTX uses a DNA query and translates this in all three reading frames, for both strands, and performs six BLASTP searches in addition to BLASTN. BLASTP uses various similarity matrices to determine which amino acids are similar. Complete textbooks have been

<sup>2</sup> [http://www.ncbi.nlm.nih.gov/sutils/genom\\_table.cgi](http://www.ncbi.nlm.nih.gov/sutils/genom_table.cgi)

written about the use of BLAST. Only a few points that have puzzled some BLAST users will be addressed here.

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 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 the content of the searched database. The 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 of 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 scores and E-values. 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 disease). The first five hits suggest that sequence similarity to an oxidoreductase subunit (an enzyme) is detected. The annotation of the sixth hit doesn't reveal the function. The next six hits suggest similarity to flagellin (a structural component of flagella). Flagellin and oxidoreductase are very different proteins, and the fact that the similarity to either class of proteins is clearly divided between the halves of the query again suggests a chimeric sequence. The graphical representation of the sixth hit suggests that





**Fig. 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 third hit, but in fact this alignment was obtained with eight 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)

there may be at least one entry in the searched database of such a chimera (though the chimera 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 at the bottom of Fig. 2.5. These are mostly (but not all) produced from different strains and are present in the non-redundant database because 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 two 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 two domains in one sequence, these will be presented as two separate hits. If so, it is important to visually inspect the alignment location to verify this possibility.

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 databases have 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 second line onwards is read as a 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 added for typographic formatting purposes only, to be continued on the next line. The end of line is indicated with a hard return, indicated by the '¶' symbol.

```
>gi7800406|gb|AAF70002.1|AF250878_163 'DNA binding protein, H-NS-like' [Salmonella typhi]¶
MSEALKSLNNIRTLRAQGRELPLEILEELLEKLSVVVEERRQEESSKEAEKARLEKIESLRQLMLE¶
DGIDPEELLSSFSKSGAPKKVREPRPAKYKYTDVNGETKTWTGQGRTPKALAEQLEAGKKLDDFL¶
```

## ***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 program.<sup>3</sup> This program first calculates the highest similarity for each possible pair of combinations, and then estimates the optimal multiple

---

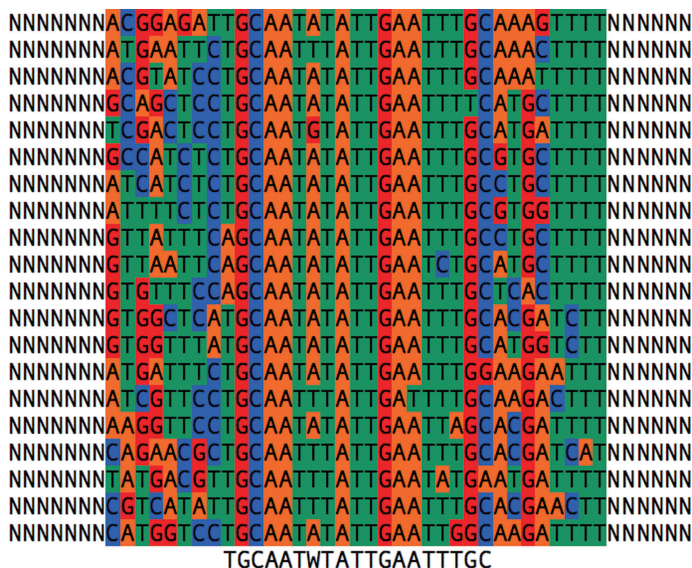
<sup>3</sup> <http://www.ebi.ac.uk/Tools/clustalw> (for example).



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 has already been established. CLUSTALW is not suitable to search databases. A better approach is first to 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.

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 locations 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.

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*

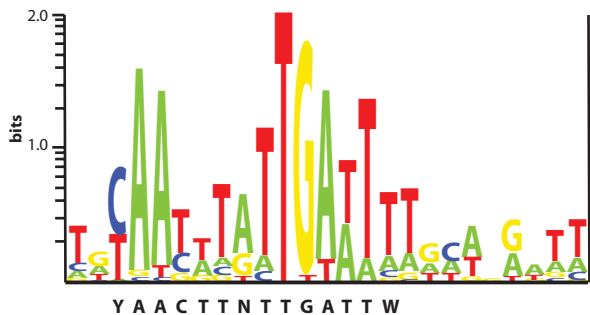


**Fig. 2.6** Multiple sequence alignment of 20 DNA sequences for IHF binding sites. Nucleotides in the region that was experimentally proven to contain the binding site are color coded. Nucleotides outside the binding site are not defined (N). Below is a consensus sequence given for this alignment

**Table 2.1** The DNA alphabet

No ambiguity	1 ambiguity	2 ambiguities	3 ambiguities
<b>G</b>	<b>S</b> =G or C	<b>H</b> =A or C or T	<b>N</b> =A or G or C or T
<b>A</b>	<b>W</b> =A or T	<b>B</b> =G or T or C	
<b>T</b>	<b>R</b> =G or A	<b>V</b> =G or C or A	
<b>C</b>	<b>Y</b> =T or C		
	<b>M</b> =A or C		
	<b>K</b> =G or T		

Note: S stands for ‘Strong’ as G and C share three hydrogen bonds; A and T share only two H-bonds, thus W for ‘Weak.’ R stands for puRine and Y for pYrimidine.

