2

Introduction to Problems and Challenges in Bioinformatics

2.1 Introduction

Chapter 1 provided an overview of the basics of molecular biology of relevance to bioinformaticians and also introduced some of the initial problems faced by researchers in the area. This chapter examines current and future challenges in bioinformatics. The problem areas and challenges are presented according to the field of molecular biology in which they occur: the genome, the transcriptome and the proteome. Also, the recently expanding area of gene silencing and interference technology will be covered.

2.2 Genome

Sequence analysis

Some of the earliest problems in genomics concerned how to measure similarity of DNA and protein sequences, either within a genome, or across the genomes of different individuals, or across the genomes of different species. DNA and proteins can be similar in terms of their *function*, their *structure* or their linear sequence of nucleotides or amino acids. The fundamental assumption for DNA is that two DNA sequences that are similar probably share the same function, even if they occur in different parts of the genome or across two or more genomes. The fundamental

Intelligent Bioinformatics: The Application of Artificial Intelligence Techniques to Bioinformatics Problems
Edward Keedwell and Ajit Narayanan
 ISBN: 0-470-02175-6

assumption for proteins is that linear sequence determines *shape* which, in turn, determines function. This is because the shape of a protein, and in particular of enzymes, determines which other molecules these proteins can lock on to and affect.

Consider the two DNA strings of equal length: **ACGTACGT** and **ACCTAGGT**. How similar are they? One way to deal with this problem is to place them one on top of the other:

A C G T A C G T
A C C T A G G T

A count is made column by column to identify the number of mismatches per position, which in the above case is two. This is the *Hamming distance*, which is the simplest measure of similarity available. The two strings **ACGTACGT** and **CCCTCCCT** would have a Hamming distance of four, and the two strings **ACCTAGGT** and **CCCTCCCT** would also have a Hamming distance of four. The two strings **ACGTACGT** and **ACCTAGGT** therefore are more similar to each other (Hamming distance of two) than **CCCTCCCT** is to either of them (Hamming distance of four). The problem is, what happens if strings are of unequal length? Consider **ACGTACGT** and **AGTACGT**. If these strands are lined up:

A C G T A C G T
A G T A C G T

the result is a Hamming distance of seven (assuming that the last base of the first string cannot be matched to a blank). Yet, if a blank is inserted in the second string:

A C G T A C G T
A - G T A C G T

the Hamming distance is one, i.e. the strings are very similar.

Now imagine that, instead of just eight bases in a DNA sequence there are hundreds and possibly thousands of bases (for example, if a whole gene is compared against other genes). Gene sequences are extremely unlikely to be of equal length, and methods must be found for inserting blanks at appropriate locations in the shorter string and stretching it out to optimize the number of matches. Shorter strings may result when the DNA replication machinery goes wrong and bases are skipped over.

Equally, some bases may need to be deleted. Consider the following three strings:

A C G T A C G T
A G T A C G T
A G G A C G T

One possibility is to insert blanks into the second and third strings at position two (two insertions) to line up the three strings. Another possibility is to delete the second base of the first string (one deletion):

A G T A C G T
A G T A C G T
A G G A C G T

Since one deletion may be preferable to two insertions this may be the preferred strategy, but now consider what would happen if the first two strings were matched without any knowledge of the third string. The strategy might well have been to insert a blank into position two of the second string to optimize similarity. However, when the third string is entered, it is now discovered that it would have been preferable to delete the second base of the first string rather than insert a blank into the third string. Backtracking may be required to undo the insertion of the blank into the second string, but backtracking will only work if there is stored information as to what was done earlier so that it can be undone. For long strings and for matching many strings, the memory requirements can quickly become large.

The above problem is easy with just a handful of strings and small numbers of bases, but already the problem with long and large numbers of sequences is apparent. There can be *pairwise comparison* of strings, where changes are made to earlier decisions as new strings are entered, or there can be *multiple comparison* of all strings at once and matches can be optimized for specific positions across all sequences. Also, there can be *local alignment* (finding alignments between parts of two or more sequences) and *global alignment* (finding an alignment for sequences in their entirety). There are now a number of publicly available tools on the web for undertaking alignments.[1,2]

The requirement for a minimal number of changes arises from the principle that, when identifying similarity between strings, as few alterations

[1] See, for example, http://www.ncbi.nlm.nih.gov/Education/ for a tutorial on Blast.
[2] See, for example, http://www.ebi.ac.uk/fasta33 for Fasta.

as possible should be made to the original strings so that optimal similarity measures are returned. This is the *unit cost* model, also known as the Levenshtein Distance, which states that the cost of an alignment of two sequences s_1 and s_2 is the sum of the costs of all the 'edit' operations required to match the two sequences, and that an *optimal* alignment of s_1 and s_2 is an alignment that has minimal cost among all the possible ways that they can be aligned. Extensions to the unit cost model include *substitution matrices* that provide variable costs for insertion, deletion and replacement of bases and amino acids, *realistic gap models* that prevent deletions and insertions in critical subsequences (such as strongly conserved subunits in protein sequences involved in protein–protein interaction, where any edit in these subsequences may destroy the desired biochemical function) and the use of an extended genetic alphabet that represents possible ambiguities in the data. The most common symbols used in an extended genetic alphabet are: **R** for **G** or **A** (Pu**R**ine), **Y** for **T** or **C** (P**Y**ramidine) and **N** (A**N**y).

A related problem here is how to find a common substring for all strings or sequences. This is known as the 'superstring' problem, where the common substring is the shortest sequence of characters shared by all sequences. This problem is, in computational terms, *intractable*, in that there is no known algorithm that will work in reasonable time to find such a superstring as the number of sequences and their length increase.

Phylogeny

Many algorithms now exist for sequence alignment, including Dynamic Programming (for both pairwise and multiple alignment) and the Carillo–Lipman method for optimal multiple alignment. The purpose of alignment is to learn about the phylogenetic and evolutionary relationships between genes with a similar function. For instance, a large number of sequences can be retrieved from a number of different genome or protein databases using a specific subsequence. Each database may store information on one or more organisms. The research task is then to discover the evolutionary relationships between these sequences and therefore the organisms on the assumption that evolution can be described as 'descent with modification'. That is, inherited similarities and differences between organisms provide the basic information needed to hypothesize evolutionary relationships between these organisms, where these similarities and differences are expressed in DNA sequences, amino acid sequences or phenotypic characteristics. The principle of *parsimony* in phylogeny

essentially states that derived similarities between sequences can be assumed to be caused by common ancestry and that inferences concerning these similarities should be kept as simple as possible.

Phylogeny and classification are important areas of biology, since they deal with the identification, naming and grouping of organisms based on shared similarities. Linnaeus introduced the 'binomial' classification system in the 18th century consisting of two Latin names, where the first name (always starting with a capital letter) denotes the *genus* and the second (always starting with a lower case) the *species* (as in *Homo sapiens*). While only two layers of taxonomy existed in Linnaeus' day, it is currently widely accepted that there are seven layers: *Kingdom, Phylum, Class, Order, Family, Genus, species*. The task of current phylogeny is to locate all organisms in a comprehensive classification scheme that reflects their evolution from a common ancestor believed to have come into existence about two and a half to three billion years ago on this planet.

To give an idea of the computational cost involved in such a comprehensive classification, imagine that all organisms have just five genes, each of which can take any number of alleles. Gene sequences can be compared base by base, as previously described, to identify similarities and differences between genes. Imagine also initially that there are just four organisms, each of which takes 1 s to compare with another organism across all five genes. To construct a set of similarity scores for these four organisms takes 6 s (3 s to compare organism 1 with organisms 2, 3 and 4; 2 s to compare organism 2 with 3 and 4; and 1 s to compare organism 3 with 4). If there are 10 organisms, the time taken is $9 + 8 + \ldots\ 1 = 45$ s. That is, to calculate similarity scores for n organisms takes $(n - 1)^*(n/2)$ s. The cost for 100 organisms is therefore 99*50 s = 4950 s, or 1 h 22.5 min. Note that the time taken for 100 organisms is not the same as 25 times the cost for four organisms. It is estimated that there are between 12 and 15 million existing organisms/species on this planet, with some claims that 99 per cent of species are extinct. To calculate similarity scores for 10 million existing species, given previous assumptions, would take 9 999 999*5 000 000 s, i.e. over one and a half million years. If this represents just 1 per cent of all species, it will take us over 150 million years to calculate similarities for all organisms that have ever existed. If it is argued that 1 s per comparison is far too long, given just four genes, it can be counter-argued that organisms contain more than just four genes, so even this figure will need amending upwards. Even if it is possible to calculate the similarities in a realistic amount of time, there is another problem which is the construction of the resulting

Table 2.1 A table of information indicating shared gene 'values' across four organisms. The gene values are assumed to be binary phenotypic values for the sake of exposition although in real life gene values can be expected to be much more complex, such as long strings of DNA, amino acids or multivalued phenotypes. '0' stands for 'ground state' and '1' for 'advanced state'.

	Gene 1	Gene 2	Gene 3	Gene 4
Organism A	0	0	0	0
Organism B	1	0	0	0
Organism C	1	1	0	1
Organism D	1	0	1	1

phylogenetic tree (a tree diagram that displays evolutionary relationships among a number of organisms or species).

Consider Table 2.1 and the four organisms with the four genes that they share. For the sake of simplicity, assume that each gene has only two phenotypic values, 0 and 1. The task here, however, is to demonstrate the complexity involved in generating phylogenetic trees for even this simple dataset.

The values for genes differ between different organisms through a variety of mechanisms. Mutations (that is, value differences) can occur through substitution (one nucleotide miscopied as another), insertions (new bases are added) and deletions (some bases are deleted altogether), resulting in different gene values, as in Table 2.1. The question arises as to whether, given the information in Table 2.1, any overall conclusions can be drawn as to how these organisms are related in evolutionary terms to each other.

