Advanced Pattern Recognition for Expressed Sequence Tag Analysis

A thesis submitted to the University of Manchester for the Degree of Doctor of Philosophy in the Faculty of Science

Crispin John MILLER

School of Biological Sciences 2000.

Table of contents

Table of figures

Abstract

Sequence databases such as EMBL are enormous, and growing rapidly (at the time of writing EMBL is doubling in size approximately every nine months). This, combined with the demands placed by new technologies such as ESTs, genomic sequencing and Single Nucleotide Polymorphism (SNP) analysis places great demands on the computers used to search and compare sequences.

This thesis describes a novel algorithm RAPID, designed to address some of these issues by performing fast word based searches of biological sequences. RAPID has been shown to be about an order of magnitude faster than BLAST, but to perform with similar sensitivity. During the development of the algorithm an investigation of the distribution of words within biological database was undertaken. The work demonstrates that a simple model of biological sequences that views them as random sequences with differing residue frequencies is unable to accurately represent the distribution of words within sequences. This has some important implications for word searching algorithms.

RAPID is supported by a set of companion tools – a pair of alignment tools that generate gapped and ungapped alignments, weighted by word frequencies, and a 'coarse grained dot plotter' that provides a view of similarity between long sequences such as genomes, chromosomes and contigs.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or institute of learning.

Copyright

Acknowledgements

Many people have helped me in the development and exploration of the ideas in this thesis. Most notably, Andy Brass for keeping me pointed in the right direction and my friends for valuable discussions – and for putting up with the appalling puns. Of these there were many, but the particularly long suffering ones: Chris Taylor, Karen Eilbeck, Richard Moore, Robert Stevens and Terri Attwood deserve a special thanks.

Much of the data in the results section were produced in collaboration with Mike Cornell and Sandra Eilbeck, to whom I am extremely grateful.

The work in chapter 5 and some of the results in 6 have been published in the journal Bioinformatics (Miller *et al.* 1999).

The work in this thesis was supported by a BBSRC CASE award with Pfizer UK Ltd. under the industrial supervision of Giles Day.

To mum and dad

1 Introduction

This thesis describes the design and implementation of a set of software tools for Expressed Sequence Tag (EST) analysis, and their generalisation to other tasks including the identification of vector contamination and the comparison between large sequence fragments such as genomes and assembled contigs. The design of these tools involved two distinct considerations: Firstly, the development of a set of algorithms to perform sensitive and efficient sequence comparison. Secondly, the intended use of these algorithms, their role within the field of bioinformatics and the resultant user-interface and software design issues.

1.1 General considerations

Bioinformatics is a term used to describe the development and use of software tools for the biological sciences. It is a broad discipline encompassing areas as diverse as protein structure prediction (Sayle and Milner-White 1995; Sussman *et al.* 1998; Sternberg *et al.* 1999), sequence analysis (Schuler 1998), taxonomy (Doolittle 1999), and expression analysis (Fields and Sternglanz 1994: Kurian *et al.* 1999; Nature Genetics Editorial 1999), amongst others. These research areas are distinct, but they are united by a number of common issues that result from the need to manage large amounts of complex data distributed across a network. These problems are compounded by the fact that the data are often poorly understood, rapidly changing and of varying quality.

Bioinformatics is an exploratory discipline in which a set of tools and data sources are used to answer *ad hoc* queries which cannot be pre-determined. Thus,

it is necessary to consider general modes of operation as well as specific queries and analyses.

Bioinformatics software is also united by its role as a service provider: it exists to support the biological scientist in his/her work. In many cases, the user has not had any formal training in computing, so that the software must be accessible to a novice with little computational experience beyond the use of a word processor or web browser.

As a result, the software designer is faced with a set of contradictory challenges which arise from the tension between the need to apply sophisticated computer science techniques and the need to render the software accessible to a non-computer scientist.

The following two chapters provide a review of the issues, principles, and current work in the field. The first considers the biological processes that are used to generate sequence data, with particular emphasis placed on Expressed Sequence Tags (ESTs) and genomic sequencing. The second, a review of current sequence analysis algorithms.

1.2 Motivations

The DNA sequence database EMBL (Stoesser *et al.* 1999) is a large repository which, at the time of writing, contains about 4 Gb of sequence data and in recent years, has been doubling in size every nine months or so. The major contributors to this growth are Expressed Sequence Tags (ESTs), which now make up about

2/3 of public domain sequences. Commercial databases such as Incyte's 'LifeSeq' (http://www.incyte.com) are entirely EST driven and are growing even more rapidly than their public domain peers – at the time of writing, Incyte contains about 4,000,000 EST sequences.

It is unlikely that the growth rate of biological databases is going to decline in the near future – ESTs will continue to be sequenced at an increasing rate, and other technologies such as whole genome sequencing and Single Nucleotide polymorphism (SNP) analysis will further add to the mass of new data.

It is also the case that the number of requests made of sequence database servers has been growing systematically – with a corresponding increase in the load placed on their hardware and software.

Thus, to simply maintain the status quo, the computers that handle sequence data need to more than double in power every nine months, and to continue to do this for the foreseeable future.

Given that Moore's law predicts that computer hardware will double in power every *eighteen* months, there is likely to be an increasing shortfall between hardware capacity and server demand. In order to meet this challenge it will either be necessary to buy increasingly expensive hardware, or to develop software that can:

1. Perform the same tasks as current algorithms but significantly faster,

2. do this in a way that scales favourably with the growth in data and

3. make efficient use of available computer hardware.

Thus, significant advances are required merely to maintain the status quo. The issue becomes even more pressing when future demands on sequence analysis tools are considered. This is because bioinformatics is a discipline that is moving beyond the analysis of individual gene sequences and into a 'post genome era' in which consideration of the behaviour and interactions of multiple genes/proteins is becoming increasingly important.

As a result, there is a requirement to make bulk queries against sequence databases in which, for example, an entire genome, a set of interacting proteins, or a gene family need to be searched against a database rather than a single sequence.

So, sequence analysis tools are faced with a significant challenge if they are to meet the demands of the next millennium – they must be capable of managing enormous and rapidly growing data sets, an increasing user base and significantly more complex queries.

In order to meet these challenges, computer technology needs to continue to increase in performance, in concert with the development of novel, faster similarity search tools.

This thesis describes the design, implementation, analysis and assessment of a set of algorithms and software tools designed to address the issues described above.

2 The generation of DNA sequences

This section describes the biological processes used to generate a DNA sequence. Its purpose is to provide a general overview of techniques so that the subsequent chapters on bioinformatics have a sufficient grounding in the principles underlying the creation of biological sequences.

The first section provides an overview of DNA cloning and the different types of vector system that are employed. The second describes Sequence Tagged Sites or STSs. The third introduces the notion of Expressed Sequence Tags or ESTs, and the fourth describes the methods used to produce the complete sequence of a large DNA molecule such as a chromosome or bacterial genome.

2.1 Cloning

DNA cloning allows multiple copies of a DNA sequence to be produced. It is essential for operations such as DNA sequencing, and allows the production of libraries. Libraries are large collections of cloned DNA fragments from a particular organism, tissue, organ or cell type, maintained within a host such as *E. coli*. Libraries production results in a large DNA sequence being broken up to make a set of smaller fragments which are suitable for sequencing. Cloning can also facilitate the production of single stranded template DNA suitable for dideoxy sequencing if a vector such as M13 is used. The cloning process can result in the production of erroneous sequences; it is introduced here in preparation for the section on vector contamination in Chapter 6.

Cloning involves the use of a host organism such as *E. coli* and a cloning vector. A cloning vector is a DNA molecule into which a DNA fragment can be introduced, *in vitro*. The resultant molecules can then be introduced into living cells in which they can be propagated. Many cloning vectors exist, most of which are based around bacteriophages or plasmids. Bacteriophages are bacterial viruses. Plasmids are genetic elements composed of DNA or RNA that exist in both prokaryotes and eukaryotes. They carry genetic information, are not part of the chromosome, but can propagate themselves autonomously.

The DNA sequence to be cloned is first digested using a restriction enzyme, or randomly sheared using a method such as sonication. This results in a set of fragments, each of which can be cloned, or alternatively, multiple copies of a single fragment can be selected by cutting its band from an agarose gel.

The cloning vector is also cut, using a restriction enzyme that has a single recognition point within the vector sequence. The DNA fragments and the linearised vector sequences are ligated together to produce a sequence containing the vector and the clone, which is referred to as the insert.

The resultant recombinant DNA molecules are introduced into the host bacteria either by transformation, enhanced by soaking the cells with calcium chloride, or via infectious bacteriophage particles (*in vitro* packaging, see below). Once the vectors have been inserted, they direct the production of multiple copies of themselves. The replication process is such that only one recombinant molecule

can propagate within a single host, so that after a period of time, each host contains multiple identical copies of a single recombinant DNA molecule.

Cloning vectors are typically less than 10Kb in length (before insertion). This makes them easier to manipulate, and also allows a restriction enzyme to be found which only has one recognition site within the sequence. This is because the longer a sequence, the more likely a given restriction enzyme site will occur more than once. For example, an 8 base pair restriction site will have an average occurrence in a random sequence of once every 4^8=65536 base pairs. Cloning vectors must also replicate efficiently. Plasmids and bacteriophages fit these criteria for bacterial hosts, but plasmids are uncommon in eukaryotes. Thus, eukaryotic vectors tend to be derived from viruses.

Vectors have marker genes that allow cells containing the vector to be identified, and also for recombinant DNA molecules to be distinguished from those which do not contain an insert. A number of different markers exist, such as antibiotic resistance, histochemical markers (which result in a colour change) and nutritional markers which allow transformed cells to grow on media lacking a specific nutrient that would otherwise be required.

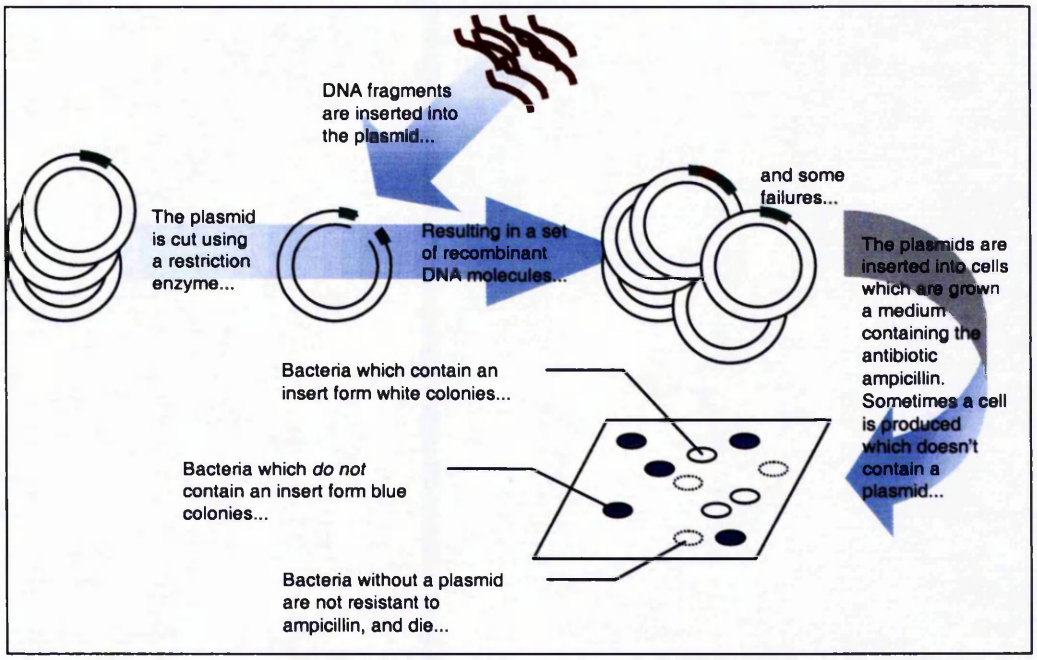

Figure 1 Diagram showing the cloning process using the plasmid vector pUC8.

For example, pUC8 is a plasmid vector with a cluster of recognition sites for the restriction enzymes *Hind*III, *Pst*I, *Sal*I, *Acc*I, *Hinc*II, *BamH*I, *Sma1*, *Xma*I and *EcoR*I. These occur within the *LacZ'* gene which codes for the enzyme β-galactosidase. In plasmids which have had a sequence successfully inserted, this gene is inactivated. β-galactosidase cleaves the disaccharide X-gal (5-bromo-4-chlor-3-indolyl-β-D-galactopyranoside), which is similar to lactose, into its component sugars – one of which is coloured blue. Thus, cells grown on agar containing X-gal form blue colonies if they do not contain an insert in the *LacZ'* gene, whilst those with an insert are white. pUC8 also contains a gene for ampicillin resistance. Together these markers allow colonies from bacteria which contain vector and insert to be distinguished from other colonies. This process is summarised in Figure 1.

2.1.1 λ phage vectors

Plasmid vectors over about 10Kb in size are liable to undergo rearrangement, and become difficult to work with. An alternative system based on the vector λ phage can handle inserts of up to 18Kb (an upper bound placed by the size of the phage particle) (Brown 1999). The λ genome is 48.5Kb, of which a 15Kb central segment contains genes that are only necessary for the integration of the phage DNA into the *E. coli* genome. Thus, this region can be removed without interfering with the phage's ability to replicate. The λ genome is linear, so that the removal of its central segment results in two arms between which DNA is inserted.

Three types of λ systems exist: insertion vectors, replacement vectors and cosmid vectors. Insertion vectors consist of the two arms joined at a restriction site that is used for the insertion of new DNA. Replacement vectors contain a 'stuffer fragment' that is replaced by the insert when it is ligated into the vector.

2.1.2 Cosmid vectors

Cosmid vectors are produced by incorporating the λ *cos* site into a plasmid. λ *cos* is a site required for the assembly of a DNA sequence into a phage particle. Ligation is arranged so that when DNA is cloned into the cosmids, they join together to form linear chains. When placed in the appropriate '*in vitro* packaging mix', these chains rearrange into 'λ-genomes'. The packaging mix is a set of proteins that self assemble into phage particles in the presence of λ *cos*.

Since the original plasmid can be as short as 8Kb, a cosmid can accept an insert of up to 44Kb.

2.1.3 M13 phage vectors

The M13 phage has a single stranded DNA genome that, after infection of the *E. coli* bacterium is converted into a double stranded replicative form. This replicates until approximately 100 copies exist, and the copy number is maintained after cell division by further replication. At the same time, the cell continues to secrete single stranded M13 phage particles – approximately 1000 per generation. M13 is of particular interest because it produces single stranded DNA which is required for Chain Termination sequencing. M13 systems suffer from rearrangement when the inserts are greater than about 3Kb.

2.1.4 Phagemids

Phagemids are produced by incorporating the origin of M13 (or another single stranded phage) into a plasmid vector, as well as the plasmid's own origin. When accompanied by a helper phage, carrying genes for the phage replication enzymes and coat proteins, an *E. coli* cell produces single stranded copies of the phagemid DNA. This system avoids the instabilities associated with M13 vectors (allowing inserts of up to 10Kb to be successfully cloned), whilst producing single stranded DNA suitable for sequencing.

Although the vectors described above are sufficient for cloning the relatively short sequences required for shotgun sequencing, vectors which can handle larger

inserts are required for techniques such as Clone Contig Sequencing[1] - see section 2.4.2 on page 12.

2.1.5 Yeast Artificial Chromosomes

Yeast artificial chromosomes or YACs (Burke *et al.* 1987) consist of a centromere, a pair of telomeres and at least one origin of replication. These are linked together with one or more marker genes and a restriction enzyme site at which new DNA is inserted. These components can be placed in a DNA molecule of about 12Kb in length.

Standard YACs are able to take inserts of 600Kb, with recent ones able to handle fragments as large as 1400Kb. Unfortunately, YACs are prone to problems with insert stability and sometimes become rearranged by recombination. (Anderson 1993).

2.1.6 Bacterial Artificial Chromosomes

For this reason, Bacterial Artificial Chromosomes or BACs (Shizuya *et al.* 1992) are the preferred vector for many tasks, such as the Human Genome Project

---

[1] Some small genomes have been sequenced using cosmid vectors, notably, *Saccharomyces cerevisiae* (Oliver *et al.*, 1992). However, the process is laborious, and the desire for larger inserts has resulted in the development of a number of vectors capable of taking much longer sequences.

(Cohen *et al.* 1993) which has abandoned YACs in favour of BACs because of fears over instability. BACs are derived from the *E. coli* F plasmid, which is relatively large. They can accept inserts of up to 300Kb.

Other vectors, such as bacteriophage P1 (Sternberg 1990), P1-derived artificial chromosomes or PACs (Ioannou *et al.* 1994) and Fosmids (Kim *et al.* 1992) are also of interest because of their ability to take relatively large inserts.

2.1.7 Summary

DNA cloning is a complex multistage process. The complexity introduces a number of potential sources of error, most notably from contamination of or by host or vector sequences. Different host organisms and vectors are used in different circumstances so that a large number of different erroneous sequences can be introduced into a sequence database. The type and level of contamination that results from cloning errors is discussed in chapter 6.

2.2 Sequenced Tag Sites or STSs

When sequences are generated manually the electrophoresis process places an upper bound of about 1500 residues (although this is rare – typically resolution is up to about 400bp) before the bands on the gel become too close to be resolved. Automatic sequencers have an upper bound of about 400 residues; with a significant increase in error rates after about 300.

The region of a template that is sequenced is determined by the type of sequencing primer that is used. Universal primers adhere to the vector adjacent to the insert. Thus, universal primers can be used to sequence either end of the insert, up to a length determined by the resolution of the autoradiograph. Internal primers anneal within the insert itself and can be used to generate the whole sequence by using a set of different primers to generate sequences distributed across the insert. These can be assembled into the complete sequence.

Once an insert has been sequenced, it is desirable to place it on a physical map of a genome or chromosome. This is typically done by relating its position to the location of a sequence-tagged site, or STS. STSs are short (~500bp) regions of DNA that have known sequence and are known to occur uniquely within a genome. There are a number of possible sources for Sequenced Tag sites:

- ESTs (see below), from genes which are known to be unique, or from the 3' non-coding region of the gene, which is less well conserved between multicopy genes than the coding region.

- Simple Sequence Length Polymorphisms or SSLPs. These are repeat regions that show different lengths.

- Random sequences taken from a database.

STS mapping takes a library of sequence fragments and identifies those that contain individual STSs. This allows the fragments to be ordered by identifying

shared STSs. The resolution of the map is determined by the fragment size. There are various possible sources of fragments, but for genomic sequencing, the contig library is eminently suitable, because the information derived from STS mapping can then be used to identify overlapping clones, which facilitates assembly.

2.3 Expressed Sequence Tags or ESTs

The work described in this thesis arose from the need to manage and interpret the large amount of data produced by EST sequencing.

ESTs are of particular interest because they represent the portion of a genome which is being transcribed into mRNA. For human beings, this represents less than 5% of the total genomic DNA (Gerhold and Caskey 1996).

ESTs are generated by the partial sequencing of randomly selected cDNA clones, and thus represent fragments of expressed genes. cDNA clones are produced by selecting mRNAs from a cell by means of their poly(A) tails and then using reverse transcriptase to copy them into cDNA. The resultant cDNAs are cloned into plasmids and replicated in *E. coli*. The relative abundance of specific cDNAs in the library is therefore related to the proportions of mRNAs expressed in a cell (Goodfellow 1995).

The correlation between mRNA and protein levels is, however, not entirely straightforward. One reason for this is that gene expression involves both transcription and translation; only the former is considered by cDNA analysis. Another reason is that post translational modification often means that the

protein transcript is not the one that is finally exhibited in a cell. Finally, mRNA degredation rates vary significantly: mRNA that exists in a cell for a long time will produce more protein than one that is rapidly degraded. For example Lange and Hengge-Aronis (1994) show that the cellular concentration of the sigma S subunit of RNA polymerase in *E. coli* is dependent on transcription, translation and protein stability. Gammie *et al.* (1999) draw the same conclusion for karyogamy transcription factor Kar4p in yeast.

Given the above provisos (especially when the *difference* in expression is being measured), the analysis of random cDNA clones is a useful technique that has a long pedigree stretching back to Costanzo *et al.* (1983), who used it to investigate liver proteins.

EST sequencing itself is a relatively new technology, originally described by Wilcox *et al.* (1991) and Okubu *et al.* (1992). Recently it has been pursued vigorously by Venter at the National Institute of Health (NIH) (Adams *et al.* 1992) and forms the basis of companies such as Incyte and Celera.

EST sequencing relies on the use of automation to rapidly produce sequences from randomly selected cDNA clones. The sequences are generated in a single pass using universal primers and are typically about 300-400 bp long – sufficient to identify the gene they code for.

Large scale sequencing projects such as Merck-WashU, and Incyte's LifeSeq database are exploiting this technology to provide rapid coverage of coding

regions in a number of genomes including *H. sapiens*. At the time of writing, LifeSeq contains 4,000,000 ESTs, representing between 100,000 and 120,000 human genes (which Incyte claim to be over 90% of the expressed genes in the human genome).

ESTs can be sequenced from either the 5' or 3' end of the cDNA insert. Typically, mRNAs are purified from total RNA by using an oligo(dT)-linked Sephadex column. This is a device which extracts mRNA by passing it through a column which contains a primer that anneals to the mRNA's poly(A) tail. The primer is physically bonded to a substrate, so that the mRNA anneals to the primer and becomes attached to the column. First strand synthesis of the cDNA is primed using an oligo(dT) that anneals to the poly(A) tail of the mRNA. Creation of the cDNA by reverse transcription results in a set of sequences all ending at the 3' end but of varying lengths. A primer containing a restriction enzyme such as *Eco*R1 is annealed at the 5' end. The oligo(dT) primer has a different restriction enzyme such as *Not*I included and, with the 5' end primer allows the cDNA to be cloned with a fixed direction into a vector. The vector has a pair of (different) primers on either side of the insert site, which allows the cDNA to be sequenced either from the 5' or the 3' end. The 3' poly(A) terminus does not normally contain a coding region. Thus, ESTs sequenced from the 3' end typically do not contain coding sequence; instead they contain the tail of the 3' untranslated region, or UTR. ESTs sequenced from the 5' end contain varying regions of the same gene, and typically represent a portion of the gene's coding region.

Many libraries are normalised to reduce the abundance of highly expressed mRNA in order to cut the amount of redundancy, which otherwise results in the same gene being sequenced many times at the expense of lowly expressed genes (Soares *et al.* 1994).

2.3.1 Differential Expression Analysis

Systematic cDNA sequencing offers the potential for differential expression analysis, but quantification at low expression levels requires a large amount of sequencing: genes at the 1:10,000 abundance level requires the sequencing of between 50,000 and 100,000 clones for each library. (Jordan, B 1998). Alternative approaches using hybridisation probes and large arrays of targets offer the chance to analyse the expression of large quantities of genes in one experiment. These technologies are mentioned briefly here because they depend on ESTs, either for their source data or in the analysis of their results.

- High density membranes are prepared by spotting bacterial colonies onto Nylon filters (Nguyen *et al.* 1995; Zhao *et al.* 1995; Pietu *et al.* 1996; Gress *et al.* 1992). The bacteria are grown and treated using standard techniques to extract, prepare and attach the bacterial DNA to the support. It is also possible to spot PCR amplified cDNA directly onto the membrane.

- Microarrays are produced by placing 0.5 – 1.0 kb cDNAs onto a glass substrate. In effect they are the high density membrane technique writ small (Southern *et al.* 1992).

- Oligonucleotide chips are generated by synthesising short oligonucleotide probes onto a glass membrane (Khrapko *et al.* 1991). Fodor *et al.* (1991) have developed a technique which allows these to be generated *in situ* using photo-lithographic techniques.

All these technologies benefit from large libraries of EST sequences, representing a high proportion of human genes, both in the decision as to which probes to place on the substrate, and in the analysis of the results of an experiment. It is likely that the trend away from expression profiling using completely uncharacterised clones will continue as EST libraries increase in size and as the ESTs themselves are combined with mapping data.

2.3.2 Mapping

ESTs themselves are useful for mapping – those representing the 3' untranslated region of genes are used as STSs in the production of a high resolution map of a genome (Boguski and Schuler 1995). 3' UTRs are used because:

- they do not contain introns – their PCR product is the same size as that produced from a genomic template.

- Their sequences are less well conserved than coding regions, so that it is easier to distinguish between multi-copy genes.

Like many other applications of ESTs, mapping is complicated by the high redundancy of EST data. For this reason, the NCBI embarked on the UniGene project which clusters ESTs around genomic sequence data.

2.3.3 Positional Cloning

Related to mapping is the desire to identify 'interesting' regions of a genome that are, for example, involved in a particular disease state. Positional cloning is a strategy used to isolate a gene whose gene product is completely unknown. It starts from a knowledge of the gene's position on the chromosome. Since ESTs represent the coding regions of a genome, they can be used to rapidly identify regions of a genome. ESTs offer the opportunity to speed up positional cloning, by allowing a region of genomic DNA to be rapidly 'skimmed' by low density EST sequencing. This allows the region to be searched for putative genes that can then be matched to known genomic/EST sequences. Combined with linkage analysis, which attempts to correlate the position of polymorphic markers with a disease gene (typically within 1-10 Mb), such an approach offers great potential for therapeutic sequence analysis.

EST sequences also offer the opportunity to localise genes to large genomic DNA clones such as YACs, BACs, and PACs by hybridising genomic probes to arrays of ESTs spotted on nylon membranes.

2.3.4 Error rates

Whilst automation and high throughput techniques are attractive in that they are able to produce large amounts of data quickly and inexpensively, the approach results in relatively 'dirty' data.

In an analysis of sequences in dbEST (Boguski *et al.* 1993), the EST subset of GenBank (Benson *et al.* 1999), Wolfsberg and Landsman (1997) searched the database using 15 human genomic gene sequences. The resulting ESTs were aligned to the genomic DNA, and the alignments studied. For one gene, 73% of the ESTs which derive from spliced or partially spliced transcripts contained introns or were spliced at previously unreported sites. Other genes showed differing amount of variation. In a related analysis of pairs of ESTs purporting to arise from the same gene, 26% do not both align with the appropriate piece of genomic DNA. The authors suggest that this is a result of artefacts in the EST process and urge caution in the treatment of EST data.

One side effect of the high throughput process by which ESTs are generated is that they have a relatively high sequencing error rate. The average fidelity is being about 97%.

The analysis of artefacts in EST data is taken further by Aaronson *et al.* (1996). They compared ESTs to a set of human transcripts from EGAD, the Expressed Gene Anatomy Database (White and Kerlavage 1996). Within the set of ESTs which showed similarity (by BLAST (Altshul *et al.* 1990) and Smith-Waterman

alignment (Smith and Waterman 1981) to EGAD, they identified a number of sequencing anomalies:

1. Reversed clones, mislabelled ends

2. Lane tracking errors

3. Insert size

4. Internal priming/alternative translation

These are discussed in more detail in the following sections.

2.3.4.1 Reversed clones/mislabelled ends.

The EST libraries used in their analysis were derived from unidirectionally cloned inserts and labelled to be 5'-3' or 3'-5'. By comparing the sequences to genomic DNA, Aaronson *et al.* were able to determine ESTs within the database which were incorrectly labelled. Labelling of the ESTs in the database was varied: either by placing "5'" or "3'" labels as unstructured text in the dbEST/GenBank entries, or by placing 's/r' in the read_id suffix to encode 3'/5'. 0.39% of the sequences they analysed used both methods of annotation, and they were conflicting – signifying errors at the annotation stage. Since sequences are represented in the direction they are read from the gel, 3' ends should match to the non-coding strand of the genomic sequence, 5' ends to the coding strand. A reversed clone appears as a 5' sequence labelled as a 3', or a 3' sequence labelled

as a 5'. In order to analyse reversal, the set of ESTs from a clone were aligned with genomic DNA, and the direction determined by the alignment compared to their annotation. Whilst the majority of clones were correctly annotated, 5% were entirely reversed and 0.5% were mixed reversed/non-reversed. In general normalised (Soares *et al.* 1994) libraries showed higher rates of reversal than non-normalised libraries.

2.3.4.2 Lane tracking errors

Associating a sequence with the correct lane on a sequencing gel is clearly a pre-requisite of successful annotation. In order to asses the level of lane tracking errors, Aaronson *et al.* identified 5' and 3' sequences from the same clone which align to different genes. Approximately 1% of sequence pairs behaved in this way – assuming that only one of each sequence pair is incorrectly assigned, this corresponds to an error rate of about 0.5%.

2.3.4.3 Insert size

The WashU-Merck project associates insert size with ESTs. If the 3' and 5' end of an insert are known, it is possible to determine the insert size by matching the ends to a known transcript sequence. This is complicated by the fact that insert size can vary as a result of alternative splicing, and care needs to be taken with multicopy genes. Aaronson *et al.* report an average error rate in the determination of insert size of 21.5%.

2.3.4.4 Internal priming/alternative translation

The 3'UTR region of mRNA is the most varied part of the sequence and it is considered to be possible to use it to uniquely map a clone to a transcript. Since 3' ESTs are generated by priming from an oligo(dT) primer that anneals to the poly(A) tail, all 3' ESTs should be anchored to the poly(A) tail. This is not always the case, for a number of reasons:

Internal priming to A-rich regions upstream of the poly(A) tail during reverse transcription can result in 3' ESTs which are generated from upstream of the 3' terminus of the sequence.

Alternative 3' ends. The poly(A) tail of an mRNA is added by the cell to pre-mRNA on recognition of a terminal signal in the pre-mRNA sequence. Some transcripts contain a number of canonical poly(A) signals, resulting in alternative 3' ends.

In order to assess the abundance of sequences resulting from these events, Aaronson *et al.* identified ESTs which did not fall at the 3' end of transcripts. These were classified into three sets: (1) near a canonical poly(A) signal, suggesting an alternative 3' end. (2) near a poly(A) rich region, suggesting an internal priming event. (3) neither near an A-rich region or near a canonical poly(A) signal. The latter subset may contain sequences that result from an internal priming event not adequately described by the poly(A) rich criterion described above. Internal priming was observed to occur with a rate approaching 3%.

2.3.5 Summary

Section 2.1 described the process by which sequences are cloned and drew attention to the process as a source of potential sequencing errors. ESTs are a rapidly generated resource and tend to be relatively unreliable. Given their potential as a resource for biological research there is an incentive to generate methods for screening ESTs for contaminations and errors. This is discussed in more detail in chapter 6.

