CHAPTER 6

Computer Analysis of Sequencing Data

Introduction

The goal of this laboratory is to introduce you to sequence analysis methods. You will analyze DNA sequences obtained by sequencing your genomic DNA fragments. Your first task will be to retrieve the file of raw sequencing data generated by the ABI automated sequencer. These data are stored as sequencing chromatograms that are represented as different color peaks. Each color represents a different base as recorded by a digital camera and read by a base-calling algorithm of the ABI automated sequencer. You will inspect these chromatograms and resolve ambiguities of base call by the sequencer. You also will remove vector sequences and store files in a form suitable for sequence analysis programs.

Next you will carry out sequence analysis using the computer. This analysis will include a search for similar sequences in the GenBank database using the local alignment analysis programs BLAST (Basic Local Alignment Search Tool) and/or FASTA. You will search the human EST (Expressed Sequence Tag) database to determine whether your sequence is expressed. Using BLAST you will search for the presence of short interspersed nuclear elements (SINEs) and long interspersed nuclear elements (LINEs) in your sequence. Single-sequence analysis will consist of a search for direct and inverted repeats using a dot matrix program and a search for GpC islands and restriction enzyme sites.

The exercise will be carried out during one laboratory period.

Background

The base sequence in nucleic acids and amino acid sequences in proteins describe the primary structures of these molecules. Sequence analysis comprises determination of sequence properties and comparison of this sequence with other known sequences. The analysis is carried out using computer programs that implement specific algorithms for describing sequence properties or which carry out comparisons with sequences present in databases.

A sequence analysis is a vast subject that incorporates large numbers of different methods and tools. These include (i) storing sequences and the construction of databases; (ii) a database search for similar sequences; (iii) sequence pair and multiple sequence alignments; (iv) prediction of the secondary structure of RNA; (v) prediction of protein structure and function; (vi) phylogenetic analysis; (vii) gene prediction analysis; and (viii) whole genome analysis and comparison.

The theoretical background of many of the methods and programs is highly complicated and not easily accessible for biologists. Recently, however, some excellent reviews of these subjects have appeared that make it possible for biologists to understand the underlying algorithms and assumptions made and the limitations existing in the application of most of these methods (Brown, 2000; Higgins and Taylor, 2000; Mount, 2001).

In this laboratory we will concentrate on two tasks: the first will be sequence alignment analysis and the second will be analysis of the properties of a single sequence.

Databases and sequence formats

DNA and protein sequences are stored in large databases. Several databases are used for sequence comparison. Table 6.1 presents a list of the main databases. In addition to these databases, there exist a number of specialized databases. These are, for example, a protein structure database, a subset of protein family databases, or databases for each fully sequenced genome. These are very useful when working with specific genomes or protein families. An easy way of accessing sequence databases on the World Wide Web (WWW) is to use ENTREZ, a resource prepared by the National Center for Biotechnology Information at http://www.ncbi.nlm.nih.gov/Entrez/.

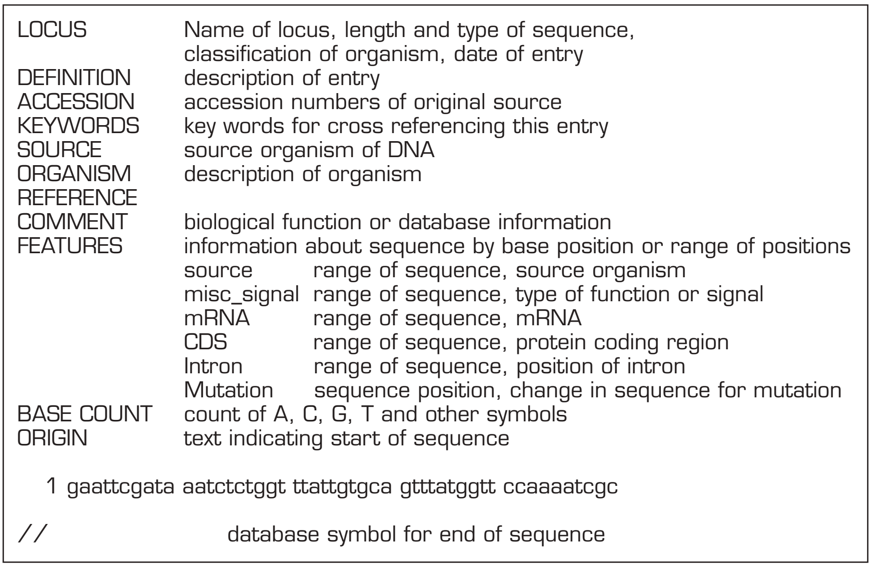
The most important databases listed in Table 6.1 are GenBank, EMBL (European Molecular Biology Laboratory), and DDBJ. Each database collects and processes new sequence data and relevant biological information from scientists in their region, e.g. EMBL collects from Europe, GenBank from the USA, and DDBJ from Japan. These databases automatically update each other with the new sequences collected from each region every 24 hours. The result is that they contain exactly the same information, except for any sequences that have been added in the previous 24 hours. This is an important consideration in your choice of database.

The databases listed in Table 6.1 store data in unique formats. These formats are standard ASCII files but, unfortunately, they differ from each other considerably. These differences are important when running sequence analysis software that may or may not recognize a particular file format.

Table 6.1 Sequence databases accessible through the Internet

Database type	Database name	Database address	Description
DNA	GenBank	www.ncbi.nlm.nih.gov/	DNA sequences (USA)
DNA	EMBL	www.ebi.ac.uk/embl/	DNA sequences (Europe)
DNA	DDBJ	www.ddbj.nig.ac.jp	DNA sequences (Japan)
Protein	SwissPort	www.expasy.ch/sprot/sprot-top.html	Highly annotated protein DB
Protein	PIR	www.gergetown.edu	Annotated protein DB
Protein	GenPept	www.ncbi.nlm.nih.gov/Entrez/protein.html	Translation of GenBank
Protein	Genomes	www.ncbi.nlm.nih.gov/Entrez/genome/org.html	Protein sequences by organisms
Protein and DNA	nr	www.ncbi.nlm.nih.gov/BLAST/	Non-redundant database*

*A non-redundant database is a database that has only one copy of a given sequence. A redundant database can have more than one copy of a given sequence. A redundant database is more comprehensive and more likely to contain recently discovered sequences.



```
LOCUS          Name of locus, length and type of sequence,
               classification of organism, date of entry
DEFINITION     description of entry
ACCESSION      accession numbers of original source
KEYWORDS       key words for cross referencing this entry
SOURCE         source organism of DNA
ORGANISM       description of organism
REFERENCE
COMMENT        biological function or database information
FEATURES       information about sequence by base position or range of positions
               source        range of sequence, source organism
               misc_signal   range of sequence, type of function or signal
               mRNA          range of sequence, mRNA
               CDS           range of sequence, protein coding region
               Intron        range of sequence, position of intron
               Mutation      sequence position, change in sequence for mutation
BASE COUNT     count of A, C, G, T and other symbols
ORIGIN         text indicating start of sequence

  1 gaattcgata aatctctggt ttattgtgca gtttatggtt ccaaaatcgc

//                database symbol for end of sequence
```

Figure 6.1 GenBank DNA sequence file format.