There are two general methods for deriving trees from such tables. The first, called *Hennig Argumentation*, considers the information provided by each gene one at a time (i.e. it works column by column). The information in Gene 1 (advanced state value 1) unites B, C and D (Figure 2.1(a)), the information in Gene 2 (advanced state value 1) is peculear to C (Figure 2.1(b)), the information in Gene 3 (advanced state value 1) is peculear to D (Figure 2.1(c)), and finally the information in Gene 4 (advanced state 1) is shared between C and D (Figure 2.1(d)). A tree is obtained that evolves as the information is included column by column.

One interpretation of the tree is that all four organisms shared an ancestor in the past (first split in the tree), but that B, C and D split from A through the sharing of a specific value for Gene 1 (common ancestor for B, C and D), that C and D split from B through the sharing of a

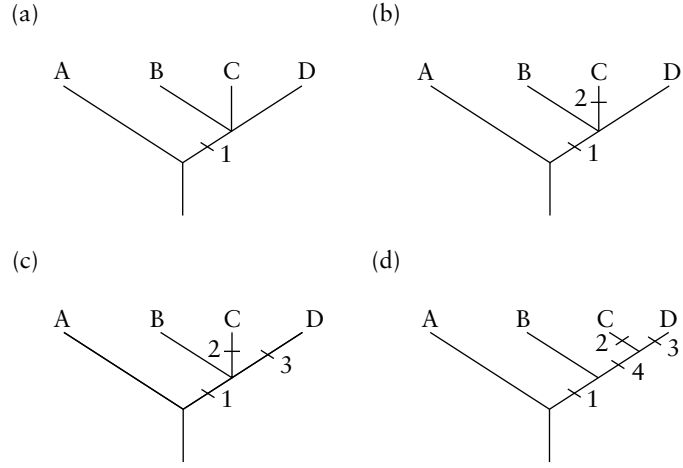


Figure 2.1 Hennig Argumentation considers the information provided by each gene one at a time

specific value for Gene 4 (common ancestor for C and D), and that C and D split from each other through the acquisition of specific values for Genes 2 and 3 (common ancestor).

Hennig Argumentation is simple but can lead to complex tree labelling when information from genes in subsequent columns conflicts with information already included from earlier columns. This can in turn lead to complex interpretations of phylogeny. For instance, if Gene 4 had united B and C rather than C and D, the label for Gene 4 would need to be moved to the same location as the label for Gene 1, and then explicitly an exception label must be inserted to signify that D does not share the value for Gene 4 (Figure 2.2). The interpretation now is that D reverted back to its original state with regard to Gene 4 after a common ancestor to B, C and D shared a common state for Gene 4.

Trees derived through Hennig Argumentation are therefore highly dependent on the first columns (genes) encountered and do not take the information in all columns into account before generating the first candidate phylogeny tree. Conflicts in subsequent columns can lead to many exception labels or even re-formatting the tree to minimize such exceptions. While the situation may not be too bad for a 'binary' gene value example, real gene values can be expected to consist of more than just binary states, and typically many more than four organisms will need to be related phylogenetically.

To overcome the problems of Hennig Argumentation, *Wagner Trees* can be used instead. Consider the information in Table 2.2, but this time a phylogenetic tree is going to be constructed organism by organism

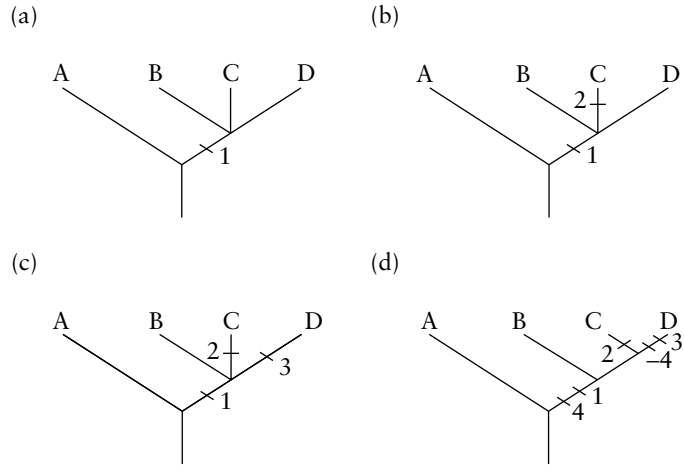


Figure 2.2 An alternative Hennig Argumentation

(row by row) rather than gene by gene (column by column), with the purpose of minimizing the number of state changes required. The first step in Wagner Tree construction is to find the organism that has fewest 'advanced' states, where 1 stands for 'advanced'. A has 0 values across all genes and therefore no advanced states.

A comparison is made between all the other organisms against A, with B having one derived or advanced state in comparison to A, while C and D have two and three derived or advanced states in comparison to A, respectively. B is linked to A first (Figure 2.3(a)) since it is most similar to A. The organism with the next lowest number of advanced states is then identified. Since C has two derived state differences, its name is written beside B and connected to the line that joins B and A (Figure 2.3(b)). At the point where the two lines intersect, the most advanced states present in B and C are listed (the intersection of state values is called an *optimization*). Since B and C both have a derived state for Gene 1 but do not share other derived states, the optimization is 1000, where the first

Table 2.2 A table of gene values for four organisms to demonstrate the Wagner method of phylogenetic tree construction

	Gene 1	Gene 2	Gene 3	Gene 4
Organism A	0	0	0	0
Organism B	1	0	0	0
Organism C	1	1	0	0
Organism D	1	1	1	0

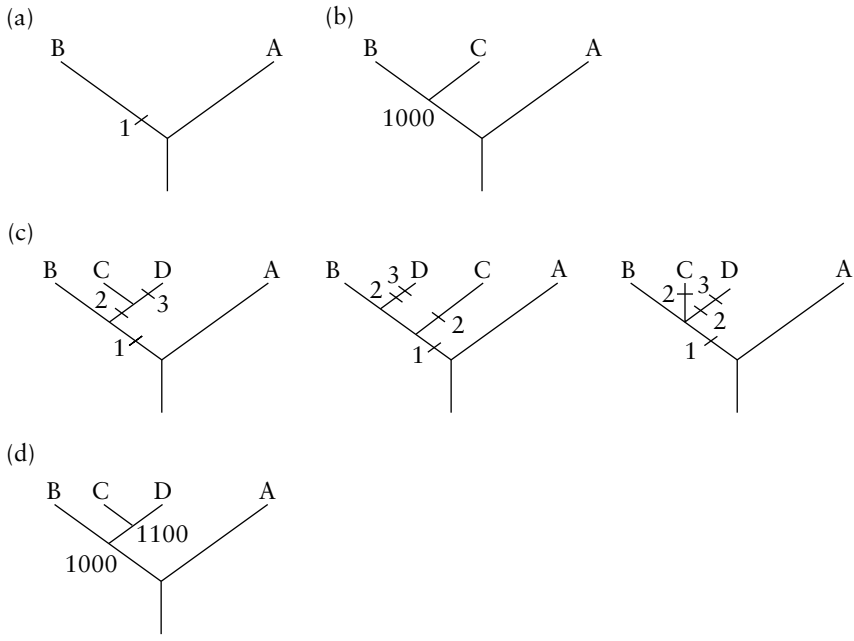


Figure 2.3 The Wagner method constructs phylogenetic trees by adding organisms one at a time based on the number of gene value differences between organisms

bit signifies Gene 1, the second bit Gene 2, and so on. Finally, D has to be linked into the tree and connected to a point that requires the fewest number of state changes. There are several possibilities, three of which are depicted in Figure 2.3 (c). Since the second and third possibilities imply that Gene 2 evolved twice, whereas the first possibility implies that Gene 2 evolved only once, the preferred most *parsimonious* tree (the first possibility) is adopted. An optimization is calculated and the analysis is complete (Figure 2.3 (d)).

To aid tree construction, an *outgroup* organism is usually used that has no shared characteristics (gene values) with any of the organisms to be classified but is nevertheless ancestrally related to the *ingroup* (the organisms to be classified). This outgroup is located in the tree first and acts as a basis for comparison as well as providing 'directionality' to the evolutionary sequence depicted by the tree. The *length* of a tree is the total number of steps or state changes in the tree, and a tree with a smaller length is to be preferred to a tree of greater length for the same organisms. Parsimony is essentially an optimality criterion, and several different methods now exist for calculating optimal tree structures, including Wagner optimality, Fitch optimality, Dollo optimality and Camin–Sokal

optimality. Building phylogenetic trees becomes complicated as datasets become larger or contain conflicts that have to be resolved, usually by re-formatting a tree. Optimality procedures usually work in a step-wise manner such that each organism is added where it optimally fits a tree, as in the Wagner method above. However, such *exhaustive* search methods that check all possible trees quickly become intractable as the number of organisms and genes grows.

The ultimate aim of phylogenetic analysis is to present a complete evolutionary history of all life on earth that shows how all organisms are related to each other, either existing or extinct. Advances in molecular biology have now allowed the use of genetic sequences (DNA or amino acid sequences) for tree construction, rather than the characteristic traits that were used in the past, since these sequences provide a more detailed and lower-level account of differences between organisms and species. In Figure 2.4, the top table describes the same stretch of DNA for the four organisms A, B, C and D. B, C and D differ from A in 3, 4 and 5 positions, respectively. B is joined first to A (Figure 2.4(a)) and the optimization is located where their lines join. The three differences between B and A are also described in the order in which the differences appear, working away from where the lines join. C is added next (Figure 2.4(b)) and again the three changes from B are described and the optimization provided. Only two possibilities for joining D are shown here in Figure 2.4(c). Since joining D to C requires fewer changes, this is the chosen tree (Figure 2.4(d)).