2.4 Genomic Sequencing

2.4.1 Shotgun Sequencing

Shotgun sequencing simply takes the results from a set of sequencing experiments and assembles them into one large sequence by finding the overlaps between fragments. It does not require any kind of physical map or prior knowledge about a sequence. However, the number of comparisons which must be made is equal to $n^2 - 2n$, where n is the number of fragments the original sequence is split into. Thus, the number of inter-fragment comparisons can become prohibitively large as the size of the sequence increases.

In 1995, Fleishmann *et al.* (1995) demonstrated the validity of the approach by sequencing the entire 1.803 Mb *Haemophilus influenzae* genome entirely by using the shotgun method.

The genome was first split into fragments using sonication. The fragments were sorted using electrophoresis, and those between 1.6 kb and 2.0 kb were cloned

into a plasmid vector, resulting in a library containing 19,687 clones. 23,643 individual sequences were produced from the library, by sequencing from the vector using universal primers. Thus, the ends of the inserts were sequenced, corresponding to a total length of 11.6 Mb. These were assembled to produce 140 contigs.

A number of methods were employed to close the gaps between contigs. Firstly, the library was searched for inserts whose ends were located in different contigs. For these inserts, the sequence was closed by sequencing across the insert using internal primers. This closed 99 of the gaps, leaving 42 physical gaps which corresponded to sequences that did not occur in the clone library – probably as a result of sequence instability.

Physical gaps were closed by preparing a new library using a different vector (λ phage). Appropriate clones to sequence were selected by a combination of two approaches. Firstly, oligonucleotide primers corresponding to the ends of the contigs were used to probe the library. Clones to which a pair of primers annealed were selected. These corresponded to a gap and were sequenced using internal primers. Alternatively, pairs of primers were used to carry out PCR. Only sequences that contained the template for the primer pair produced a PCR product. The PCR product corresponded to a gap and was sequenced.

Shotgun sequencing is eminently suitable for sequencing small genomes; its strength is its speed and lack of dependence on a physical map.

Its principal shortcomings are the complexity of the data analysis, and the issues that arise from repetitive regions within a sequence - which can result in mis-assembly. Directed shotgun sequencing (see section 2.4.3) offers potential solutions to some of these problems.

2.4.2 Clone Contig Sequencing

The clone contig approach breaks a large sequence into smaller fragments (of about 5 Mb) which are then sequenced using a shotgun approach. These large fragments are generally anchored onto a physical map so they can be identified and analysed using features such as STSs. The clone contig approach relies on cloning vectors such as YACs and BACs that are capable of handling large inserts.

2.4.2.1 Chromosome Walking

Chromosome walking starts with a clone, identifies a second clone from the library which overlaps with the first clone, a third clone which overlaps with the second, and so on. One way to do this is to use the insert from a clone as a hybridisation probe with which to screen the other clones in the library. Problems arise when the probe contains repetitive DNA which hybridises with DNA in a number of different clones. This can be reduced by pre-hybridising with unlabelled, repetitive, genomic DNA, but this is not effective with the large inserts from YACs or BACs. In this situation, the end of an insert is used as a probe (it is shorter, so less likely to contain a repetitive sequence). The probe can be sequenced in advance to confirm that it does not contain a repetitive region. If

the end has been sequenced, PCR can be used to identify a neighbouring clone instead of hybridisation.

This approach is generally used for positional cloning, where the objective is to walk to a gene that is some distance (15 Mb, for example) away in the physical map. It is generally too slow to be applied in situations where there are more than 15 contigs to be walked.

2.4.2.2 Clone Fingerprinting

Clone fingerprinting identifies neighbouring clones by generating a coarse map of some features in each clone and then identifies similar patters between clones. The map can be generated by a number of methods, such as:

- restriction patterns,

- repetitive fingerprints, which carries out Southern hybridisation experiments using probes for various known genome wide repeats

- Repetitive DNA PCR which generates PCR products from the clone using primers to genome wide repeats. Since repeats are not evenly spaced, different sized products are produced, and these can be used as a fingerprint.

- STS content mapping. STS mapping uses primers directed at different STSs. If each STS is unique, then two clones that contain the same STS must

overlap. STS mapping is also appealing because it allows contigs to be anchored to a physical map.

2.4.3 Directed Shotgun Sequencing

Shotgun sequencing results in about 8 times as much sequence being produced as the original genome being sequenced. For the human genome, this corresponds to about 70 million 500 base pair sequences. This is a tractable amount of data, but it is considered too hard to assemble the fragments in the correct order, partly because of repetitive regions and partly because of the computational effort required to perform the assembly. For this reason, the directed shotgun approach has been proposed, which makes use of the physical map during assembly (Venter *et al.* 1998). This is currently underway using a combination of 2Kb and 10kb clones in a different plasmid vectors. The 10Kb clones are large enough to entirely contain the majority of repetitive sequences found in human DNA; they should help resolve the problems of mis-assembly when sequencing around these regions. It is believed that by using the STS map of the human genome, it will be possible to assemble the master sequence correctly. However, doubts as to whether this is achievable mean that the human genome project continues, using BACs as the principal vehicle for cloning.

2.5 Summary

This chapter has described the principal methods used for obtaining DNA sequences from biological samples. From a bioinformatics perspective, it is important to recognise the complexity of the process. Errors can arise at many

stages and vary in range from mis-reading of a sequencing gel or trace, contamination by vector or host DNA, through to complex biological processes such as alternative splicing and inversion. As a result, sequence data are not the pure repository of information a computer scientist would like them to be, and there are many pitfalls to traps a naïve user. Part of the motivation for the work described in this thesis was to generate tools capable of searching through the vast amount of sequence data in order to identify errors such as contamination, and to generate cleaner DNA databases. The results of these endeavours are presented in chapter 6. Fast algorithms are also required to locate the overlapping ends of sequences and to assemble them into contigs. This is particularly true with shotgun and directed shotgun sequencing, where there is a lack of mapping information to help with the ordering of the sequences.

3 Current bioinformatics sequence analysis tools

At the time of writing, the EST subset of EMBL (release 60) contains 3.95Mb nucleotides and is continuing to grow exponentially as it has done since its inception – see Figure 2. (Stoesser *et al.* 1999). It is of a size that defies manual analysis.

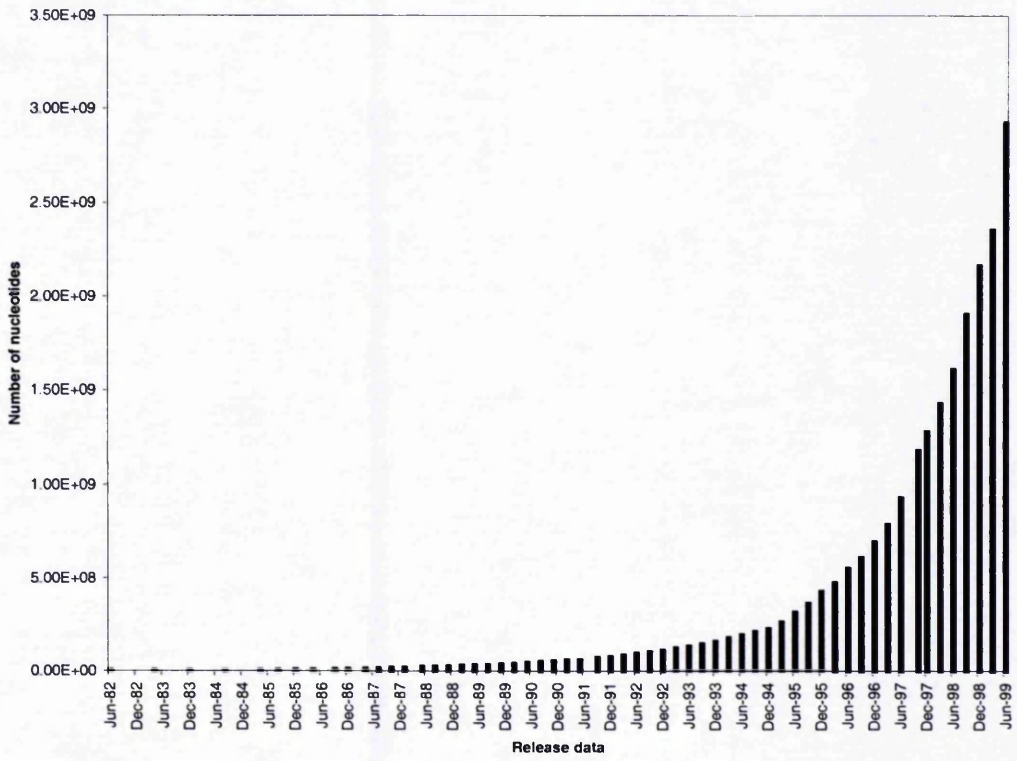

Figure 2 Growth of the EMBL DNA database since its inception in 1982.

Databases perform a number of services to the biological community. At one extreme, they are simply a repository of experimental results in which scientists can submit their data for peer review and analysis. At the other, they are valuable

data sources for performing *in silico* biology. Both views of databases provide significant challenges to the information management community.

3.1.1 Repositories

A database which is being used as a repository needs to represent the complex, diverse and interconnected information which drives biology. This is exemplified by the fact that only about 30% of a database such as EMBL is sequence: the rest is annotation describing things such as gene function, tissue type, organisms and bibliographic references. Simply *representing* such information is sufficient if the only mode of access is by browsing, or by reference to a previously identified entry. For example, a user might follow a link to a specified entry by using an accession number quoted in a paper, or presented in the output of a similarity search tool (see below).

However, a database such as EMBL, cannot be browsed – it contains over 4.7 million entries. For this reason it is necessary to do more than just represent the data: infrastructure is also required to allow it to be searched for records which match a specified criterion.

Unfortunately, this is difficult with biological systems. One reason is that, with a few notable exceptions (Rodriguez-Tome and Lijunzaad 1997; Skupski *et al.* 1999; Blake *et al.* 1999; Attwood *et al.* 1999), the majority of bioinformatics databases are supplied as flat files without a generic query engine or a published schema. This is unfortunate given the amount of effort that has been spent over the last 29 years (Codd E.F. 1970; Chen P 1976) developing relational database

technologies for representing large amounts of information in a principled and structured manner, and for extracting data from the resulting databases. The situation is made more complex by the fact that some databases, such as EMBL, are stored in a relational database, but are not available in that form. Instead, *ad hoc* systems have developed for the indexing and querying of flat files. Currently, the most widely used of these is the indexing and search software SRS (Etzold *et al.* 1996), which is a sophisticated system for performing text based searches on partially structured text files.

It is also the case, however, that bioinformatics data is significantly more challenging to represent than the kind of information that is carried in typical relational databases such as airline booking and payroll systems. This is because biology is a discipline based on knowledge, and that knowledge is complex – describing amongst other things, the form, function and interaction of proteins and small molecules.

This kind of information is generally represented in databases as a mixture of free text and partially structured hierarchies based on keyword taxonomies, such as that found in SWISS-PROT (Bairoch & Apweiler 1999). The expressive nature of the English language makes this kind of representation less than ideal for databases that are going to be searched by computer. For example, a search to retrieve all transposable elements from EMBL requires searches for *transposon*, *transposase*, *mariner*, *ty element* and probably many others. Further, the content and integrity of some databases are not effectively controlled, so that, for

example, fields intended to be annotated using a controlled vocabulary contain unauthorised entries.

Consequently, there is considerable interest in the bioinformatics community in representing biological knowledge in a formally structured framework that allows relationships such as '*transposon* and *mariner* are both types of *transposable element*' to be expressed unambiguously. In order to do this effectively it is necessary to represent relationships other than *isAKindOf*, such as *isAPartOf*, *interactsWith* and *isImplicatedIn*.

For example, EcoCyc (Karp *et al.* 1998) uses a frame based system developed by Artificial Intelligence researchers to represent metabolic pathways in *E. coli*. RiboWeb (Chen *et al.* 1997) is a knowledge base of the most common experiments used to study the structure of RNA/Protein complexes. TAMBIS (Baker *et al.* 1998;1999) uses a representation of key concepts in bioinformatics and the relationships between them to facilitate the generation of database queries which can then be performed across multiple information sources. INTERACT (Eilbeck *et al.* 1999) uses an Object Oriented Database Management System (OODBMS) to represent information on protein-protein interactions derived from a number of sources such as yeast 2-hybrid and co-immunoprecipitation. The system uses 'wrappers' to provide a relatively uniform interface to a set of resources distributed across the Internet. Information is extracted from these resources and placed in a data warehouse built around the interaction data. This warehouse contains information such as protein homologues, motifs, and bibliographic references.

3.1.2 Resources for *in Silico* biology

Sequence databases are very much more than simply being electronic filing cabinets, they are resources in their own right. They can be used to help identify the function, behaviour or properties of proteins, genes and other sequences elements such as polymorphisms and promoter sites. The principal methods employed involve comparing protein or DNA sequences in an attempt to find ones which are similar. The hypothesis is that similar sequences have similar properties and hence are likely to perform a similar function. Similarity searches across sequences in a database are the main method of identifying database records and can be viewed as providing the 'hooks' that allow entries to be retrieved and browsed by a user. Thus, an uncharacterised sequence is searched against a database of sequences with known function to provide a set of annotations, which, it is hoped, will provide information that can be used to infer the function of the initial query.

A number of pattern-matching algorithms have been developed for comparing DNA and protein sequences against each other; this section details the principal ones, after a brief discussion of the relationship between sequence similarity and homology.

3.2 Similarity vs. homology

States and Boguski (1991) highlight the distinction between similarity and homology:

"Similarity is a descriptive term which implies that two sequences, by some criterion, resemble each other and carries no suggestion as to their origin or ancestry. Homology refers specifically to similarity due to descent from a common ancestor."

They point out that although the words 'similarity' and 'homology' are often used interchangeably they have different implications, and that whilst it may be possible to infer homology from sequence similarity, *"...outside of an explicit laboratory model system, descent from a common ancestor remains hypothetical."* They use an information theoretical argument to show that the length and combinatorial nature of a protein sequence implies that it is unlikely that two random sequences would carry 'the same message'.

They argue - with reference to (Patterson, 1988) - that convergent evolution at the molecular scale should be extremely rare, and that the observation of similar sequences of sufficient length can be accepted as evidence for homology. The same argument can be made to generalise to (longer) DNA sequences.

They also explore what can be deduced from an implied homology between two sequences. In the examples that follow it is demonstrated that:

- homologous sequences do not necessarily perform a similar function: Haptoglobin is not a functional proteinase although it is often considered part of the Serine Proteinase family.

- even in proteins where the majority of the sequence has been replaced, it may be possible to establish homology from the sequence which remains (e.g. trypsin vs. chymotrypsin).

- homology may occur between whole proteins, or just between one domain amongst many: trypsin vs. prothrombin.

- structural similarity need not infer homology, because migration of exon/intron junctions can introduce or delete sequences and structural motifs (Craik *et al.,* 1983).

In a brief discussion of the meaning and implications of 'homology', three different biological properties are considered: function, structure and evolutionary relationships. Only the latter is a direct consequence of sequence homology, the other two are properties which might (or might not) be predicted as a result of an inferred homologous relationship.

When other properties of sequences are taken into account (such as: cleavage sites, promoter sites, areas of contamination, repeat regions, likelihood of primer adhesion, protein sorting, gene transposition and intron/exon prediction), it is apparent that there is a large set of biological questions which might be asked of a sequence analysis program.

Thus, there are many interesting biological relationships apart from homology – and homology is not *necessarily* the most useful basis for their investigation.

This means that whilst percentage similarity is often a useful basis with which to infer homology, in situations where the question that is *really* being asked is a different one – such as function prediction, for example, - an alternative metric of similarity might be more useful. One motivation behind the work described in this thesis was to explore whether alternative methods of similarity searching might in some situations perform better than an alignment based approach.

States and Boguski, by exploring the difference between similarity and homology, also draw the distinction between a mathematical algorithm and the biological relationship the algorithm is intended to infer. A necessary implication of this is that *any consideration of the ability of an algorithm to answer a biological question must be judged in terms of biology, not just a statistical analysis of the algorithm.*

The rest of this chapter considers some of the different tools and techniques that have been produced for sequence analysis.

Sequence comparison techniques can be divided into five distinct types: graphical methods, alignments, word-searching, profiling and motif searching.

3.3 Graphical methods

Graphical methods make use of a human being's ability to spot patterns in images. The most common technique – the Dot Plot – is generally attributed to Gibbs and McKintyre, with numerous modifications and improvements by other groups (Gibbs and McKintyre 1970; McLachlan 1971, 1972, 1983; Maizel and

Lenck 1981; Staden 1982; Pustell and Kafatos 1982, 1984; Argos 1987; Reisner and Bucholtz 1988).

In its simplest form, the dot plot represents similarity between a pair of sequences by:

1. placing each sequence on the axis of a plane, and
2. plotting a point on the plane $p(i, j)$ whenever residue i of sequence x is the same as residue j of sequence y.

In the resultant image, diagonal lines with a gradient of 1 represent regions of similarity between the two sequences, and repetitive regions appear as a grid-like section of the image containing a regular pattern of diagonal lines which appear off the main diagonal – refer to Figure 3.

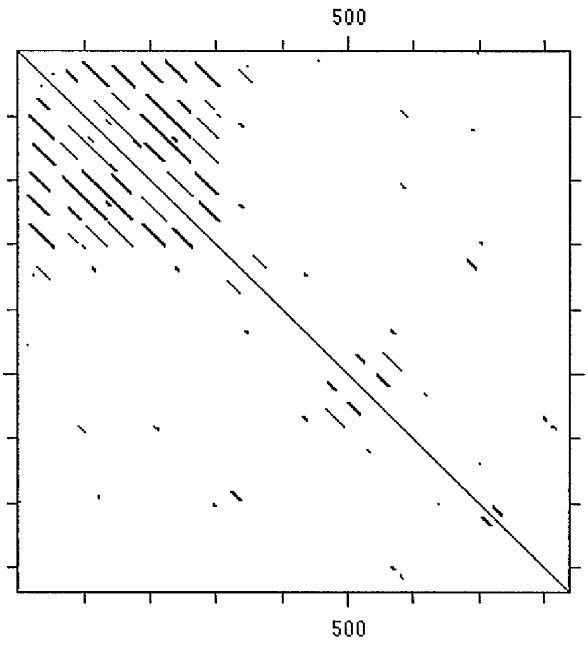

Figure 3 A dot plot of human LDL receptor against itself showing a repeat region.

This simple form of dot plot suffers from problems with noise – particularly at the DNA level where (assuming equal base composition) one in four residues are likely to match. Improvements can be made by scanning a window across each sequence and scoring the match between the windows. Points are then plotted with intensity as a function of score. The simplest scoring scheme simply counts the number of matching residues between each window so that, for example:

'ACGACA'
| | || |
'AAGAGA'

scores 4/6.

Dot plots provide an instantaneous and easy to interpret view of similarity, but in their simplest form do not offer any kind of automation. Attempts have been

made to estimate the statistical significance of dot plots by performing a Monti Carlo analysis of windowing scores (Lipman and Pearson 1985; Pearson 1990). The analysis contrasts the distribution of scores produced by comparing a pair of sequences with the distribution of scores produced after randomly shuffling the sequences (which maintains residue frequencies). The result of such a comparison is a set of outlier points representing 'interesting' sequences, or alternatively a set of z-scores. Other techniques estimate significance by comparing the scores to a theoretical model based on random sequences. Argos uses real protein sequences to generate statistics for significance tests (Argos 1987).

3.4 Optimal alignments

Alignments represent similarity by writing a pair of sequences next to each other and highlighting matching residues. Generally, alignments allow gaps to be made in the sequences. Since gaps may be of arbitrary length, and may be inserted between any pair of residues, a large number of different alignments may be produced from a pair of sequences. Alignments are scored; the highest scoring one being referred to as the optimal alignment.

If it is assumed that the sequences to be aligned have length N, a comparison without gaps has a time complexity of $O(N^2)$. When gaps are allowed, this comparison must be repeated 2N times giving a complexity $O(N^3)$. Needleman and Wunsch (1970) introduced 'dynamic programming' to significantly reduce the time complexity of a gapped alignment to $O(N^2)$.

Dynamic programming is a standard computational technique that keeps a record of previous steps in a calculation so that they might be used in the future, avoiding the same result being recomputed many times. It is often used to solve optimisation problems that satisfy the *principle of optimality*: in an optimal sequence of decisions or choices, each subsequence must also be optimal (Brassard and Bratley 1988).

The scoring system used in the Needleman and Wunsch algorithm satisfies this by using the following recursive definition of the alignment score:

3.5 Optimal alignment scores

The score S_{ij} for an alignment ending in residue i from sequence 1 and residue j from sequence 2 is calculated as follows:

$$S_{ij} = s_{ij} + \max\left\{\begin{array}{c} S_{i-1\,j-1} \\ \max\limits_{2<x<i}(S_{i-x\,j-1} + w_{x-1}) \\ \max\limits_{2<y<j}(S_{i-1\,j-y} + w_{y-1}) \end{array}\right\} \qquad [1]$$

where s_{ij} is the score for aligning residue i with j, and

W_x, W_y are the scores for making gaps of length x and y, respectively, in sequences 1 and 2.

Dynamic programming is often referred to as a 'bottom up' technique, in that it usually starts with the smallest subinstances of a bigger problem. The Needleman and Wunsch algorithm proceeds by calculating the scores for alignments of increasing size, and recording those scores in a matrix M, where each cell

$M(i, j) = S_{ij}$. Thus, the addition of a pair of residues to the alignment can be found by computing s_{ij}, and finding $S_{i-1\,j-1}$ from the matrix. The insertion of a gap in sequence 2 corresponds to a horizontal jump through the matrix of length x, (to find $S_{i-x\,j}$), and a gap in sequence 1 corresponds to a vertical jump of length y.

When the algorithm is complete, the optimal alignment score is represented by the largest cell at the edge of M.

If the alignment itself (rather than just its score) is required, it is necessary to maintain a record of the path taken through the matrix. This can be done by using a 'traceback' matrix T, which records the move made to generate each cell in M.

A diagonal move is typically represented by the value '0', a horizontal one by recording $-x$, and a vertical one by recording y.

3.6 Gap penalties

The way the gap penalty W is calculated can have a significant effect on the alignment produced. Needleman and Wunsch applied a single penalty that was the same irrespective of the length of the gap. Sellers used a penalty that was proportional to length (Sellers 1974), and Smith and Waterman (1981) use a penalty w_x:

$$w_x = g + lx \qquad [2]$$

where g is a gap opening penalty,

l is a gap extension penalty, and

x is the length of the gap.

This is considered to provide a better model of insertions and deletions because gaps can be represented as being hard to open initially, but easy to extend once open. Modifying the gap opening and extension penalties can have a significant effect on the alignment that is produced: it is often advisable to perform an analysis a number of times with different penalties.

3.7 Global vs. local alignments

The Needlman and Wunsch algorithm described above produces a 'global alignment' in which all the residues in the sequences are aligned. Such an alignment can result in a situation where a short but well conserved region is missed because it is out-weighed by the rest of the alignment. In a significant modification to Needlman-Wunsch, Smith and Waterman produced a 'local alignment' algorithm designed to find short conserved regions between pairs of

sequences (Smith and Waterman, 1981). The algorithm modifies Needlman-Wunsch by requiring that:

1. Mismatch scores must be negative.
2. The minimal value a cell in the alignment matrix takes is zero.
3. The optimal alignment may end anywhere in the matrix, not just in the final row or column.

3.8 Scoring schemes

The score assigned to a match between two residues also has a significant effect on an alignment. For global alignments of DNA sequences it is common to use an identity scheme that simply scores 1 for a match and 0 for a mismatch. For local alignments, where the mismatch score must be negative, it is usual to score a match 1 and a mismatch -1. For protein sequences, identity matrices ignore the fact that different amino acids share similar biological properties, so that, for example, it is common to see one hydrophobic residue substituted by another. As a result, a number of scoring schemes have arisen based on the chemical or mutational properties of amino acids.

3.8.1 PAM matrices

Point Accepted Mutation (PAM) matrices (Dayhoff *et al.* 1978) model evolutionary change as a set of uncorrelated amino acid point mutations. The PAM-1 matrix tabulates the observed probability for each amino acid that it will mutate to each of the other amino acids, when the average rate of mutation is 1 in 100 residues. This data set was produced by examining the alignments of a set of

proteins that were more than 85% similar (closely related sequences were chosen so that they could be unambiguously aligned). Since the mutations are considered to be uncorrelated, the PAM model of evolution allows the PAM-1 matrix to be used to generate matrices for situations where the average rate of mutation is greater than 1%. For example, the PAM-120 matrix can be produced by multiplying the PAM-1 matrix by itself 120 times. PAM matrices are often represented as a log odds matrix, where each cell corresponds to the probability of the mutation occurring divided by the probability that the two residues may be aligned by chance. For mathematical convenience, the logarithm of this value is taken.

PAM matrices have been criticised for a number of reasons:

Firstly, the model assumes that all residues in a protein are equally likely to mutate. This is certainly not the case, as can be seen by examining a multiple alignment of proteins, where certain residues are clearly conserved. Secondly, by starting with a set of proteins that were highly conserved, Dayhoff *et al.* produced data for the most mutable residues in the proteins – which may be an incorrect starting point, given it is the highly conserved residues that are of interest when comparing diverse sequences. Finally, the initial data set was based on small globular proteins; the applicability of data derived from this set to other protein families has been questioned (States, DJ. and Boguski, MS. 1991).

3.8.2 BLOSUM matrices

Blocks Substitution Matrices (BLOSUM) matrices provide an alternative to PAM matrices for representing amino acid substitutions (Henikoff & Henikoff 1992). The matrices are often preferred to PAM matrices because they are designed to represent mutations for distant relationships - something that can only be inferred from a PAM matrix.

BLOSUM matrices are derived from 'blocks' in the Blocks database (Henikoff *et al.* 1999a; Henikoff *et al.* 1999b). A block is an ungapped section of a multiple alignment representing a conserved region of a protein family. The first step in the production of the matrix is to compile a table of the observed amino acid substitution frequencies for columns in a block. This is used to calculate a log odds matrix representing the odds ratio between observed frequencies and those to be expected by chance. In order to reduce multiple contributions to amino acid pair frequencies from the most closely related members of a family, sequences in a block are clustered according to percentage ID. Thus, for example, with a threshold of 80%, sequences A and B will be placed in a cluster if they are more than 80% similar. Sequence C will be placed in the same cluster if it is similar to either A, or B.

All sequences in a cluster are counted as a single sequence in the computation of the BLOSUM matrix, so that their contribution to the matrix is reduced.

By varying the clustering threshold, a family of matrices can be produced - BLOSUM80 for example, represents a matrix built with a threshold of 80%.

3.9 Heuristic alignment tools

Even with the use of dynamic programming to reduce the complexity of optimal algorithms such as Smith-Waterman, the size of biological databases has resulted in a desire for faster methods of alignment generation.

Heuristic algorithms such as BLAST (Altschul *et al.* 1990) and FASTA (Pearson & Lipman 1988) satisfy this demand by trading optimality for speed.

The algorithms use a heuristic (a rule of thumb) to rapidly identify potentially high scoring alignments from the much larger set of possible alignments. This prunes the search space that has to be considered by the algorithm, reducing its time complexity. However, because the algorithm uses a heuristic, it is no longer guaranteed to produce the optimal alignment. Generally, the alignment is good enough – particularly since the speed of the algorithm allows it to be produced in circumstances where it would otherwise be too expensive to generate.

Both BLAST and FASTA use 'word' searches to identify short matching regions between sequence pairs. A word is simply a k residue sub-sequence of a larger one. Thus, a word that is shared between a pair of sequences can be viewed as a short (k long) alignment. Both BLAST and FASTA use matching words to 'seed' alignments which are then scored and ranked. The difference between the algorithms lies in the scoring method they employ and the way the initial word-matching phase is performed.

3.9.1 BLAST

BLAST (Altschul *et al.* 1990) is a program that finds high scoring alignments between a query sequence and a target database. BLAST is able to do this very quickly because it is only required to find a good alignment, rather than an optimal one.

BLAST works on the principle that an optimal alignment is likely to contain at least one short region of identities. It uses this heuristic to first generate a set of short sequences which would match the query sequence with a score greater than a specified threshold (for DNA these sequences are typically 11 bp long, for proteins, 3 residues). The words are subsequently compared against the database to be searched. Each time a match is found, the algorithm attempts to extend the match at either side to generate an ungapped local alignment. The alignments are scored, sorted and presented to the user as a text file – although a number of front ends now exist which parse the output file and make it more attractive, for example, Power BLAST (Zhang and Madden 1997).

3.9.1.1 BLAST statistics

BLAST's similarity measure begins with a matrix of similarity scores for all possible residue pairs. For protein sequences the PAM family of matrices (Dayhoff *et al.* 1978), or BLOSUM (Henikoff and Henikoff 1992), are generally used; for DNA, identities are typically scored +5, mismatches -4.

A maximal segment pair (MSP) is defined as the highest scoring pair of identical length segments chosen from two sequences. An MSP is scored by generating the

sum of the matrix scores for each residue pair in the MSP. BLAST calculates this score heuristically, and attempts to find the set of MSPs which score above a specified threshold.

The statistical significance of an MSP score is calculated with respect to a random model (Karlin & Altschul, 1990; Altshul *et al.* 1990) which considers MSP scores to be an extreme value distribution. For random sequences this can be shown to be the case.

3.9.1.1.1 Extreme value distributions

If two sequences are compared using a tool such as BLAST, a set of Segment Pairs are produced, each with its own score. If it is assumed that the Segment Pairs are independent of one another, their scores can be considered to be independent and identically distributed. This means that the sum of their scores should tend to a normal distribution.

BLAST, however, generates and scores *Maximal* Segment Pairs, or MSPs – which are the *highest* scoring Segment Pairs in a search. It can be shown that these tend to an Extreme Value Distribution (Altshul and Gish 1996; Altshul *et al.* 1994):

$$P(S < x) = \exp(-e^{-\lambda(x-u)}) \qquad [3]$$

The equation gives the probability that the optimal sub-alignment score from the comparison of two random sequences of length m and n is less than some score, x.