The most common file formats that are recognized by nearly all sequence analysis programs are the GenBank format, EMBL format, FASTA format, NBRF format and Stanford University Intelligenetic format.

In most of these formats information is organized in fields that are recognized by an identifier word or letter at the beginning of each text line.

The format of the GenBank file is shown in Fig. 6.1. The GenBank format starts from the word LOCUS on the first line of the text. This is followed by a number of fields: DEFINITION, ACCESSION, SOURCE, etc. The word ORIGIN delineates the last line of the text. All lines after this are base or amino acid sequences. The file ends with the sign "//."

The EMBL sequence entry format is similar to the GenBank format and is shown in Fig. 6.2. The identifier words are substituted by two-letter abbreviations of the fields. The first line identifier is ID, which is equivalent to the LOCUS line in the GenBank format. The last line of the text has the identifier SQ and all lines after that are DNA or protein sequences. The symbol for identifying the end of a sequence is "//."

The FASTA sequence format is shown in Fig. 6.3. The first line begins with a ">" character as the identifier. That can be followed by the name or origin of a sequence 60 characters long. No other fields are included in this format. The second line is a line with DNA or amino acid sequences. No spacing or numbering is allowed in the description of the DNA or protein sequences. If two or more sequences are listed in a single file, each sequence is ended by the character "*." The presence of this character may or may not be essential for reading the FASTA format by some sequence analysis

```
ID                  identification code for sequence in the database
AC                  accession number giving origin of sequence
DT                  dates of entry and modification
KW                  key cross-reference words for lookup up this entry
OS, OC              source organism
RN, RP, RX, RA, RT, RL literature reference or source
DR                  i.d. in other databases
CC                  description of biological function
FH, FT              information about sequence by base position or range of positions
                    source range of sequence,  source organism
                    misc-signal range of sequence, type of function or signal
                    mRNA range of sequence, mRNA
                    CDS range of sequence, protein coding region
                    intron range of sequence, position of intron
                    mutation sequence position, change in sequence for mutation
SQ                  count of A, C, G, T and other symbols
gaattcgata aatetctggt ttattgtgca gtttatggtt ccaaaatcgc cttttgctgt 60

//                        symbol to indicate end of sequence
```

Figure 6.2 EMBL sequence file format.

```
>Cp Chloroplast region 2
ACTTGTTGCCATGGTACGTACGTACGGT
TGGCCCATTCGGTACCTGCCATTGCATT*
```

Figure 6.3 FASTA sequence file format.

programs. It is therefore customary to include this character even in files containing a single sequence.

The NBRF sequence entry format is identical to the FASTA format except that the second line contains information about the sequence and the third line contains the sequence.

The Intelligenetic sequence entry format is very similar to the NBRF format except that a semicolon is placed before the first line instead of the “>” character. The second line contains an identifier describing the sequence. The third and subsequent lines contain sequences. A number 1 is placed at the end of the sequence if the sequence is linear or a number 2 if the sequence is circular.

A number of programs exist that will convert one format into another format. The most popular is the READSEQ program developed by Dr Gilbert at Indiana University. The program is available on the Internet at http://dot.imgen.bcm.tmc.edu:9331/seq-util/readseq.html or it can be downloaded from the FTP site ftp.bio.indiana.edu/molbio/readseq.

Sequence alignments

Comparing two or more sequences to each other is called sequence alignment. Sequence alignment is the most important and commonly used method of sequence analysis. The first thing to do with a newly determined sequence is to compare it to all known sequences. The goal is to determine whether this sequence is identical to known sequences or does this sequence have some degree of similarity to known sequences.

Sequence similarity or identity may indicate a similar structure and suggest the function of an unknown sequence. Moreover, finding dissimilar regions in sequences that are otherwise identical is also very important. These dissimilarities can have a different origin, such as population polymorphism, differences in multiple copies of a gene in a single individual, or evolutionary divergence of genes in different organisms. Thus, sequence analysis is a necessary first step in detailed experimental studies of structure, function, evolutionary origin, and relations between biological molecules.

Alternatively, sequence alignment can be used "in reverse," that is one can use a sequence with known function for searching through the sequence database (e.g. whole genome sequence) of a particular organism in order to identify a gene that may have the same function.

The process of sequence alignment involves one-to-one matching of two strings of letters (nucleotides or amino acids) so that each letter in a pair of sequences is associated with a single character of the other sequence or with a null character or gap. As its basis, the process of comparison can be imagined as writing two sequences across a page in two rows in a way that identical characters are placed in the same column and non-identical characters are placed as the gaps or mismatches. An optimal alignment is considered an alignment that places the maximum number of identical characters under each other in both sequences. The alignment of two sequences without gaps requires an algorithm that performs a number of comparisons that are proportional to the square of the sequence length. If an alignment is to include gaps at any position and over any length in each sequence, the number of combinations of gaps and matches, even for two short sequences, becomes very large and it is impossible to find the best alignment by trying all possibilities. It was calculated that the number of comparisons that would be required to compare two sequences with 300 characters would be 10^{88} (Waterman, 1989). In order to realize how big this number is it should be compared to the estimated number of elementary particles in the universe: 10^{80}. The essence of alignment methods (programs) is to solve this problem in a realistic time and to give statistical measurements for the quality of the alignment.

There are several different alignment algorithms used in sequence analysis. Most of these algorithms use a dynamic programming method. The number of combinations in dynamic programming is limited by the following approach. The alignment of one sequence with another is represented as a grid, with each sequence on an axis. Each cell of this matrix ties a pair of units (amino acids or nucleotides) in the two sequences. The best alignment of two sequences is the path from one end of the matrix (upper left

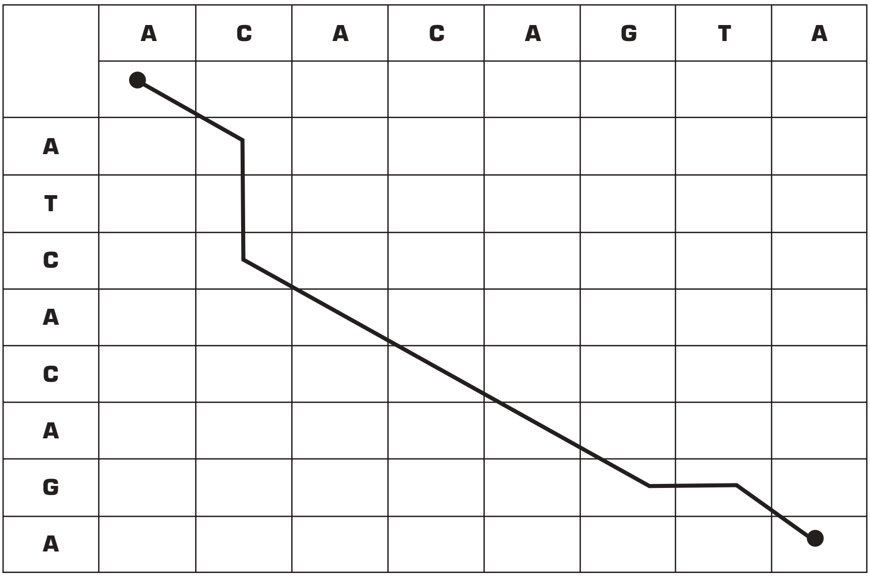


Figure 6.4 Graph of the dynamic programming decision matrix.

corner) to the other (lower right corner) that passes through most matching cells.