2.3 Transcriptome

As previously described in Chapter 1, the total collection of mRNA and their alternative splice forms represents the transcriptome of a cell or organism. The transcriptome can be considered the complete set of instructions for deriving all the different proteins found in a cell or organism. By analysing the transcriptome, it may be possible to discover new proteins that are present in specific tissues or produced only by certain cells under certain conditions. If the genome provides us with the complete set of genes of a cell or organism, and the proteome tells us all of the proteins that can be produced by the genome, the transcriptome is the bridge between the two. If there are more proteins than genes, something must be happening between the genome and proteome to make this possible. By measuring the transcriptome during certain cell development stages, it is possible to identify which genes are switched on or are switched off

A	**a**	**a**	**a**	**a**	**a**
B	**a**	**a**	**c**	**c**	**g**
C	**c**	**c**	**c**	**c**	**a**
D	**c**	**c**	**c**	**c**	**t**

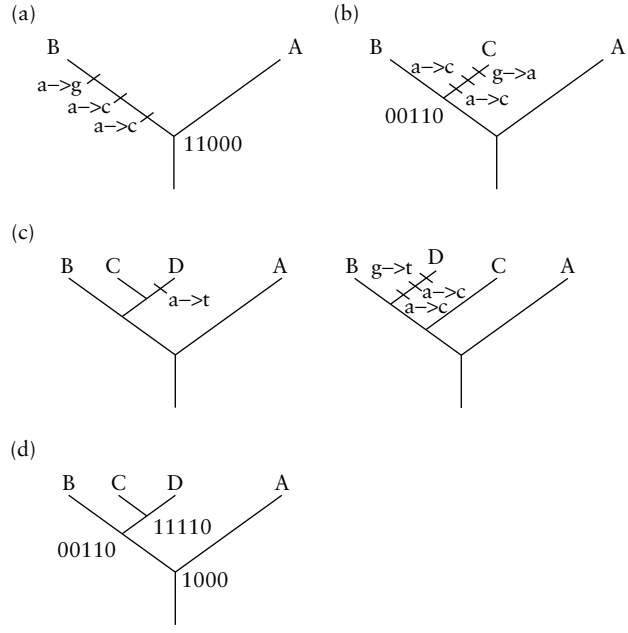


Figure 2.4 Constructing a phylogenetic tree from example DNA sequences for four organisms A, B, C and D, using the Wagner method

at various points during the process. Also, if the transcriptome can be measured during the development of stem cells, it may also be possible to identify exactly how and when genes are switched on and off so that the cells specialize to become one of the 200 or so different types of cell found in the human body. Such measurement will help answer one of the most profound mysteries in molecular biology, since there is no 'central control' of stem cell division that specializes cells. Specialization of cells must therefore be through some form of signalling pathway through genes.

Interest in the transcriptome (the total set of transcripts possible from the genome, including alternative splice variants) has grown significantly since the arrival of a new technology that allows us to measure both the amount and nature of these transcripts. *DNA arrays* are devices that contain DNA probes that allow complementary mRNA or complementary DNA (cDNA) samples to be bound to the probes. Assume for the

moment that the probes are short fragments of each gene that can be found in the genome of an organism, and that the mRNA or cDNA samples are taken from cells or tissues of that same organism under some condition. If the samples are applied to the DNA array and 'stick' to some probes but not others through complementary base pairing, that tells us which genes are expressed in the sample and which genes are not expressed in the sample (Figure 2.5).

The total mRNA from an individual (cell or tissue) is extracted and purified. Since mRNA does not remain stable for long, cDNA versions of the mRNA are reverse transcribed so that the mRNA and cDNA form a stable structure. The strands are then further amplified or transcribed to generate further cDNA or mRNA (called cRNA) strands before being 'labelled'. Typically, samples from one cell or individual are labelled green and samples from another cell or individual red to allow for differential comparison between the samples. The samples are then fragmented into smaller substrands, and the gene chip/microarray is applied. The gene chip/microarray will contain probe nucleotide sequences that uniquely

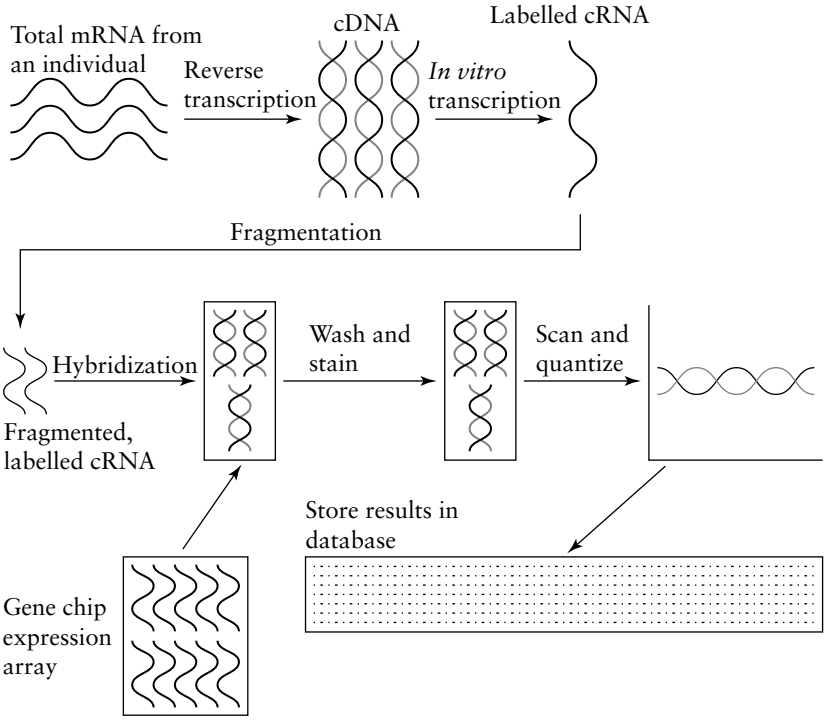


Figure 2.5 Microarray and gene chip measurement (see http://www.affymetrix.com)

detect the presence of its cDNA or mRNA counterpart, if it is present in the sample. The samples are washed over the gene chip/microarray and allowed to 'hybridize' (form short complementary base pairings) with the probes. The gene chip/microarray is then 'read' with a laser that is tuned to measure probes hybridized with green or red samples. If both samples contain equal amounts of the same mRNA/cDNA, the probe will fluoresce an orange/yellow colour. If one of the samples contains more of one form of mRNA/cDNA than another, it will fluoresce either green or red, depending on which sample it came from. If there are no mRNA/cDNA samples for a particular probe, the probe will reflect black or the background colour of the gene chip/microarray. Because the laser reads probes at a certain frequency, the intensity of reflected light can be converted into measures of amount of mRNA/cDNA and stored in a database for further analysis.

There are two main types of DNA array: *microarray* and *DNA* or *gene chip*, depending on how probes (nucleotide sequences) are put onto the chip. Microarrays use presynthesized DNA (about 100 bases) for probing, whereas DNA chips use *in situ* synthesized oligonucleotide probes (25 bases for Affymetrix gene chips). More recently, types of array are distinguished by the amount of genes that can be measured, since DNA chips allow for increased numbers of probes due to their shorter length (between 30 000 and 4 million probes for DNA chips, as opposed to about 20 000 probes for microarrays). Microarrays generally use *spot* technology, whereby a robot places spots (roughly 0.1 μm to 0.5 μm) of DNA on a glass slide (the microarray) and each spot is a DNA counterpart to one of the mRNAs to be measured. These DNA spots act as probes and are generally between 100 and 200 bases long. The advantage of this method is that specialized microarrays can easily be fabricated to search for specific genes. However, given the size of the spots, there are limits on the number of probes that can be put onto one spot of the microarray. For this reason, the use of smaller probes is generally preferred, and these are put on the chip using photolithographic techniques adapted from semiconductor technology. The probes are built 'bottom-up' and in parallel in the same way that circuits are, so that nucleotides are added to multiple growing chains simultaneously. A 'spot' ('well' or 'cell') on a gene chip can contain a thousand probes for one specific gene.

After the mRNA samples (control and experiment) are reverse transcribed into cDNA, labelled (dyed) and allowed to hybridize with the probes on the microarray or gene chip in the form of cRNA, lasers are used to produce an emission signal for each dye. It is not yet possible for computers to be linked directly to gene chips and microarrays so that

the amount of mRNA in a sample can be read directly from the probe cells. DNA probes and mRNA fragments are far too small to be read in this manner. Instead, the array or gene chip has to be converted into a fluorescent image which is sufficiently detailed at the pixel level to allow inferences to be made about the quantity of sample in a cell. Confocal array scanners are currently the most popular method of measuring the fluorescence. A gene chip probe cell is currently beween 25 μm and 50 μm, and pixel sizes used by confocal lasers are about 5 μm. Confocal lasers can therefore produce six-by-six or eight-by-eight pixel images of a gene chip well or spot. Each pixel will have a certain colour attached to it, and the overall 'colour' of the spot or cell is determined by the colour of the individual pixels making up the spot. For instance, if two colours are used (say, red for experimental mRNA sample and green for control mRNA sample), and cRNA of both samples hybridize with the probes of a cell, all pixels will give off a yellow/orange diffraction pattern. If, however, mRNA of only one sample is present and hybridizes with the probes in a cell, a diffraction pattern which represents red or green will be produced which is broken down by the pixel matrix (Figure 2.6 (1)). The outermost pixels are removed from analysis and the intensity of pixels plotted to arrive at an average intensity value for the cell as a whole to determine whether enough sample is present in a cell.