The distribution is described by two parameters, the *characteristic value, u*, which can be thought of as the centre of the distribution, and the *decay constant*, λ, which is a scaling parameter. u and λ can be determined analytically: λ is the unique positive solution for x in:

$$\sum_{i,j=1}^{r} p_i p_j e^{s_{ij}x} = 1, \qquad [4]$$

where the distribution of residues in the sequences are defined by $p_1, p_2 \ldots p_r$, and the score for a pairwise match between residues is s_{ij} (as determined by a scoring matrix such as a PAM or BLOSUM matrix).

u may be calculated from the size of the sequences, m, and n:

$$u = (\ln Kmn)/\lambda \qquad [5]$$

K is a constant which, like λ, is dependent on the sequence composition of the database and the scoring matrix employed.

Combining [3] and [5] allows u to be eliminated; the probability that the optimal ungapped alignment score $S \geq x$ can then be written:

$$P(S \geq x) = 1 - \exp(-Kmn\, e^{-\lambda x}) \qquad [6]$$

3.9.1.1.2 Expectation Score

It is possible to compute the expected number of MSPs for a pair of sequences with lengths m and n respectively using the following formula:

$$E = Kmne^{-\lambda S} \qquad [7]$$

3.9.1.1.3 p-score

The following formula gives the probability of a score greater than or equal to x occurring by chance:

$$p = 1 - \exp(e^{-\lambda(x-u)}) \qquad [8]$$

p gives the probability that a pair of sequences would result in a score greater than x. For a search against an entire database containing D sequences, it is necessary to consider the fact that a high p-value can occur between the query sequence and any of the database sequences. If it is assumed the sequences in the database are random, this can be modelled using a Poisson distribution:

$$P \approx 1 - e^{-Dp} \text{ which, for } p < 0.1 \text{ approximates to } Dp. \qquad [9]$$

An alternative approach considers the database to be populated by a set of proteins which consist of multiple domains. With this view of the data, D should be replaced by N/n, where N is the number of residues in the database, and n is

the length of the region of interest. For DNA sequences, such a normalisation is particularly relevant - since the database entries do not typically represent natural units of sequence.

Finally, a pair of sequences may generate more than one MSP. BLAST assumes that the number of MSPs matching between a pair of sequences with a score greater than x is distributed as a Poisson distribution $e^{-\lambda(x-u)}$. Note that p-scores and E values are basically the same thing: p-scores are simply E values scaled by the database size.

3.9.2 FASTA

The other widely used heuristic algorithm is FASTA, (Pearson, W.R. & Lipman, D.J. 1988: Pearson, W.R. 1990). FASTA proceeds first by identifying all identically matching words between a pair of sequences via a lookup table. For DNA, words are typically 4-6 bp long, for proteins, 1 or 2 residues.

Once matching words have been identified, the algorithm places them in an xy plane, one sequence per axis. The diagonals of this plane are searched to find regions that contain a high density of matching words, using the 'diagonal method' (Pearson, W.R. & Lipman, D.J. 1988) which counts word matches on a diagonal whilst penalising intervening mismatches. These regions constitute an ungapped local alignment between the sequence pair. The ten highest scoring regions are re-scanned using a similarity matrix (such as PAM250 for proteins); the top scoring region is referred to as the *init1* score. This is treated as a measure of pair-wise similarity and is used for ranking the hits against database

sequences. Finally, FASTA attempts to combine regions together by looking on nearby diagonals for another region that could be incorporated by means of an insertion or deletion. This is assessed by using a 'joining' penalty which is similar to the gap penalty employed by other algorithms. This alignment of initial regions is computed using a dynamic programming algorithm, and produces the *initn* score which FASTA uses to rank its hits.

3.9.2.1 FASTA statistics

FASTA presents a histogram of scores generated by searching a query sequence against a library of target sequences. It also calculates the score's mean and standard deviations. The FASTA package comes with the program RDF2 that compares the query sequence with randomly shuffled versions of the potentially similar database sequence. It can be used to further explore the statistical significance of the match.

3.9.3 A discussion of statistical significance

Much has been made of the various levels of statistical rigor employed by different sequence comparison algorithms, and it is worth considering this in some detail. Firstly, Needlman-Wunsch, Smith-Waterman, BLAST and FASTA all score their matches in a similar way - by generating an alignment score based on summing the match/mismatch scores for the aligned residues. In this respect, the difference between the algorithms is dependent only on how close they get to the optimal alignment score for a pair of sequences.

It is in their estimates of statistical significance that the algorithms differ. So, when Altschul *et al.* say "This tractability to mathematical analysis is a crucial feature of the BLAST algorithm" they are referring to the fact that a statistical model has been created which allows an estimate to be made of the probability that a given match occurred by chance. BLAST uses this estimate (the so-called *p*-value) to rank its hits, and to provide an appropriate cut-off below which matches are not returned.

Karlin and Altschul (1990) refer to their statistical model as 'appropriate'. It is built on a theory for a single sequence which is produced by sampling from an alphabet $A=\{a_1,a_2...a_r\}$, with probabilities $\{p_1, p_2...p_r\}$. Thus, the sequence is random, with no Markov dependency between successive elements (although it has generalisations to models that do have a Markov dependence). The results in chapter 4 of this thesis show that the composition of sequences in biological databases is far from random, and that the independence of residues assumed by the Karlin-Altschul model is not an accurate representation of biological sequences. It cannot be used to predict the distribution of subsequences within a database.

It is also interesting to consider the relationship between *p*-scores and alignment scores. In general, as an alignment score is increased (either by raising the % identity, or the length of the alignment), the *p*-score will decrease. This is because the better the alignment, the less likely it is to have occurred by chance. Thus, there is a rough inverse-mapping between alignment score and *p*-score. The mapping is not exact because individual residue frequencies are used to

generate the random model upon which *p*-scores are based. The use of residue frequencies means that an alignment that contains a high proportion of common residues will have a higher *p*-score than other alignments consisting of rarer residues. So, *p*-scores provide a similar metric to alignment scores *except* that:

- they make a correction for biases in residue frequencies

- they provide a method for determining a cut-off below which hits can not be considered to be statistically significant.

This draws into question the whole utility of *p*-value statistics in the evaluation of similarity between biological sequences. Firstly, the cut-off required by the Karlin-Altschul model is generally ignored by biologists because *biologically* interesting relationships do not typically appear with *p*-values above about 0.001 - three orders of magnitude greater than predicted by the model. In other words, the cut-off is almost always set to be much more stringent than required by Karlin-Altschul statistics. Secondly, we must consider the effect that a *p*-value's correction for residue bias has on the ranking of a BLAST hit, and the user's perception of it. The correction has the effect of changing a sequence's ranking in the set of hits. When sequences are highly similar, the difference in *p*-value is inconsequential – the match is definitely interesting, and the difference between a *p*-score of 10^{-162} and 10^{-165} is meaningless. When the matches are much less similar (~10^{-4} for example) a biologist must apply their expertise to the problem, and again, the relative ordering imposed by the *p*-value scoring system is not a relevant discriminator. Thus, rigorous statistics such as Karlin-Altschul, are of

interest mathematically, but should not be used as a reason to choose one algorithm over another.

3.9.3.1.1 Bit scores

Perhaps in recognition of the problems associated with *p*-scores, recent versions of BLAST (Altschul *et al.* 1997) no longer report them in their output. Instead, they produce a *bit-score*. Bit scores attempt to unite different scoring schemes by normalising them onto a common curve defined by the extreme value distribution. Placing an arbitrary scoring scheme into a framework which has a common set of units allows the results of using a particular algorithm to be assessed without knowing its inner workings. Thus, for example, an alignment score computed with a PAM matrix may be directly compared with one generated using a BLOSUM matrix.

Bit scores can be calculated using the formula:

$$S' = \frac{\lambda S - \ln K}{\ln 2} \qquad [10]$$

3.10 Word searching

In the introduction to this thesis, a need for high-speed sequence comparison algorithms was identified. The previous sections of this chapter described alignment methods and the use of heuristics to improve the speed of database searches.

The approach also avoids the computational complexity of computing an alignment – again offering a significant performance increase over alignment programs.

3.10.1 EMBLSCAN

Bishop & Thompson (1984) devised a sequence comparison program, EMBLSCAN which did not attempt to compute alignments. Instead, it counted the number of common words between two sequences and used this as a measure of sequence similarity. Avoiding the overhead required to produce alignments results in an algorithm which approaches a time of $O(n)$, where n is the product of the sequence lengths. Since unrelated sequences are likely to share a number of words by chance, it ranked scores according to the predicted number of chance hits on the matching words.

The search process works on a pre-computed datastructure representing the position and identities of all the unique subsequences of length 7 which occur in the database to be searched. The data-structure represents the database as an array of lists - one for each possible word - allowing all the sequences which contain a particular word to be found by supplying the correct index into the array.

A statistical model is described which allows the number of matches expected by chance to be estimated.

3.10.1.1 EMBLSCAN's statistics

Consider a specific sub-sequence, q, length d. Each nucleotide in q is independently assigned a type (A,C,G or T) with possibly varying probabilities. The probability of q occurring in a sequence is defined as $P(q)$.

For a long sequence of n nucleotides, a, a binary sequence I is defined such that:

$I_j(q) = 1$ if the subsequence starting at the j^{th} position of S is q, and

$I_j(q) = 0$ otherwise.

The number of times q occurs within a is then:

$$\sum_{j=0}^{n-d+1} I_j(q) \qquad [11]$$

Further, since we assume nucleotide independence $P(I_j(q) = 1) = P(q)$ for each j, and the *expected* number of occurrences of q is

$$m(q) = (n - d + 1)P(q) \approx nP(q). \qquad [12]$$

The *distribution* of repeats is harder to deal with, since the overlapping of words renders them non-independent. It is modelled as a Poisson distribution, justified by the following argument about the independence of words in a.

Each $I_j(q)$ is dependent only on the $I_k(q)$ with $1 \le |k - j| \le d$, since only then do the words starting at j and k have positions in common. The occurrence of a

word at position j, has a strong influence on the chances of it occurring at position k.

For most words, which cannot overlap,

$$P(I_j(q)=1|I_k(q)=1)=0 \qquad [13]$$

However, the effect of a word *not* occurring is minimal. That is:

$$P(I_j(q)=1|I_k(q)\neq 1)=P(q) \qquad [14]$$

The combination of limited length dependence and the small effect of zero values together imply that the probability that only one of the $I_j(q)$ is 1 differs very little from that given by independence of $I_j(q)$.

The probability that a word occurs exactly once in a sequence is modelled by a Poisson distribution with mean $m(q)$:

$$P(q,1)=s(q)=m(q)e^{-m(q)} \qquad [15]$$

Bishop & Thompson note that "such a distribution is not accurate when large numbers of repeats occur because, in this situation, many of the $I_j(q)$ are 1". They also point out that under the assumption of independent nucleotide types, such probabilities are minute - where large numbers of repeats occur, these assumptions are invalid.

Using these statistics, Bishop and Thompson derive the expected number of matches M between two sequences:

$$M = \sum_{q} s_1(q)s_2(q) = \sum_{q} m_1(q)m_2(q)e^{-(m_1(q)+m_2(q))} \qquad [16]$$

EMBLSCAN functions with a word length of 7, and assumes that all words occur with equal probability (of 4^{-7}). This allows values of M to be rapidly computed, reaching a maximum when the sequence length is equal to the number of distinct sequences - for 7-mers this is $4^7 = 16384$.

Hits where the number of matches are greater than the expected value M are then selected for further analysis.

3.10.2 FLASH

FLASH is an algorithm for generic pattern recognition which uses 'probabilistic indexing' into a Hash Table (Rigoutsos and Califano 1994). It has been applied to tasks such as:

- speech recognition
- fingerprint recognition
- text retrieval
- searching DNA and protein sequence databases

It is the latter application that will be discussed here.

FLASH proceeds through two stages. The first, an offline single pre-processing step, computes a hashtable representing the database to be searched using an indexing scheme described below. In the second phase it searches that hashtable for a given query string using the same indexing string. The aim is to find all sequences in the database that are within a specified edit distance, m, from the query sequence. Edit distance is defined simply as the number of residues that must be changed to mutate from one string to another.

For a given string, S, FLASH uses the hashing function λ to generate an index into the hashtable, A, for each token S_i in S. λ may create more than one index for each token; this is represented by:

$$\left\{\lambda_k\left(S_i\right)\right\}_{k=1}^{d_I} \qquad [17]$$

d_I is the index density, describing the number of indexes generated per token. Each index defines a bin into which is stored the position, i, of the token being indexed and the identity of the string from whence it came. This process is repeated for all the strings in the database.

Once the hashtable has been built, the same index function is used to generate the set of indexes for a query sequence, Q. These indexes are used to index into A, and identify the location of all matching tokens to Q.

Since FLASH is aiming to find the alignment between Q and S which has the minimal edit distance (and hence the optimal %ID), a further step is required. The indexing positions on the query and database sequences define a point in a dot plot representing where both sequences match. Thus, by subtracting the indexing position on the query sequence from those on the database sequence and plotting the resultant 'absolute alignment positions', it is possible to find the alignment with the largest number of matching tokens. Note that this is similar to the 'diagonal method' employed by FASTA.

In its simplest form, FLASH generates a single index for each token, which is just the integer value of the word beginning at i. This approach has a number of problems; three of which are listed by Rigoutsos and Califano:

- The algorithm's speed is proportional to the size of the database.

- Most of the hits against A do not correspond to actual matches; rather, they are false matches between sequences which happen to share a word. As the database size increases the number of false matches increases.

- There is a maximum of one index per token. As a result, a minimal change in the query sequence can drastically reduce the number of matching tokens.

For these reasons, FLASH uses an alternative indexing function that generates a set of tokens for each position in the query sequence. It does this by scanning a window across the sequence to generate a set of words of size w. From this it

generates the set of v-tuples (where $v < w$) that start with the same residue as w, and are built from contiguous residues in w. For example, the 5-mer ACGTG generates the 3-mers ACG, ACT, ACG, AGT, AGG and ATG. The use of multiple indexes per token increases redundancy and hence allows the algorithm to find matches between more dissimilar sequences than the hashing scheme described above. This is done at the expense of significantly increased database size. It is not clear from the paper why increasing the number of indexes serves to reduce the speed/size dependency.

FLASH provides fast computation at the expense of high memory overhead. Like BLAST and FASTA, it attempts to generate an alignment – and by using the 'diagonal method', that alignment is a global one centred on the diagonal with the largest number of matching words.

3.10.3 The D^2 algorithm

D^2 is another word-based algorithm that works on the hypothesis that similar sequences share words (Torney *et al.* 1990).

D^2 proceeds by counting the number of times (multiplicity) each word occurs within a sequence and contrasting distributions between pairs of sequence to generate a similarity score. The algorithm allows words to be weighted to adjust their significance within the scoring system. The difference in multiplicity distributions, d^2 is used as a measure of dissimilarity between sequences:

$$d^2 = \sum_{n=1}^{u} \sum_{i}^{4^n} \rho_n(w_i)\left\{m_D(w_i) - m_Q(w_i)\right\}^2 \text{ , where} \qquad [18]$$

u =maximal word size,

n =subsequence length,

$\rho_n(w_i)$ =weight for word i,

$m_D(w_i)$ =multiplicity for word i in the database sequence and

$m_Q(w_i)$ =multiplicity for word i in the query sequence.

Thus, if two sequences share identical distributions they will generate a score of 0, and as the distributions become less and less similar, the score increases.

3.10.3.1 An application of D^2

The square root of the score is a distance metric suitable for clustering sequences, leading to the algorithm being employed in the Sequence Tag Alignment and Consensus Knowledgebase (STACK) project (Hide *et al.* 1999). STACK adds value to data in the Genome Sequence Database (GSDB) by performing analysis of EST sequence data. In particular it:

- Provides ESTs clustered according to 90% similarity.

- Alignments of EST clusters.

- Consensus ESTs (contigs) mapped to parent cDNAs or genomic sequences (where available).

- Annotation of alternate splice sites and consensus alternate splice sites

D^2 is used near the start of the STACK pipeline to generate initial clusters of ESTs which are then subjected to further analysis by other algorithms. Since D^2 does not consider the position of words within a sequence it is able to spot similarity between EST sequences that are similar but have undergone inversions and alternative splicing. This is a significant advantage over alignment algorithms such as FASTA.

3.10.4 Summary

Word searching algorithms avoid the need to compute alignments, a computationally intensive task. As such, they offer a potential route to the solution of the problems described in the introduction: the need to successfully search and manipulate the vast amount of information that is starting to swamp bioinformatics. It is generally assumed that word-searching algorithms, whilst being faster, are not as sensitive as alignment methods. The next chapter explores the way words are distributed in biological sequences in an attempt to generate a model of word distributions which can be used to augment a word-searching algorithm and increase its sensitivity.

4 The distribution of sub-sequences within biological databases

This chapter examines the distribution of sub-sequences within biological databases. It develops a statistical model designed to represent the distribution of words observed in biological databases, and shows that a simple model such as that upon which Karlin-Altschul statistics is based (Karlin & Altschul 1990) is not necessarily appropriate for modelling word distributions in databases. This has important consequences for the design and analysis of word matching algorithms and forms the basis of RAPID, which is described in Chapter 5.

4.1 A simple statistical model

As an initial step, a simple model is created which predicts the occurrence of sub-sequences by assuming biological sequences to be essentially random, with independence between adjacent residues. This is similar to the model upon which Karlin-Altschul statistics is based (Karlin & Altschul 1990).

Consider a random sequence, S, of length L, sampled from an alphabet of letters $A = \{a_1, a_2 ... a_r\}$ with probabilities $\{p_1, p_2 ... p_r\}$.

This section derives a simple model based on S, which is used to predict the distribution of k-mers within a biological database. In the next section, this distribution is compared to the real distribution observed in the yeast genome.

First, consider a specific subsequence, w, of length k.

It is assumed that the effect of overlapping words is not significant (with the same justification as that made in section 3.10.1.1).

4.1.1 Even letter composition

If the composition of letters in S is equal, then the distribution of w in S can simply be modelled by a Poisson distribution with mean L/r^k. This is confirmed by Figure 4 which shows a random sequence of length 13,390,000 and a Poisson distribution with mean $13{,}390{,}000/4^9$. The graph (and the subsequent graphs in this chapter) shows a histogram of word frequencies normalised to have an area of 1. Thus, in Figure 4 the majority of words occur just over 50 times in the sequence, and almost all words occur between 30 and 75 times. The graph also justifies the assumption that the effect of word-overlaps does not significantly change the chance of their occurrence – and hence their distribution within sequences.

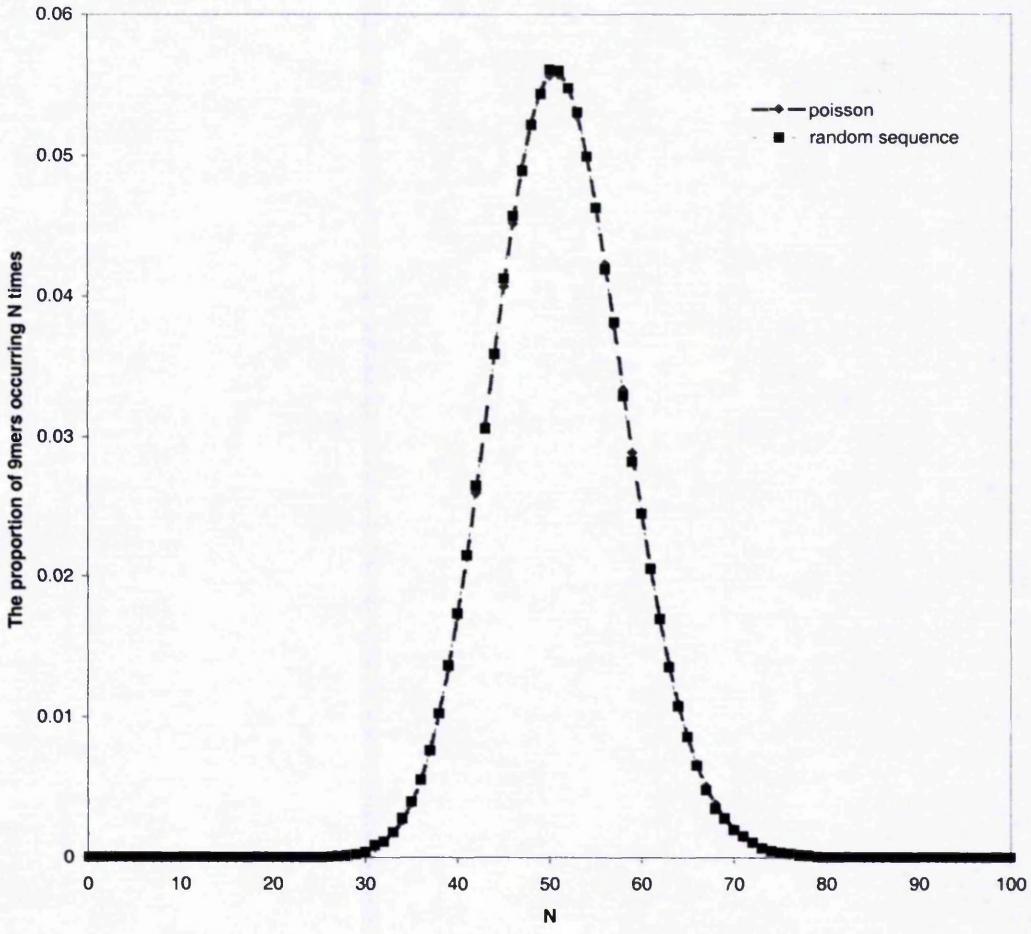

Figure 4 Comparison between a Poisson distribution mean 13,390,000/4^9, and the distribution of 9mers in a random DNA sequence length 13,390,000 built with equal residue composition. The graph is normalised to have an area of 1. The graph show the proportion of words that occur a specified number (N) of times For example, the majority of words in yeast occur just under 50 times, and almost all words occur between about 30 and 75 times.

4.1.2 Uneven letter composition

If the distribution of letters in a sequence is not equal, then a Poisson distribution is not sufficient to model the distribution of *k*-mers within that sequence. This is shown in Figure 5 which depicts the distribution of 9-mers in the yeast genome: yeast has a skewed residue composition (19% 'c', 19% 'g', 31 % 'a', 31% 't').

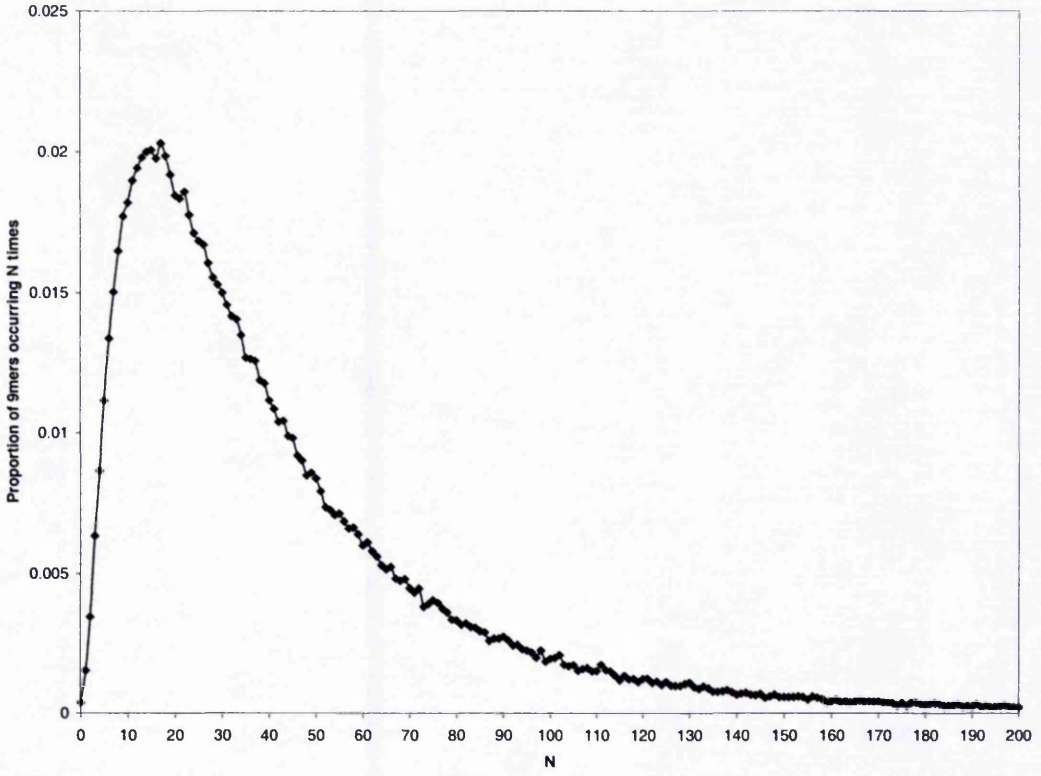

Figure 5 Distribution of 9mers in the yeast genome. The data in this figure and the others in this section were generated using RAPID, a word searching algorithm described in the next chapter.

The rest of this section develops a simple representation of word distributions that attempts to model an uneven residue composition and compares that model with the real observed distribution for yeast.

Since letters in S are assumed to be independent, the probability of w is dependent only on the relative abundance of letters, rather than their ordering.

Let $\bar{n}$ be a vector such that n_i is the number of times letter a_i occurs in w. Then,

$$p(w) = p(\bar{n}) = \prod_i p_i^{n_i} \qquad [19]$$

In a sequence, length L, the probability $p(w,m)$ that w occurs m times is represented by the binomial distribution:

$$p(w,m) = p(w)^m (1 - p(w))^{(1-m)} \frac{L!}{m!(L-m)!} \qquad [20]$$

Since $p(w,m) << 1$ and $L >> 1$ this can be approximated by a Poisson distribution with mean $\lambda = p(w)L$:

$$p(w,m) = \frac{\lambda^m e^{-\lambda}}{m!} \qquad [21]$$

Since, typically, $0 < m < 500$, $m!$ can be large. As a result m is approximated by Stirling's approximation:

$$\ln(m!) = m\ln m + \frac{\ln m}{2} + \frac{\ln 2\pi}{2}, \qquad [22]$$

and the logarithm of the Poisson distribution is computed:

$$\ln p(w,m) = m\ln\lambda - \lambda - m\ln m - \frac{\ln m}{2} - \frac{\ln 2\pi}{2} \qquad [23]$$

This gives the distribution of a specific word, *w*. As a consequence of assuming residue independence, *p(w)* is dependent on the number and probability of the word's constituent letters, not their order. Thus, all words with the same letter composition have the same probability of occurrence ('AACCGGTT' and 'TTGGCCAA', for example). The consequence of this is a number of sets of words, where each member of a given set has the same probability of occurrence.

The overall distribution of words in *R* is the sum of *P(m)* for all such sets – i.e. the sum of the probability distributions where $\bar{n} = \{(0,0,0,k), (0,0,1,k-1), (0,1,1,k-2), ...\}$.

The number of words with a given letter composition is C_k^n and can be calculated as follows:

$$c(\bar{n}) = \frac{k!}{\prod_i n_i!} \qquad [24]$$

Thus, the probability of finding *any* word with a given base composition occurring *m* times in *R is*:

$$P(\bar{n},m) = c(\bar{n}).p(\bar{n},m) \qquad [25]$$

This is shown in Figure 6.

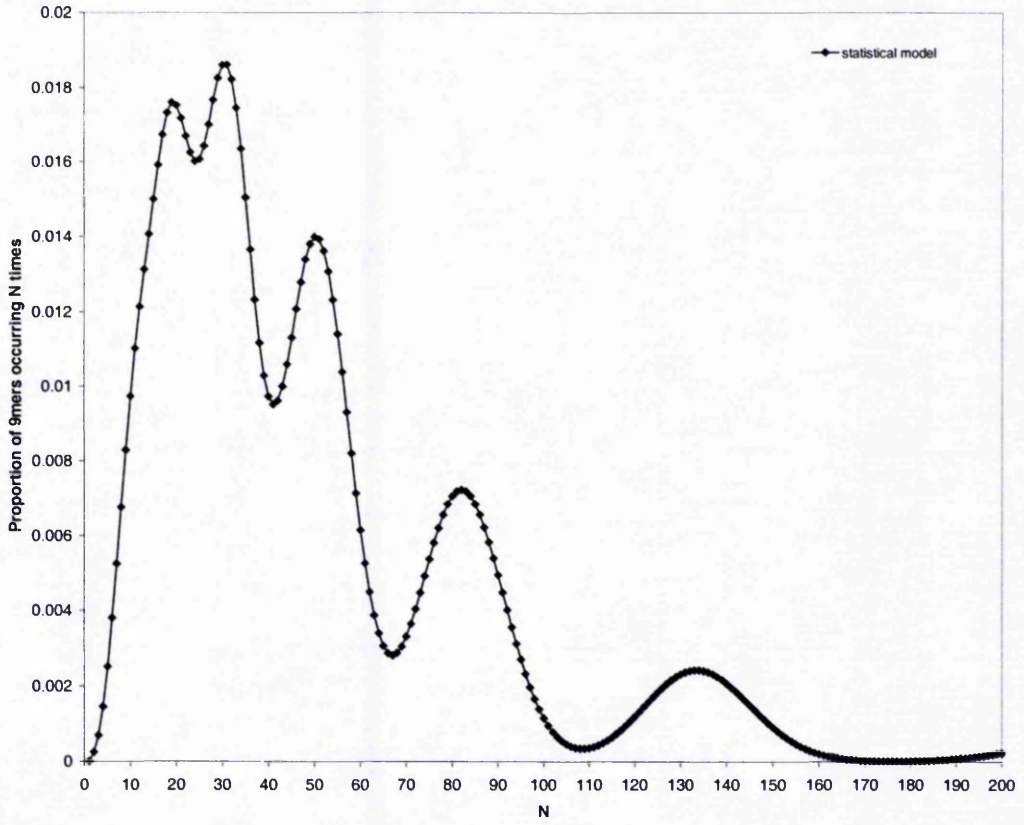

Figure 6 Statistical model of 9mer distribution in yeast. The curve shows a histogram of 9mer frequencies for a sequence the same length as the yeast genome built with 19% a, 31% c, 31% g, and 19% t.

In order to confirm that the model above is correct, and that even with a highly skewed base composition the effect of overlapping words is insignificant, the graph in Figure 7 was generated. It shows a comparison between the distribution derived above and a random sequence with the same parameters.