Figure 6.4 shows a dynamic programming decision matrix for alignment of two sequences. Wherever sequences are identical, the path moves diagonally. When the sequences differ, the path can move vertically or horizontally, indicating the insertion of gaps in one or another sequence. At each step, the computer chooses the path through the most identical cells of the matrix that has highest score from all previous cells that brought the path to this point. Thus, for any given comparison of two sequences, there can be more than one optimal path, i.e. alignment. It is important to realize that only a few of these optimal alignments may have biological significance and the decision as to which one of them has biological importance cannot be made by the computer.

In order to carry out dynamic programming analysis, an alignment must use a scoring matrix that assigns values for identical scores and mismatched scores and assigns a penalty for gaps. There are several different scoring matrixes in use for nucleic acid and protein alignments. The use of one matrix over another determines the stringency of the alignment, e.g. finding closely related sequences or evolutionary distant sequences, etc. The dynamic programming method was introduced to biology by Needleman and Wunsch (1970) and is usually referred to as the Needleman–Wunsch algorithm. Smith and Waterman (1981a,b) extended and enhanced this method to include an improved scoring system. This algorithm is referred to as the Smith–Waterman algorithm.

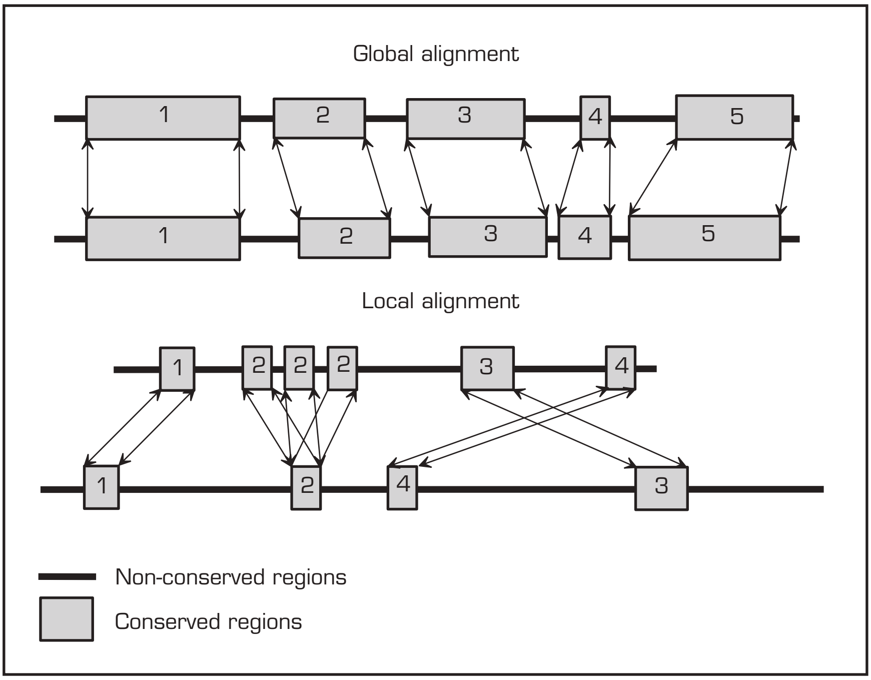


Figure 6.5 Principle of global and local alignments. Two DNA sequences are compared for each alignment. Five regions of similarity are indicated in global alignment. These regions are at approximately the same positions in both sequences. Local alignment of two sequences shows four conserved short regions. They are not at similar positions in both sequences.

There are two types of sequence alignment: global alignment and local alignment. The principle of both alignments is illustrated in Fig. 6.5.

In **global alignment** one attempts to derive an optimal alignment between two sequences over their entire length. Sequences that are similar and approximately the same length are usually compared using this type of alignment. The alignment is implemented by using the Needleman–Wunsch algorithm and finds global similarities between two or more sequences. This type of analysis is not sensitive enough for comparing highly divergent sequences and cannot be used for similarity searches with databases. Its most frequent use is in the construction of evolutionary trees or an analysis of closely related proteins.

Most sequences cannot be compared using global alignment algorithms. This is because, in most cases, the similarities between two sequences are limited to specific short regions or domains. Indeed, most proteins are constructed from a combination of specific "modules" called domains. One can imagine the structure of protein as a building constructed from Lego building blocks (domains). Thus, one can construct an almost infinite number of Lego houses using a small number of building blocks! This modular evolution played a major role in the evolution of most protein and DNA sequences.

In order to analyze these structures the **local alignment** method is used. Optimal alignments are made over short regions of similarity that may exist in two sequences rather than a comparison of their entire length. Thus, conserved regions can be found in two sequences even if most of the sequence is dissimilar. The Smith–Waterman algorithm is used for this type of alignment. Routine database searches are possible using a modification of the Smith–Waterman algorithm, but the searches are approximately 50-fold slower than when a search is carried out with heuristic algorithms (based on a process of successive approximations). Database searches can be carried out on the net using this implementation of the Smith–Waterman algorithm with the program SSEARCH at http://fasta.bioch.virginia.edu/fasta/cgi/searchx.cgi?pgm=fa.

The Smith–Waterman algorithm is not used for everyday database searches because it runs very slowly, particularly when very large databases are searched. Routine searches use heuristic algorithms that are very fast. Heuristic algorithms are not guaranteed to find the optimal alignments and might result in some loss in the rigor of comparison by missing weak similarities or identifying similarities that are biologically irrelevant. It is therefore very important to pay close attention to the statistical significance of the results, understand the options presented for optimizing ones search, and be aware of the limitations of each option.

The most popular heuristic programs for similarity searches are BLAST and FASTA. They differ in sensitivity and speed, BLAST being less sensitive but faster and FASTA being slower but more sensitive. Since BLAST performs a faster search than FASTA, it is usually the first choice for searching large databases. FASTA is used if BLAST searches are not successful or give misleading results.

BLAST

The BLAST algorithm was developed for performing fast similarity searches using very large databases (Altschul et al., 1990, 1994, 1997). Access to the BLAST system is possible through the Internet at http://www.ncbi.nlm.nih.gov/BLAST/. There are also numerous mirror sites that provide a BLAST database search. Since we will be using the BLAST program extensively, it is important to understand how this program works and how its options may affect the results.