Quantitiation (converting fluorescence intensities into amounts of sample) usually results in large numbers that are conventionally converted into $\log_2$ ratios. For instance, if after laser analysis there are 200 transcripts of red cRNA for a gene and 10 000 transcripts of green cRNA, $\log_2(10\,000/200) = 5.64$. If the expression values are identical, the result is 0. Minus $\log_2$ values would signify more red cRNA than green. Such $\log_2$ ratios are easier to work with as well as provide absolute values, even if they have to be subsequently normalized to overcome skewed frequency distributions. Interpreting $\log_2$ ratios can, however, be difficult. Also, determining how reliable both $\log_2$ ratios and raw intensity values are is difficult. Different amounts of the two samples and of labelling concentrations may have been used, for instance, which will affect the quantitiation process.

Alternatively, Affymetrix gene chips use a *perfect-match/mismatch* strategy to help identify the reliability of the readings as well as produce an *absolute call* value for each gene which expresses whether the gene probed for is 'present', 'absent' or 'marginal' (Figure 2.6 (2)). Affymetrix use two types of probe in a cell: a 25-nucleotide sequence which is identical to a fragment of a sample mRNA and a 25-nucleotide sequence which is identical to the probe except that the middle base is different. If the

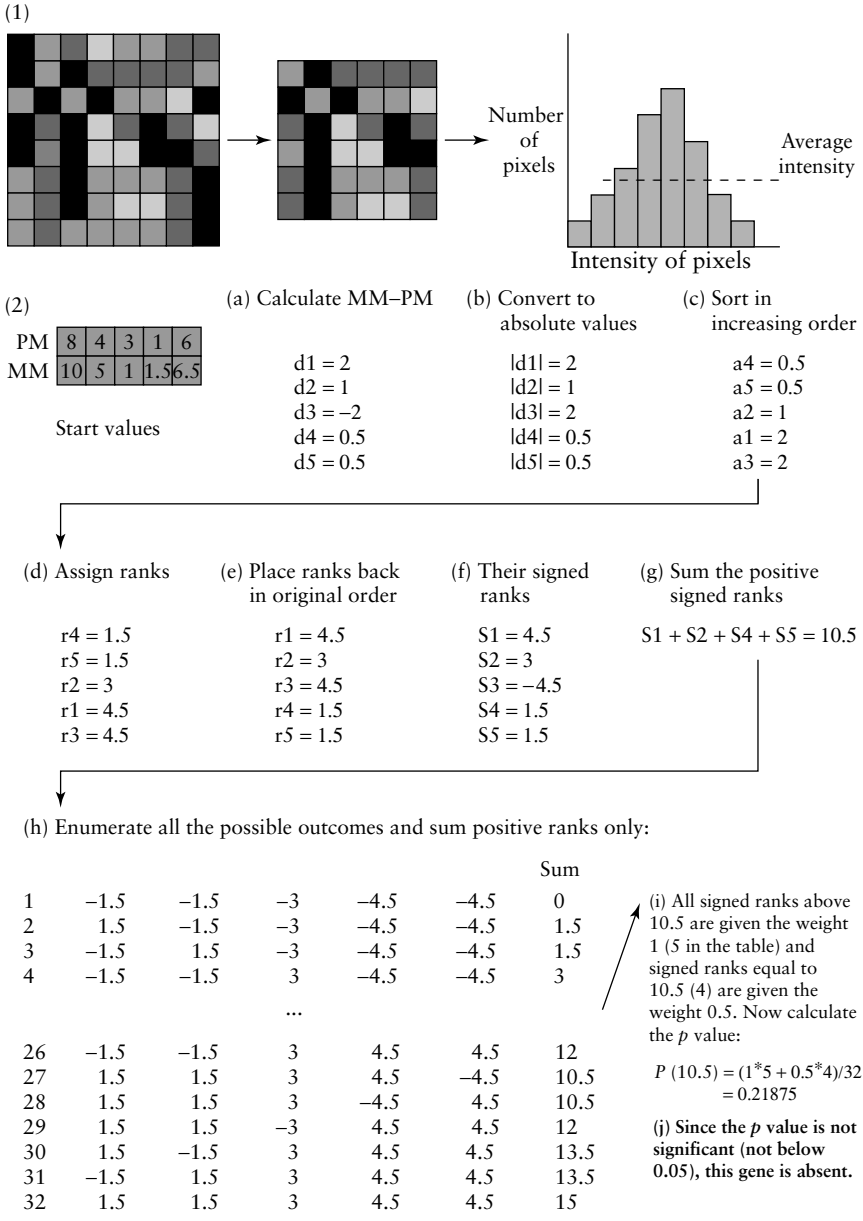


Figure 2.6 Affymetrix gene chip technology

base in the middle of a probe sequence is not complementary to the base in the middle of the sample sequence, the repulsion forces between just these two bases should be sufficient to ensure that the sample sequence does not hybridize with the probe sequence. Mismatch probes therefore

allow for checks on non-specific cross-hybridization in the sample. That is, outside of the human body mRNA nucleotides are not always guaranteed to bind to their complementary base pairs, due to heat differences and degradation, for instance. These mismatch probes are also used to generate absolute call values in that the fewer mismatches there are, the more confidence one has in the accuracy of the perfect matched figures. In Figure 2.6 (2) a gene is probed across several 'probe pairs' (typically 10–15 on Affymetrix gene chips), where each pair is made up of 'perfect-match' probe sequences and 'mismatch' probe sequences. To determine whether a gene is present in a sample, Wilcoxon's Signed Rank Test is used. Imagine there are five probe pairs for a gene (each probe pair consists of a perfect match and a mismatch beneath it) and the values are as indicated in Figure 2.6 (2), where these values represent the number of samples hybridized in each of the cells. The first step is to calculate the difference between each pair (a), followed by a conversion to absolute values (b), which are then sorted and ranked (c, d). The ranked values are placed back in their original order (e) and re-allocated their signs (f). The sum of the positive signed ranks is calculated (g) and a full enumeration of all possible signed outcomes is listed (h), with only positive ranks summed. All signed ranks above the sum calculated at step (g) are given the weight 1 and equal to the sum the weight 0.5 (i). The p value is then calculated as the sum of the weighted values divided by the total number of enumerated outcomes. If the value is below 0.045 a value of 'present' is attached to the gene, if the value is above 0.055 a value of 'absent' is returned, and otherwise 'marginal'.

Gene chips now exist for measuring the expression levels of all genes in the human genome. They can also be used to check whether genes are being expressed in specific tissue and which genes are expressed in response to drugs. One particular application of gene chips and microarrays is in the identification of single nucleotide polymorphisms (SNPs) that express common genetic variances among people, caused by a single nucleotide change every 300 bases or so in both the coding and non-coding parts of the human genome. For a nucleotide change to be an SNP, it should occur in at least 1 per cent of the population, and it is believed that, while SNPs do not affect the normal function of cells, they do affect the way that individuals react to drugs or predispose individuals to certain diseases. Microarrays and gene chips can be purpose-designed to identify SNPs and detect their presence in individuals.

While DNA arrays and gene chips are among the most exciting genomic tools to have been developed within the last few years, it has to

be remembered that mRNA levels do not always correlate with protein levels. It is not currently known how much mRNA actually makes it to protein.

Alternative splice variants of genes that are not measured on a DNA chip mean that a gene may not be accurately measured. Also, DNA chips cannot identify post-translational modifications of a protein. However, perhaps the biggest problem with DNA chips concerns current gene expression analysis techniques. The sheer volume of data (gene expression datasets can be several megabytes) leads to the need for fast analytical tools; but more importantly, there are many more attributes (genes) than records (samples). Typically, 12 000 to 25 000 genes are measured for each sample (subject or individual), and only 50 to 100 samples are collected. In database terms this leads to a hugely sparse data space. Gene expression analysis (G) can be defined to be concerned with selecting a small subset of relevant genes from the original set of genes (the S problem) as well as combining individual genes in either the original or smaller subsets of genes to identify important causal and classificatory relationships (the C problem). That is, G = S + C. In later chapters it will be shown how artificial intelligence techniques are making promising progress in analysing gene expression data and mining the data for useful knowledge.

The analysis problem becomes even more acute when dealing with temporal gene expression data, i.e. the repeated application of DNA chips to measure the transcriptomic state of an individual over time. So far it has been assumed that DNA chips are used to measure an individual just once and that the database will consist of several samples, measured once, where each sample falls in a clearly designated and independently observed class (e.g. a cancerous sample versus a normal one). Imagine that an individual cancer patient's mRNA is measured at time 0 and then a drug added which, it is believed, will 'cure' the patient. The individual's mRNA is measured after 30 min, then 1 h, then 2 h, then 4 h, etc., to see how the drug is affecting gene expression of the immune system and whether cancerous cells are being targeted for attack by the immune system. What is of interest here is the network of gene activation over time, as expressed not just for one patient but for several patients. The task is to 'reverse engineer' this gene network from not just one dataset but several. Reverse engineering means identifying which genes at one time point affect which other genes at the next time point. Given the large numbers of genes measured, if each gene is allowed to affect every other gene, a search space will rapidly be generated that is too complex for

computers to analyse. If a gene at one time-step is restricted to affecting only five other genes at the next time step, or a gene at a subsequent time-step to be affected by only five other genes at the previous time-step, the question is how to identify just these small numbers of affected or affecting genes from the huge number measured. Reverse engineering gene networks from gene expression data, where there is confidence that the correct causally influencing and causally influenced genes have been identified, is one of the biggest unsolved problems in bioinformatics.