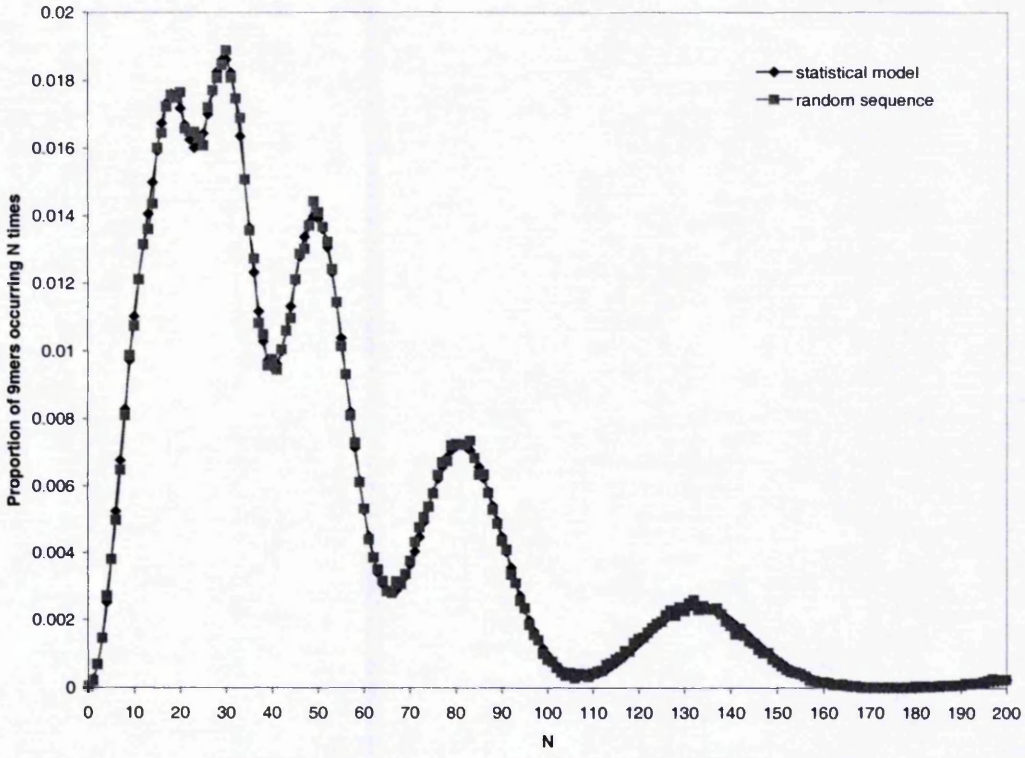

Figure 7 A comparison between the statistical model in Figure 6 and a random sequence the same length, with the same residue composition.

In order to assess the quality of the model, the curves were compared with that produced by the real yeast genome. This can be seen in Figure 8. Clearly, the model is only a crude approximation to distribution of words in real biological sequences.

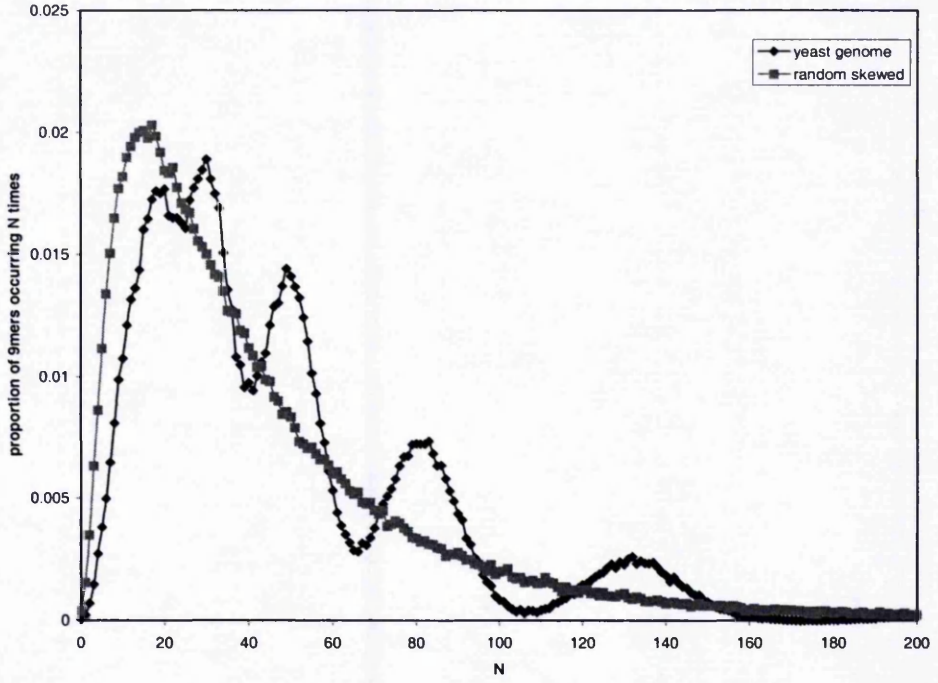

Figure 8 Comparison between 9mer distribution in yeast and a random sequence of the same length, with the same residue composition.

4.2 Reality

The peaks in the curve occur because it was assumed that the adjacent residues in *R* are independent. This results in sets of words, each containing words with the same probability of occurring, and the same distribution of occurrence. Each of these sets corresponds to one of the peaks in Figure 8. Thus, the simple model that assumes residue independence is not capable of modelling the distribution of words in a skewed sequence such as yeast. The next section considers the model's ability to represent words in databases with an even residue composition such as the human subset of EMBL.

4.2.1 Even residue composition revisited

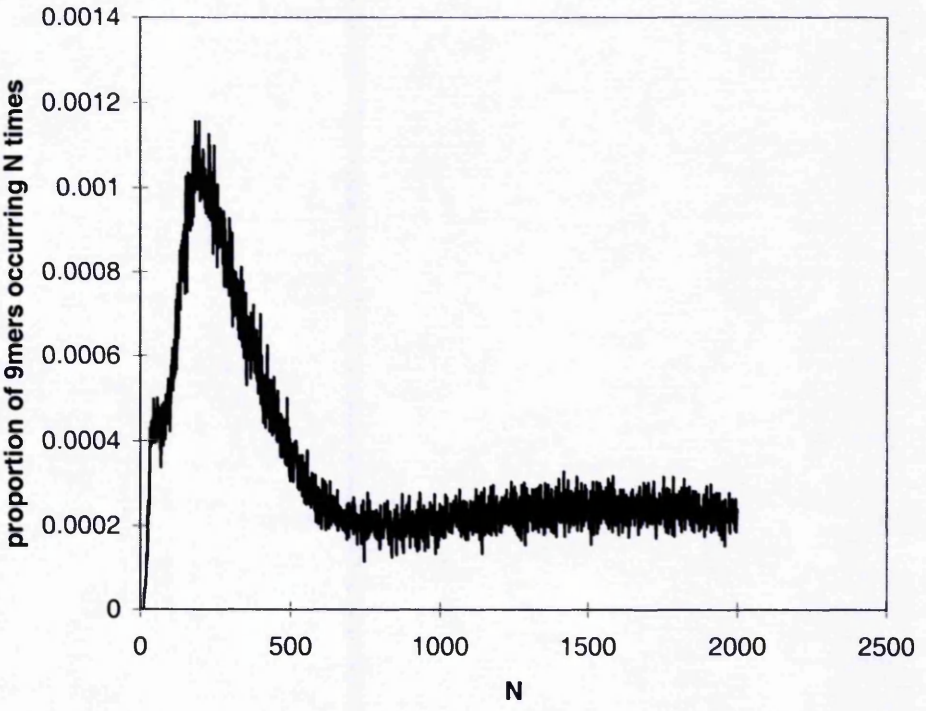

Figure 9 Distribution of 9mers in the human subset of EMBL.

The graph in Figure 9 shows the observed distribution of 9mers in the human subset of EMBL, which has an even residue composition. It can be seen that this distribution is significantly different from the distribution in Figure 4 generated for a random sequence with even residue composition. Thus, the simple model which assumes independence between adjacent residues is not sufficient to model DNA with either an even or a skewed residue composition.

In the next section, a more sophisticated model is derived which better predicts the distribution of words within the database.

4.3 A more complex statistical model

The above analysis makes the assumption that the chance of a base occurring in a DNA sequence is independent of the bases that preceed it. One consequence of this is that a number of different words have the same probability of occurring – leading to the peaks in Figure 8. If the probability of a symbol's occurrence is related to adjacent symbols, two words with the same base composition no longer occur with the same frequency. This has the effect of widening and flattening each peak in Figure 8.

This section develops a model that allows for dependencies between residues in a sequence starting with the simplest case: di-mers.

The non-independence of residues is modelled as follows:

For a string w, of length k, the probability of the j^{th} residue being x, given that the $j\text{-}1^{th}$ residue was y is:

$$p(s_j = x | s_{j-1} = y) \text{ for } 1 < j \leq k . \qquad [26]$$

This is calculated using Bayes' formula:

$$p(s_j = x | s_{j-1} = y) = \frac{p(s_{j-1} | s_j)}{p(s_j)} \qquad [27]$$

The probability of a word occurring is now:

$$p(w) = p(s_2 \mid s_1).\prod_{2\le i\le k} p(s_i \mid s_{i-1}) \qquad [28]$$

This can be used as before, to calculate the probability of a word, w, occurring m times, and to calculate the probability of finding *any* word occurring m times.

The probability, $p(s_{j-1}|s_j)$, is determined empirically by counting di-mer frequencies in the sequence to be modelled.

In order to test the model above, the di-mer composition of the yeast genome was determined empirically, and used to generate the probabilities required by the above model. The distribution that arises is plotted in Figure 10.

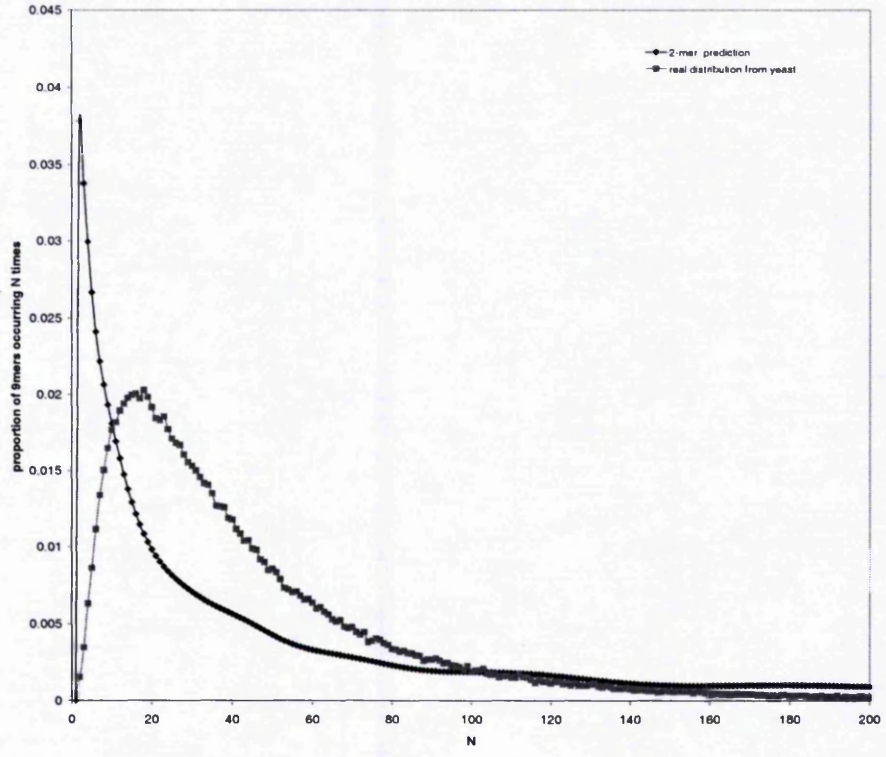

Figure 10 The predicted distribution of 9mers in the yeast genome generated by using a set of empirically determined di-mer frequencies contrasted to the actual distribution.

The di-mer model can be generalised to arbitrary length dependencies as follows:

Let l be the length of dependency – in the previous model, l=2. Y is a consecutive sequence in S, of length l.

$$p(s_j = x | s_{1,2,\ldots,l} = Y) = \frac{p(s_{1,2,\ldots,l})}{p(s_{1,2,\ldots,l-1})}. \qquad [29]$$

This can be calculated from l-mer and $(l-1)$-mer frequencies.

Once again, the model was assessed by producing real *l* and *(l-1)*-mer distributions and generating a predicted *k*-mer distribution.

Figure 11 shows 3-mer predictions and Figure 12 shows 4-mer predictions. These graphs are misleading because of yeast's highly skewed base composition; they are included for completeness.

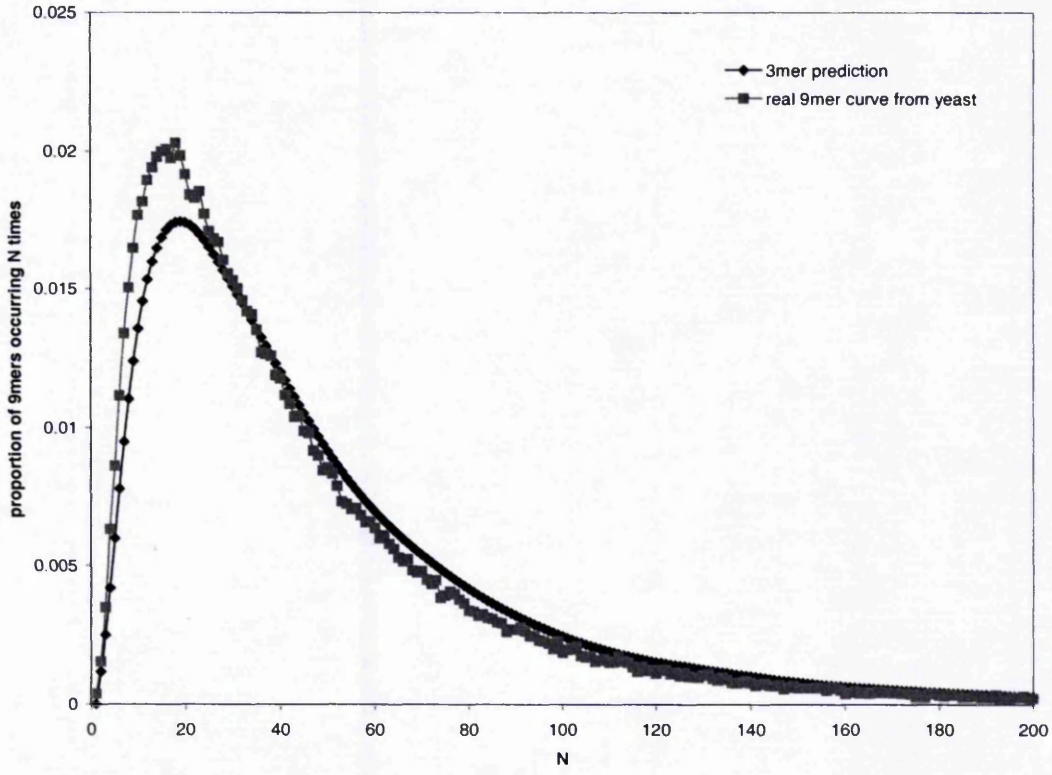

Figure 11 The predicted distribution of 9mers in the yeast genome generated by using a set of empirically determined tri-mer frequencies contrasted to the actual distribution.

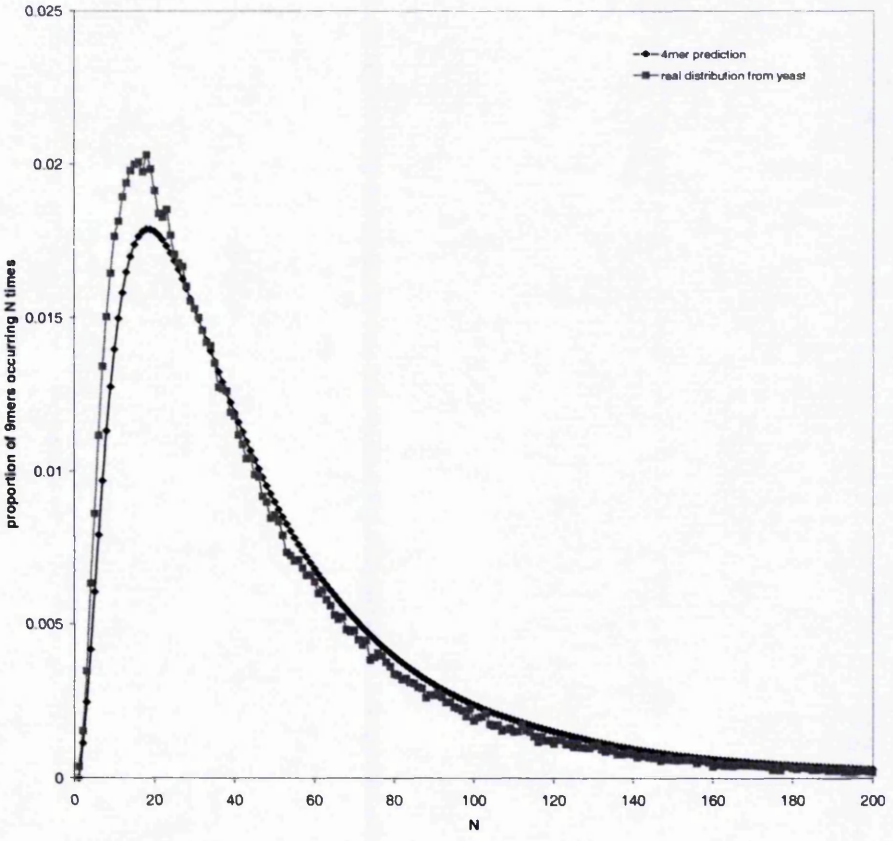

Figure 12 The predicted distribution of 9mers in the yeast genome generated by using a set of empirically determined quad-mer frequencies contrasted to the actual distribution

If the same method is used to predict the 9-mer composition in the human subset of EMBL (in which all bases occur with approximately equal abundance), the distribution that arises is not an accurate model of reality, even if 5-mers are used to predict the 9-mer distribution. Figure 13 shows how the model performs with 5-mers.

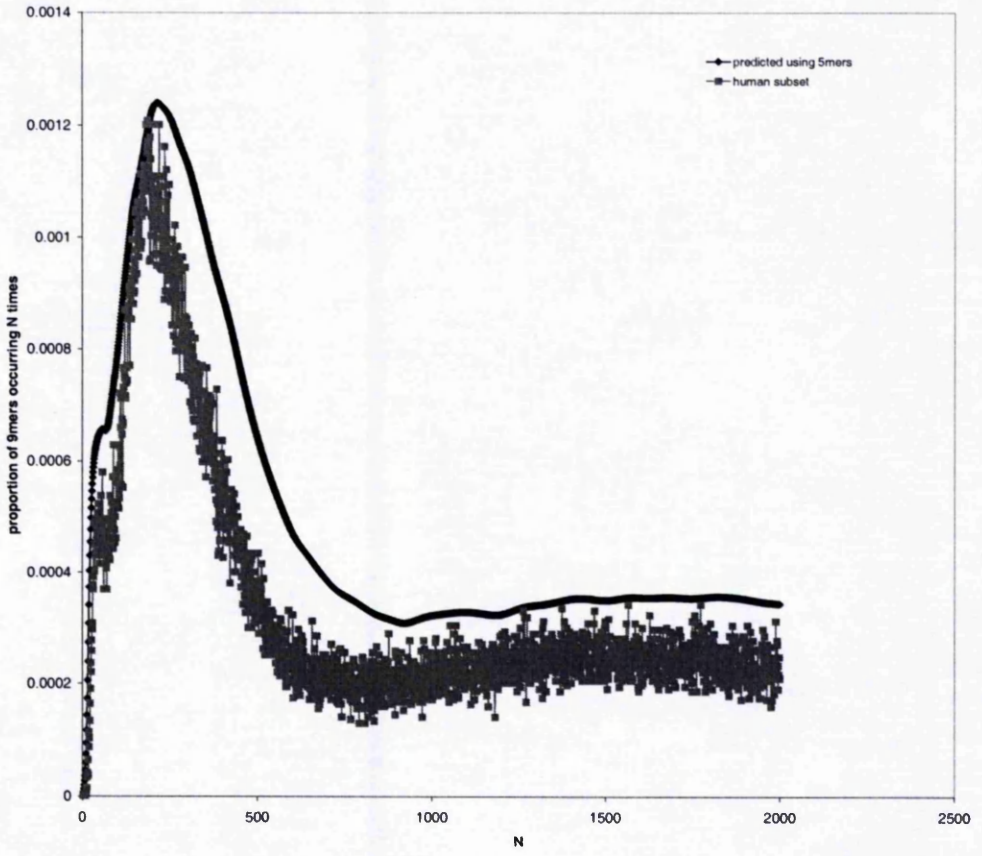

Figure 13 Distribution of 9mers in the human subset of EMBL compared to that predicted by using 5mers. Both curves have been normalised to have an area of 1.0.

4.4 Summary

As word lengths are increased the shape of the word distribution changes dramatically (see Figure 14 which shows the distribution of different sized words in the yeast genome).

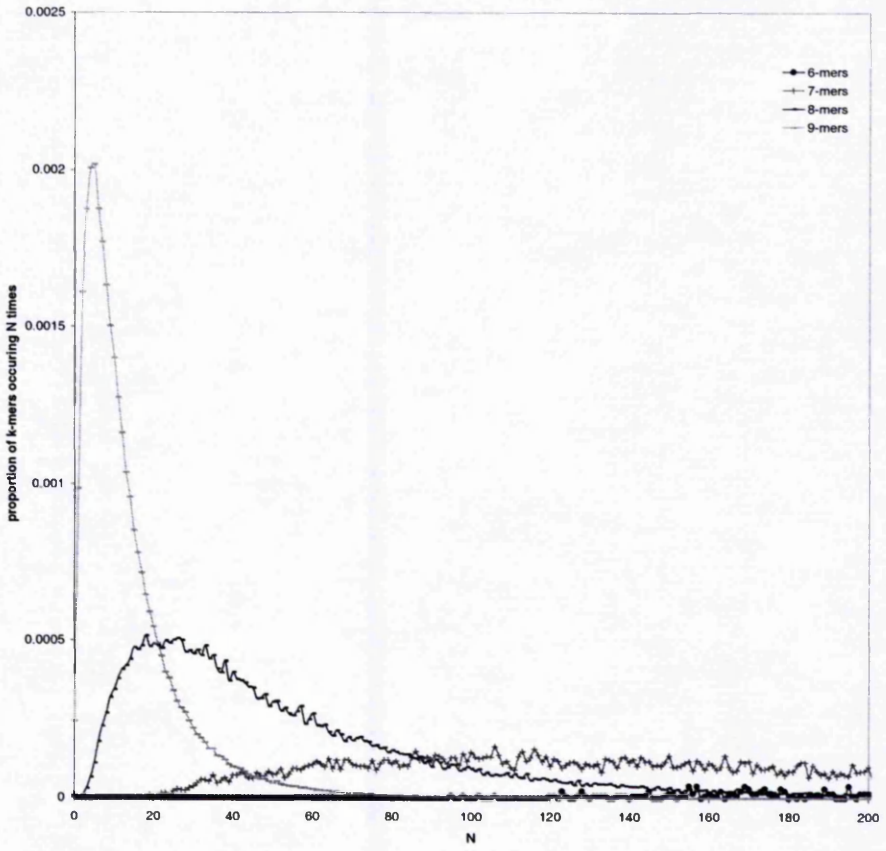

Figure 14 Distribution of words of different lengths in the yeast genome. The curves show the number of words which occur a given number of times – so that, for example, the majority of 9mers occur about 17 times. All the curves are normalised to have an area of 1.0.

When words are short, the distribution is essentially flat, with all words occurring with similar abundance. However, as their length is increased, the distribution changes shape, taking on a skew.

One of the reasons for this is the non-independence of bases within DNA. This suggests that statistical models which assume biological sequences to be random and composed of independent residues should be treated with a good deal of caution.

A residue's influence is quite far ranging, preventing models based on short words (less than seven residues) to be used to predict the distribution of 9-mers in a biological database.

In summary, the distribution of residues in biological databases is complex and cannot be effectively modelled. This is not really that surprising given that sequences exist to carry complex biological information. The analogy between biological sequences and language is persuasive, as is the analogy to information-carrying sequences (Shannon 1948). Both suggest the application of mathematical techniques to the analysis and representation of biological sequences.

However, these analogies suggest that the message and the medium are independent – the information carried in a piece of text is the same irrespective of whether it is typed, printed on a computer screen or spoken. This is not the case with biological sequences. For example, RNA and DNA both fold into 3D structures which affect the way the sequences are treated by a cell. Codon biases affect annealing temperature and hence the optimal temperature in which organisms can live. Coding regions can contain signals which result in alternative splice sites, they contain alternative consensus sequences which instruct the cell to attach poly(A) regions to pre-mRNA and genes can contain multiple promoters within introns. Whilst it might be possible to generate a model that took all of these (and the multitude of others that have not been mentioned here) into account, such a model would require a large amount of context and high level knowledge – such as how the 'meaning' of a sequence changes depending on

whether it is occurring in DNA, transcribed mRNA or translated to a protein sequence.

The results of this chapter show that a statistical model of word frequencies is hard to generate, and suggest that the best way to obtain a distribution is simply to measure it empirically. This has profound implications for word searching algorithms and the statistics used to evaluate the significance of their results. The following chapter describes RAPID, a word-searching algorithm that makes use of an empirically determined distribution.

5 RAPID Analysis of Pre-Indexed Databanks

In the first chapter of this thesis, it was established that there is a requirement for a highly efficient method of performing similarity searching at the DNA level. This chapter describes the design and implementation of a novel word-based search algorithm that is an order of magnitude faster than BLAST on the same hardware, but which performs with similar sensitivity.

5.1 The RAPID algorithm

Word matching can be seen as a very rough approximation to computing an alignment - it is the number of *k*-long alignments that can be made between two sequences. By counting the number of short alignments between two sequences without considering any kind of relative position, a similarity search algorithm can reduce the amount of information it needs to consider and, as a result, make considerable gains in speed. Unfortunately, throwing so much information away also results in an algorithm which is rather insensitive.

In order to increase the sensitivity of a word-searching algorithm it is necessary to augment the raw matches with something else. Rather than use positional information to compute local alignments (as BLAST and FASTA do), RAPID uses statistical measures of word frequencies based on an n-gram analysis of a large amount of real data. This allows word matches to be scaled according to the likelihood of them occurring by chance and results in a surprisingly sensitive algorithm.

An analogy can be made to the task of comparing two newspaper articles: if they share a number of rare words like 'Michael', 'Howard', 'creature', and 'night', they are likely to be talking about the same thing, but if they only share common words such as 'the', 'and', and 'because' they are probably not. The scoring system employed by RAPID is based on an analogous assumption.

5.2 RAPID's scoring system

RAPID compares two sequences, a and b, by counting the number of words, N, occurring one or more times in a which also occur one or more times in b. This is compared to an estimate, E, of the number of such "matches" we would expect to occur by chance.

A DNA sequence of length L contains $L-k+1$ overlapping words, which we consider as a list $K_1, K_2 \ldots K_{L-k+1}$. Consecutive words in this list share $K-1$ bases. The algorithm ignores words containing unknown bases (normally represented by the letter 'n' in sequence databases) with the result that a sequence containing a large number of unknowns has a relatively small number of unique words.

It is assumed that the probability of a word, K_i, being w is simply $P(w)$, the probability of w occurring next in an arbitrary DNA sequence, and we model the distribution of words within a DNA sequence as a Poisson distribution in a fashion similar to EMBLSCAN (Bishop & Thompson 1984). Thus, the probability of a word, w, occurring n times in a sequence of length L is given by a Poisson distribution with mean $P(w)L$:

$$P(w,n) = \frac{(P(w)L)^n e^{-P(w)L}}{n!} \qquad [30]$$

So that the probability of a word occurring one or more times is:

$$P(w,n \geq 1) = 1 - P(w,0) = 1 - e^{-P(w)L} \qquad [31]$$

With typical values, $e^{-P(w)L}$ is of the order $e^{1/500}$ so [31] can be approximated by expanding $e^{-P(w)L}$ and ignoring all but the first two terms. Thus the probability of a word occurring one or more times in a sequence of length L reduces to:

$$P(w,n \geq 1) = P(w)L \qquad [32]$$

The experiments in chapter 4 demonstrate that this assumption is reasonable when k is of the order 9 – the sort of word size which RAPID is designed to use.

5.2.1 The number of matches to be expected by chance

Let W^a and W^b, sizes L_a and L_b respectively, be the sets of words which occur one or more times in two sequences, a and b.

The total number of matches E between a sequence, a, and an unrelated sequence, b, is estimated using [32]:

$$E = \sum_{i=0}^{L_a} L_b P(W_i^a) = L_b \sum_{i=0}^{L_a} P(W_i^a) \qquad [33]$$

The significance of a match is estimated by taking the ratio S of the number of matches actually found to the number of matches expect by chance:

$$S = N / E \qquad [34]$$

S is highly dependent on the length of the sequences. With long sequences, small but significant matching regions are masked by chance matches from the rest of the sequences. Conversely, matches on very short sequences are assigned a higher score than they appear to warrant. For this reason, RAPID treats a long sequence as a set of independent, short, overlapping fragments (typically 500bp long), and adopts a modification of S which normalises it for the lengths of the subsequences:

$$S' = \frac{L_a L_b}{C_a C_b} . S \, , \qquad [35]$$

where C_a and C_b are the size of the segments. Dividing long sequences into shorter segments places an upper bound on the size of segment that is considered by the algorithm. When segments are smaller than the segment size, the scaling factor reduces the significance of their scores.

Note that S and S' are the same for all but very short sequences, where S is considerably larger than S'. Whilst S is a statistically robust estimate of significance, it is highly length dependent. The modification resulting in S' produces a score has been found to be more useful in practice.

Crucial to the calculation of E are the chosen values of $P(w)$. Chapter 4 of this thesis compared the distribution of words in biological sequences to the distribution in random sequences generated with statistical models of varying complexity. The chapter showed that the distribution of long words (> 8 bp) is skewed in a complex fashion. It cannot be modelled by assuming that adjacent residues in a sequence are independent of one another, nor can it be modelled by using the distribution of short words (<6 bp) to predict the effects of neighbouring residues in a sequence.

For this reason, RAPID estimates the probability of a pair of sequences matching by chance using an empirically determined table of word frequencies. Ideally, these would be found by counting the occurrence of words in a large, representative and non-redundant set of DNA. One possible source would appear to be the yeast genome, but its highly skewed base composition (38% A+T) is reflected in an unusual word distribution. At present, the problems of redundancy

are accepted; the probability table is generated by simply determining the distribution of words in an EMBL subset. Redundancy has the effect of making words which occur in sequences repeated in the database appear more likely than they should. It is necessary to put this in perspective; some words occur thousands of times more often than others, and it is matches due to these words that really need to be discounted. It is shown in the next chapter that the program functions well with less than perfect word probabilities.