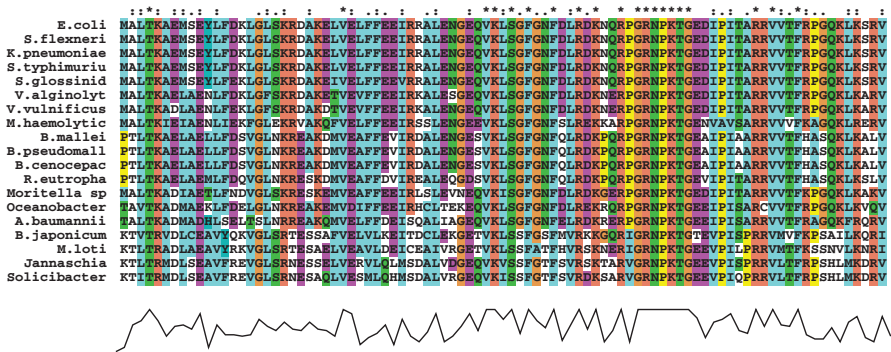


**Fig. 2.7** Sequence logo for the IHF consensus binding sites. The highest value on the bits scale is 2 bits, representing a 100% conserved nucleotide

*logo* plot. An example is shown in Fig. 2.7 for the same IHF binding site alignment, but this time showing a longer section of the sequence. In this figure the size of the letter is a measure of 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, Fig. 1.7). Sequence logo plots are further applied in Chapter 10.

### From Alignments to Phylogenic Relationships

Multiple alignments can also be done for proteins, and Fig. 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 Fig. 2.7. But from other segments

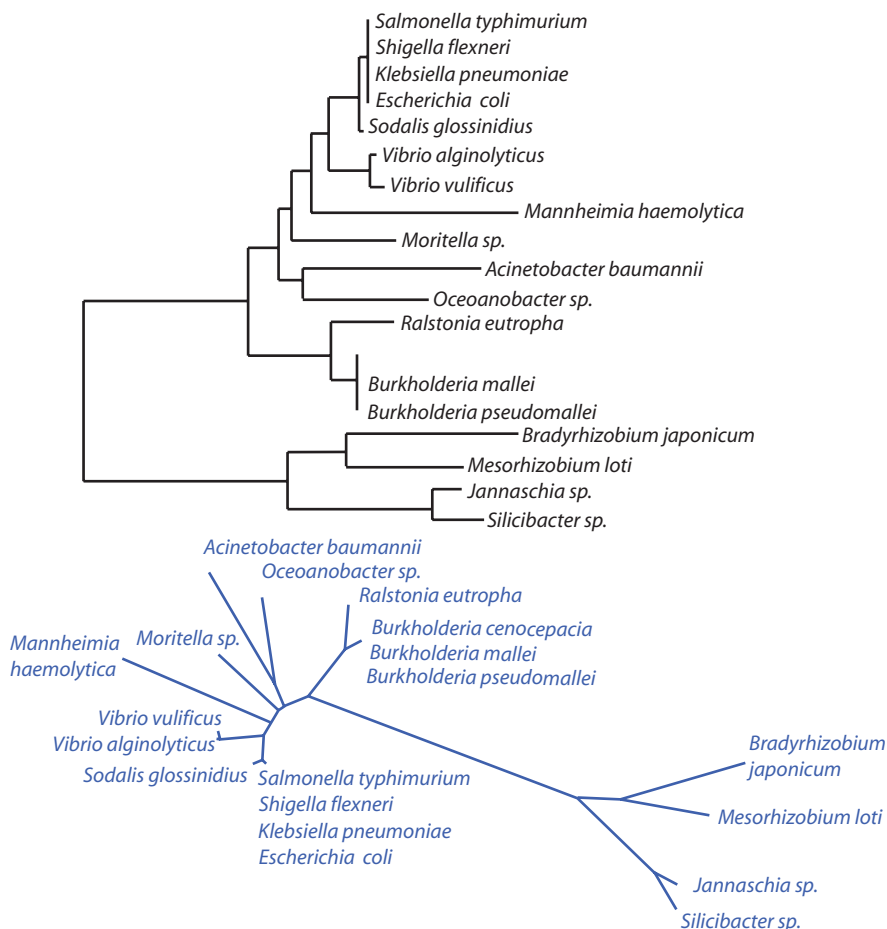


**Fig. 2.8** Multiple sequence alignment for 19 different IHF alpha proteins produced using CLUSTAL. For sake of clarity, only the first part of the protein sequence is shown. The original FASTA file contained the proteins in alphabetical order by the species from which they were derived (starting with *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

in the alignment, it can also be seen that some sequences (derived from various organisms) are more like each other, and others cluster to different groups. The proteins in Fig. 2.8 were originally arranged in alphabetical 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, designed to perform multiple alignments, normally does). This illustrates that the alignment conservation hints at how closely the proteins are related to each other.