Ethical considerations

There is also an ethical dimension to gene expression analysis. First, measuring the gene expression of an individual gives us information on not just that individual but also that individual's closest relatives. So while an individual may well permit their gene expression to be measured and a genetic profile for that individual to be stored in a database, there are fundamental questions about the rights of that individual's relatives to have information about their genetic profiles not stored in a database. Identifying through gene expression analysis that an individual has a predisposition to a particular inheritable disease provides information about other members of that individual's family. Secondly, while it may be acceptable to measure the gene expression of individuals who are suffering from a disease, there are fundamental questions concerning the scope of gene expression analysis. Should embryonic stem cells be monitored for gene expression, for instance, so that important information is obtained about how cells are differentiated during the early stages of fetal development? One of the most puzzling of all mysteries in biology is the way in which, from one fertilized cell, a multi-trillion cellular organism called a human results, where billions of cells have somehow 'agreed' to express only certain genes that allow them to form tissue and cooperate with each other. The fertilized cell and its daughter cells after initial division within a few hours are *totipotent*, i.e. they have the ability to become any cell in the body. After about four days some of these cells become a blastocyst (hollow sphere) and have lost their totipotency, whereas the other cells inside the blastocyst form an inner cell mass. These inner cells are *pluripotent* in that they have the ability to become one of several different types of cell. After further division pluripotent cells become *multipotent*, whereby a multipotent brain cell, for instance, has the ability to become any one of the different types of brain cell (Figure 2.7).

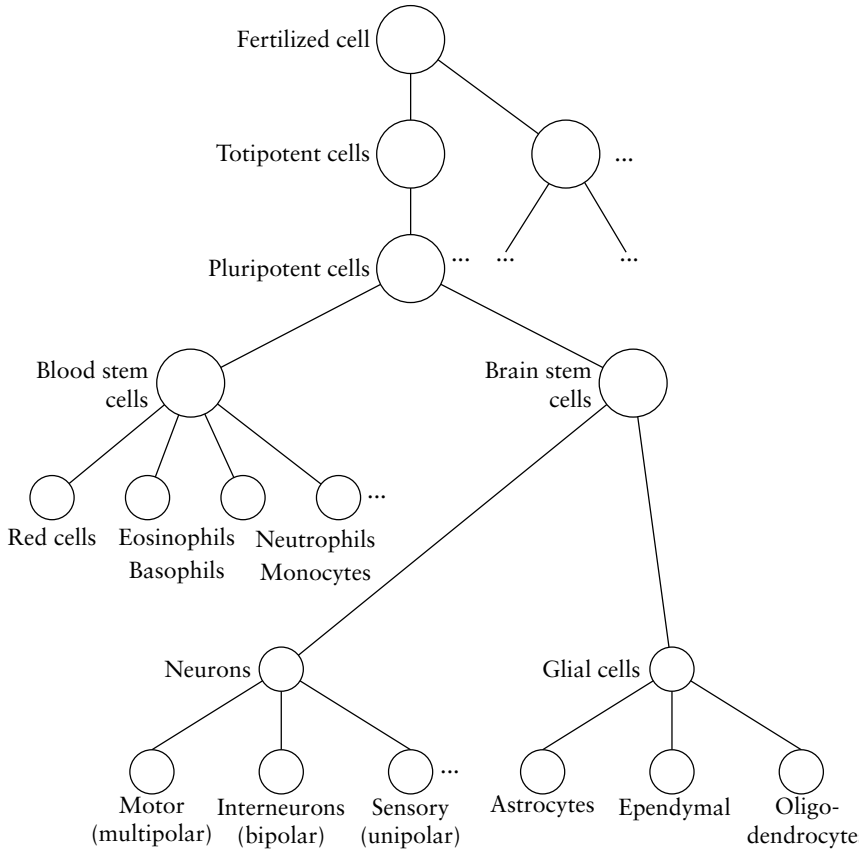


Figure 2.7 Embryonic stem cells

Currently there are two methods for developing pluripotent cells: from inner cell mass at the blastocyst stage and from fetal tissue from terminated pregnancies. While DNA chips provide an unprecedented opportunity to measure early gene expression (within a few hours of conception), this may mean that embryos are 'farmed' for research purposes. The promise of stem cells lies in their possible ability, when located next to damaged tissue, to become one of the cells in that tissue by expressing the same genes as those expressed in the tissue. The mechanisms whereby this happens are not known, but the potential to repair parts of the body where cells no longer divide in sufficient numbers to overcome damage (such as the brain or liver) is huge. However, before stem cells can be used there needs to be an understanding of their gene expression and differentiation mechanisms. Different countries are taking different ethical and legislative stances on this important ethical topic.

2.4 Proteome

Secondary and tertiary structure prediction

Proteins are the end result of translation of mRNA by ribosomes. Once protein sequences of amino acids leave the ribosomes they fold in complex ways to achieve a 'native' state or conformation in the cell. The native state of a protein is a highly stable three-dimensional structure that helps determine its biological function. In other words, a protein cannot function unless it folds in the right way. For instance, catalytic proteins must fold in such a way that they can lock onto another molecule (substrate), thereby lowering the energy threshold required to start a reaction in the substrate. Once the reaction takes place, the catalytic protein is released to find other molecules to attach to so that further reactions can take place. If the catalytic protein misfolds, it will not be able to start the catalytic reaction. In particular, the *active site* of the protein which locks onto the appropriate section of the target molecule (the *substrate*) to start a reaction may not be revealed and so the protein cannot function.

Protein misfolding is associated with several diseases, and to understand the nature of the disease at the molecular level involves understanding the way that amino acids both locally and distantly affect the folding. That is, while it may not be possible to predict how a specific sequence of amino acids folds locally, once it folds it comes into contact with other regions of amino acids elsewhere in the sequence. Folding is determined by the chemical and physical properties of the amino acids making up the protein, but such chemical and physical explanations of folding have to take into account 'long distance' relationships between different parts of the same sequence. Determining the way that proteins fold into specific shapes is called the 'protein folding problem'. Laboratory experiments have shown that if a protein is gently denatured (that is, unfolded by, say, raising the temperature or changing the salt concentration of the surrounding fluid) and then allowed to refold, it resumes its original structure, thereby demonstrating that the ability of the protein to fold into its correct shape is intrinsic (all the information required to fold a protein is in the protein constituents).

While one obvious use of computers in bioinformatics is the storing of DNA sequence information and constructing the correct DNA sequences from fragments identified by restriction enzymes (enzymes which break up the DNA at certain points), protein sequences and the polypeptide[3]

[3] The term 'peptide' is used to refer to short sequences of amino acids, while the term 'polypeptide' refers to sequences of length 50 or more.

sequences that make up that protein also need to be stored. New protein sequences are being added to protein databases as a result of analysing mRNA sequences, where redundantly transcribed DNA (introns) have been removed, and by translating codons via the genetic code into letters of the amino acid alphabet. However, these linear sequences of amino acids (polypeptide sequence) do not tell us anything about the *structure* of the protein or how it folds. The protein folding problem is important because it takes a lot of effort to determine the structure of an actual protein. A real protein has to be to denatured (unfolded) so that its amino acid sequence can be described, but denaturing a protein *and* sequencing its amino acid content are much more difficult than simply denaturing a protein. In the act of denaturing the structure of the protein is affected so that information is lost about the structure as amino acids making up the protein are sequenced. Identifying the structure of a protein requires complex measurement, typically through X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy techniques, neither of which may be readily available to biologists. In any case, not all proteins are susceptible to crystallization, and NMR is constrained to deal with small proteins because of the computational costs involved in trying to model complex proteins. Finally, to determine the structure of a protein means removing it from its natural environment – the cell or organism. There is no guarantee that a protein being experimentally investigated *in vitro* will have the same structure as *in vivo*. As a consequence, the number of experimentally determined protein sequences is far fewer than the number of protein sequences that have been 'translated' by a computer from DNA and mRNA sequences.

The structure of a real protein is conventionally described in four ways (Figure 2.8). The *primary* structure of a protein (Figure 2.8(a)) is the sequence of amino acids produced at ribosomes. Since there are 20 amino acids, the primary structure describes the precise order of amino acids in the protein. The *secondary* structure of a protein (Figure 2.8(b)) describes those parts of the primary structure (subsequences of amino acids) that fold into regular and repeated patterns, such as α-*helices*, β-*sheets*, or *turns* (see Figure 2.9 for conventional computer-generated graphical ways of describing secondary structure). The *tertiary* structure (Figure 2.8(c)) consists of those elements of the secondary structure that build more complex units, such as an $\alpha-\beta$ *motif*, and provide a three-dimensional shape of the protein. The tertiary structure of enzymes is typically a compact, globular shape, for instance. Finally, many proteins consist of more than one polypeptide chain. The *quaternary* structure of a protein (Figure 2.8(d)) is a description of how several separate polypeptide sequences have come together to form a complex protein.