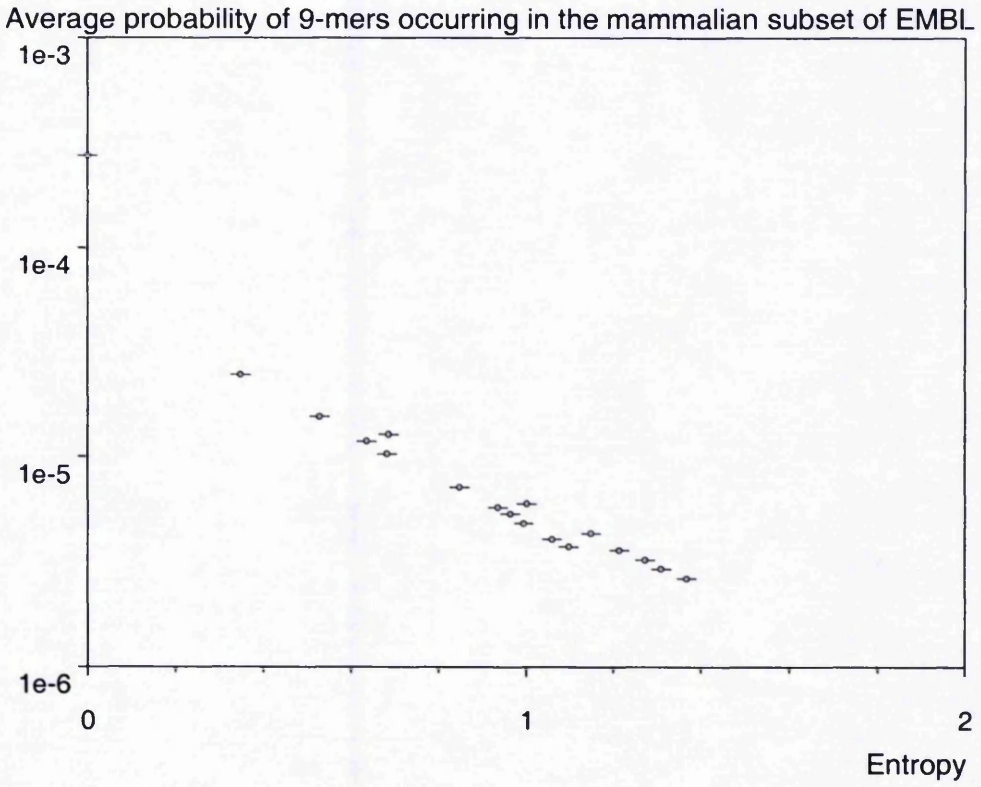

Figure 15 The average probability of words occurring in the mammalian subset of EMBL against Shannon Entropy (Shannon, C. 1948) used as a measure of complexity. Entropy is calculated by $H = \sum_{i \in a,c,g,t} -p_i \log p_i$, where p_i is the number of times a base occurs in the word, divided by its length. Since a number of different words have the same entropy, we plot the mean word probability for each of the values of H. The standard error in the mean is insignificant.

Intuition suggests that a search tool should scale down matches between low complexity regions (such as a telomere repeat). This would occur if *P(w)* was inversely proportional to complexity. Figure 15 shows that this is indeed the case. The use of probabilities rather than entropy has the advantage that common

regions which have a relatively high complexity (such as microsatellite repeats) are also scaled down. This is demonstrated in the next chapter.

5.3 Implementation of RAPID

The mathematical system described above provides a method for measuring pair-wise similarity between sequences. It can be summarised as a two step process:

1. Measuring the number of words which occur in sequence *a* that also occur in sequence *b*, and

2. Scaling that score by a statistical estimate of the number of words that occur by chance.

5.3.1 Counting the matches

In order to perform the first task it is necessary to determine, for each word in sequence *a*, the presence or absence of that word in sequence *b*. For a search of a single sequence against a database, it is necessary to determine the presence or absence of each word in *a* for *every* sequence in the database. Since a number of pairwise comparisons need to be made, the list of words in *a* needs to be used many times (one for each comparison), whilst the list of words in each of the database sequences is required only once. This suggests generating a datastructure which 'remembers' the words in *a* so that they only need to be generated once. This is the approach taken by BLAST, which generates a Trie. Refer to Figure 16. A Trie represents lists of symbols (such as DNA sequences)

as a tree structure, in which each arc of the tree represents a transition from one symbol to the next. So, the grey nodes in Figure 16 represent the sequence 'acccg'. Tries are very efficient data structures which allow a *k*-residue sequence to be looked up in *k* operations. However, the need to store a pointer for each arc makes them fairly memory intensive.

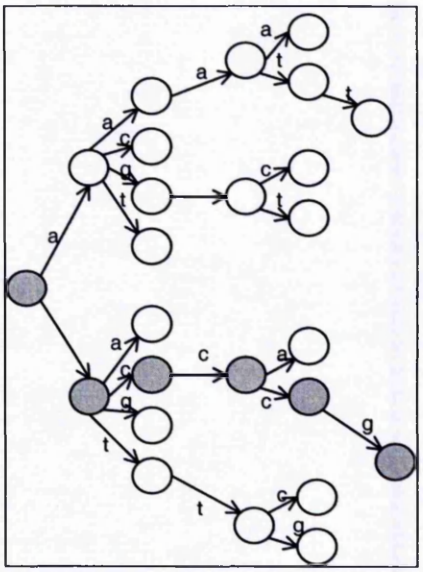

Figure 16 A Trie, with the sequence 'acccg' highlighted by the grey nodes.

BLAST generates a Trie for the query sequence to be searched, and then matches each database sequence against it to identify 'seeds' from which to generate MSPs. This is a very efficient operation for a single-sequence against database search.

Unlike BLAST, however, RAPID is designed to be particularly fast for database-against-database rather than single-sequence-against-database comparisons. In this situation, not only is each word in the query-sequence-set re-used many

times, each word in the database-sequence-set is also required more than once. This suggests that it would be efficacious to produce a structure which represents the words that occur in each database sequence in such a way that they can be rapidly found during a search. One possibility would be to use a Trie, but given the potential size of a database, a more compact datastructure is appropriate. For this reason RAPID generates a simple array which represents, for every possible word, the list of the database sequences in which it can be found. The array is arranged so that indexes into this array can be rapidly generated. The approach taken is to treat each word as a base four number, so that 'aaaaaaaaa'is represented by the number 0, and 'ttttttttt' is represented by the number 4^9. Even though this results in some very large indexes (4^9=262144) this approach is valid because computer memory is relatively cheap. To put this in perspective, it is possible to search all of EMBL against a database of vector sequences in a few hours on a desktop PC with 256Mb of memory, which costs less than £1500 at today's prices.

In order to evaluate the algorithms described above an initial implementation was produced which uses the hashing technique described above to locate word lists within a table.

5.3.2 Recording the statistics

The same index which is used to find the appropriate word list is also used to index into a table that lists the probability of occurrence for each possible word. For each query sequence, the sum of all its word probabilities are recorded by

incrementing a variable as the search proceeds. This is used in concert with the query and database lengths to generate an *E* value for the query sequence.

The whole process is summarised in Figure 17.

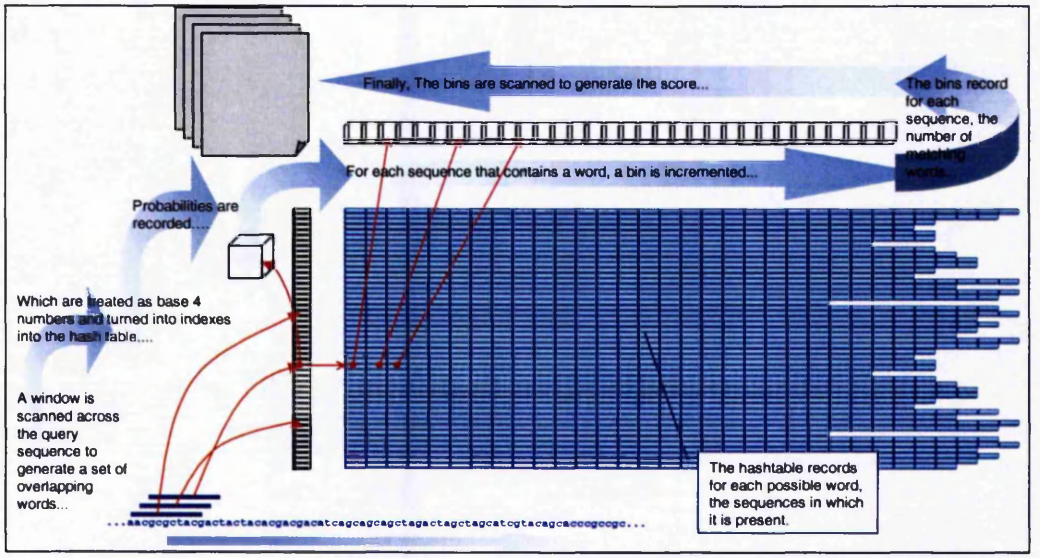

Figure 17 Summary of RAPID's implementation.

5.3.3 Management of the Hashtable

In advance of a search, the software builds a database of word lists which records, for each of the 4^k possible words, the sequences in which that word occurs. The database is stored as a disk file. The use of this pre-computed data significantly reduces the amount of work required during a search.

For database against database searches, the low cost of memory would make it feasible to load the entire database into memory, where searches can be performed with no disk accesses. In the case of a single sequence search, where

only a small proportion of the possible words need to be looked up, the time taken to load the entire database into memory would be prohibitive; a preferred solution would be only to load a word list when it was required. RAPID meets these conflicting demands by using memory mapped IO to map the database file into its (virtual) address space. When the mapping takes place, no disk to memory transfer is initiated. Instead, page faults are generated each time the software accesses a word list which is on a page of the address space that has not been previously loaded into physical memory (RAM), and the operating system loads the required page into physical memory.

Memory mapped IO has a number of advantages. Firstly, only the sections of the database which are required are loaded into RAM. Secondly, the number of actual disk accesses is reduced to one-per-page as opposed to one-per-read with traditional file IO and, thirdly, traditional file operations perform a certain amount of buffering which results in repeated copying of the data being read. Direct memory access is, by its nature, unbuffered, and does not incur these costs.

5.4 Time complexity

Given a sequence a length L_a , the L_a - k +1 words it contains can be generated in O(L_a). These are used to index into a table comprising of 4^k lists that record the sequences in the database which contain them. In a database containing n sequences and N nucleotides these lists are, on average:

$\frac{N}{\langle P(w) \rangle}$ elements long, if the mean probability of a word occurring is $\langle P(w) \rangle$.

For each element the only operation required is to increment a few counters in the results array which is scanned at the end of the search in $O(n)$.

This gives a time complexity:

$$C = O(LN) + O(n). \qquad [36]$$

Since n is proportional to N, the search scales linearly with the length of the search sequence and the database size. If the output is required to be sorted, the complexity is $O(N \log N)$, but it is possible to do better than this because it is not necessary to sort the entire set of results, only the significant ones, which form a far smaller subset.

A pair of graphs showing how RAPID scales with database size can be found in Figure 26 page 142.

5.5 Input & Output

The software has a simple command line interface, and accepts an EMBL or FASTA file as input. Different probability tables can be loaded allowing the word weighting scheme to be changed if desired. One problem associated with a large search is the amount of data produced; it is not appropriate to present all of this simply as a long text file. In order to address this, RAPID generates a tree of

web pages containing the results of a search. The root page shows the top hit for each query sequence which has matched against at least one database sequence. Below this is a set of pages describing, for each query sequence, all the hits that have been found. These pages intentionally resemble the list produced at the top of a BLAST output file, and links through to a set of pages, each containing an alignment for a query/database pair.

5.6 Summary

RAPID is implemented using a highly efficient set of data-structures that exploit the relative cost advantages of computer memory with respect to processor speed. The algorithm is also one that lends itself to cheap parallel hardware. The nature of the search process is such that it can be divided into a set of coarse-grained computations that can be conducted independently of one another. The final answer is produced by combining the results from each sub-computation. This means that such an implementation will run efficiently on a multi-computer system (which is effectively a set of individual computers, or nodes, connected by a fast network).

A number of alternative programming libraries exist to allow this kind of hardware to be used: the most appealing of which is the Linux-Beowulf project (http://www.beowulf.org) which enables a multi-computer system to built out of a set of cheap PCs connected by a fast Ethernet link. RAPID could be implemented on such a system by dividing the hashtable into fragments, each of which is placed on a separate node. One node acts as a 'master' that serves to co-ordinate the work performed by the other nodes, which are referred to as the

'slaves'. The master sends a message to each slave containing the query sequence which is then searched by the salves in parallel against each of the database fragments. The result of each individual sub-search is then sent back to the master, which then combines them to produce the final result. If inter-process communication is too great (resulting in inefficiencies) more than one sequence can be searched before the results for *all* the sequences are combined. Thus, the only traffic on the network is to distribute a packet of sequences, followed by a set of partial results being returned to the master by each slave node. To make sure that some nodes aren't sitting idle whilst others are computing, care needs to be taken to divide the database into appropriate sized fragments. The probability table can be used to perform this calculation, because it states, for each word, the probability of its occurrence – and hence the likelihood that it is going to be looked up in the hash table.

The current implementation has not been designed to run on parallel hardware, although work is underway to produce one that does.

5.7 User Interface considerations

The algorithm, RAPID, presents its result as nothing more than a score. Subsequent sections describe the design and implementation of a set of companion tools that provide alternative ways of viewing similarity between sequences, and their use to augment and corroborate the results generated by RAPID. The principal method of presenting similarity to users is via an alignment. The rest of this section describes why this choice was made.

Although several different approaches exist for similarity searching – dot-plots, word searching and alignments – it is alignments that predominate. One reason for this is the mathematical rigor that has been applied to the analysis of their algorithms, allowing alignments to be scored effectively, and that score to be used to rank database hits. Another reason, which has been somewhat neglected in reviews of similarity searching techniques, is that an alignment provides a justification of the score assigned to a pair of sequences.

Experience from the realm of expert systems, shows that software which provides an 'audit trail', documenting why a particular decision has been made, is more likely to be accepted by a user base than software which simply presents its result in an opaque fashion. For example, MYCIN was an interactive program that diagnoses certain infectious diseases and prescribes anti-microbial therapy. One of its key features was its ability to explain its reasoning in detail - it would have been unrealistic to expect a physician to prescribe treatments without being able to understand the reason why a particular drug had been chosen (Shortliffe, 1976).

Part of the role played by alignments in a BLAST/FASTA report is similar – they provide an accessible explanation of why a score has been assigned to a pair of sequences. The reassurance which results from this is an important feature of alignment algorithms and one that needs to be considered in the design of alternative sequence searching strategies. One implication from this is that search tools which are not alignment-based may still use alignments as 'corroborative evidence' when presenting their results. In this situation, alignments exist to

improve the user interface, rather than to generate a score that is used during the database search. As a result, alignments (which have a significant computational overhead) need only be generated for those sequences in which a user is interested.

5.8 Visualisation and analysis tools

The previous section discussed the role that alignments play in the *user interface* of a tool such as BLAST – that they helped a user understand the score that the algorithm had assigned to a pair of sequences. Given that RAPID only provides a numeric score detailing the strength of a match, it was decided that an important part of its user interface would be the generation of alternative mechanisms for depicting sequence similarity.

Three strategies were explored: ungapped alignments, gapped alignments and 'coarse grained dot-plots'. The alignment tools are described in the next two sections, an analysis of coarse grained dot-plots and their use in comparative studies of genomes and gene fragments can be found in section 6.1.

5.8.1 Alignment tools

These algorithms are designed to form part of a user interface – they do not exist to produce similarity scores for use in actual comparing sequences. This means that they do not have to produce optimal alignments, and do not have to use scoring schemes that are statistically rigorous. Instead, they are required to show the similarity between a pair of sequences in a way which helps a user rapidly

assess the nature of match, and in a manner they are comfortable with. One reason for developing a pair of alignment tools was that they are a ubiquitous user interface device that the majority of bioinformatics users will be familiar with.

The fact that RAPID appears to be a fast and sensitive algorithm for performing similarity searching, and that the use of word weighting is part of the reason for this, suggests that the use of word probabilities might be a way of improving a standard alignment tool.

Two tools are described: the Probabilistic Hough Alignment Tool (PHAT) which generates ungapped alignments similar to a BLAST Maximal Segment Pair (MSP), and the Smart Probabilistic Alignment Tool (SPLAT) which is derived from the Smith-Waterman algorithm, and generates gapped alignments.

5.8.1.1 PHAT (Probabilistic Hough Alignment Tool)

One of the earliest techniques used to determine sequence similarity was the dot plot (Gibbs, A. & McIntyre, G. 1970). Given two sequences a and b, a point $p(x, y)$ is placed in the xy plane whenever $a_x = b_y$. This results in regions of similarity appearing as diagonal lines angled at 45 degrees to the axes.

Rather than plot such an image, it is possible to apply a spatial transformation and, for each point in xy space, plot lines in mc space which satisfy the equation c = y-mx.

A set of points which lie on a line in conventional space appear as a set of intersecting lines in *mc* space. The point where these lines intersect gives the gradient and the *y*-axis crossing point of the original line.

This technique, known as a Hough transform (Hough, P. 1962), provides an efficient method for finding regions of similarity, when it is recognised that only lines with a gradient of 1 are of interest. These correspond to points in *mc* space on the line $m = 1$. It is similar to the 'diagonal method' employed by FASTA.

When DNA sequences are compared in this way, the small number of symbols (A,C,G,T) result in a large number of points occurring by chance. In order to avoid this, PHAT only considers *k*-mer matches between two sequences.

PHAT plots histogram H_c:

$$H_c = \sum_x \sum_y s_{W_x W_y} \qquad [37]$$

where $c = y - x$, and s is the score assigned to a match between the *k*-mers starting at a_x and b_y.

A large value of H_c corresponds to a significant match on the line $y = x + c$.

Once an interesting diagonal has been determined, PHAT runs along it searching for the most significant alignment, offsetting the sequences by an amount determined by the point at which the diagonal crosses the *y* axis. The alignment is found by computing a vector M, where:

$$M_r = \begin{cases} M_0 = 0 \\ M_{r-1} + m_{a_x b_y} . I_{a_x} I_{b_y} / 2 \end{cases} \quad [38]$$

The alignment ends where M_r is maximum, and starts at the first preceeding point where M_r is zero.

$m_{a_x b_y}$ is the score assigned to a match between the bases at a_x and b_y.

I_{a_x} and I_{a_y} are 'interest factors' assigned to each nucleotide, determined as follows:

$$I_i = \frac{\sum_{i \in w} \left(1 - \frac{p(w)}{p(\max)} \right)}{k} \quad [39]$$

where k is the word length, $p(w)$ is the probability of a word occurring (as used by RAPID), and $p(\max)$ is the probability of the most common word occurring.

Thus, a base in a common region, such as a telomere repeat, is assigned an interest factor close to zero, whereas bases occurring in less common regions are assigned higher interest factors. The interest factor is used to weight the match and mismatch scores, both in the computation of the histogram, and the alignment matrix. In both cases, the raw (mis)match score is multiplied by the average interest factor of the two bases being compared.

PHAT displays an alignment as coloured text, with each nucleotide being assigned a colour temperature according to its interest factor. Thus, bases in rare regions are coloured yellow, whilst those in common regions are coloured blue.

5.8.1.2 SPLAT (Smart Probabilistic Local Alignment Tool)

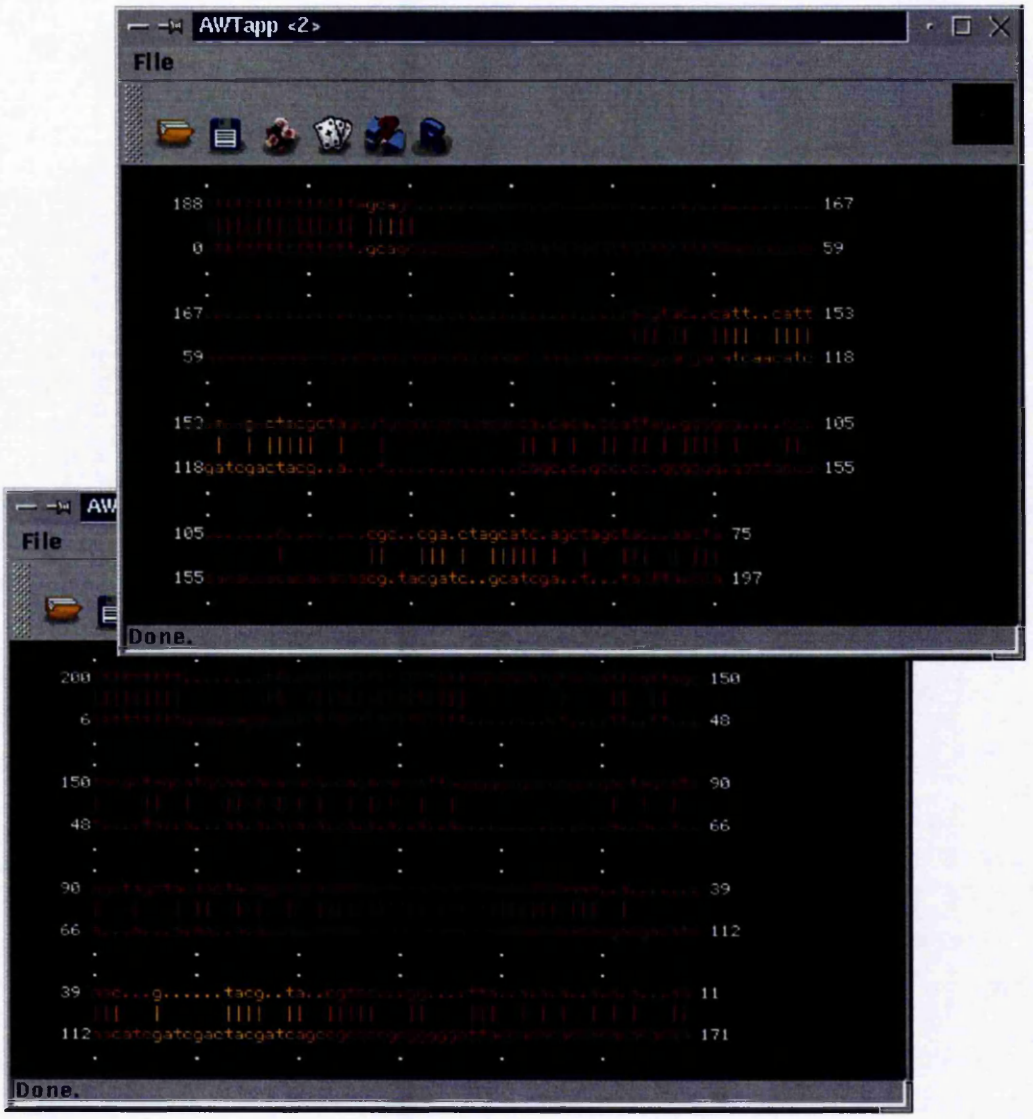

Figure 18 Alignments produced by the SPLAT applet showing how word probabilities change the alignment produced for a pair of sequences. The bottom alignment is produced by an implementation of Smith-Waterman, the top by SPLAT, using word frequencies (both are generated with the same pair of query sequences). Word frequencies cause SPLAT to insert gaps opposite low complexity regions, allowing shorter higher complexity regions to be aligned in preference.

SPLAT is a modification of the Smith-Waterman (Smith, T.F. & Waterman, M.S. 1981) alignment algorithm. Match/mismatch scores are modified using interest factors, as described in the previous section, and alignments are displayed

using a colouring scheme similar to that of PHAT. In an optimisation similar to that employed by FASTA, SPLAT limits the maximum gap length which the program will attempt to identify. The tool performs differently from Smith-Waterman when aligning sequences containing low complexity regions and repeats – see Figure 18.

6 Results

RAPID is able to present the similarity between a pair of large sequences such as genomes by generating 'coarse-grained dot plots'. These are described in the next section.

RAPID has also been used to conduct a systematic survey of sequence contamination in the DNA database EMBL. An evaluation of RAPID's ability to perform this task can be found in section 6.2. Contamination due to vector sequence is discussed in section 6.3; genomic *E. coli* in section 6.4.

6.1 Coarse Grained Dot-Plots

Some of the earliest software tools for sequence analysis represented similarity by graphical methods, such as those of Fitch, McLachlan and Gibbs and McIntyre (Fitch 1966, Fitch 1969, McLachlan 1971, McLachlan 1972, Gibbs and McIntyre, 1970). In their simplest form, these tools constructed dot plots by placing two sequences on the axes of a plane and plotting a point whenever a matching residue occurred. Thus, regions of similarity appear in such a tool as diagonal lines. Other features such as repetitive regions, insertions and deletions are also easily identified. The early dot plotters did not offer any computational measure of similarity, they simply attempted to represent the relationship between a pair of sequences in a way which helped a user visualise it. They were successful because they were able to rely on a human being's own perceptual apparatus to perform the pattern recognition necessary to identify interesting regions. Although dot plotters were developed which produced a numerical

measure of similarity (Staden 1982, Pustell and Kafatos 1982, Pustell and Kafatos 1984, Argos 1987), tools such as Smith-Waterman (Smith, T. F. and Waterman, M. S. 1981) BLAST (Altschul *et al.* 1990) and FASTA (Lipman, D.J. and Pearson, W.R. 1985) became the tools of choice for tasks that required the comparison of a single sequence against a set of other sequences (such as similarity searching within a database).

At the time of writing, 32 genomes have now been completely sequenced. This provides the opportunity for comparative studies between genomes, and a desire to perform a detailed analysis of the similarities and differences between a pair of gene sequences. Part of such a study can be seen as analogous to the kind of analysis that can be performed using dot plots – except that now the sequences are at least three orders of magnitude longer.

This section describes a novel tool that produces 'coarse grained dot plots' showing similarity between sequence pairs. It shows that graphical methods similar to those used for the detailed analysis of short sequences can be fruitfully employed in the comparison and analysis of entire genomes.

The tool takes a pair of sequences which it segments into a set of fragments (typically 1000bp long). These are compared using RAPID. The resulting set of matching sequences are represented as dots within a plane, which are colour-coded to represent the level of similarity. The results of the analysis are represented as an HTML image map; dots are clickable and link to an alignment of the relevant region.

The tool is interesting for a number of reasons. Firstly, each dot represents a match somewhere within a fragment, computed by a similarity search tool which is both fast and sensitive. Thus, even weak matches between regions are displayed within the image, but by virtue of their colour coding, they can be distinguished from stronger matches. This makes the tool different from that of Delcher *et al.* (1999) which relies on sequences being of high similarity.

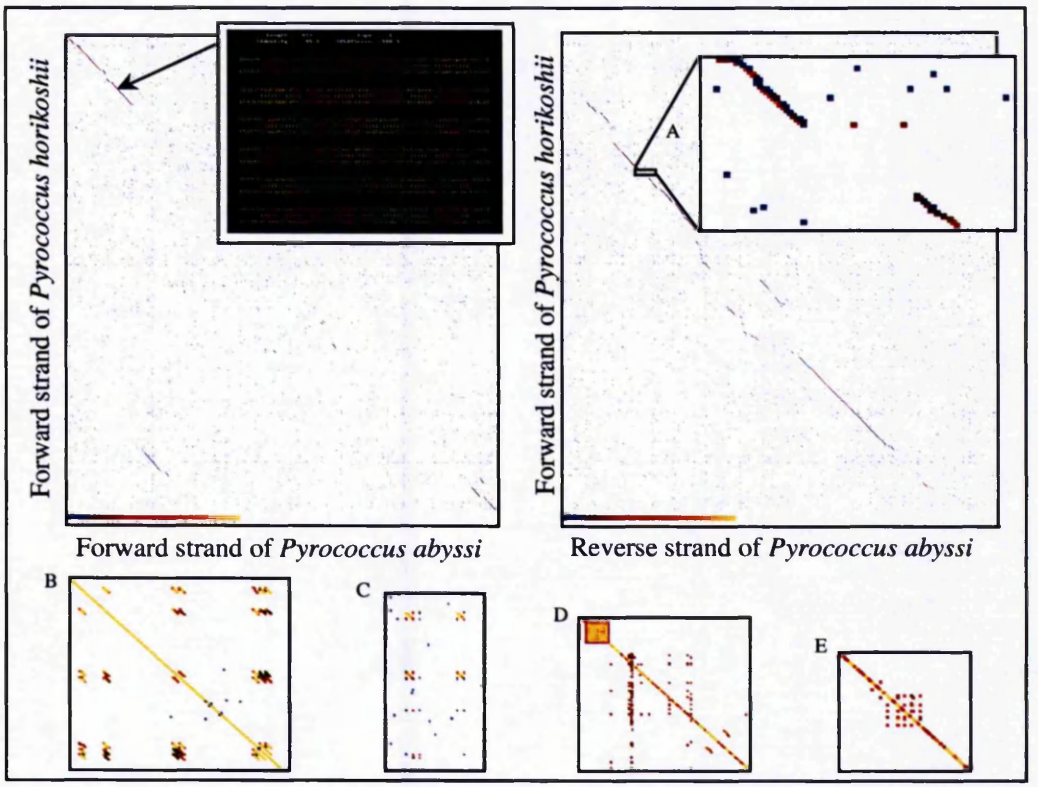

Figure 19 shows genome comparisons performed with RAPID and the coarse grained dot plotter, on two species of Pyrococcus, abyssi and horikoshii. Inset is an alignment produced when one of the dots is clicked. Box A is an enlargement showing repeat tRNA sequences which show up clearly as a line of red dots. Box B shows repeated sequences from a comparitive genome plot of Bacillus subtilis. Box C is a feature from the comparison of chromosomes 12 and 16 from Saccharomyces cerevisiae illustrating the Ty1 transposable elements. Box D is an example of telomeric sequences from the Plasmodium falciparum chromosome 3 against itself. Box E shows a cluster of five similar SERA antigen/papain like proteases from Plasmodium falciparum which are identified by searching chromosome 2 against itself.

Like the original dot plots, many relationships between sequences become instantly apparent. Figure 19 shows some examples of the kind of features that become apparent when a set of RAPID hits is presented using a dot-plot. Conserved 'gene strings' appear as diagonal lines in the plot (see Figure 21), with insertions or deletions appearing as 'staggers' in the diagonal line. Horizontal and vertical lines show evidence of repeat regions: refer to Figure 20.