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 signals 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 (because 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 Fig. 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 looser set of ‘other organisms,’ which are taxonomically more diverse. There are several methods for producing a tree plot, and many web sites



**Fig. 2.9** Phylogenetic tree of the IHF protein alignment shown in Fig. 2.8; the tree on the *top* (black) is rooted, and the one on the *bottom* (in blue) is unrooted. Both trees represent the same phylogenetic data

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 been around only 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 Fig. 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 represents an outlier. Another method for rooting the tree is to use the molecular clock assumption, which also has problems in that it is perhaps assuming 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 many IS10 sequences (an insertion sequence typical of prokaryotes) would be found in plants? Not many, 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 Fig. 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 following 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-fetched. 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 to 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 with 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 (or 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 helps to screen which ORF is a good gene candidate. However, if we were only to 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 computer 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 two 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 to 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 is 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 for ongoing research. Genome sequences also have generated, apart from *in silico* research (the designation for computer-based research, to complement *in vivo* and *in vitro* analysis), a whole new area of wet lab research that was unthinkable 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 members of the same species. We can now define the so-called *pan-genome*, considering all genes that can possibly be 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 bioinformatic 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 that bioinformatics has become an essential tool to greatly enhance the possibilities 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 from both an informatical and a biological viewpoint are needed to produce data that help microbiology progress.

## References

- Altschul SF, Gish W, Miller W, Myers EW and Lipman DJ, “Basic local alignment search tool”, *J Mol Biol*, **215**:403–410 (1990). [PMID: 2231712]
- Felsenstein J. “Inferring Phylogenies” (Sinauer Associates, Inc. Sunderland MA, 2004).
- Kawarabayasi Y, *et al.*, “Complete sequence and gene organization of the genome of a hyper-thermophilic archaeobacterium, *Pyrococcus horikoshii* OT3”, *DNA Res*, **5**:55–76 (1998). [PMID: 9679194]
- Kawarabayasi Y, *et al.*, “Complete genome sequence of an aerobic hyper-thermophilic crenarchaeon, *Aeropyrum pernix* K1”, *DNA Res*, **6**:83–101 (1999). [PMID: 10382966]
- Lipman DJ and Pearson WR, “Rapid and sensitive protein similarity searches”, *Science*, **227**:1435–1444 (1985). [PMID: 2983426]
- Thompson JD, Higgins DG and Gibson TJ, “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice”, *Nucleic Acids Res*, **22**:4673–4680 (1994). [PMID: 7984417]
- Ussery DW, Larsen TS, Wilkes KT, Friis C, Worning P, Krogh A, and Brunak S, “Genome organisation and chromatin structure in *Escherichia coli*”, *Biochimie*, **83**:201–212 (2001). [PMID: 11278070]
- Yamazaki S, Yamazaki J, Nishijima K, Otsuka R, Mise M, Ishikawa H, Sasaki K, Tago S and Isono K., “Proteome analysis of an aerobic hyperthermophilic crenarchaeon, *Aeropyrum pernix* K1”, *Mol Cell Proteomics*, **5**:811–823 (2006). [PMID: 16455681]

## Books on Bioinformatics

- Baldi P and Brunak S, “Bioinformatics – The machine learning approach” (MIT Press, Cambridge, Massachusetts, USA, 2nd Edition 2001).
- Claverie J-M and Notredame C, “Bioinformatics for Dummies” (Wiley Publishing Company, New York, 2003).

- Durbin R, Eddy SR, Anders Krogh, and Gaeme Mitchison, "Biological sequence analysis – probabilistic models of proteins and nucleic acids" (Cambridge University Press, Cambridge, UK, 1998).
- Gibas C and Jambeck P, "Developing bioinformatics computer skills" (O'Reilly & Associates, Sebastopol, California, USA, 2001).
- Korf I, Yandell M, Bedell J, "BLAST" (O'Reilley Media, Inc., Sebastopol, California, USA, 2003).
- Lund O, Nielsen M, Lundegaard C, Kesmir C, and Brunak S, "Immunological Bioinformatics" (The MIT Press, Cambridge, Massachusetts, USA, 2005).

Computing for Comparative Microbial Genomics

Bioinformatics for Microbiologists

Ussery, D.W.; Wassenaar, T.M.; Borini, S.

2009, XIV, 270 p. With online files/update., Hardcover

ISBN: 978-1-84800-254-8