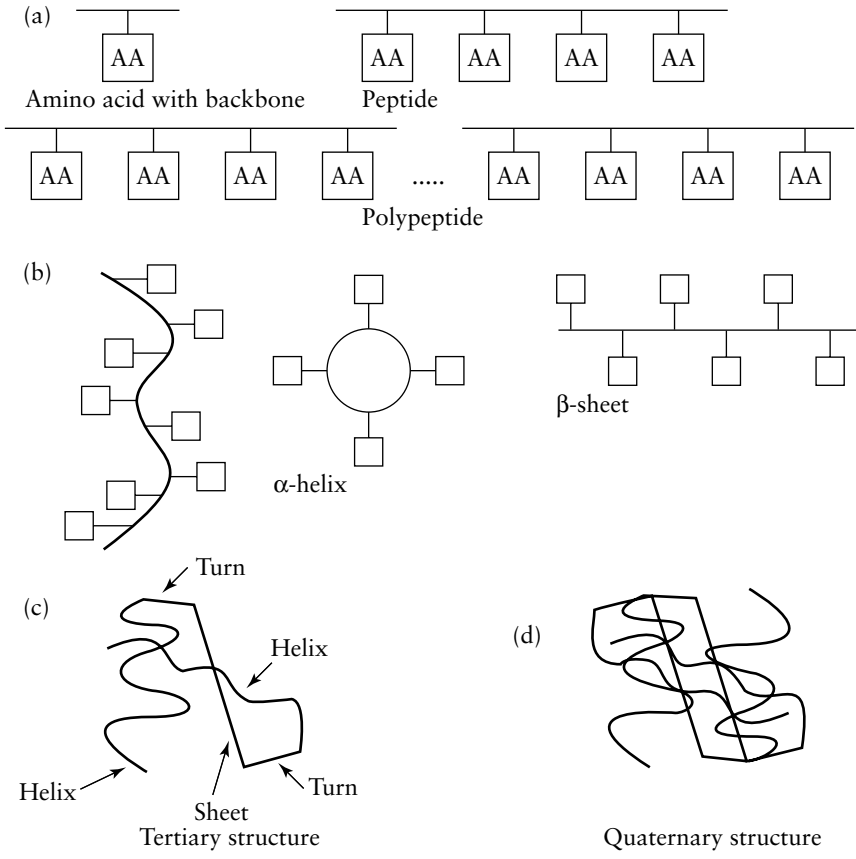
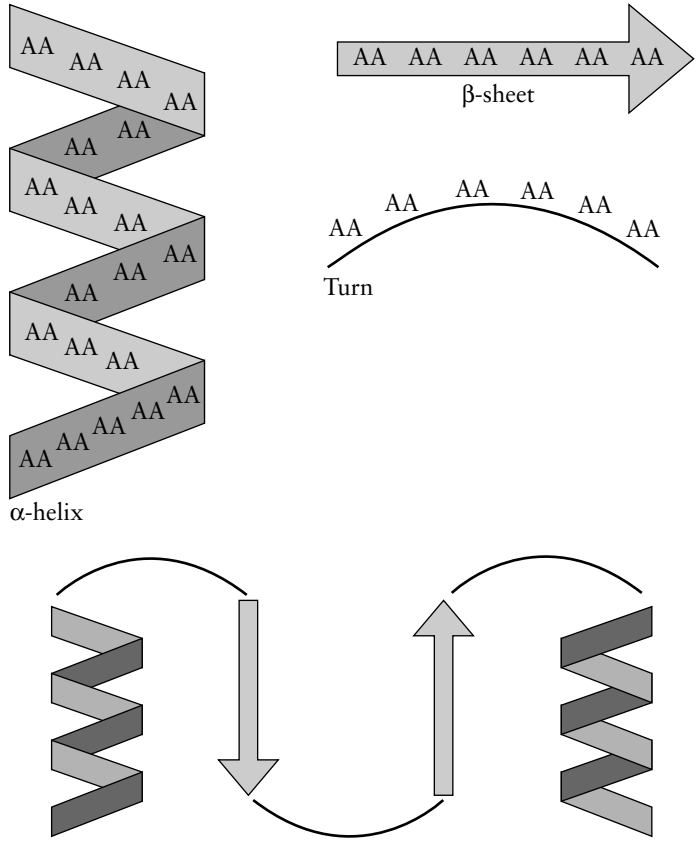


Figure 2.8 Protein structure

For instance, human haemoglobin consists of four separate polypeptides that come together to form a complex molecule that takes up oxygen from the lungs and delivers it to the cells of the body. These four peptides result from the translation of four separate genes. For experimental biologists, identifying all four levels of structure from an actual protein is very difficult, since not all parts of the protein are available for analysis. A real protein has to be dissected into smaller parts so that amino acids hidden by folds are revealed. There is therefore a great need to work from primary structures of proteins (as revealed by mRNA) to the three-dimensional and quaternary structure of the protein. Currently, this task is proving a great challenge to computer scientists because of the complexity of predicting secondary, tertiary and quaternary structures from primary structures. Folding arises because of basic charges (attraction and repulsion) of atoms and molecules, and modelling these for long sequences of amino acids is proving difficult.



Secondary structure consisting of α-helix, turn, β-sheet, turn, β-sheet, turn, α-helix

Figure 2.9 Computer visualization of a secondary structure

There are currently three approaches to protein folding prediction. *Comparative modelling* (also known as modelling by homology or knowledge-based modelling) uses structural data from experimentally determined protein sequences. An amino acid alignment is first made between protein sequences with unknown structure (typically derived from DNA or mRNA translation by the computer) with protein sequences with known structure. Then where the alignment agrees, the conformation of the sequence with known structure is allocated to the sequence with unknown structure for that part of the alignment. The main problem with comparative modelling is that there can be significant similarities between two proteins with known structure where the structures are significantly different from each other. Similarity of primary sequence is therefore no guarantee of similarity of structure and therefore of function. Similarly,

there are also examples of two proteins with known structure with similar function where the primary sequence information is significantly different in each sequence. Typically a threshold value of 30 per cent sequence identity is required to be exceeded before two sequences are considered homologous for modelling. While this figure may appear to be low, the argument is that the three-dimensional structure of proteins is conserved to a greater extent than the primary sequence. That is, a high degree of primary sequence similarity between a protein with known structure and a protein with unknown structure is not needed, since the function of a protein, as given by its structure, is more likely to be preserved through its shape than its amino acid sequence. The homologies being searched for are assumed to reflect structurally conserved regions of the protein.

Fold recognition, or *threading*, techniques are similar to homology modelling techniques but use a database of proteins with known structure and folds (called templates) against which to compare the protein of unknown structure. A scoring function is used to rank the folds and the folds with the best scores are then adopted for the protein with unknown folds and structure.

The final method is *ab initio*, where a structure is predicted for a protein with unknown structure by using physical principles of folding. One of the main assumptions of this method is that the native structure of a protein reflects its global free energy minimum, and the task of *ab initio* methods is to search the space of possible conformations of the amino acids (residues) making up a sequence to find optimal conformations that achieve low energy levels. While some *ab initio* methods work at the atomic level for residues, in practice residues are modelled using only a few interaction centres within the residue. Many molecular dynamics optimization methods now exist, using lattice-based enumerations and diffusion equation methods. The basic approach is to minimize the energy of the system, add a structural change, minimize the energy, add a structural change, and so on. *Ab initio* methods may have to be used when suitable template structures are not available.

Problems with *ab initio* methods include a minor conformational change at one residue having major implications for the entire sequence, which may not be captured by the simulation models used. For instance, a bond between two residues may be rotated for local minimization of energy, but given that the structure as a whole is three-dimensional there may be unfavourable effects on the whole structure that cannot be captured by the simulation. Also, the complexity involved in predicting the structure of a large protein may be too hard for a computer. Nor has it

escaped the attention of some researchers that large proteins naturally fold within seconds of translation, whereas computer models take hours or even days to predict the structure of less complex proteins.

Protein folding is perhaps the biggest problem in bioinformatics currently. Even if good techniques and methods for predicting the structure of proteins from primary sequences are discovered, this may not reveal anything about how the biological function or activity of that protein is carried out. There is also increasing interest in the actual stage-by-stage process by which a protein naturally folds to identify causes of misfolding. Current protein folding methods may not actually reflect this natural folding process. Yet there is increasing evidence that many diseases, such as Alzheimer's, cystic fibrosis, sickle cell anaemia, bovine spongiform encephalopathy (BSE) and its human equivalent Creutzfeldt–Jakob disease (CJD), are due to misfolding. It is currently estimated that of the several hundred thousand protein sequences stored in databanks (derived from DNA and mRNA), only about 1 per cent have an experimentally determined structure. As genome projects provide increasingly more protein sequences in their databases, this mismatch between proteins of known structure and unknown structure is bound to grow. *In silico* methods of accurately predicting the structure of proteins are still at an early stage of development and present one of the most profound challenges in bioinformatics.

Protein identification

Another current challenge in bioinformatics is to determine how large the human proteome (the total collection of all proteins produced by the genome) actually is. While many prokaryotic cells have small numbers of genes in comparison to the human cells (about 5000, typically), there is little evidence of significant alternative splicing. However, post-translational modification of proteins as they emerge from the ribosomes may increase the number of proteins so that anywhere between 10 000 and 20 000 proteins are actually produced by a prokaryotic cell. For a human (eukaryotic) cell containing 30 000 genes, it is currently estimated that each gene can be alternatively spliced anywhere between three and 100 times. Even assuming the lower figure, that gives about 90 000 different polypeptide sequences. However, several different types of post-translational modification can be carried out, such as cleavage of polypeptide sequences at different points to give different proteins,

including removal of the initial methionine residue. Many proteins are *inactive precursors* that are activated under appropriate physiological conditions. Their task is to be present in the body should a situation arise when they are suddenly required, for instance, enzymes for forming clots in the blood in the case of a wound. Such *proproteins* are typically activated by the removal of certain amino acids at the ends of a protein, allowing the protein to function by revealing the active site of the protein. The task of proteomics is to identify not just all the different proteins that can be produced by a genome but also to detect those proteins that are associated with disease because of misfolding of proteins or different amounts of protein.

The biggest problem for proteomics currently is a suitable technology for measuring the variety and abundance of protein in a cell or organism. The most common form of measurement is *protein electrophoresis*. Proteins have an electrical charge, and the basic method is to place all protein from a sample on a gel and apply an electrical current to the gel so that the proteins move to different parts of the gel depending on their electrical charge; they then form bands that indicate the relative proportion of each protein fraction. Proteins are separated because at some point in the migration there is no net charge, and the protein is then stationary. While this form of measurement is appropriate when comparing different samples, the technology does not allow for the individual identification of proteins in a sample. Also, small proteins move through the gel more quickly than large proteins and may end up in regions of the gel that cannot be measured accurately because of smearing or distortion. Many proteins also react unpredictably with the gel and may migrate to wrong parts of the gel matrix. Gel electrophoresis also requires a great deal of expert human manipulation, leading to increased possibility of error. However, automated protein identification techniques using gels are increasingly appearing on the market. Nevertheless, gel-based techniques by themselves may not be sufficiently accurate to identify individual proteins.