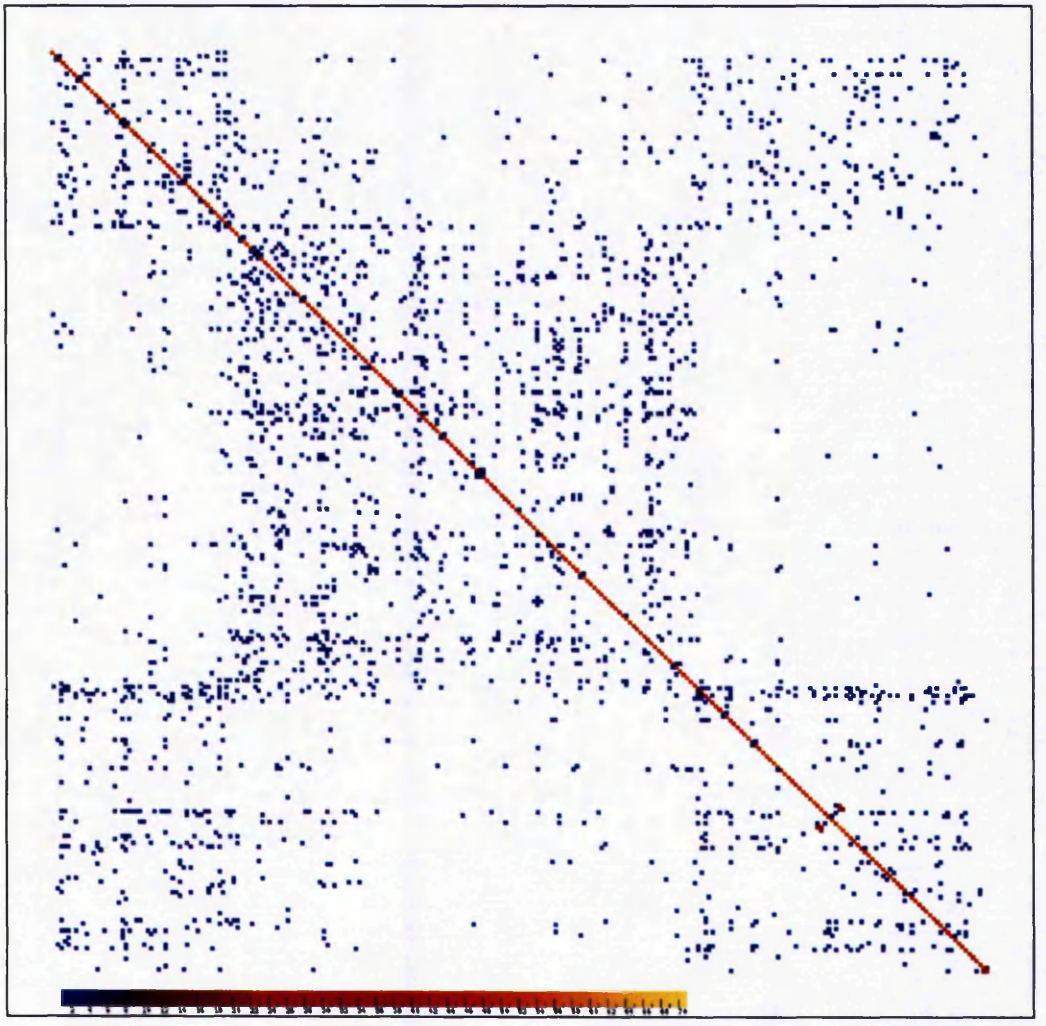

Figure 20 Dot plot of *Chlamydia trachomatis* against itself. It is possible to discern a coarse structure in the genome where the first and last 170,000 base pairs show a set of very weak matches to each other, and the central region of the genome shows a set of weak matches to itself.

Many of the points (such as those which make up the horizontal lines of weakly repetitive sequence) are of extremely low similarity. The dot plot allows these weak regions to be treated as a set, because they can be related pictorially (see Figure 21). If the same set of results were presented in linear form (such as a BLAST output, for example) these matches could not easily be distinguished from statistical noise.

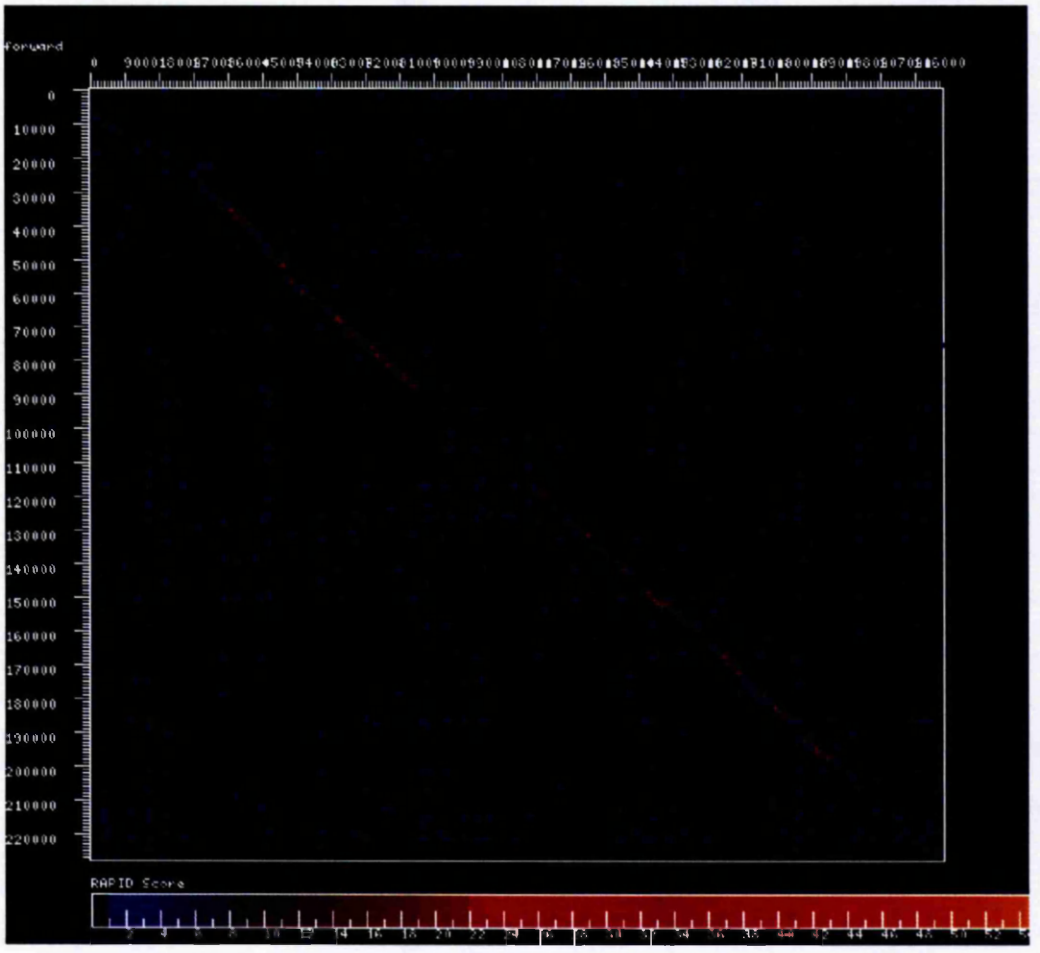

Figure 21 A plot of a human chromosome 12p13 vs. mouse chromosome 6.

Many more such plots can be found at http://www.bioinf.man.ac.uk/rapid, including *E. coli, Bacillus subtilis,* and a set of all sixteen yeast chromosomes against each other.

6.2 Evaluation

This chapter describes the methods that were used to assess RAPID performance both in terms of its speed/memory usage, and also its sensitivity as a similarity search tool. In section 3.2, it was noted that 'similarity' between biological

sequences can mean a number of different things apart from an implied homology. In order to test the ability of an algorithm to perform a comparison, it is necessary to determine the particular question being asked. For example, the test set used to evaluate a tool's ability to predict function by similarity ought to be different from the test set used to determine a tool's ability to correctly identify sequences which contain erroneous vector sequence.

Database search engines operate by analysing the arrangement of residues, represented as lists of letters. This process is purely syntactic - it involves pattern recognition without any attempt to represent the biological 'meaning' of a particular piece of DNA.

Since the questions normally asked of a database search are about biological form or function, a database search carries with it an implicit assumption that similar syntactic structure indicates similar semantic content. This is sensible, because:

1. It is not possible to accurately predict function from DNA, so that there is no way of generating a semantic representation with which to make comparisons.

2. It has been shown to work relatively well.

Given that the majority of database searches are looking for semantic relationships such as function, a test set should address the ability of a syntactic tool to find *semantic* relationships.

6.2.1 Functional classification

6.2.1.1 Artificial Test Sets

Artificially constructed test sets have the potential to be rigorously defined, characterisable and precise.

They are either produced from scratch or derived from real biological data using a mathematical model. The model is designed to produce a set of sequences which simulate the kind of patterns which could be expected in the real world.

Since the semantics of DNA are not well understood, it is not possible to produce model sequences which would be biologically viable; all that can be done is to produce sequences with letters arranged in a way which, according to the statistical methods available, are representative of real DNA.

The algorithm Evolve (Slater 1996) attempts to simulate evolution, and so test a tool's ability to determine similarity due to homology. It translates a coding region to the protein level and then uses a PAM matrix to randomly 'mutate' amino acids before translating the sequence back to the DNA level. The use of PAM data lends the process a minimal amount of semantics, but with no understanding of structural and functional constraints, the resulting sequences are unlikely to be biologically representative.

The algorithm has been used by Anderson and Brass to test the sensitivity of three algorithms – BLAST, FASTA and BIC_SW, a parallel implementation of Smith-Waterman running on a biocelerator, a specialist piece of hardware

designed for running fast Smith-Waterman searches (Anderson and Brass 1998). In order to do this, Evolve was used to create a set containing artificial sequences at varying PAM distances away from a real primate sequence. These sequences were searched against the primate subset of EMBL (which contained the original unmutated sequence) in order to determine the level of mutation required before a search tool was no longer able to find a match. This process was repeated ten times with different sequences.

The results were in line with expectation - Smith-Waterman identified the largest number of correct relationships, BLAST with a word size of six identified less, but, by virtue of its statistical model, had the smallest number of false positives. FASTA came a close third with default parameters. All tools were capable of identifying sequences at an artificial distance of about PAM130.

An attempt to evaluate RAPID's performance, with the same test set yields different results – not one of the artificial sequences was correctly identified. A discussion as to why this is the case can be found in section 6.2.1.3, on page 130.

6.2.1.2 Real data

RAPID has also been evaluated using a test set based on real DNA sequences – again based on the work of Anderson and Brass (1998). The evaluation was designed to assess a tool's ability to assign function to a sequence, and was conducted as follows: a database was created by removing all the tyrosine phosphatases from the primate subset of EMBL. Eight of these sequences were selected and used for the evaluation. Seven of the sequences were added to the

database and the eighth used to search the database for the other seven. This process was repeated eight times, so that each of the sequences was used to search for its peers. Ideally, a tool should, given one sequence, be able to find the other seven and return them as the top hits. It should also be possible to correctly pick a score threshold that allows sequences to be classified into two sets – 'tyrosine phosphatase', and 'not tyrosine phosphatase'. The task is a demanding one because some of the query sequences were a sizeable evolutionary distance apart (the PAM distances of their corresponding protein sequences have been estimated as being between 60 and 400), and none of the current tools performed this search particularly well.

RAPID however, performed marginally better than its contemporaries, identifying 13 out of the 64 possible matches, whilst the other tools only identified 10. The results of the search are detailed in Table 1.

	ID	A	B	C	D	E	F	G
A	M33685	✓						
B	L11329		✓		✓	✓		
C	S78086			✓				
D	U16996		✓		✓	✓		
E	X68277					✓		
F	L18983						✓	
G	D13540							✓

Table 1 Shaded squares are found as top hits by RAPID. Ticks are found as top hits by Smith-Waterman, BLAST, BLAST2 and FASTA.

6.2.1.3 Conclusion and discussion

The stark contrast between RAPID's performance on the real test set and the artificial one demonstrate that the artificial test set is not appropriate for testing a word based search tool. This is a consequence of the artificial test set which was designed to test the performance of alignment algorithms, and should not necessarily be expected to work well with a word based algorithm. The approach makes it possible to determine the maximum edit distance at which alignment tools can be expected to find similarity, and to define this in terms of PAM distances.

Evolve assumes that the rate of evolution across a sequence is constant. As a result, it generates DNA sequences in which increasingly more bases have changed, distributed evenly across the sequence. This is exactly the kind of relationship that alignment algorithms ought to be able to find. By contrast, a

word search tool with a long word size is particularly bad at finding this kind of relationship; instead, it is able to find small highly conserved regions.

In conclusion, mathematically generated test sets must be treated with caution since they carry with them implicit assumptions which were made when the mathematical model was conceived.

6.2.1.4 Vector Contamination

One potential application for an algorithm designed to rapidly compare databases against each other is the task of identifying database entries that show high similarity to vector sequence. It is unfortunate that a small, but significant, proportion of the data is contaminated by vector sequences. Vector contamination can result in a number of errors including incorrect contig assembly and false functional assignment due to spurious matches on vector sequence.

The problems of vector contamination have been studied by a number of researchers (Lamperti *et al.* 1992; Harger *et al.* 1998). Harger *et al.* found that of nearly 100,000 sequences from GSDB, 0.36% contained vector contamination. Although the overall level of vector contamination was found to remain constant over a 5 year period (at less than 1 %) more than 50% of the contamination incorporated into the database came from EST and STS sequences. However, even though more than 50% of the contaminated sequences identified by Harger *et al.* were EST and STS sequences, 43.8% of the sequences added to EMBL between March 1996 and March 1998 were EST and STS sequences (see Table

1). This implies that EST and STS sequences are only marginally more contaminated than other sequences submitted to the database.

	Total size in nucleotides	*ESTs*	*STSs*	*Total growth*	*growth of ESTs+STSs*	*%growth due to EST+STS*
EMBL Rel. 46 Mar '96	473M	152M	9M	-	-	-
EMBL Rel. 54 Mar '98	1428M	561M	18M	954M	418M	43.8

Table 2 An analysis of the growth of the EMBL database showing the proportion of the last two years' growth which can be attributed to EST and STS sequences.

Clearly, the correct identification and annotation of vector contaminations is a task which is important if the integrity of sequence databases is to be improved. At present, this involves comparing each database entry against a database of vector sequences using a tool such as BLAST or FASTA, and examining any sequences which show similarity to a vector sequence; a time consuming task.

The algorithms described in this thesis are particularly well suited to the rapid identification of vector contamination. This section describes the evaluation process that was undertaken to see how well the algorithms performed the task of identifying vector contamination, and the methods used to determine the optimal parameters for identifying vector contamination.

In order to assess the software's efficacy, a test set was produced by artificially introducing progressive amounts of vector sequence into a set of uncontaminated DNA sequences. The test set was used to determine RAPID's ability to correctly classify sequences as being contaminated or uncontaminated, and to determine the optimum score threshold and word size for identifying vector contamination.

The test set was also used to compare the performance of raw matches vs. statistically weighted ones. The specific task addressed in this thesis is that of identifying vector contaminations; a different task is likely to demand different parameters.

The test set was produced by taking an uncontaminated EST and replacing progressive amounts of the sequence with that of a vector, resulting in twenty one sequences containing between zero and two hundred base pairs of contamination. This process was repeated with six ESTs and five different vectors, producing 630 entries. 612 uncontaminated sequences were added to this, to give a test set containing 1242 sequences. No attempt is made to mimic sequencing errors (such as mutations or insertions/deletions). Since sequencing error rates are generally about 3%, and RAPID can identify matches of 30bp, we do not consider this to be a significant flaw in the test set. A sequence was considered to be contaminated if it contained over 30bp of contamination.

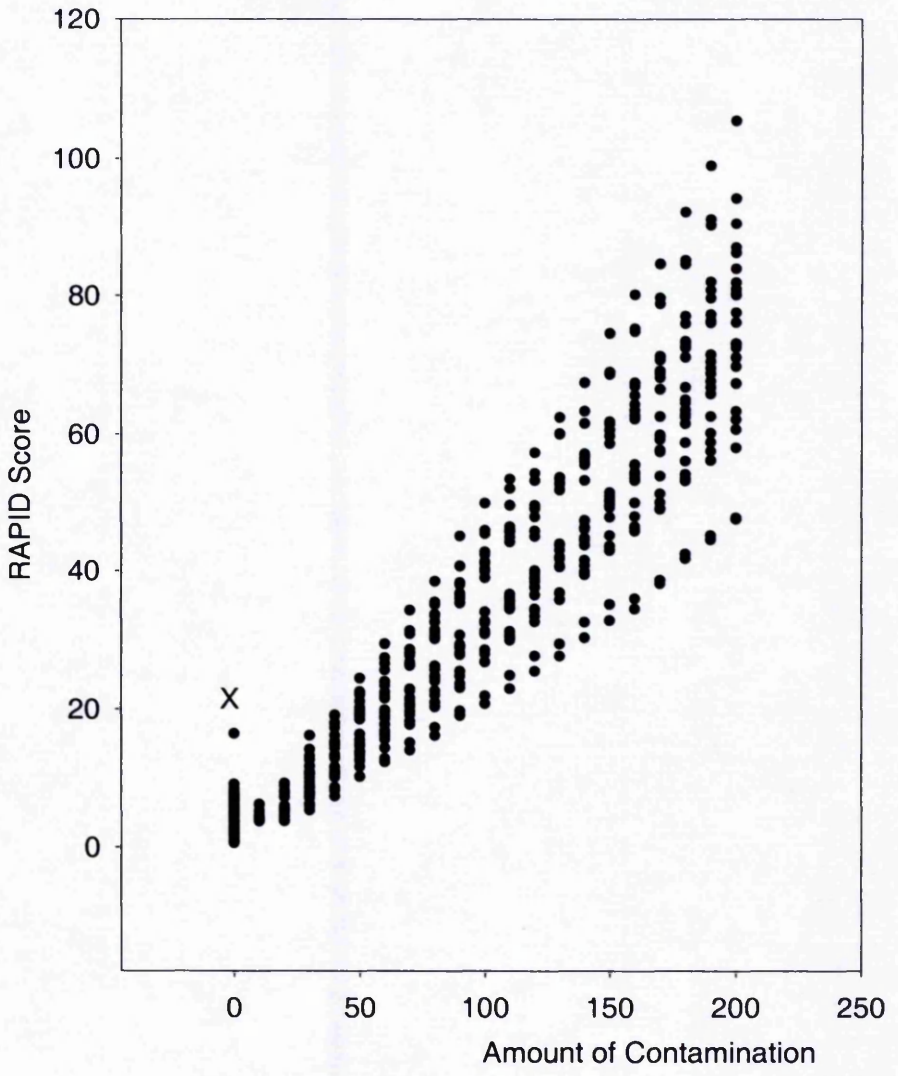

Figure 22 RAPID scores from a search against vector-ig with an artificial test set containing different levels of contamination. The point X refers to the sequence AF011925 which was included in the test set before its similarity to vector was discovered.

Figure 22 shows RAPID scores resulting from a comparison of the artificial test set against vector-ig.

6.2.1.5 Receiver Operator Characteristic (R.O.C.) analysis

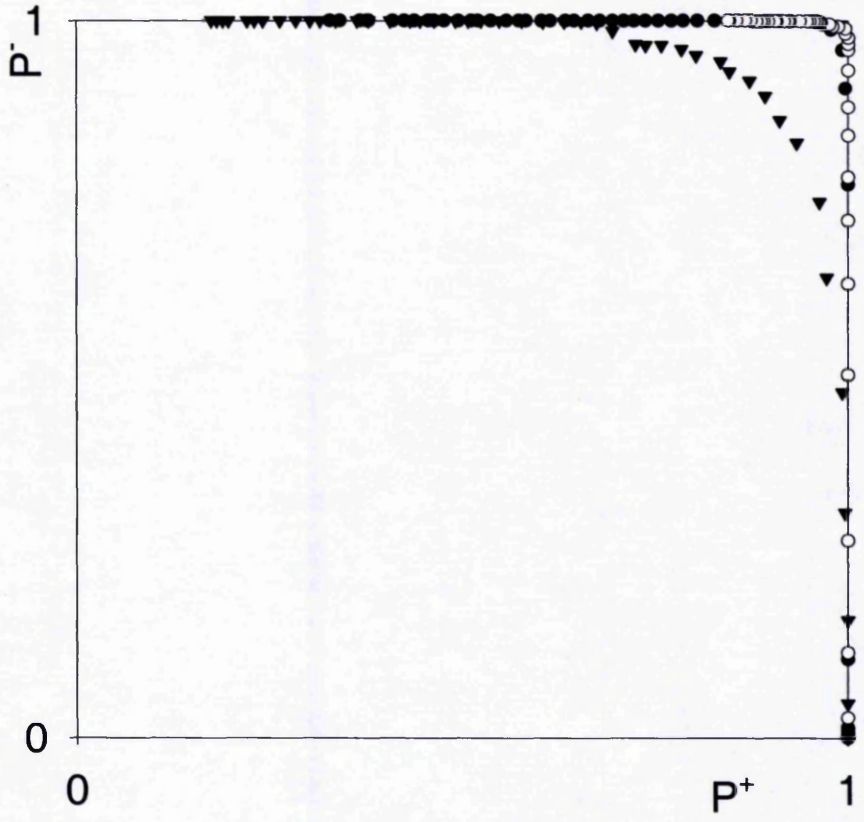

Figure 23 Receiver Operator Characteristic (R.O.C.) curves for searches against vector-ig with an artificially contaminated test set. A contaminated sequence contained at least 30bp of contamination. Filled circles represent weighted 9mers, triangles 8mers. Unfilled circles represent unweighted 9mers. 10mers were left off the graph for clarity.

Receiver Operator Characteristic (R.O.C.) curves (Figure 23) can be used to determine a search tool's ability to correctly classify sequences by calculating the tool's sensitivity and selectivity for different score thresholds.

Given a score between a test sequence Q and a vector sequence, Θ, and a score threshold Θ_c, the test sequence can be assigned to one of four sets:

t^+: true positive; $\Theta > \Theta_c$, Q is contaminated.

t^-: true negative; $\Theta < \Theta_c$, Q is not contaminated.

f^+: false positives; $\Theta > \Theta_c$, Q is not contaminated.

f^-: false negatives; $\Theta < \Theta_c$, Q is contaminated.

The number of sequences in each of these sets (T^+, T^-, F^+, F^-) can be determined for a particular value of Θ_c, allowing P^+ and P^-, sensitivity and selectivity, to be determined for different score thresholds:

$$P^+ = \frac{T^+}{T^+ + F^-} \qquad [40]$$

$$P^- = \frac{T^-}{T^- + F^+} \qquad [41]$$

With a sufficiently low value of Θ_c, every sequence is judged to be contaminated (they have a score above Θ_c), resulting in $P^+ = 1$ and $P^- = 0$. Conversely, for a sufficiently high threshold, every sequence is judged to be uncontaminated, so that $P^+ = 0$ and $P^- = 1$. An ideal tool would have a score threshold which allowed it to correctly identify all contaminated sequences without mis-classifying any uncontaminated ones (P^+=1 and P^-=1). In reality, such a tool does not exist, and it is useful to investigate the relationship between P^+ and P^- for different score thresholds. Such a curve is known as a receiver

operating characteristic (R.O.C.) and ideally should have an area of 1.0. (Shah, I. & Hunter, L. 1997), (Swets 1982).

R.O.C. curves of RAPID scores were produced for k=8, 9, 10 and for the raw number of matches for k=9 (i.e. without any probabilistic weighting). The area under the curve was 0.96 for k = 8-mers, 1.00 for 9-mers and 0.99 for 10-mers. Unweighted 9-mers also produced a curve with area 1.00. However, the test set did not contain any sequences with significant low complexity regions. When these are considered it is evident that probabilistic weighting significantly reduces the scores due to these matches (see Figure 24).

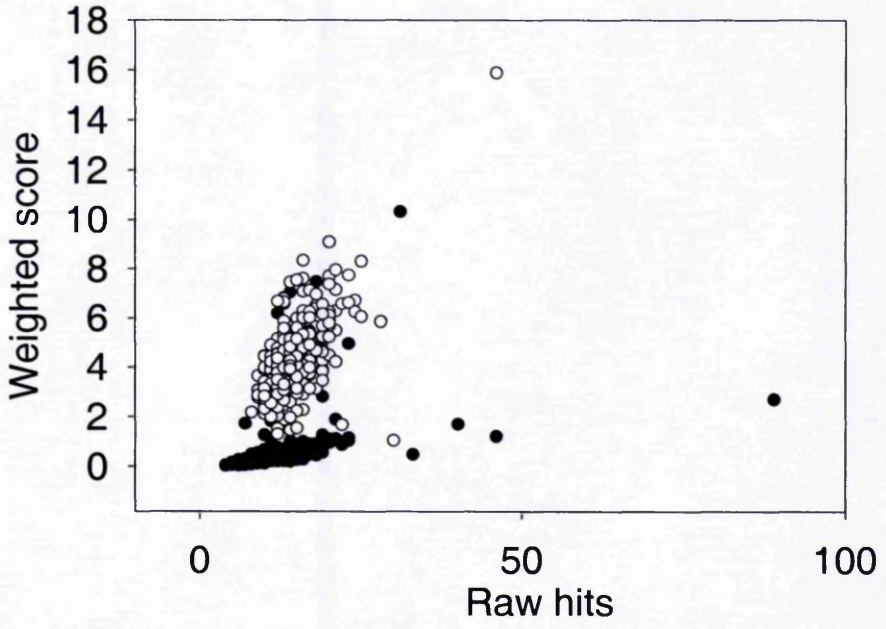

Figure 24 A set of microsatellite repeats (filled circles) and non-repeat DNA sequences (unfilled circles) were searched against vector-ig. Repeat regions which hit against the database were consistently assigned a much lower score than non-repeat regions which matched with a similar number of words.

Although 9-mers and 10-mers have similar discriminatory ability, 9-mers place smaller demands on memory. Thus, k = 9 with probabilistic weighting was selected for spotting vector contaminations. The optimal value for the score threshold Θ_c was determined to be 10 for 9-mers. This is in keeping with the results in Figure 22.

6.2.1.6 Comparison with BLAST

Having established RAPID's ability to correctly classify sequences from the test set, we compared the scores produced by RAPID and BLAST for a number of real sequences.

This was done by taking five randomly selected ESTs which showed varying amount of similarity to vector sequences and using them to search against vector-ig with both algorithms. Each EST hit against a number of vector sequences, resulting in a total of 1803 pair-wise comparisons. A graph of RAPID vs. BLAST scores for each EST/vector pair was plotted (see Figure 25). The approximate straight line demonstrates that RAPID and BLAST identify the same set of matches with a given probe sequence and that the matches are ordered in an equivalent way.

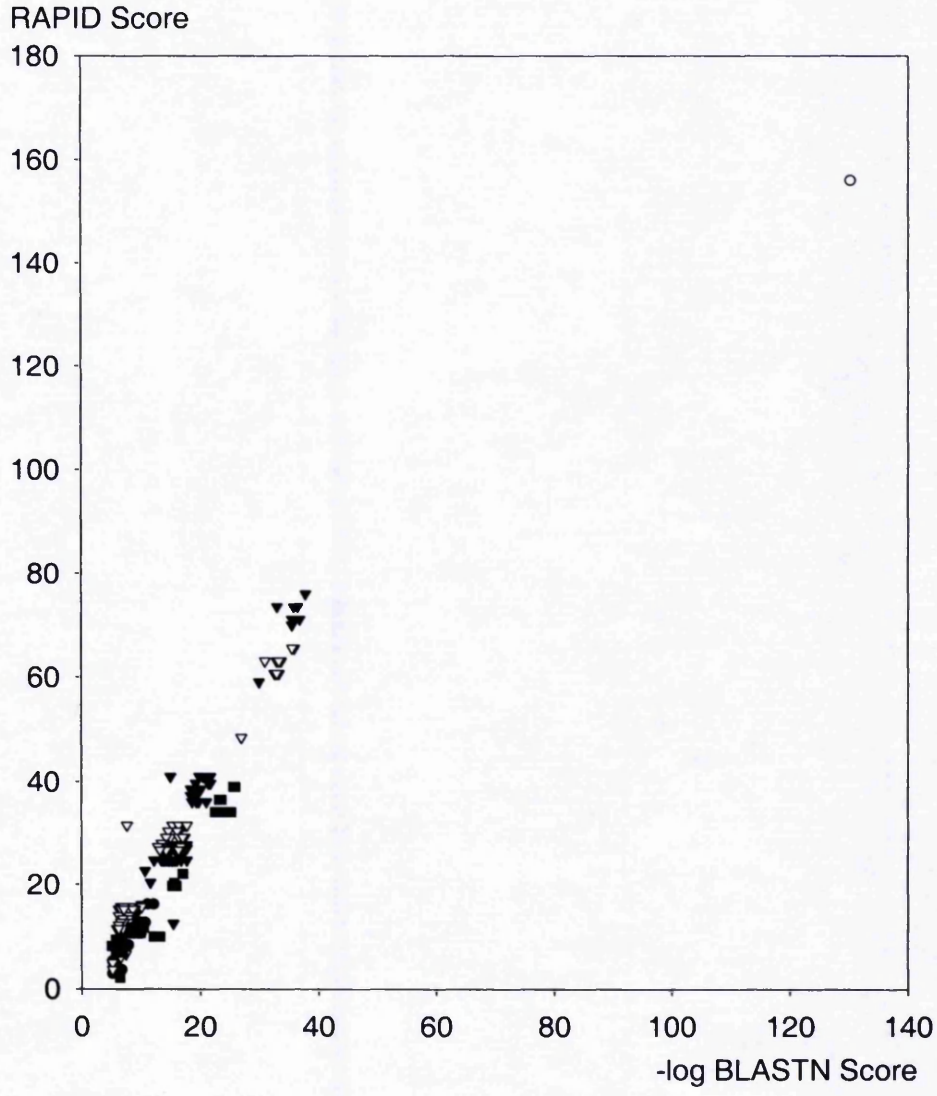

Figure 25 A comparison of RAPID and BLAST scores for sequences with different levels of similarity to those in vector-ig. Each point represents a hit between a query sequence and a particular database sequence. Filled circles: C15000, unfilled circles: X93604, filled triangles: C14014, unfilled triangles: C15706, filled squares: C14077.