In the initial scanning step, BLAST compares a query sequence (your sequence) to each sequence in the database. The algorithm breaks each sequence into short fragments designated as words and then looks for closely matching pairs of words between the two sequences. All matching words with similarity scores exceeding a certain preset threshold are saved. These segment pairs are sequences of the same length, one from each sequence. A score is assigned using a designated scoring matrix (usually BLOSUM62 for proteins and PAM for nucleic acids). The sum score is used for determining the degree of similarity. Sequences with a high score are referred to as **high-scoring segment pairs (HSPs)**. The program extends the best HSPs (those with the highest score, i.e. the best matches) in both directions until the maximum possible score for the extension is reached. Those sequences with higher similarity scores are reported as **MSPs** (maximal-scoring segment pairs). Finally, multiple MSP regions are combined and the statistical significance of the similarity score is calculated using the Poisson or sum statistic (Altschul et al., 1994). The most significant hits and their statistical significance (*E* value) are reported. The value *E* describes the number of hits one can "expect" to see just by chance when searching a database of a particular size. It decreases exponentially with the score (*S*) that is assigned to a match between two sequences. Essentially, the *E* value describes the random background noise that exists for matches between sequences. Thus, the smaller this number is the more probable that the similarity is not random and has biological significance.

The word size option

One of the most important options in BLAST is the **word size**. The lengths of the word determine a fragment size that must have a perfect match to be extended. The default is 11 for BLASTn (nucleic acid search). The BLAST program will scan the database until it finds words that are 11 letters long exactly matching a word of 11 letters in the query. This match will be extended. The 11 letter word is used as a default because it will exclude even moderately diverged homologs from extension and therefore will exclude almost all chance alignments.

Changing the word size will change the speed of the program execution, as well as its output. A small word size will increase the speed and obtain a large number of short exact matches that might not have biological significance. For example, the BLAST search type "Search for short nearly exact matches" uses a word size value of seven.

The filter option

BLAST version 2.0 enables the application of a filter. The filter masks regions of the query sequence that have low compositional complexity (e.g. *Alu* sequences). Masking is achieved by replacing the sequence with a string of Ns (NNNNNN). N is the IUB (International Union of Biochemistry) code for any DNA base. Only the query sequence is masked. The sequences in the database will not be masked. Filtering is necessary because of the large number of repeated or identical sequences (e.g. poly-A tails, proline-rich sequences, etc.) that are dispersed throughout the genome and, therefore, also throughout the database. They will return artificially high scores and misleading results. By default, filtering is turned on to "low complexity." Filtering eliminates statistically significant but biologically uninteresting reports from the BLAST output. When working with human sequences, one can turn default "human repeats" instead of "low complexity." This option masks human repeats (LINEs and SINEs) and is particularly useful for human sequences that may contain these repeats.

The choose database option

This option will limit the search to a particular database. The available databases are as follows.
nr: all GenBank plus EMBL plus DDBJ plus Protein Data Bank (PDB) sequences (but no expressed sequence tags (ESTs), sequence tagged sites (STSs), genome survey sequences (GSSs), or phase 0, 1, or 2 high throughput genomic sequences (HTGSs). No longer "non-redundant."
month: all new or revised GenBank plus EMBL plus DDBJ plus PDB sequences released in the last 30 days.
Drosophila genome: *Drosophila* genome.
est_others: EST sequences of GenBank plus EMBL plus DDBJ.
est_human: human expressed sequence tags.
est_mouse: mouse expressed sequence tags.
dbsts: STSs from database from Bank plus EMBL plus DDBJ
htgs: unfinished HTGSs.
gss: GSS, includes single-pass genomic data, exon-trapped sequences, and *Alu* polymerase chain reaction sequences.
S. cerevisiae: yeast (*Saccharomyces cerevisiae*) genomic nucleotide sequences.
E. coli: *Escherichia coli* genomic nucleotide sequences.
pdb: sequences derived from the three-dimensional structure from the Brookhaven Protein Data Bank.
vector: vector subset of GenBank.
mito: database of mitochondrial sequences.
alu: select *Alu* repeats from REPBASE, suitable for masking *Alu* repeats from query sequences.
Epd: Eukaryotic Promotor Database.

The expect value option

This value is used as a convenient way of creating a significance threshold for reporting results. When the expect value is increased from the default value of ten, a larger list with more low-scoring hits will be reported. The meaning of ten is that, in a database of the current size, one might expect to see ten matches with a similar score simply by chance.

FASTA

The FASTA algorithm was developed by Lipman and Pearson (1985) and Pearson and Lipman (1988). It uses an algorithm that is similar in concept to a dot plot. Similarly to BLAST, FASTA makes a list of all words in each sequence. The words are called KTUP values and are usually two for amino acids and four to six for nucleotides. Then the program identifies words that are identical between the two sequences and a check is made if these words are located close to other identical words in these pairs of sequences. Only non-overlapping words are counted. The next program tries to join high-scoring words, introducing gaps. Whereas BLAST relies on the sum match probability for each local alignment for the sequence, FASTA scores only exact matches. FASTA allows gapped searches to be made. Like BLAST, FASTA is heuristic, sacrificing some speed for sensitivity.

Sequences as short as ten nucleotides in length can be queried using FASTA. The speed of the alignment is largely determined by the **KTUP** value, which is used to limit the word length. In BLAST, a "word" is a short region of the query sequence that is compared against the database. In FASTA, the word is not scored, but must be an exact match if it is to be processed further.

The FASTA output is essentially very similar to the BLAST output. A list of sequences is presented with the most significant alignments first. The best region of the match is reported as the "initial score" (init1). The optimized score represents the score from joining all scoring regions and applying statistical treatment in order to extend the size of the match. Naturally, if the sequences are related, the optimized score is much higher than the initial score because these sequences will have more than one identity region. Actual base alignments are shown in the context of the database sequence that matches it. The numbers of bases that match exactly are reported as a percentage of identity. In the list of all reported sequences the last value is the score and the last number given on each line is the **expect** value (scoring *E* value). The maximum (threshold) *E* value is 2.0 by default. As with BLAST, the smaller the expected value, the lower the probability that the reported alignment is a chance finding. Or expressing it another way, **the lower the reported expected value is for a reported sequence, the more likely it is that it is true**. The expect values should be regarded as guidance tools only for identifying the origin of the query sequence.

BLAST versus FASTA

FASTA and BLAST both perform an identical function – to search databases for similar sequences. The way this is achieved by both programs is quite different. FASTA in general is more sensitive than BLAST but this comes with the penalty of speed because the FASTA search is much slower than the BLAST search. However, the choice of each program does not depend only on its speed. Some other considerations might favor one program over the other. FASTA is more sensitive for DNA–DNA searches, particularly for diverged sequences. Moreover, FASTA is better for finding long regions of similarities. However, BLAST is better for finding short regions of high similarity. In general, any search should always start with the BLAST program, and if the search is negative, FASTA should be used.