New techniques being explored currently for individual protein identification include peptide-mass fingerprinting and peptide sequencing. The former uses *proteases* (special proteins that cut other proteins) to dissect specific proteins into fragments that have a unique 'fingerprint' when subjected to NMR spectroscopy techniques. The correct identification of these fingerprints requires access to a database of protein fragments and their signatures under specific NMR spectroscopy conditions. However, as more proteins and their fragments are included in such databases, the

chances of finding unique fingerprints begin to worsen! Ideally, it would be helpful if a protein could be sequenced in the same way that a gene can be sequenced (through complementary base pairing techniques). Amino acids do not have complements, however. Peptide sequencing attempts to identify the amino acids of a protein or protein fragment either by working from one end of the fragment (*terminus* sequencing), one residue at a time, by cutting the residue from the sequence and then using complex methods for identifying the residue that has been cut off, or if the terminus is not visible by cutting the sequence into a number of fragments and then identifying each residue, as before (*internal* sequencing). Again, NMR or other mass spectrometry techniques are used for identifying residues, and many biologists do not have easy access to such facilities. Also, fragmentation processes are not sufficiently advanced to ensure that a protein is cut at the correct locations.

High-throughput peptide sequencing analogous to nucleotide high-throughput sequencing is a fundamental requirement for identifying novel proteins and novel ways in which proteins are translated from their mRNA sequences. The future bioinformatics problem, once high-throughput protein identification techniques are made available, is to map the actual proteins and their sequences found in cells with genome databases. Given the variety of alternate splicing of mRNA and post-translational modifications, the identification of exactly which gene is the source for which protein sequences is not likely to be an easy task, especially given the redundancy in the genetic code (several different ways of DNA mapping onto amino acid).

2.5 Interference technology, viruses and the immune system

Interference technology

Proteomics is considered one of the most important ways of understanding gene function. That is, even if a gene is fully sequenced and located on a chromosome, this does not mean that there is a full understanding of the gene unless it is known what its translated products do. So even if there is full knowledge of a genome and full knowledge of all the proteins derivable from that genome, a full understanding of the genome and proteome will only come with a detailed understanding of how genes

affect other genes through proteins, of how proteins affect other proteins. While the genome is static, in that once it is characterized it can be assumed to be constant, the proteome is dynamic and reflects the state of the cell and the conditions under which it survives. Some proteins are produced only when the cell's environment is stressed (e.g. by heating). It is possible that there is a specific stress gene for that condition that only comes on when the stress condition is apparent, but it is also possible that the cell deals with the stress either by producing more quantity of a protein or by modifying a product of an already expressed gene. One way to study the effects of proteins is through 'knock-out' technology that effectively silences genes. If genes can be silenced under controlled conditions, the effects of the absence of the gene on the proteome can be studied. While one method for silencing genes is to look upstream of a gene and at its transcription regulatory elements to see if promoter and enhancer regions can be blocked, not enough is known currently about these regions to determine effective gene silencing mechanisms at the transcriptional level. However, *interference* technology provides a mechanism for regulating the translation of mRNA even if transcription takes place.

Antisense technology is an mRNA interference technology that blocks the translation of 'sense' mRNA (see Figure 1.5) and is based on the idea of introducing an antisense gene or antisense RNA into cells. The effect of antisense technology has been known for over 20 years but its mechanisms were not understood. Introducing a short piece of antisense RNA, that is, a sequence that is complementary to part of an mRNA sequence, produced the obvious result that the gene giving rise to the mRNA was silenced due to its mRNA being partly double-stranded when the antisense RNA paired with the appropriate sequences of complementary bases in the transcribed mRNA. Such double-stranding was assumed to prevent the ribosomes from effectively translating the sequence of amino acids in the mRNA. In other words, it was assumed that the ribosomes 'jammed' when the mRNA transcript was found to contain double-stranded codons rather than the linear sequence of single-stranded codons expected. However, it was also found that introducing a *sense* RNA subsequence (that is, a subsequence that is identical to part of the mRNA) produced the same silencing effect. Sense RNA cannot pair with sense mRNA, since the bases are identical. Finally, it was also discovered that introducing a small section of *double-stranded RNA* was more effective at silencing the target gene than introducing either a sense or antisense RNA strand. To understand the mechanisms at work, viruses and the immune system will need to be explored.

Viruses and the immune system

A virus is not a living entity or cell, since it lacks many of the essential components of a cell, such as translation machinery and cellular transport systems. It is between 20 and 100 times smaller than a typical single cell organism and attacks all types of cell or organism. Viruses that attack bacteria are called *bacteriophages*. A virus is a piece of genetic sequence (either DNA or RNA) with some proteins, wrapped up in a protein coat (*capsid*) and with the ability to recognize specific prokaryotic and eukaryotic cells through sites on the capsid that are complementary to receptors on the target cell. When a virus recognizes the cell it is specifically tuned for, it attaches itself to the cell and injects its genetic material (DNA or RNA sequence together with any viral proteins). The cell processes (transcribes and translates) the viral genetic material which contains the information on how to make components of the virus (such as the capsid, recognition sites and the genetic material). As the components are produced, they assemble into complete copies of the original virus (*virions*) and are released from the cell to target other cells. The host cell's transcription and translation machinery may be so overcome with the task of reproducing the virus that it stops making the essential components required to enable it to survive, or the virions are released from the cell by puncturing a hole in the membrane of the cell, thereby killing the cell as its contents leak out.

Viruses come in many different forms, and the *Baltimore Classification* identifies viruses according to the nature of the genetic material they contain. Viruses can contain, for example, (a) double-stranded DNA (typically 5000 base pairs (bp) to 300 000 bp), (b) single-stranded DNA, (c) double-stranded RNA, (d) positive sense single-stranded RNA, and (e) negative sense single-stranded RNA. Of these, the positive sense single-stranded RNA class is the best known to humans, causing the common cold (*rhinoviruses*) and meningitis (*enterovirus*). A viral infection is dangerous to an organism because, if the infection goes unchecked, a sufficiently large number of cells can be killed which leads to the organism as a whole dying. An example of HIV (human immunodeficiency virus, considered to be the main cause of AIDS (Acquired Immunodeficiency Syndrome)), is provided in Figure 2.10.

The HIV virion consists of two single-stranded negative sense RNA sequences (about 9000 bases each) containing at least nine genes, plus three proteins – a reverse transcriptase, an integrase and a protease (Figure 2.10(a)). The HIV virion attaches itself to lymphocytes (helper and killer T cells) of the immune system through the CD4 and CCR5 receptors

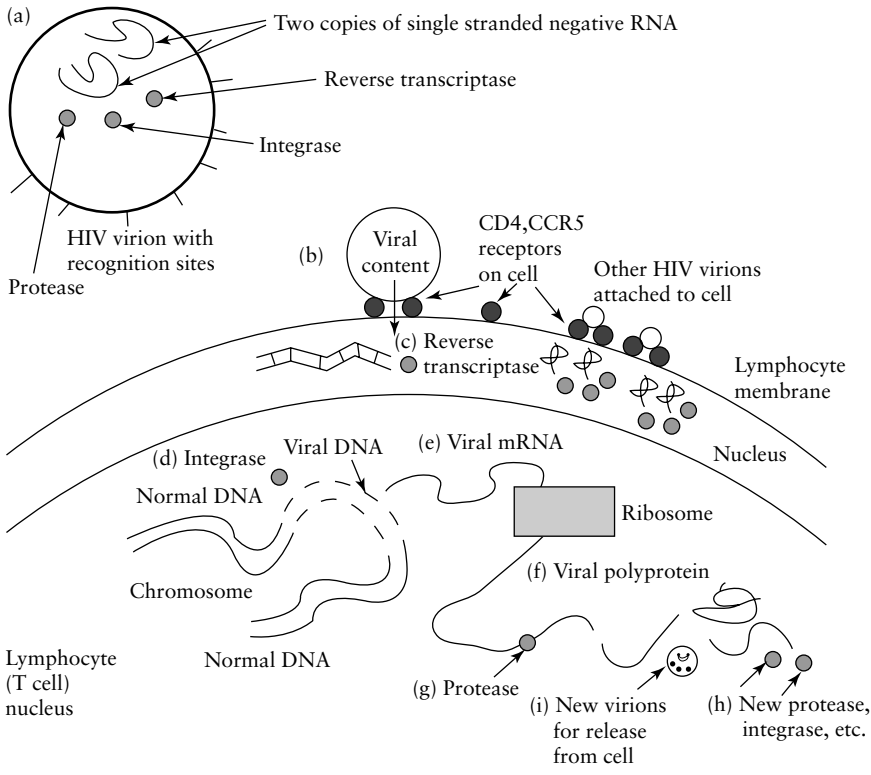


Figure 2.10 The life cycle of HIV

on the surface of the cell (Figure 2.10(b)). The viral content (RNA and proteins) is injected into the cell, and the reverse transcriptase makes a positive copy of one of the negative strand viral RNA to form a double strand (Figure 2.10(c)). The viral integrase takes the double strand into the nucleus and splices it into the cell's DNA (Figure 2.10(d)). Normal cellular machinery then transcribes (Figure 2.10(e)) and translates the viral mRNA to form one long viral polyprotein sequence (Figure 2.10(f)). The third viral protein, protease (Figure 2.10(g)) has the task of cleaving the viral polyprotein into constituent parts (new copies of viral protein, capsid, etc., Figure 2.10(h)) so that new virions can be assembled for further infection (Figure 2.10(i)).