6.2.1.7 Speed, memory, and disk usage

Two factors which contribute to the time and space performance of the algorithm are the query database size (which affects the number of times the algorithm

needs to look up a word in the target database) and the target database size (which affects the length of the word lists the algorithm has to process).

The number of table lookups made by the RAPID algorithm is proportional to the number of *k*-mers in the query database, which in turn is approximately proportional to the number of nucleotides it contains.

The time taken to process a table lookup is proportional to the number of sequences which contain that *k*-mer. If it is assumed that the *k*-mer composition of the query database is uniform then the time taken to perform a search should be proportional to the number of *k*-mers in the query.

The size of a given word list, assuming even *k*-mer composition, is slightly sub-linear with respect to database size. This is because words which occur more than once in a sequence are only recorded once by the algorithm. Thus search time should be linear with respect to target database size, and the overall memory usage should also be linear.

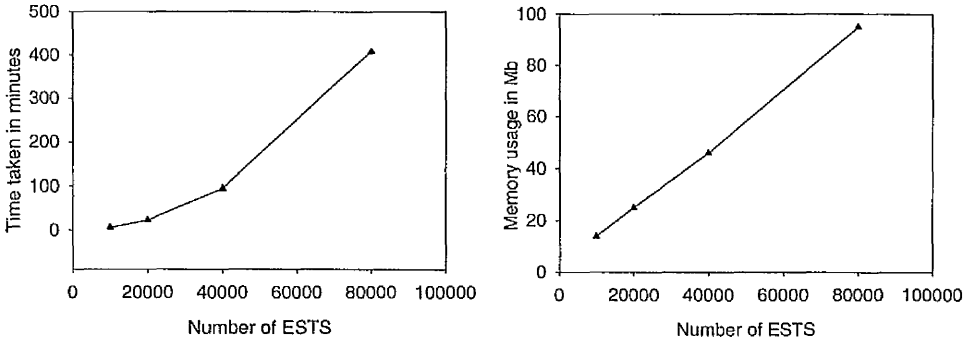

Figure 26 The time taken and memory usage when clustering 10k, 20k 40k and 80k ESTs on a P200 Pro running RAPID.

This is confirmed by the results shown in Figure 26, produced by clustering varying numbers of ESTs.

If it is assumed that the databases are of even composition, then the total number of query-target matches which score above a threshold should also be proportional to query and target database sizes. Thus, if alignments are required, the number of calls to the alignment algorithm should also be proportional to query and target database sizes.

A conservative estimate based on the results in Figure 26 suggests that the EMBL DNA database (currently containing about 1.9Gb) could be clustered in about 6Gb of RAM.

6.2.1.8 Index files

Since the implementation uses memory-mapped IO, the index file on disk is similar in size to the program's memory image. In addition to the index file, the implementation stores the sequence description lines, a probability table, a compressed representation of the DNA sequences in the database and their constituent k-mers (the last two files are used by the alignment tools). For the EMBL EST subset, est10.dat (which contains 50,000 ESTs), this totals 282Mb: about 30% bigger than the original EMBL file.

Indexing times are also fast: est10.dat is indexed in 187 seconds on a P200Pro running LINUX.

6.2.1.9 Comparison with BLAST

Timings were taken using the UNIX `time` command. In order to compare the speed of RAPID and BLAST, est10.dat was searched for vector contaminations by using each tool to compare it against vector-ig. Parameters were set so that each tool only identified and aligned the top hit for each matching EST. RAPID generated ungapped alignments using PHAT. On a Sun Ultra 5 with 256Mb, RAPID takes approximately 33 minutes to perform this search; NCBI BLAST version 2.05, 493 minutes.

6.3 Vector contamination in EMBL

A search of the entire EST subset of EMBL release 54 (which contained 1,506,038 sequences) against vector-ig identified 4096 sequences. This corresponds to an estimated error rate of 0.27%, which is broadly in keeping with Lamperti *et al.* (1998). Parameters were as determined in section 6.2.1.4 – a score threshold of 10.0 and a word size of 9.

A more detailed analysis was conducted for one of the EST subsets: est10.dat. The subset contained 50,000 sequences, of which 412 were identified as having significant similarity to vector – a contamination rate of 0.82%.

A total of 66% of the contaminated sequences identified were found to have been submitted as part of two batch sequencing projects (Nathans J., 1996 unpublished; Lanfranchi *et al.* 1997). This proportion is greatly in excess of their overall contribution to the est10.dat subset (14.5%). Two percent of Lanfranchi's sequences and 2.8% of Nathan's showed significant match to vector.

Of the sequences identified, 171 (41.5%) contained <100 bp of vector, and are likely to be simple editing errors where regions flanking the insert have not been removed before submission. If a restriction site is present in one of these sequences, the vector/insert junction can be identified and the vector cleanly removed.

A total of 241 (58.5%) of the sequences contain >100bp of vector. 131 of these were submitted by Nathans. 55 of these showed significant similarity to the OP region of λ phage (λgt10 was used as the cloning vector).

6.4 *A systematic survey of EMBL database contamination by* E. coli

Since the majority of DNA sequences are cloned into *E. coli*, there is a possibility of sequences being contaminated by fragments from the *E. coli* genome. The recently completed *E. coli* genome sequence makes it possible to perform a systematic search to assess the degree of such contamination and to compare it to previously reported levels of contamination caused by the failure to remove vector sequences.

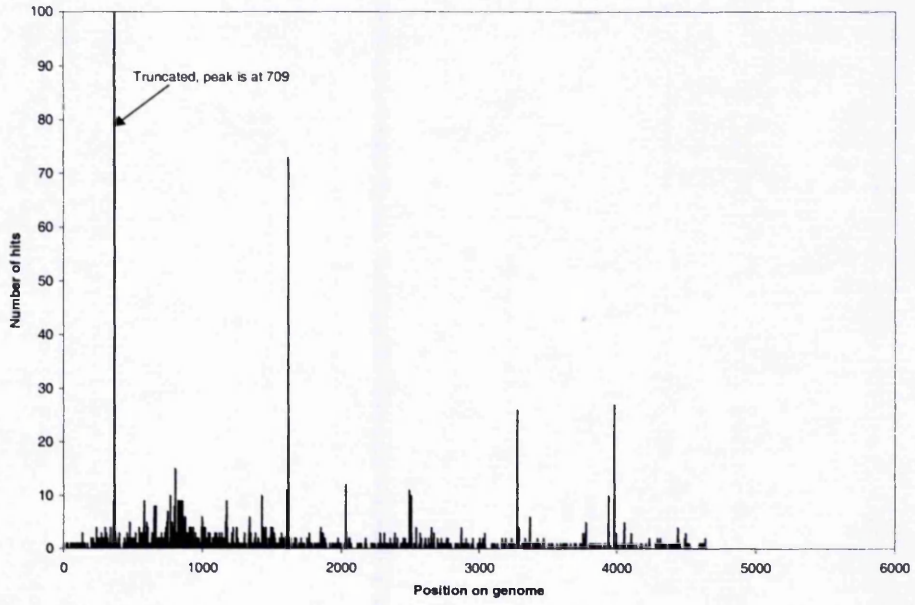

Figure 27 Histogram illustrating the distribution of *E. coli* contaminations in EMBL. The graph shows for a given position on the genome, the number of EMBL sequences which matched. The largest peak corresponds to sequences that hit against the *LacZ* gene, and is due to vector contamination. In order to allow the smaller peaks to be seen more clearly, it has been truncated on this graph – it extends to 709 hits.

Figure 27 was generated by searching the EST subsets of EMBL release 59 against the *E. coli* genome and recording the position where they matched. The *x* axis shows position on the genome, the *y* axis, the number of contaminations that were found due to *E. coli* sequence from that region. The largest peak corresponds to *LacZ* and is due to sequences that have not had vector beyond their restriction enzyme sites removed before being submitted to the database. From the graph, it is clear that a significant amount of contamination arises from regions of *E. coli* that are not near the insert site: an alternative mechanism is responsible for sequencing contamination by genomic *E. coli.* In order to differentiate between contaminations caused by vector sequences and those caused by genomic *E. coli*, sequences which showed a high similarity to vector

were first identified and removed using the methods described in the previous section.

SOURCE OF CONTAMINATION	NUMBER OF HITS	% CONTAMINATION
Vector	8005	0.32
Insertion sequences	123	0.005
E. coli genomic (not vector/LacZ or insertion)	1736	0.06
	Total	0.4

Table 3 Levels of contamination determined in the combined EST subsets of the EMBL databank.

The remaining sequences were then searched against a database of transposable elements; 123 sequences matched and were also eliminated. The final subset was compared against the entire *E. coli* genome. Table 3 shows the distribution of matches found against the combined EST subsets of EMBL containing 2,516,840 sequences.

Further analysis of this data showed that a number of the *E. coli* matches are associated with particular EST projects. For example, 73 sequences from an *Arabidopsis* cDNA sequencing project match a region between positions 1617000 and 1618000 on the *E. coli* genome, bounded by *Sal*I and *Not*I

restriction enzyme sites - the restriction enzymes used in the cloning of the *Arabidopsis* library. It therefore seems likely that the vector DNA was contaminated with a small amount of *E. coli* genomic DNA, fragments of which were then cloned into the vector. In order to investigate this further, other contaminated sequences were examined to determine whether this might in fact be a relatively common mechanism for producing sequencing artifacts. To do this the sequences were searched to see whether there were commonly used restriction enzyme sites near to the start of the sequences. Analysis showed that 31% of the *E. coli* matches had such a site within 20 bases of the start of the sequence.

From this study it can be concluded that the degree of sequence contamination is higher than previously reported. Whilst the majority of contaminations arise from vector sequence, a significant but previously unreported subset are probably caused by contamination of materials used in the production of libraries. In order to maintain the integrity of sequence databases it is important that submissions should be routinely screened against both the *E.coli* genome and a database of vector sequences.

6.5 EMBLScalar

As a consequence of the two surveys described above, it was decided to produce a database EMBL*Scalar*, derived from EMBL by removing contaminated sequences. Producing a database such as this requires a number of significant decisions to be made. Firstly, because it is not possible to find an ideal discriminant between contaminated and not-contaminated, there is a choice

between whether to produce a database which contains (ideally) no contaminated sequences, but from which a number of uncontaminated sequences have also been removed, or alternatively one which has not lost any clean sequences but still contains contaminants. It was decided with EMBLscalar to produce a database that minimised the number of mis-classifications. Therefore, sequences which scored above 10.0 using a RAPID search, and which aligned with greater than 95% similarity were considered contaminated.

The second decision was whether to attempt to clean sequences or to simply eliminate them from the database. Given that often sequences are not annotated with vector or restriction enzymes, it was decided that cleaning up sequences was a difficult task to perform with a sufficient degree of reliability. Since only a small minority of database sequences are contaminated, it was decided to eliminate any sequences which were selected by the search, rather than to try to clean them up.

The database is available at http://www.bioinf.man.ac.uk/emblScalar, along with the software used to generate it, and a list of the restriction enzyme sites used in the analysis of genomic *E. coli* contamination.

6.6 *Discussion of issues arising from the surveys of vector and genomic* E. coli *contamination*

An analysis of the vector contamination in the entire EST subset of EMBL found an overall contamination rate of 0.27%. The approach relies on the vector database having sufficient coverage to hit against every contaminated sequence in the EST database. As a result, it is possible that a number of sequences were missed. However, given that many sequences are submitted without the cloning vector, lab host, and restriction enzyme site being included in their annotation, it is not possible to assess how many.

In many cases, even if the actual vector used for sequencing a given clone is not in the database, a similar (but non-identical) sequence will exist. The λgt10 contaminations discussed earlier are an example of this: the hits found were actually against λ phage.

The work on searching sequences against *E. coli* shows that there is a significant amount of contamination arising from genomic *E. coli* sequence, probably as a result of impure vector DNA being used in library production.

It is important to stress the role that sequence annotation plays in the whole process: poor annotation results in the need to search each sequence against a database of possible contaminations, rather than just the relevant host and cloning vector. The lack of consistent, machine-readable documentation describing the restriction enzyme used during cloning makes it impossible to automatically remove vector sequences from partially dirty database entries.

Analysis of the contaminations shows that the majority arise from a small number of projects which have submitted data without an appropriate level of quality control. Although the increasing use of well designed cloning kits should help reduce the number of errors, quality control is still necessary, and should routinely involve scanning sequences for vector and genomic *E. coli* before submission.

It could be argued that a sequence database acts as a repository for experimental results, allowing an experimental scientist to submit their data to a public site for peer review and analysis. If this view is taken, then it is correct to submit contaminated sequences. However, this should not be an excuse to submit contaminated sequences to a database without annotating them clearly and unambiguously as such.

7 Discussion and conclusions

The initial motivation behind the work described in this thesis was to develop an algorithm for fast and sensitive comparison of high volumes of DNA sequence, such as those generated by the automatic sequencing of Expressed Sequence Tags. In order to do this, a word based approach was considered – leading to the analysis of word distributions in EMBL described in chapter 4. The analysis produced results that suggest that DNA sequences are complex, context sensitive, and hard to model mathematically. This is not surprising given the nature and purpose of DNA. One consequence of this study is that the statistical models used to assess significance of matches need to be considered carefully: a simple random model is probably not an appropriate basis to start with when designing the statistics for a word matching algorithm such as RAPID.

Another interesting outcome is that the distributions of subsequences described in chapter 4 are reminiscent of word distributions for the English language; as seen in the 'Word Frequency Book' by Carroll, Davies and Richman (1971), a pivotal text for the Information Retrieval community. This similarity suggested that a word based algorithm that scaled matches according to their rarity might prove to be an interesting tool, and led to the development of RAPID, the algorithm described in chapter 5. Analysis of RAPID's performance showed that it is an order of magnitude faster than BLAST but that it performs with similar sensitivity. Although this meets the initial aims of the project, a number of improvements were made. Firstly, RAPID presents its output as simply a score. It was decided that this was not acceptable, and that a user would require

significantly more justification before they were prepared to consider that two sequences were similar. For this reason three visualisation tools were developed. Of these, two were alignment tools – partly because alignments are a ubiquitous device within bioinformatics, and therefore familiar to the majority of users, and partly because there was a desire to investigate whether word probabilities could be used to lend context to an alignment and thus increase their utility. The third tool, and perhaps the most interesting one, was a 'coarse grained dot-plotter' that allows similarity between two large sequences such as complete genomes, chromosomes and large gene fragments to be presented in a way that allows many feature to be rapidly identified.

The work in this thesis shows that an extremely fast search tool allows a number of useful tasks to be performed that would otherwise be effectively impossible. The database EMBLscalar can only be produced by performing a database against database sequence comparison: and it is possible to perform this on a PC. The comparative dot plots also require a database against database search, and can also be performed on a PC. The analysis of genomic *E. coli* contamination in EMBL has not been performed before – it was made possible by the existence of a tool fast enough to do the job.

Throughout this thesis, emphasis has been placed on the need for speed. This is, of course, not the only way to deal with some of the problems presented by the size of databases. An alternative approach is to partition the database into subsets and search against these. One possibility is to perform this partitioning by clustering at the sequence level, but this imposes a subjectivity on the databases

which results from the particular algorithm used to perform the partitioning. It is also the case that, when trying to assign function to an unknown sequence, what is required is a set that is diverse at the sequence level, but similar in terms of biological behaviour. For example, a database of G-protein coupled receptors (GPCRs) would be useful for predicting the function of a putative GPCR, but, because the families are sequentially distant, it is unlikely that such a database could be built solely by similarity searching. It is also the case that the appropriate database is dependent on the task in hand. There are a large number of possible databases and they cannot be pre-determined.

Instead, a more desirable approach would be to select sequences in terms of their annotation so that the molecular biologist can build *ad hoc* databases for the specific query they wish to make. This kind of approach is possible using tools such as SRS, but is limited by the unstructured text that forms the majority of sequence annotation. An interesting approach to the problems of size is to use formal, structured representations such as Description Logics, Conceptual Graphs, First Order Predicate Calculus or Frame Systems to represent the knowledge carried in sequence annotation. Presenting the information in a structured form offers the opportunity to significantly enhance database retrieval tasks. It also has the advantage of rendering the information accessible to algorithmic attack, and hence allows that information to be used by analysis tools.

Both RAPID and BLAST both perform with a similar level of sensitivity, even though they work in significantly different ways. This implies that there is not

much more information that can be deduced from sequence data alone, in order to improve the performance of a search. Instead, it will be necessary to use higher level knowledge such as (and this is not meant to be an exhaustive list) 2D structure, co and contra expression, metabolic pathways, phylogenetics, protein interactions, linkage analysis and bibliographic data.

It is also important to note bioinformatics tools do not work in isolation. They are used in a heterogeneous, distributed environment. Results produced by a set of different tools, in different locations on a network, often need to be considered together, so that for example, sequence analysis packages are required to produce data that is subjected to further analysis by other software. Currently, all this must be undertaken in an environment that is almost totally devoid of standards. Different programs require different file formats, typically inter-converted using PERL scripts, and the standard technique for inter-operation across a network is the parsing and production of HTML forms. This is less than ideal. Devising systems and standards that allow a software tool access to such information, and that allow the tool to render it's data accessible to its peers is one of the significant challenges facing bioinformatics as we approach the next millennium.

Appendix A – Glossary

This glossary has been adapted from Terri Attwood's 'Protein sequence analysis A practical guide "A taste of bioinformatics"' which can be found at http://www.bioinf.man.ac.uk/bioactivity/prefacefrm.html. It is included with her kind permission.

Adenine (A) A nitrogenous base, one member of the base pair A-T (adenine-thymine).

Algorithm The logical sequence of steps by which a computational task can be performed.

Alleles Alternative forms of a genetic locus; a single allele for each locus is inherited separately from each parent (e.g., at a locus for eye color the allele might result in blue or brown eyes).

Amino acid Any of a class of 20 molecules that are combined to form proteins in living things. The sequence of amino acids in a protein and hence protein function are determined by the genetic code.

Amino acid sequence The order of amino acids in a protein molecule.

Amplification An increase in the number of copies of a specific DNA fragment; can be in vivo or in vitro. See cloning, polymerase chain reaction.

Antibody An immunoglobulin molecule that forms part of the body's response to a foreign substance (see antigen), to which it binds specifically.

Antigen A substance recognised as 'foreign' by the immune system and which is bound by the variable regions of an antibody.

Arrayed library Individual primary recombinant clones (hosted in phage, cosmid, YAC, or other vector) that are placed in two- dimensional arrays in microtiter dishes. Each primary clone can be identified by the identity of the plate and the clone location (row and column) on that plate. Arrayed libraries of clones can be used for many applications, including screening for a specific gene or genomic region of interest as well as for physical mapping. Information gathered on individual clones from various genetic linkage and physical map analyses is entered into a relational database and used to construct physical and genetic

linkage maps simultaneously; clone identifiers serve to interrelate the multilevel maps. Compare library, genomic library.

Autoradiography A technique that uses X- ray film to visualize radioactively labeled molecules or fragments of molecules; used in analyzing length and number of DNA fragments after they are separated by gel electrophoresis.

Autosome A chromosome not involved in sex determination. The diploid human genome consists of 46 chromosomes, 22 pairs of autosomes, and 1 pair of sex chromosomes (the X and Y chromosomes).

Backup A copy of data or a program made in case a computer crashes.

Bacteriophage See phage.

Base pair (bp) Two nitrogenous bases (adenine and thymine or guanine and cytosine) held together by weak bonds. Two strands of DNA are held together in the shape of a double helix by the bonds between base pairs.

Base sequence The order of nucleotide bases in a DNA molecule.

Base sequence analysis A method, sometimes automated, for determining the base sequence.

Bioinformatics The study of the application of computer and statistical techniques to the management of information. In genome projects, bioinformatics includes the development of methods to search databases quickly, to analyze DNA sequence information, and to predict protein sequence and structure from DNA sequence data.

Biotechnology A set of biological techniques developed through basic research and now applied to research and product development. In particular, the use by industry of recombinant DNA, cell fusion, and new bioprocessing techniques.

Boot Boot-strap - to restart a computer after it has crashed.

bp See base pair.

Bug An error within a program that causes it to misbehave or crash.

Carbonyl group >C=O, occurring in the peptide groups of the main chain of a protein and also in some side chains. Carbonyl groups are polar because the oxygen atom is strongly electronegative.

Catalytic site The region of an enzyme where a chemical reaction takes place, so changing the structure of the enzyme's substrate.

cDNA See complementary DNA.

Centimorgan (cM) A unit of measure of recombination frequency. One centimorgan is equal to a 1% chance that a marker at one genetic locus will be separated from a marker at a second locus due to crossing over in a single generation. In human beings, 1 centimorgan is equivalent, on average, to 1 million base pairs.

Centromere A specialized chromosome region to which spindle fibers attach during cell division.

Chromosomes The self-replicating genetic structures of cells containing the cellular DNA that bears in its nucleotide sequence the linear array of genes. In prokaryotes, chromosomal DNA is circular, and the entire genome is carried on one chromosome. Eukaryotic genomes consist of a number of chromosomes whose DNA is associated with different kinds of proteins.

Clone bank See genomic library.

Clones A group of cells derived from a single ancestor.

Cloning The process of asexually producing a group of cells (clones), all genetically identical, from a single ancestor. In recombinant DNA technology , the use of DNA manipulation procedures to produce multiple copies of a single gene or segment of DNA is referred to as cloning DNA.

Cloning vector DNA molecule originating from a virus, a plasmid, or the cell of a higher organism into which another DNA fragment of appropriate size can be integrated without loss of the vectors capacity for self-replication; vectors introduce foreign DNA into host cells, where it can be reproduced in large quantities. Examples are plasmids, cosmids, and yeast artificial chromosomes; vectors are often recombinant molecules containing DNA sequences from several sources.

cM See centimorgan.

Code See genetic code.

Codon See genetic code.

Complementary DNA (cDNA) DNA that is synthesized from a messenger RNA template; the single-stranded form is often used as a probe in physical mapping.

Complementary sequences Nucleic acid base sequences that can form a double-stranded structure by matching base pairs; the complementary sequence to G-T-A-C is C-A-T-G.

Configuration The arrangement of connecting bonds in a molecule. The configuration of a molecule can only be changed by breaking and remaking covalent bonds (e.g., L- and D-amino acids differ in configuration). See conformation.

Conformation The shape of a protein molecule created by rotations about single bonds. See configuration.

Conserved sequence A base sequence in a DNA molecule (or an amino acid sequence in a protein) that has remained essentially unchanged throughout evolution.

Contig map A map depicting the relative order of a linked library of small overlapping clones representing a complete chromosomal segment.

Contigs Groups of clones representing overlapping regions of a genome.

Cosmid Artificially constructed cloning vector containing the cos gene of phage lambda. Cosmids can be packaged in lambda phage particles for infection into E. coli; this permits cloning of larger DNA fragments (up to 45 kb) than can be introduced into bacterial hosts in plasmid vectors.

Covalent bond A bond between atoms formed by sharing of their electrons.

Crash An unexpected failure of a computer program, or of the operating system itself.

Crossing over The breaking during meiosis of one maternal and one paternal chromosome, the exchange of corresponding sections of DNA, and the rejoining

of the chromosomes. This process can result in an exchange of all eles between chromosomes. Compare recombination.

Cytosine © A nitrogenous base, one member of the base pair G-C (guanine and cytosine).

Debug To remove bugs from a program. See bug.

Deoxyribonucleotide See nucleotide.

Diploid A full set of genetic material, consisting of paired chromosomes one chromosome from each parental set. Most animal cells except the gametes have a diploid set of chromosomes. The diploid human genome has 46 chromosomes. Compare haploid.

Discriminator An abstraction of a conserved motif, or motifs (e.g., a regular expression pattern, or a fingerprint), within an alignment used to search either an individual query sequence or a full database for the occurrence of that same, or similar, motif.

Discriminating power or diagnostic performance A measure of the ability of a discriminator to identify true matches, either in an individual query sequence or in a database.

Disk drive The apparatus that contains a hard disk, or into which a floppy disk is inserted.

DNA (deoxyribonucleic acid) The molecule that encodes genetic information. DNA is a double-stranded molecule held together by weak bonds between base pairs of nucleotides. The four nucleotides in DNA contain the bases adenine (A), guanine (G), cytosine (C), and thymine (T). In nature, base pairs form only between A and T and between G and C; thus the base sequence of each single strand can be deduced from that of its partner.

DNA probes See probe.

DNA replication The use of existing DNA as a template for the synthesis of new DNA strands. In humans and other eukaryotes, replication occurs in the cell nucleus.

DNA sequence The relative order of base pairs, whether in a fragment of DNA, a gene, a chromosome, or an entire genome. See base sequence analysis.

Domain A discrete portion of a protein with its own function. The combination of domains in a single protein determines its overall function.

Double helix The shape that two linear strands of DNA assume when bonded together.

Down The condition of a computer system following a crash.

E. coli Common bacterium that has been studied intensively by geneticists because of its small genome size, normal lack of pathogenicity, and ease of growth in the laboratory.

Electrophoresis A method of separating large molecules (such as DNA fragments or proteins) from a mixture of similar molecules. An electric current is passed through a medium containing the mixture, and each kind of molecule travels through the medium at a different rate, depending on its electrical charge and size. Separation is based on these differences. Agarose and acrylamide gels are the media commonly used for electrophoresis of proteins and nucleic acids.

Endonuclease An enzyme that cleaves its nucleic acid substrate at internal sites in the nucleotide sequence.

Enzyme A protein that acts as a catalyst, speeding the rate at which a biochemical reaction proceeds but not altering the direction or nature of the reaction.

EST Expressed sequence tag. See sequence tagged site.

Ethernet System for the connection of computer networks.

Eukaryote Cell or organism with membrane-bound, structurally discrete nucleus and other well-developed subcellular compartments. Eukaryotes include all organisms except viruses, bacteria, and blue-green algae. Compare prokaryote. See chromosomes.

Evolutionarily conserved See conserved sequence.

Exogenous DNA DNA originating outside an organism.

Exons The protein-coding DNA sequences of a gene. Compare introns.

Exonuclease An enzyme that cleaves nucleotides sequentially from free ends of a linear nucleic acid substrate.

Expressed gene See gene expression.

False positive A sequence incorrectly identified by a discriminator as possessing a particular motif or pattern.

FISH (fluorescence in situ hybridization) A physical mapping approach that uses fluorescein tags to detect hybridization of probes with metaphase chromosomes and with the less-condensed somatic interphase chromatin.

Flow cytometry Analysis of biological material by detection of the light-absorbing or fluorescing properties of cells or subcellular fractions (i.e., chromosomes) passing in a narrow stream through a laser beam. An absorbance or fluorescence profile of the sample is produced. Automated sorting devices, used to fractionate samples, sort successive droplets of the analyzed stream into different fractions depending on the fluorescence emitted by each droplet.

Flow karyotyping Use of flow cytometry to analyze and/or separate chromosomes on the basis of their DNA content.

Format To prepare a floppy disk for use.

Gamete Mature male or female reproductive cell (sperm or ovum) with a haploid set of chromosomes (23 for humans).

Gene The fundamental physical and functional unit of heredity. A gene is an ordered sequence of nucleotides located in a particular position on a particular chromosome that encodes a specific functional product (i.e., a protein or RNA molecule). See gene expression.

Gene expression The process by which a genes coded information is converted into the structures present and operating in the cell. Expressed genes include those that are transcribed into mRNA and then translated into protein and those that are transcribed into RNA but not translated into protein (e.g., transfer and ribosomal RNAs).

Gene families Groups of closely related genes that make similar products.

Gene library See genomic library.

Gene mapping Determination of the relative positions of genes on a DNA molecule (chromosome or plasmid) and of the distance, in linkage units or physical units, between them.

Gene product The biochemical material, either RNA or protein, resulting from expression of a gene. The amount of gene product is used to measure how active a gene is; abnormal amounts can be correlated with disease-causing alleles.

Genetic code The sequence of nucleotides, coded in triplets (codons) along the mRNA, that determines the sequence of amino acids in protein synthesis. The DNA sequence of a gene can be used to predict the mRNA sequence, and the genetic code can in turn be used to predict the amino acid sequence.

Genetic engineering technologies See recombinant DNA technologies.

Genetic map See linkage map.

Genetic material See genome.

Genetics The study of the patterns of inheritance of specific traits.

Genome All the genetic material in the chromosomes of a particular organism; its size is generally given as its total number of base pairs.

Genome projects Research and technology development efforts aimed at mapping and sequencing some or all of the genome of human beings and other organisms.

Genomic library A collection of clones made from a set of randomly generated overlapping DNA fragments representing the entire genome of an organism. Compare library, arrayed library.

Guanine (G) A nitrogenous base, one member of the base pair G-C (guanine and cytosine).

Haploid A single set of chromosomes (half the full set of genetic material), present in the egg and sperm cells of animals and in the egg and pollen cells of plants. Human beings have 23 chromosomes in their reproductive cells. Compare diploid.

Head crash Collision of the reading head of a disk drive with the surface of the disk.

Heterozygosity The presence of different alleles at one or more loci on homologous chromosomes.

Homeobox A short stretch of nucleotides whose base sequence is virtually identical in all the genes that contain it. It has been found in many organisms from fruit flies to human beings. In the fruit fly, a homeobox appears to determine when particular groups of genes are expressed during development.

Homologous chromosomes A pair of chromosomes containing the same linear gene sequences, each derived from one parent.

Homologous sequences Sequences that are related by divergence from a common ancestor. Homology is not a synonym for similarity.

Human gene therapy Insertion of normal DNA directly into cells to correct a genetic defect.

Human Genome Initiative Collective name for several projects begun in 1986 by DOE to (1) create an ordered set of DNA segments from known chromosomal locations, (2) develop new computational methods for analyzing genetic map and DNA sequence data, and (3) develop new techniques and instruments for detecting and analyzing DNA. This DOE initiative is now known as the Human Genome Program. The national effort, led by DOE and NIH, is known as the Human Genome Project.

Hybridization The process of joining two complementary strands of DNA or one each of DNA and RNA to form a double-stranded molecule.

Hydrogen bond A weak electrostatic bond between a hydrogen atom attached to an electronegative (e.g., oxygen or nitrogen) atom, and another electronegative atom.

Hydrophilic group A chemical group that makes favourable interactions with water, generally through hydrogen bonds.