Single-sequence analysis

Finding the position of features in a DNA sequence is an important step in establishing its function. This analysis is frequently called single-sequence analysis. Single-sequence analysis of DNA usually involves the following.

1. Analysis of the DNA base composition. Genomes of different organisms vary considerably in their base composition. The base composition of various regions of the same genome can also be very different. Mammalian genomes are organized into large regions of similar base composition called isochores. There are AT-rich isochors, which are usually referred to as paleo-isochores and GC-rich isochors, which are referred to as neo-isochores. The human genome contains five isochores (two paleo-isochores and three neo-isochores) that not only differ in their base composition, but also in their positions on chromosomes and the presence of specific types of genes and repeating elements. Thus, the GC composition of a DNA fragment can point to its position on the chromosome, the presence of SINEs or LINEs, or specific genes (e.g. housekeeping genes).

2. Analysis of the distribution of nucleotide doublets or triplets. Distribution of these features is highly characteristic for particular genomes and is not uniform across a single genome. For example, CpG nucleotide pairs are less common than expected in vertebrates, including humans. The gene-coding sequences usually have a low frequency of CpG. However, first introns and 5′ upstream regions of most human genes have higher than average concentrations of CpG. These regions are called “CpG islands.” The presence of a CpG island is considered a good indicator of the presence of coding sequences, i.e. the presence of a gene in close vicinity.

3. Search for sequences coding for proteins. This is the most important task in the analysis of whole genomes, which is frequently referred to as annotation of the genome. Since in higher eucaryotes only 5 percent of the DNA sequence is coding for proteins, identifying these DNA sequences is not a trivial task. The protein-coding regions do have an effect on the composition of the DNA, largely due to three factors: (i) uneven use of an amino acid, since proteins have a restricted range of amino acid composition (e.g. tryptophan is a quite rare amino acid); (ii) uneven numbers of codons for each amino acid, with this number varying from one to six; and (iii) uneven use of codons. Different organisms and different genes in a single organism have different codon usage. These and other considerations are usually taken in to account when designing genome annotation programs (for a discussion see Mount (2001)).

4. Mapping the positions of various site-specific sequences. These include restriction enzymes recognition sites, promoter sites, ribosome binding sites, regulatory motifs, etc. Specific programs performing these tasks are widely available.

5. Analysis of repetitive sequences. Direct and inverted repeats are common in DNA. In addition, each eucaryotic genome contains a large number of repetitive elements, i.e. tandem repeats and interspersed repeats. These repeats are the subject of many of the experiments described in this book. A genome-wide search for tandemly repeated elements can be carried out using the program Tandem Repeat Finder (Benson, 1999). Searching short DNA sequences for tandem and inverted repeats is usually carried out using dot matrix programs.

Dot matrix analysis

This method is the oldest method of sequence analysis and was first introduced by Gibbs and McIntyre (1970). Later, Maizel and Lenk (1981), Staden (1982), and Pustell and Kafatos (1982) introduced its popular implementations. The method does not align sequences, but it is used for finding similarities. Dot matrix analysis is used for two tasks: (i) detailed pairwise sequence comparison; and (ii) revealing the presence of direct repeats and inverted repeats in DNA.

Pairwise analysis is usually performed using sequences (proteins or nucleic acids) that have been identified by BLAST or FASTA as having some region(s) of similarity to a query sequence. Dot matrix analysis will uncover the exact positions of these regions in the query sequence and find regions of lesser identity that neither BLAST and FASTA can identify.

In a search for repeats the sequence is analyzed against itself and repeats are revealed as diagonal in the plot. RNA folding programs are a specific implementation of a self-analysis dot plot.

What are the basic principles of dot matrix analysis? In order to compare two sequences using this method one sequence is written out vertically with each base (amino acid) representing a row and the second sequence is written out horizontally with each base (amino acid) representing a column. Each letter in a row is compared to each letter in a column and a dot is placed at the corresponding intersection when the letters are identical. The diagonal stretch of dots will indicate regions where analyzed sequences are identical. Direct repeats will show up as diagonal lines of dots and inverted repeats as vertical lines. A break in a diagonal line and its displacement

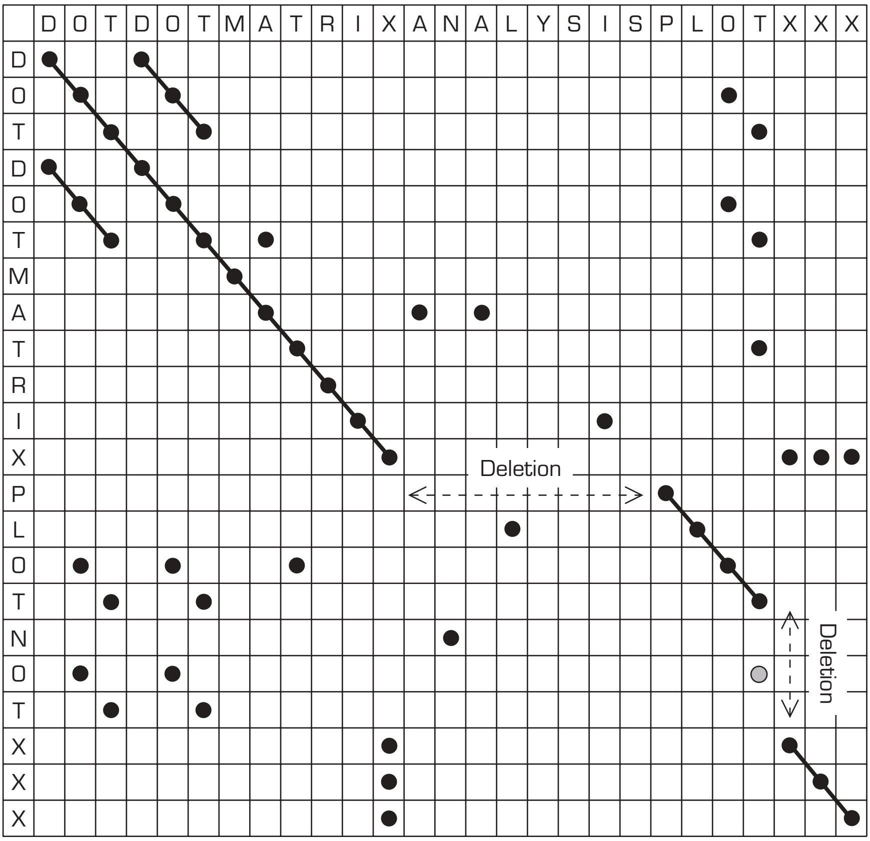

D O T D O T M A T R I X A N A L Y S I S P L O T X X X
D O T D O T M A T R I X P L O T N O T X X X
Deletion
Deletion

Figure 6.6 Principle of dot matrix analysis. Two "sequences" are compared. The horizontal sequence is DOTDOTMATRIXANALYSISPLOTXXX and the vertical sequence is DOTDOTMATRIXPLOTNOTXXX.

represents a deletion or insertion in one or another sequence. These are common features in pairwise alignment analysis and are important indicators of gap placement. Figure 6.6 illustrates these principles.