The human immune system has developed a number of methods for detecting and eradicating viruses and other *pathogens* (any disease-producing agent including bacteria) by activating both an *innate* and *adaptive* response. Innate responses are general responses to a limited number of pathogens and include *phagocytes* (scavenger cells) and

macrophages (either fixed to specific locations in the body or circulating with the blood) that 'swallow' whole pathogens or clear up debris. Such cells are directed to pathogens through the stimulation of *antibodies* (immunoglobulins) in response to *antigens* and other substances produced by the pathogen. Also part of the innate response are the *natural killer* cells that destroy cells in the body that have been infected to prevent the infection from spreading. If the innate system cannot deal with the pathogen, the adaptive system takes over. One important part of the adaptive system consists of *lymphocytes* (white blood cells) binding approximately to pathogens. This can result in *B-lymphocytes* (cells produced in bone marrow) producing antibodies to bring the pathogen to the attention of macrophages and phagocytes for destruction, or cloning themselves in large numbers with even more specialized binding mechanisms so that they can inactivate the pathogens directly. Approximate binding and cloning by B-cells provides us with the ability to identify and deal with any new pathogen. However, since approximate binding and cloning can lead to the production of B-lymphocytes that inadvertently attach themselves to healthy *self-cells* (cells that are part of the body and not foreign to the body), the immune system requires *helper T-cells* (cells produced in the thymus) to co-stimulate B-cells only if the B-cell is not attached to a healthy (non-antigen presenting) self-cell. This is particularly important in the case of viruses that have infected self-cells. Such infected cells produce fragments of the virus on their surface through the use of *major histocompatability* (*MHC*) *molecules*. If helper T-cells recognize these viral fragments on the surface of self-cells, it produces a co-stimulus to the B-cell which then destroys the infected cell. One of the critical properties of HIV is that it attacks these helper T-cells (Figure 2.10). If these immune system cells become infected, they can no longer provide the co-stimulation required for B-cells to work. The immune system then becomes sufficiently weakened (Acquired Immunodeficiency Syndrome – AIDS) that any pathogen that would normally be non-dangerous to us becomes lethal. With this basic understanding of viral and immune system behaviour, gene silencing can be described in more detail.

Post-transcriptional gene silencing in multicellular organisms is considered to be an evolutionary conserved, single cell defence mechanism for dealing with foreign genes and RNA introduced typically by a virus. That is, before multicellular organisms – with their complex immune systems requiring the cooperation of many different types of cell – developed from single-cell organisms, such single-cell organisms had to fight

pathogens on their own and without the help of other cells. Both positive-sense and negative-sense RNA are produced by different types of virus, and the cell had to find a mechanism to prevent their expression. Also, double-stranded RNA can be produced by viruses using reverse transcriptase. Since all three types of sequence were found to silence genes in multicellular organisms, the current hypothesis is that the underlying gene silencing mechanisms reflect the manner in which single cells prevented infection.

The current model of interference is that an enzyme called *Dicer* (Figure 2.11(a)) takes the introduced double-stranded RNA and cuts it into small (20–25 bp) sequences called *small interfering RNA* (siRNA) (Figure 2.11 (b)), which in turn – after separating into single strands – bind to an RNA-inducing silencing complex (*RISC*) (Figure 2.11(c)).These

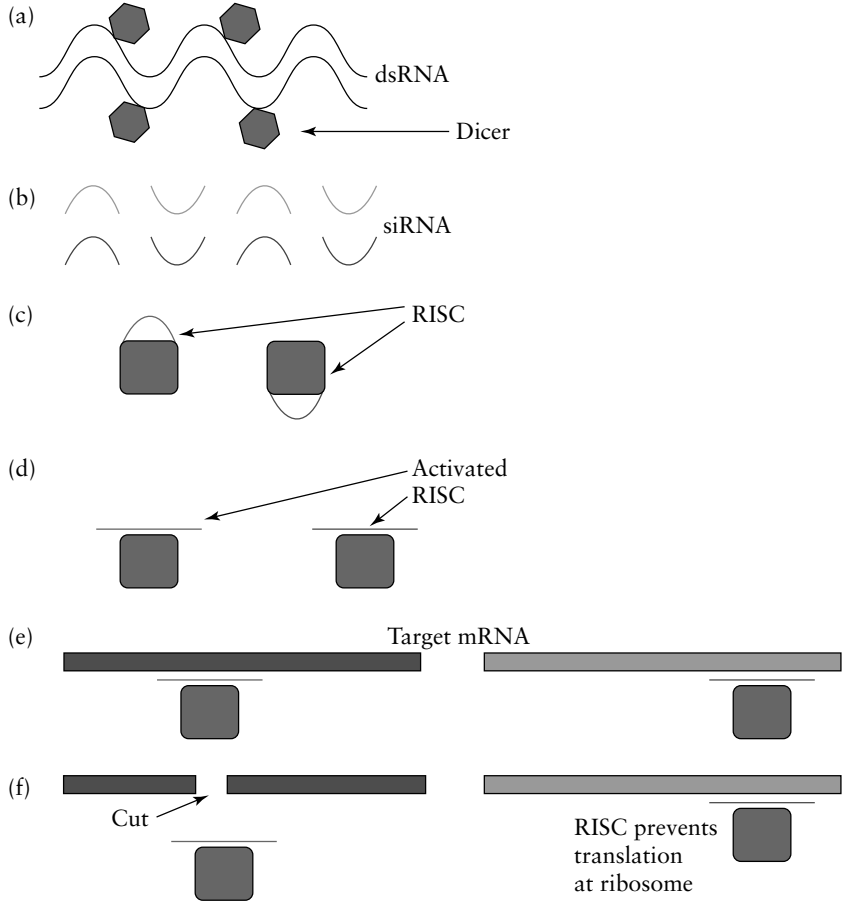


Figure 2.11 Interference technology

become activated when the siRNA unfolds (Figure 2.11(d)) and the activated RISCs then target mRNA transcripts through complementary base pairing (Figure 2.11(e)). If transcripts containing the appropriate complementary sequences are found, they are cut or the RISC binds to the transcript to prevent translation of the transcript at the ribosome (Figure 2.11(f)). In some organisms a 'spreading' effect has been found, whereby the cut mRNA is copied to form secondary siRNA for use in RISCs. This copy process is used to explain why introducing a sense RNA strand can also lead to gene silencing. However, for effective gene silencing, dsRNA is rarely used, since such strands can trigger an anti-viral response from the immune system leading to the cell's death. Instead, siRNA is currently used to silence genes. Such siRNA can be produced synthetically and injected into cells, or they can be transported into the cell with the help of viral 'vectors' (safe viruses that have been genetically engineered to contain a DNA sequence which, when inserted into a cell and transcribed, produce the siRNA). Current research points to whole genome functional analysis being possible in the near future, where all genes are individually screened by siRNA and the resulting transcriptomes and proteomes measured to identify the effects. It is currently unclear as to exactly what sort of bioinformatics resource will be needed to support systematic functional analysis of genomes. Also, current research into RNA interference (RNAi) technology is directed towards fighting viral diseases (the production of siRNA that prevents viral mRNA from being translated) and silencing cancer-associated genes (e.g. siRNA to silence cell division). Many of these problems are so complex that standard modelling and simulation tools may not be adequate. Novel methods and techniques may have to be developed to take bioinformatics into the next generation.

2.6 Summary of chapter

1 The major problems in bioinformatics can be distinguished according to the areas into which these problems fall: genomics, transcriptomics and proteomics.

2 Current problems in the post-genomic era deal with sequence analysis and phylogenetic analysis to make clear the relationships between organisms as the number of fully sequences genomes grows. However, there are problems in being able to compare organisms in such a way that clear and unambiguous phylogenetic relationships emerge.

3 Transcriptomics is a relatively new problem area arising from recent technological advances in DNA arrays (microarrays and gene chips). The major problems here, apart from obtaining the data, is the analysis of the data given the large number of genes measured for a comparatively small number of samples. Novel techniques may need to be developed to reverse engineer gene networks from temporal data so that the interrelationships between genes are clearly identified.

4 Protein folding prediction is one of the oldest known problems in proteomics and hence bioinformatics. Problems exist in sequencing a protein without affecting its nature, and techniques for predicting the structure of proteins from their linear sequence need improving.

5 A new problem area concerns interference technology and the way that genes can be silenced to measure their effect. Of great interest is the application of interference technology to immune systems, since it is by observing the effect of switching off genes and interfering with genes of the immune system that a greater understanding will be obtained of how the body fights infections, thereby leading to future drugs that can be more carefully targeted for particular viruses.

6 Finally, embryonic stem cell research provides a novel way to understand cell differentiation for possible future cures of diseases currently believed to be untreatable. There are, however, ethical considerations with regard to embryonic stem cell research that will need discussion before approval can be given to such research.

2.7 Further reading

Baldi, P. and Hatfield, G.W. (2002) *DNA Microarrays and Gene Expression*, Cambridge University Press.

Coico, R., Sunshine, G. and Benjamini, E. (2003) *Immunology: A Short Course*, 5th edn, Wiley–Liss.

Mount, D.W. (2001) *Bioinformatics: Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press.

Parson, A.B. (2004) *The Proteus Effect: Stem Cells and Their Promise*, National Academies Press.

Ridley, M. (2003) *Evolution*, 3rd edn, Blackwell.

Sternberg, M.J.E. (ed) (1996) *Protein Structure Prediction*, IRL Press.