Dots that are not on the diagonal will also be present and represent random matches that do not form any significant alignment. This is particularly prevalent in nucleic acid dot plot analysis since these molecules have only four "letters." Various filters are employed for removing this noise. Most popular is the "sliding window" filter. A comparison is made not between individual "letters," but between several of them at the time (e.g. ten). This group is called a window. A dot is only placed if all of the letters or some percentage of them (e.g. 80 percent, i.e. eight bases in a ten-base window) are identical in both sequences. Next the window is moved in both sequences by one or more bases and the comparison is repeated. This process is continued until the entire sequence pair has been analyzed.

This method not only eliminates random dots, but also permits the application of statistics to dot plot analysis as a predetermined percent of the match and the detection of more distant similarities. This can be done by increasing the size of the window and decreasing the percent of matches that will result in the creation of a dot. In more sophisticated dot plot programs, scoring matrix tables are used for determination of an identity between two windows.

Technical tips

The success of sequence analysis depends critically on two steps in the chromatogram-editing task. The first step is to remove poorly sequenced regions at the beginning of the chromatogram (usually ten to 20 bases) and at the end of the chromatogram. Usually sequences after 400 bases are not correct. If both of these sequences are not removed first, it will be very difficult to identify plasmid sequences in the sequence.

Another important step is the removal of plasmid sequences. A plasmid sequence can be present at the beginning of a file (sequences close to the primer) and at the end of an entire sequence. The presence of a plasmid sequence at the end of a file will occur when a fragment inserted into the sequencing plasmid is shorter than 400 bases. Submitting to a BLAST search sequence file with plasmid sequences present will result in a very large output containing all plasmid sequences present in the database. In most chromatogram-editing programs plasmid sequences can be removed automatically. Otherwise they should be removed "by hand."

If a sequence contains tandem repeats or dispersed repeats (LINE or SINE), the output after a BLAST search will also be very large since there are a very large number of these elements in the human genome. In this case, the sequence can be resubmitted for a BLAST search after removing these sequences. LINE and SINE elements are usually located close to protein-coding regions and, thus, their presence can indicate the rest of the sequence codes for proteins.

Removing LINE and SINE sequences from the query file is even more important when searching for the chromosome position of this sequence.

There are a number of non-commercial DNA analysis packages than can be used instead of the Sequencher and DNASIS programs described here. The Staden package can run on Mac and PC computers and can be downloaded from http://www.mrc-lmb.cam.ac.uk/pubseq/staden_home.html. The AnnHyb package for Windows can be downloaded from //annhyb.free.fr/download.php3. DNATools is a shareware package that can be downloaded from www.dnatools.dk. In addition to the usual analysis of a single sequence, this package contains a chromatogram-editing module. Another excellent shareware package for PC computers is "DNA for

Windows" with an excellent module for chromatogram editing. It can be downloaded from http://website.lineone.net/~molbio/.

Most of the packages do not incorporate dot plot analysis. There are several java applets for dot matrix analysis that can be run on any computer. These are as follows.

1. Dottlet at www.isrec.isb-sib.ch/java/dotlet/Dotlet.html.
2. DNA dot at http://arbl.cvmbs.colostate.edu/molkit/dnadot/.
3. DotPlot at http://www.geneart.com/dotplot.php3 (best).

Protocol

Editing chromatograms

The chromatogram editor is used for viewing and editing the raw sequence data produced by automated DNA sequencers. You will do this analysis using the program **Sequencher**. The chromatogram editor displays colored peaks as interpreted by a base-calling algorithm of the ABI sequencer. The chromatogram can be analyzed manually when resolving ambiguities (which may become apparent when assembling sequences) and changes can be made to the derived DNA sequence. In the chromatogram editor (i) sequence data is displayed graphically; (ii) the DNA sequence derived from the chromatogram is freely editable; (iii) each trace can be dragged up or down to help clarify base calls at the beginning and extreme ends of a chromatogram; and (iv) the vertical scale of the chromatogram traces can be adjusted.

1. Download your sequences from the server. The file extension will be abi. Place this file into a folder with your group number. Remove the long header from the file name that was introduced by the ABI sequencer and change it to a file name preferred by you. Do not remove the "abi" extension. Repeat this procedure with each file of your sequences.

2. Open one of the sequencing files by double clicking on it. You will see windows with DNA sequences. Each base is represented by a different color: green = **A**, blue = **C**, black = **G**, and red = **T**.

3. At the beginning and end of your sequence there will be many letter **N**s. This letter is colored light blue. These are the positions in your sequence that the computer could not assign to any specific base (N stands for u**N**known). The beginning and end of the sequence will have the most bases designated N.

4. In the upper right corner of the DNA sequences window, there is a button labeled "**show chromatogram**." Click on it. The window opens showing chromatogram peaks labeled in different colors. On the top of this window, you will see letter designations for each base peak presented in the chromatogram. The base numbers will be indicated also.

5. Inspect the beginning of the chromatogram. Try to correct N to the appropriate base if this is possible. If too many Ns are present in a particular region and they cannot be corrected, you need to remove this entire segment.

6. In order to remove an ambiguous stretch, move the mouse over the **top base letter line**. The cursor will change to a small square. Holding the mouse button down, outline the unreadable segment. The outlined segment should turn light blue. Delete this segment by pushing **delete** on the keyboard. This will delete all the letters (but not the chromatogram) of this region. This letter will also be deleted from the sequence window and the base numbering will be changed. **Be very careful with this deletion, you will not be able to undelete it**.

7. Scan the chromatogram to the end. At approximately position 300 or 400 bases the quality of the chromatogram will start to deteriorate and be nearly unreadable. Base peaks will become wide and flat. The ABI interpreter frequently recognizes this as a long stretch of identical bases (e.g. AAAA). You will need to remove all of these sequences to the end of chromatogram as described in step 6.

8. Next you will inspect other ambiguous positions in your sequence. Move the cursor to each base designated N. Inspect the chromatogram at this position. It is frequently possible to guess the correct base directly from the chromatogram picture. Change N to your guessed base by typing the letter.

9. After all corrections are finished, save the corrected sequence. Click on "Save sequencing project," give it an appropriate name, and click OK.

10. Transfer other sequencing files (files with extension abi) to the window of the sequencing project and correct them as described above. Save the sequencing project.

Removing vector sequences

Sequencing files might contain vector sequences at the beginning or end. These should be removed before the BLAST or FASTA programs can analyze the sequences. In order to screen for vector contamination, you must specify the vector used for amplifying your fragment. You can type this information in yourself or load it from a file. To enter or load vector information perform the following.

1. Choose the **trim vector** command from the **sequence** menu. The vector contamination window will open. Choose the "**choose insertion site now**" button from it. The vector insertion site window will open.

2. This window has two entry fields. The first one allows you to "**load the sites**." The second allows you to "**save your choice**." The program can load vector sequences directly from a file in **VecBase** format.

3. Click the button labeled "**use VecBase file**." The window with vector names will open. Find the file for the **pUC18** vector and open it. When the polylinker window is displayed, click the site where your fragment was inserted into the vector. Since we cloned our insert into *Sma*I sites, highlight it and click the **OK** button.

4. Save this vector by clicking on the **save sites** button (top of the window). Choose the name for your file in **save as** window (for example, my puc18) and **store it in your folder where the rest of your sequences are located**.

5. Highlight your sequence and choose the "**trim vector**" command from the "**sequence**" file. The computer searches the selected fragments for overlap with the vector bases entered. Any of the searched fragments that are found to contain vector bases will be displayed in the **vector screening** dialog box.

6. The window will show your sequence as a single line in two colors. Blue color indicates your sequence, whereas vector sequences are red. A scissor icon will separate the sequences.

7. Click on the "**show sequence**" button and, instead of a line, you will see base sequences. The *Sma*I site should be displayed on the border between the vector and insert. This site has the sequence CCC|GGG and, thus, you should see three Gs (GGG) in the sequence colored red.

8. Trim unwanted vector sequences by clicking on the "**trim checked items**" button.

9. Perform this analysis with all of your sequences and save the project again.

References

Altschul, S.F., Gish, W., Miller, W., Mayers, E.W., and Lipman, D.J. (1990) Basic local alignment tool. *J. Mol. Biol.*, **215**, 403–10.

Altschul, S.F., Boguski, M.S., Gish, W., and Wooton, J.C. (1994) Issues in searching molecular databases. *Nature Genet.*, **6**, 119–29.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, W., Miller, W. et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–402.

Benson, G. (1999) Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res.*, **27**, 573–80.

Brown, M.S. (2000) *Bioinformatics: A Biologist's Guide to Biocomputing and the Internet*. Eaton Publishing, A BioTechniques Book Publication, Natick, MA.

Gibbs, A.J. and McIntyre, G.A. (1970) The diagram, a method for comparing sequences. Its uses with amino acid and nucleotide sequences. *Eur. J. Biochem.*, **16**, 1–11.

Higgins, D. and Taylor, W. (2000) *Bioinformatics: Sequence, Structure and Databanks. A Practical Approach*. Oxford University Press, Oxford.

Lipman, D.J. and Pearson, W.R. (1985) Rapid and sensitive protein similarity search. *Science*, **227**, 1435–41.

Maizel, J.V. and Lenk, R.P. (1981) Enhanced graphic analysis of nucleic acid and protein sequences. *Proc. Natl Acad. Sci. USA*, **78**, 7665–9.

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

Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–53.

Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–8.

Pustell, J. and Kafatos, F.C. (1982) A high speed, high capacity homology matrix: zooming through SV40 and polyoma. *Nucleic Acids Res.*, **10**, 4765–82.

Smith, H.O. and Waterman, M.S. (1981a) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–7.

Smith, H.O. and Waterman, M.S. (1981b) Comparison of biosequences. *Adv. Appl. Math.*, **2**, 482–9.

Staden, R. (1982) An interactive graphic program for comparing and aligning nucleic acid and amino acid sequences. *Nucleic Acids Res.*, **10**, 2951–61.

Waterman, M.S. (1989) Sequence alignment. In *Mathematical Method for DNA Sequences*, M.S. Waterman (ed.), pp. 53–92. CRC Press, Boca Rotan, FL.

Sequence Alignment with BLAST

BLAST is the alignment program for finding sequences in a database similar to your sequence. Reported alignments (i.e. sequences in the database that show some statistically significant similarity to your sequence) are reported in order of significance. BLAST does not try to match the whole sequence. Look for more details about the BLAST program in the introduction to this chapter.

1. In order to perform a BLAST analysis you first need to copy one of your sequences into computer memory. Open your sequencing project, if it is not yet open. Highlight the file with the corrected DNA sequence and double click on it. The window with your DNA sequence will open. Outline this sequence with the mouse and click on **edit**. In the **edit** window choose **copy selection**. This will copy your sequence (**query sequence**) to computer memory.
2. Open **Internet explorer** and type the address that will open the WWW-based BLAST search www.nci.nlm.nih.gov/BLAST/. Open the BLAST search by clicking on "**standard nucleotide–nucleotide BLAST**."
3. A new page will open with space provided for the query sequence (**search**). Click in this window to move the cursor into it.
4. Copy the sequence in the space provided. Open **edit** and choose **paste**. Your sequence will appear in the search window.

Search of "nr" database

1. Choose the **nr** database in the "**choose database**" window.
2. Start the search with the **BLAST!** button.
3. A window will open that will have **query** = (base number) on the top. The

length of your sequence will be indicated in the brackets under this title, e.g. 246 letters. Change 100 to 50 in the window "**description**." Change 50 to ten in the "**alignment**" window. Click on the "**format**" button to see the results.
4. After a while the "result of BLAST" window will open. It will be updated automatically until the search result is ready.
5. Your sequence will be called **query sequence** by the BLAST program. Scroll down this window until a list of identical sequences appears. The first sequence in the list is the best-matched sequence. Click on this sequence and you will see the exact alignment of the matching region(s).
6. Print the results of the BLAST search. Use the **print** function present in the **file** menu. Choose the page number to print. Print only three to four pages of your results.
7. If one of your sequences has high homology to a sequence in the GenBank, download this sequence to your folder.
8. The name of the file to download is located at the left side of the text and is printed in blue. Click on this name. That will open the window with the GenBank file to which your sequence is homologous.
9. Save this file in your folder. Click on **file** at the top of the window and choose **save as**. **Do not click on the text or save buttons in the window**. The query window will appear. Choose your folder. Next, in the window "**save file as**" give a name to this file. In the **format** window, choose **plain text**. Save the file by clicking on the **save** button.

Search for an *Alu* SINE element

1. Return to the search page and initiate a new search with the same sequence for *Alu* repeats. Change the database from nr to ALU. Proceed with the search as described in the steps for the nr database search. Print the results.

Search for expressed sequences

1. If your sequence is part of an expressed protein sequence it should be present in the human EST database. Return to the search page and initiate a new search using the same sequence. Change the database from the ALU database to the human_EST database. Proceed with the search as described above. Print your results.

Search for the Chromosome Position of the Query Sequence

In order to find the chromosomal position of your sequence we will use a human sequence database located in the Sanger Center. Type the new address in the Internet explorer window: www.ensembl.org/Homo_sapiens/blastview/. Open this site.

1. Click on the "submit a BLAST query" window and pass your sequence to it. Open **edit** in Internet explorer and click **paste**. Your sequence should appear in the window.

2. Initiate a search by clicking on the **search** button. A new window will appear with a "BLAST retrieval ID" box. Read the explanation on how to retrieve BLAST results and follow them.

3. Click on the ID number. A new window will appear. It will have a schematic picture of all human chromosomes. Colored boxes or arrowheads indicate the best scored regions. Red colors indicate the best alignment. Arrowheads (blue or green) indicate other positions of partial homology.

4. Outline the entire picture of chromosomes with the mouse pointer.

5. Print the result. Click on the **file** button of Internet explorer. Choose **print** and the radio button **selected**. Start printing by clicking on the **OK** button.

6. Return to the result window and scroll down to the list of matched sequences. The first number in each row indicates the chromosome number and the next (in red) is the name of the sequence, which is followed by the score value and *E* value. Click on the name of the sequence. You will see alignment between the query sequence and the sequence in the database. Print this page.

7. Return to the window with the chromosome picture. Move the cursor over the arrowhead of the box with the best alignment (red box). A little window will appear. Click on the **show in Contigview** sign. A new window will appear with a detailed view of the position of your sequence on the chromosome, i.e. its region and band numbers. The positions of nearby known genes and DNA marker sequences (labeled D) are also indicated. Outline the **overview** window with the cursor and print it as described above.

8. Scroll down to the **detailed view** box. It contains a picture representation of details of the protein-coding regions, the positions of exons and introns, etc., located close to your sequence, as well as the positions of known mRNA and a list of homologous proteins from other organisms. This window will also indicate whether your sequence is part of a known or predicted human protein. Move the cursor over any filled rectangle that indicates an exon sequence. First, click on any exon (if it exists) in the line "human proteins." A little window will appear, indicating the name of a protein. Click on the **protein homology** sign. The sequence of the protein will appear. Record this data. Next click on any rectangle in the **protein** row. Click on the **protein homology** sign and the window that lists all homologous proteins from other organisms will appear. Make a note as to what these proteins are and what are their functions.

9. Move to the next step of analysis, single-sequence analysis.

You will perform a single-sequence analysis. First you will determine the base frequency of your sequence. Next you will make a restriction enzyme map and perform dot matrix analysis. You will use the DNASIS program for this analysis. This program is available both for Mac and PC platforms. Any other program suites such as Staden or GCG can also be used for this analysis.

Converting file formats

In order to perform single-sequence analysis you need to export your sequences from the **Sequencher** program. We will use the GenBank file format. You will export your sequences to your folder in this format. To do so, follow this procedure.

1. Click on a file to be exported in the Sequencher project window.

2. Open the **file** menu and choose the **import & export** window. Choose **export sequence(s)** from this window. The export window will appear.

3. Choose the folder in this window to which you will export this file. Choose your folder and highlight it. Next, change the extension of your file in the **export as** window from **abi** to **seq**. Do not change the name of your file only its extension (for example, your file name may be myfile.seq).

4. Open the **file format** window and change the file format to the **GenBank** format. Click the select button.

5. Export the file by clicking the **save** button.

6. Print your file. Open your folder with the exported file. Highlight the file that you want to print and drag it to the word-editing program icon (e.g. BBedit for Macintosh computers or Word Edit for PCs). Open the **file** menu and choose **print**.

7. Alternatively, you can convert the file using the READSEQ program. Outline the sequence in the Sequencher window and copy it into computer memory.

8. Open the WWW site html://searchlauncher.bcm.tmc.edu/seq-util/seq-util.html in Internet explorer. Choose the **ReadSeq** button and click on **O** (full option button). In the new window open **edit** and click on **paste**. Your sequence will appear in the window. Choose the format GenBank and click on the **perform conversion** button. Click on **save as**, change the file name, and give it seq extensions (e.g. myfile.seq). Change the file format to text and save it into your folder. You can also print this file using the **print** command.

9. Export all your files following one of the described procedures.

Base content analysis

Open the **DNASIS** program by clicking on its icon. Import your files into this program.

1. Open the **file** menu and click on **open**.

2. Open your folder and click on your file with extension ***.seq**. A note will appear that indicates that the program cannot identify this file type. Click the **open** button anyway. A window will open for file identification. Click on the **DNA** button and click **OK**. A window with your DNA sequence will appear. Make sure that there is no other text incorporated other than the DNA sequences (i.e. only A, C, G, or T) at the beginning of this file.

3. Import all your files into DNASIS following the procedure described.

4. Click on the **function** menu and choose **content** and from the content menu choose **base content**. A window will open to set the parameters for this function. It will contain the name of the file and several parameter settings. Choose **window size** 50 and bases G and C.

5. Click on **go**. A graph of base content will appear. You can increase the size of the window by dragging one of its corners. You can also increase the size of the graph (red) in the window by dragging its corner. **Print this graph**.

Restriction enzyme site analysis

Next you will analyze restriction enzyme sites. Use only one of your files for this analysis. Choose one of the sequences by clicking on it and follow the procedure described below.

1. Open the **function** menu and choose **search**. From the **search** menu, choose **restriction enzymes**. A window will open for setting the parameters of the restriction enzyme search. Do not change any parameters. Click on the **go** button. After a short time, a window will open with results. This window tabulates all of the results. Increase the size of the widow by dragging its corner and inspect the results. You do not need to print them.

2. In the upper left corner of the result window, you can see **several icons**. These icons control how data are presented. The first icon, which depicts **tables**, is marked. Right under it is another icon that depicts **circle/line**. Click on it. You will see a presentation of the data in graphic form. Each enzyme-cutting site is presented as a single line with vertical marks. The number of cuts is also indicated. Print this graph by clicking on **print** in the **file** menu. In the print window that opens select **pages from** 1 to 2. This will print only **two pages**. Close the window.

Dot matrix analysis

In order to perform a dot matrix analysis choose one of your sequences that was imported into the DNASIS program. Choose the sequence by clicking on it and follow the procedure described below.

1. Click on the **function** menu and choose **compare** and click on **homology plot**.

2. A window will appear that controls the parameters of this function. **Vertical sequence** will have the file name that you chose to analyze. **Horizontal file** does not contain any sequence. Click on **file selection** and choose the same file that is listed in **vertical sequence**. Change the **check size** to ten and **matching base** to eight or nine. Press the **go** button.

3. A window will appear displaying the results. You should increase the size of this window by dragging one of it corners. The graph will remain small. You should increase the graph size to the size of the window by dragging the corner of the graph.

4. If present a region of direct repeats will appear as a parallel line to the diagonal lines some distance from it (see Fig. 6.6 for an explanation). The distance of the line from the diagonal indicates the distance between tandem repeats. In order to see more detail of the plot in this region, click on the graph icon located in the upper left corner of the data window. Using the arrow, outline the region of the graph that you want to see in detail (hold the mouse button when you make this outline). Release the button and an enlarged graph in the specified region will appear. If you want to move back to the previous size; double click in any area of the graph.

5. Print the enlarged graph. Click on **file** and choose **print**. **Choose print page 1 to 1 only**.