Hydrophobic group A chemical group that cannot make favourable interactions with water, generally because it is non-polar and cannot form hydrogen bonds.

imino group >N-H, occurring in the peptide groups of the main chain of a protein and also in some side chains.

Informatics See Bioinformatics.

In situ hybridization Use of a DNA or RNA probe to detect the presence of the complementary DNA sequence in cloned bacterial or cultured eukaryotic cells.

Interphase The period in the cell cycle when DNA is replicated in the nucleus; followed by mitosis.

Introns The DNA base sequences interrupting the protein-coding sequences of a gene; these sequences are transcribed into RNA but are cut out of the message before it is translated into protein. Compare exons.

In vitro Outside a living organism.

Karyotype A photomicrograph of an individuals chromosomes arranged in a standard format showing the number, size, and shape of each chromosome type; used in low-resolution physical mapping to correlate gross chromosomal abnormalities with the characteristics of specific diseases.

kb See kilobase.

Kilobase (kb) Unit of length for DNA fragments equal to 1000 nucleotides.

Library An unordered collection of clones (i.e., cloned DNA from a particular organism), whose relationship to each other can be established by physical mapping. Compare genomic library, arrayed library.

Linkage The proximity of two or more markers (e.g., genes, RFLP markers) on a chromosome; the closer together the markers are, the lower the probability that they will be separated during DNA repair or replication processes (binary fission in prokaryotes, mitosis or meiosis in eukaryotes), and hence the greater the probability that they will be inherited together.

Linkage map A map of the relative positions of genetic loci on a chromosome, determined on the basis of how often the loci are inherited together. Distance is measured in centimorgans (cM).

Localize Determination of the original position (locus) of a gene or other marker on a chromosome.

Locus (pl. loci) The position on a chromosome of a gene or other chromosome marker; also, the DNA at that position. The use of locus is sometimes restricted to mean regions of DNA that are expressed. See gene expression.

Macrorestriction map Map depicting the order of and distance between sites at which restriction enzymes cleave chromosomes.

Mapping See gene mapping, linkage map, physical map.

Marker An identifiable physical location on a chromosome (e.g., restriction enzyme cutting site, gene) whose inheritance can be monitored. Markers can be expressed regions of DNA (genes) or some segment of DNA with no known coding function but whose pattern of inheritance can be determined. See RFLP, restriction fragment length polymorphism.

Mb See megabase.

Megabase (Mb) Unit of length for DNA fragments equal to 1 million nucleotides and roughly equal to 1 cM.

Meiosis The process of two consecutive cell divisions in the diploid progenitors of sex cells. Meiosis results in four rather than two daughter cells, each with a haploid set of chromosomes.

Messenger RNA (mRNA) RNA that serves as a template for protein synthesis. See genetic code.

Metaphase A stage in mitosis or meiosis during which the chromosomes are aligned along the equatorial plane of the cell.

Mitosis The process of nuclear division in cells that produces daughter cells that are genetically identical to each other and to the parent cell.

Motif A consecutive string of amino acids in a protein sequence whose general character is repeated, or conserved, in all sequences in a multiple alignment at a particular position. Motifs are of interest because they may correspond to structural or functional elements within the sequences they characterise.

mRNA See messenger RNA.

Multifactorial or multigenic disorders See polygenic disorders.

Multiplexing A sequencing approach that uses several pooled samples simultaneously, greatly increasing sequencing speed.

Mutation Any heritable change in DNA sequence. Compare polymorphism.

Nitrogenous base A nitrogen-containing molecule having the chemical properties of a base.

Non-polar molecule A molecule that has uniform distribution of electronic charge.

Nucleic acid A large molecule composed of nucleotide subunits.

Nucleotide A subunit of DNA or RNA consisting of a nitrogenous base (adenine, guanine, thymine, or cytosine in DNA; adenine, guanine, uracil, or cytosine in RNA), a phosphate molecule, and a sugar molecule (deoxyribose in DNA and ribose in RNA). Thousands of nucleotides are linked to form a DNA or RNA molecule. See DNA, base pair, RNA.

Nucleus The cellular organelle in eukaryotes that contains the genetic material.

Oncogene A gene, one or more forms of which is associated with cancer. Many oncogenes are involved, directly or indirectly, in controlling the rate of cell growth.

Overlapping clones See genomic library.

PCR See polymerase_chain reaction.

Phage A virus for which the natural host is a bacterial cell.

Physical map A map of the locations of identifiable landmarks on DNA (e.g., restriction enzyme cutting sites, genes), regardless of inheritance. Distance is measured in base pairs. For the human genome, the lowest-resolution physical map is the banding patterns on the 24 different chromosomes; the highest-resolution map would be the complete nucleotide sequence of the chromosomes.

Plasmid Autonomously replicating, extrachromosomal circular DNA molecules, distinct from the normal bacterial genome and nonessential for cell survival under nonselective conditions. Some plasmids are capable of integrating into the host genome. A number of artificially constructed plasmids are used as cloning vectors.

Polar molecule A molecule that has non-uniform distribution of electronic charge, resulting in partial positive charge in one part of the molecule and complementary negative charge in another part.

Polygenic disorders Genetic disorders resulting from the combined action of alleles of more than one gene (e.g., heart disease, diabetes, and some cancers). Although such disorders are inherited, they depend on the simultaneous presence of several alleles; thus the hereditary patterns are usually more complex than those of single-gene disorders. Compare single-gene disorders.

Polymer A large molecule formed by joining small molecules (monomers) in a long chain.

Polymerase chain reaction (PCR) A method for amplifying a DNA base sequence using a heat-stable polymerase and two 20-base primers, one complementary to the (+)-strand at one end of the sequence to be amplified and the other complementary to the (-)-strand at the other end. Because the newly synthesized DNA strands can subsequently serve as additional templates for the same primer sequences, successive rounds of primer annealing, strand elongation, and dissociation produce rapid and highly specific amplification of the desired sequence. PCR also can be used to detect the existence of the defined sequence in a DNA sample.

Polymerase, DNA or RNA Enzymes that catalyze the synthesis of nucleic acids on preexisting nucleic acid templates, assembling RNA from ribonucleotides or DNA from deoxyribonucleotides.

Polymorphism Difference in DNA sequence among individuals. Genetic variations occurring in more than 1% of a population would be considered useful polymorphisms for genetic linkage analysis. Compare mutation.

Primary structure See amino acid sequence.

Primer Short preexisting polynucleotide chain to which new deoxyribonucleotides can be added by DNA polymerase.

Probe Single-stranded DNA or RNA molecules of specific base sequence, labeled either radioactively or immunologically, that are used to detect the complementary base sequence by hybridization.

Prokaryote Cell or organism lacking a membrane-bound, structurally discrete nucleus and other subcellular compartments. Bacteria are prokaryotes. Compare eukaryote. See chromosomes.

Promoter A site on DNA to which RNA polymerase will bind and initiate transcription.

Prosthetic group A chemical group additional to the polypeptide chain of a protein that is essential for its activity.

Protein A large molecule composed of one or more chains of amino acids in a specific order; the order is determined by the base sequence of nucleotides in the gene coding for the protein. Proteins are required for the structure, function, and regulation of the body's cells, tissues, and organs, and each protein has unique functions. Examples are hormones, enzymes, and antibodies.

Protein fingerprint A group of conserved motifs excised from a sequence alignment, used to build a characteristic signature of family membership.

Protein pattern A single consensus expression derived from a conserved region of a sequence alignment, used as characteristic signature of family membership.

Purine A nitrogen-containing, single-ring, basic compound that occurs in nucleic acids. The purines in DNA and RNA are adenine and guanine.

Pyrimidine A nitrogen-containing, double-ring, basic compound that occurs in nucleic acids. The pyrimidines in DNA are cytosine and thymine; in RNA, cytosine and uracil.

Quaternary structure The arrangement of subunits in a protein molecule.

Rare- cutter enzyme See restriction enzyme cutting site.

Recombinant clones Clones containing recombinant DNA molecules. See recombinant DNA technologies.

Recombinant DNA molecules A combination of DNA molecules of different origin that are joined using recombinant DNA technologies.

Recombinant DNA technologies Procedures used to join together DNA segments in a cell-free system (an environment outside a cell or organism). Under appropriate conditions, a recombinant DNA molecule can enter a cell and replicate there, either autonomously or after it has become integrated into a cellular chromosome.

Recombination The process by which progeny derive a combination of genes different from that of either parent. In higher organisms, this can occur by crossing over.

Regular expression See protein pattern.

Regulatory regions or sequences A DNA base sequence that controls gene expression.

Resolution Degree of molecular detail on a physical map of DNA, ranging from low to high.

Restriction enzyme, endonuclease A protein that recognizes specific, short nucleotide sequences and cuts DNA at those sites. Bacteria contain over 400 such enzymes that recognize and cut over 100 different DNA sequences. See restriction enzyme cutting site.

Restriction enzyme cutting site A specific nucleotide sequence of DNA at which a particular restriction enzyme cuts the DNA. Some sites occur frequently in DNA (e.g., every several hundred base pairs), others much less frequently (rare-cutter; e.g., every 10,000 base pairs).

Restriction fragment length polymorphism (RFLP) Variation between individuals in DNA fragment sizes cut by specific restriction enzymes; polymorphic sequences that result in RFLPs are used as markers on both physical maps and genetic linkage maps. RFLPs are usually caused by mutation at a cutting site. See marker.

RFLP See restriction fragment length polymorphism.

Ribonucleic acid (RNA) A chemical found in the nucleus and cytoplasm of cells; it plays an important role in protein synthesis and other chemical activities of the cell. The structure of RNA is similar to that of DNA. There are several classes of RNA molecules, including messenger RNA, transfer RNA, ribosomal RNA, and other small RNAs, each serving a different purpose.

Ribonucleotides See nucleotide.

Ribosomal RNA (rRNA) A class of RNA found in the ribosomes of cells.

Ribosomes Small cellular components composed of specialized ribosomal RNA and protein; site of protein synthesis. See ribonucleic acid (RNA).

RNA See ribonucleic acid.

Secondary structure Regions of local regularity in the fold of a protein sequence, such as alpha-helices and beta-sheets.

Sequence See base sequence or amino acid sequence.

Sequence alignment A linear comparison of amino (or nucleic) acid sequences in which insertions are made in order to bring equivalent positions in adjacent sequences into the correct register. Alignments are the basis of sequence analysis methods, and are used to pin-point the occurrence of conserved motifs.

Sequence tagged site (STS) Short (200 to 500 base pairs) DNA sequence that has a single occurrence in the human genome and whose location and base sequence are known. Detectable by polymerase chain reaction, STSs are useful for localizing and orienting the mapping and sequence data reported from many different laboratories and serve as landmarks on the developing physical map of the human genome. Expressed sequence tags (ESTs) are STSs derived from cDNAs.

Sequencing Determination of the order of nucleotides (base sequences) in a DNA or RNA molecule or the order of amino acids in a protein.

Sex chromosomes The X and Y chromosomes in human beings that determine the sex of an individual. Females have two X chromosomes in diploid cells;

males have an X and a Y chromosome. The sex chromosomes comprise the 23rd chromosome pair in a karyotype. Compare autosome.

Shotgun method Cloning of DNA fragments randomly generated from a genome. See library, genomic library.

Single- gene disorder Hereditary disorder caused by a mutant allele of a single gene (e.g., Duchenne muscular dystrophy, retinoblastoma, sickle cell disease). Compare polygenic disorders.

Somatic cells Any cell in the body except gametes and their precursors.

Southern blotting Transfer by absorption of DNA fragments separated in electrophoretic gels to membrane filters for detection of specific base sequences by radiolabeled complementary probes.

STS See sequence tagged site.

Supersecondary structure The arrangement of alpha-helices or beta-strands in a protein sequence into discrete folded structures: e.g., beta-barrels, or beta-alpha-beta-motifs.

Tandem repeat sequences Multiple copies of the same base sequence on a chromosome; used as a marker in physical mapping.

Technology transfer The process of converting scientific findings from research laboratories into useful products by the commercial sector.

Telomere The ends of chromosomes. These specialized structures are involved in the replication and stability of linear DNA molecules. See DNA replication.

Tertiary structure The overall fold of a protein molecule.

Thymine (T) A nitrogenous base, one member of the base pair A-T (adenine-thymine).

Transcription The synthesis of an RNA copy from a sequence of DNA (a gene); the first step in gene expression. Compare translation.

Transfer RNA (tRNA) A class of RNA having structures with triplet nucleotide sequences that are complementary to the triplet nucleotide coding sequences of mRNA. The role of tRNAs in protein synthesis is to bond with amino acids and transfer them to the ribosomes, where proteins are assembled according to the genetic code carried by mRNA.

Transformation A process by which the genetic material carried by an individual cell is altered by incorporation of exogenous DNA into its genome.

Translation The process in which the genetic code carried by mRNA directs the synthesis of proteins from amino acids. Compare transcription.

tRNA See transfer RNA.

True positive A sequence correctly identified by a discriminator as possessing a particular motif or pattern.

Uracil A nitrogenous base normally found in RNA but not DNA; uracil is capable of forming a base pair with adenine.

van der Waals forces Weak forces that occur between any pair of atoms. The forces are attractive when the atoms are close, but become repulsive when they are too close.

Vector See cloning vector.

Virus A noncellular biological entity that can reproduce only within a host cell. Viruses consist of nucleic acid covered by protein; some animal viruses are also surrounded by membrane. Inside the infected cell, the virus uses the synthetic capability of the host to produce progeny virus.

VLSI Very large-scale integration allowing over 100,000 transistors on a chip.

YAC See yeast artificial chromosome.

Yeast artificial chromosome (YAC) A vector used to clone DNA fragments (up to 400 kb); it is constructed from the telomeric, centromeric, and replication origin sequences needed for replication in yeast cells. Compare cloning vector, cosmid.

References

Aaronson JS, Eckman B, Blevins RA, Borkowski JA, Myerson J, Imran S, Elliston KO (1996) Toward the development of a gene index to the human genome: an assessment of the nature of high-throughput EST sequence data. *Genome Res* **6**(9):829-45

Adams MD, Dubnick M, Kerlavage AR, Moreno R, Kelley JM, Utterback TR, Nagle JW, Fields C, Venter JC (1992) Sequence identification of 2,375 human brain genes. *Nature* **355**(6361):632-4

Altschul SF, Boguski MS, Gish W, Wooton JC (1994) Issues in searching molecular sequence databases *Nat. Gen.* **6** 119-128.

Altschul SF, Gish W (1996) Local Alignment statistics *Methods in Enzymology* **366** 460-480.

Altschul SF, Gish W, Miller W, Myers E, Lipman D (1990) Basic Local Alignment Search Tool. *J. Mol. Biol.* **215** 403-410.

Altschul SF, Madden TL Schäffer AA, Zhang J, Zhang Z Miller W Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs *Nucleic Acids Research* **25(17)** 3389-3402.

Anderson C (1993) Genome Shortcut Leads to Problems. *Science* **259** 1684 – 1687.

Anderson I and Brass A (1998) Searching DNA databases for similarities to DNA sequences: when is a match significant? *Bioinformatics* **14**(4):349-356.

Argos, P. (1987) A sensitive procedure to compare amino acid sequences *J. Mol. Biol.* **193**:385-396.

Attwood T. K., Flower D. R., Lewis A. P., Mabey J. E., Morgan S. R., Scordis P., Selley J. N. and Wright W. (1999) PRINTS prepares for the new millennium *Nuc. Acid. Res.* **27**(1) 220 -225

Bairoch A & Apweiler R (1999) The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999 *Nuc. Acid. Res.* **27**(1) 49-54

Baker PG, Brass A, Bechhofer S, Goble C, Paton N, Stevens R (1998) TAMBIS--Transparent Access to Multiple Bioinformatics Information Sources. *Ismb 1998*;**6**:25-34.

Baker PG, Goble CA, Bechhofer S, Paton NW, Stevens R, Brass A (1999) An ontology for bioinformatics applications.*Bioinformatics* **15**(6):510-20

Benson D.A, Boguski M.S, Lipman D.J., Ostell J., Francis B.F., Ouellette, Rapp B.A. and Wheeler W. (1999) GenBank *Nuc. Acid. Res.* **27**(1)

Bishop M, Thompson E. (1984) Fast computer searches for similar DNA sequences. *Nucleic Acids Res.* **12**(13):5471-5474.

Blake JA, Richardson JE, Davisson MT, Eppig JT (1999) The Mouse Genome Database (MGD): genetic and genomic information about the laboratory mouse. The Mouse Genome Database Group. *Nucleic Acids Res* **27**(1):95-8

Boguski MS, Lowe TM, Tolstoshev CM (1993) dbEST – database for "expressed sequence tags". *Nat. Genet.* **4**(4):332-3

Boguski MS, Schuler GD (1995) ESTablishing a human transcript map. *Nat Genet* **10**(4):369-71

Brassard, G. & Bratley, P. (1996) Fundamental Algorithms *Addison Wesley NY USA.*

Brown T.A (1999) The genome. *John Wiley NY USA.*

Burke DT, Carle GF and Olson MC (1987) Cloning of large segments of exogenous DNA into yeast by means of artificial chromosome vectors. *Science* **236** 608 – 812.

Carroll JB, Davies P and Richman B (1971) The American heritage word frequency book. *Houghton Miflin* Boston.

Chen P. (1976) The Entity Relationship Model: Toward a Unified View of Data *ACM Transactions on database Systems* **1**:9-36.

Chen RO, Felciano R, Altman RB (1997) RIBOWEB: linking structural computations to a knowledge base of published experimental data. *Ismb 1997 AAAI Press.* **5**:84-7

Codd E. F. (1970) A relational model of data for large shared data banks. *Communications of the ACM* (13).

Cohen D, Chumakov I and Weissenbach J (1993) A first generation map of the human genome. *Nature* **366**:698-701.

Collins FS, Patrinos A, Jordan E, *et al.* (1998) New goals for the human genome project: 1998 –2003. *Science* **282,** 682 – 689.

Costanzo F. *et al.* (1983) Cloning of several cDNA segments coding for human liver proteins. *EMBO J.* **2**:57-61.

Craik, C. Rutter, W. Fletterick, R. (1983) Splice junctions: association with variation in protein structure. *science* **220**:1125-1129.

Dayhoff MO (1974) Computer analysis of protein sequences. *Fed Proc* **33**(12):2314-6

Dayhoff MO, Schwartz RM, Orcutt BC (1978) A model of Evolutionary change in proteins *in* Atlas of Protein Sequence and Structure **5**:345-352. *ed.* Dayhoff MO *Nat. Biomed. Res. Found.*, Washington D.C.

Delcher AL, Kasif S, Fleischmann RD, Peterson J, White O, Salzberg SL (1999) Alignment of whole genomes. *Nucleic Acids Res* **27**(11):2369-76

Doolittle WF (1999) Phylogenetic classification and the universal tree. *Science* **284**(5423):2124-9

Eilbeck K, Brass A, Paton N, Hodgman C (1999) INTERACT: An object oriented protein-protein interaction database *ISMB 1999*:87-94.

Etzold T, Ulyanov A, Argos P (1996) SRS: information retrieval system for molecular biology data banks. *Methods Enzymol* **266**:114-28

Fields S, Sternglanz (1994) The two-hybrid system: an assay for protein-protein interactions. *Trends Genet* **10**, 286-92

Fitch WM (1966) An improved method for testing for evolutionary homology. *L. Mol. Biol.* **16**:9-16

Fitch WM (1969) Locating gaps in amino acid sequences to optimize the homology between two proteins. *Biochem. Genet.* **3**:99-108.

Fleishmann RD, Adams MD, White O, *et al.* (1995) Whole genome random sequencing and assembly of *Haemophilus influenzae Rd. Science* **269** 496-512.

Fodor SP, Read JL, Pirrung MC, Stryer L, Lu AT, Solas D (1991) Light-directed, spatially addressable parallel chemical synthesis. *Science* **251**(4995):767-73

Gammie AE, Stewart BG, Scott CF, Rose MD (1999) the two forms of karyogamy transcription factor Kar4p are regulated by differential initiation of, transcription, translation, and protein turnover. *Mol. Cell Biol.* **19**(1):817-25

Gerhold D and Caskey T (1996) It's the genes! EST access to human genome content *BioEssays* **18**(12):973-981.

Gibbs, AJ. and Mcintyre, GA. (1970) The diagram, a method for comparing sequences. its use with amino acid and DNA sequences. *Eur. J. Biochem.* **16:**1-11.

Goodfellow P (1995) A big book of the human genome. Complementary endeavours. *Nature* **377**(6547):285-6

Gress TM, Hoheisel JD, Lennon GG, Zehetner G, Lehrach H (1992) Hybridization fingerprinting of high-density cDNA-library arrays with cDNA pools derived from whole tissues. *Mamm Genome* **3**(11):609-19

Harger C, Skupski M, Bingham J, Farmer A, Hoisie S, Hraber P, Kiphart D, Krakowski L, McLeod M, Schwertfeger J, Seluja G, Siepel A, Singh G, Stamper D, Steadman P, Thayer N, Thompson R, Wargo P, Waugh M, Zhuang JJ, Schad PA (1998) The Genome Sequence DataBase (GSDB): improving data quality and data access. *Nucleic Acids Res.* **26**(1):21-26

Henikoff JG, Henikoff S & Pietrokovski S (1999a) New features of the Blocks Database servers *Nucl. Acids Res.* **27**:226-228

Henikoff S, Henikoff JG & Pietrokovski S, (1999b) Blocks+: A non-redundant database of protein alignment blocks dervied from multiple compilations *Bioinformatics* **15**(6):471-479

Henikoff S., Henikoff J.G.(1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U.S.A.* **89**:10915-10919.

Hide W. *et al.* (1999) Sequence Tag Alignment and Consensus Knowledgebase (STACK) documentation. *http://psytrance.sanbi.ac.za/stack/stackdoc/index.html*

Hough P.V.C. (1962) Methods and Means for Recognizing Complex Patterns, *US Patent* 3,069,654.

Ioannou PA, Amemia CT, Garnes J, Kroisel PM, Shizuya H, Chen C, Batzer MA and de Jong PJ (1994) P1 derived vector for propogation of large human DNA fragments *Nature Genet.* **6** 84-89.

Johnson RA, Wichern DW (1998) Applied multivariate statistical analysis *Prentice Hall* p279.

Jordan B. (1998) Large-Scale Expression Measuurment by Hybridization Methods: From High-Density Membranes to "DNA Chips". *J. Biochem.* **124**:251-258.

Karlin S, Altschul SF. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes *proc. Natl. Acad. Sci. USA* **87** 2264-2268.

Karp PD, Riley M, Paley SM, Pellegrini-Toole A, Krummenacker M (1999) Eco Cyc: encyclopedia of Escherichia coli genes and metabolism. *Nucleic Acids Res* **27**(1):55-8

Khrapko KR, Lysov YuP, Khorlin AA, Ivanov IB, Yershov GM, Vasilenko SK, Florentiev VL, Mirzabekov AD (1991) A method for DNA sequencing by hybridization with oligonucleotide matrix. *DNA Seq* **1**(6):375-88

Kim U-J, Shizuya H, de Jong Pj, Birron B and Simon MI (1992) Stable propagation of cosmid and human DNA inserts in an F factor based vector. *Nucleic Acids Research* **20** 1083 – 1085.

Kolakowski LF (1999) GCRDb: A G-protein coupled receptor database *Receptors and channels: The International Journal of Receptors, Channels and Transporters* (in press).

Kurian KM, Watson CJ, Wyllie AH (1999) DNA chip technology. *J Pathol* **187**(3):267-71

Lamperti ED, Kittelberger JM, Smith TF, Villa-Komaroff L. (1992) Corruption of genomic databases with anomalous sequence. *Nucleic Acids Res.* **20**(11):2741-2747

Lanfranchi G, Muraro T, Caldara F, Pacchioni B, Pallavicini A, Pandolfo D, Toppo S, Trevisan S, Scarso S, Valle G (1996) Identification of 4370 expressed sequence tags from a 3'-end-specific cDNA library of human skeletal muscle by DNA sequencing and filter hybridization. *Genome. Res.* **6**(1):35-42.

Lange R and Hengge-Aronis R (1994) The cellular concentration of the sigma S subunit of RNA polymerase in *Escherichia coli* is controlled at the levels of transcription, translation and protein stability. *Genes Dev.* **8**(13):1600-12.

Lipman DJ, Pearson WR (1985) Rapid and sensitive protein similarity searches. *Science* **227**:(4693):1435-41.

Maizel, JV. Lenk RP. (1981) Enhanced graphic matrix analysis of nucleic acid and protein sequences *PNAS USA* **86**:4412-4415.

McLachan AD. (1971) Tests for comparing relayted amino acid sequences: cytochrome c and cytochrome c551. *J. Mol. Biol.* **61**:409-424.

McLachan AD. (1972) Repeating sequences and gene duplication in proteins *J.Mol.Biol.* **72**: 417-437.

McLachan AD. (1983) Analysis of gene duplication repeats in the myosin rod. *J. Mol. Biol.* **169**:15-30.

Miller C.J., Gurd, J., Brass A. (1999) A RAPID algorithm for sequence database comparison: application to the identification of vector contamination in the EMBL databases *Bioinformatics* **15**(2):111-121

Nature Genetics Editorial (1999) The chip challenge. *Nat. Genet.* **21**(1):61-2

Needleman, SB. and Wunsch CD. (1970) A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.* **48:**443-453.

Nguyen C, Rocha D, Granjeaud S, Baldit M, Bernard K, Naquet P, Jordan BR (1995) Differential gene expression in the murine thymus assayed by quantitative hybridization of arrayed cDNA clones *Genomics* **29**(1):207-16

Okubo K. *et al.* (1992) Large scale cDNA sequencing for analysis of quantitative and qualitative aspects of gene expression *Nature Genetics* **2**:173-179.

Oliver SG, van der Aart QJM, Agostini-Carbone ML *et al.* (1992) The Complete DNA sequence of yeast chromosome III *Nature* **357** 38 – 46.

Patterson, C. (1988) Homology in classical and molecular biology. *Mol. Biol. Evol.* **56:**603-625.

Pearson WR. (1990) "Rapid and Sensitive Sequence Comparison with FASTP and FASTA" *Methods in Enzymology* **183**:63- 98.

Pearson WR., Lipman DJ. (1988), Improved Tools for Biological Sequence Analysis, *PNAS* **85**:2444- 2448.

Pietu G, Alibert O, Guichard V, Lamy B, Bois F, Leroy E, Mariage-Sampson R, Houlgatte R, Soularue P, Auffray C (1996) Novel gene transcripts preferentially expressed in human muscles revealed by quantitative hybridization of a high density cDNA array. *Genome Res* **6**(6):492-503.

Pustell, J. Kafatos, FC. (1982) A high speed, high capacity homology matrix: zooming through SV40 and polyoma. *Nucleic Acid Res.* **10:**4765-4782.

Pustell, J. Kafatos, FC. (1984) A convenient and adaptable package of computer programs for DNA and protein sequence management, analysis and homology determination. *Nucleic Acid Res.* **12**:643-655.

Reisner, AH. Bucholtz, CA. (1988) The use of various properties of amino acids in colour and monochrome dot-matrix analyses of protein homologies. *CABIOS* **4**:395-402.

Rigoutsos, I. Califano, A. (1994) Searching in Parallel for similar strings *IEEE Comp. Sci. Eng.* Summer 1994:60-75

Rodriguez-Tome P, Lijnzaad P (1997) The Radiation Hybrid Database. *Nucleic Acids Res* **25**(1):81-4

Sayle RA, Milner-White EJ (1995) RASMOL: biomolecular graphics for all. *Trends Biochem Sci* **20**(9):374

Schuler GD (1998) Sequence alignment and database searching. *Methods Biochem Anal* **39**:145-71

Sellers, P. (1974) On the theory and computation of evolutionary distances *SIAM J. Appl. Math.* **26**:787-793.

Shah I., Hunter L. (1997) Proceedings Fifth International Conference on Intelligent Systems for Molecular Biology. *AAAI Press California.* 276 - 283.

Shannon C.E., (1948) A mathematical theory of communication, *Bell System Technical Journal*, **27**:379-423 and 623-656, July and October.

Shizuya H, Birren B, Kim UJ, Mancino V, Slepak T, Tachiiri Y and Simon M (1992) Cloning and stable maintenance of 300 kilobase-pair fragments of hman DNA in *Escherichia coli* using an F-factor-based vector. *Proc. Natl. Acad. Sci.* **89**:8794-8797.

Shortliffe, E.H. (1976). Computer-Based Medical Consultation: MYCIN. *New York, American Elsevier.*

Skupski MP, Booker M, Farmer A, Harpold M, Huang W, Inman J, Kiphart D, Kodira C, Root S, Schilkey F, Schwertfeger J, Siepel A, Stamper D, Thayer N, Thompson R, Wortman J, Zhuang JJ, Harger C (1999) The Genome Sequence DataBase: towards an integrated functional genomics resource. *Nucleic Acids Res* **27**(1):35-8

Slater G. (1996) Modelling of molecular evolution as an approach to a quantitative evaluation of sequence comparison algorithms. *MSc Thesis, University of Manchester.*

Smith, TF. and Waterman, MS. (1981) Identification of common molecular subsequences *J. Mol. Biol.* **147**:195-197.

Soares MB, Bonaldo MF, Jelene P, Su L, Lawton L, Efstratiadis A (1994) Construction and characterization of a normalized cDNA library. *Proc Natl Acad Sci U S A* **20**:9228-32.

Southern EM, Maskos U, Elder JK (1992) Analyzing and comparing nucleic acid sequences by hybridization to arrays of oligonucleotides: evaluation using experimental models. *Genomics* **13**(4):1008-17

Staden, R. (1982) An interactive graphics program for comparing and aligning nucleic acid and amina acid sequences. *Nucleic Acids. research.* **10**:2951-2961.

States, DJ. Boguski, MS. (1991) Similarity and Homology. *Sequence Analysis Primer eds. Grobskov M, Devereux J.* 89-157. Stockton press UK.

Sternberg MJ, Bates PA, Kelley LA, MacCallum RM (1999) Progress in protein structure prediction: assessment of CASP3. *Curr Opin Struct* **9**(3):368-73

Sternberg N (1990) Bacteriophage P1 cloning system for the isolation, amplification and recovery of DNA fragments as large as 100 kilobase pairs. *Proc. Natl. Acad. Sci, USA.* **87**:103-107.

Stoesser G, Tuli MA, Lopez R, Sterk P. (1999) The EMBL Nucleotide Sequence Database. *Nucleic Acids Res* **27**(1):18-24.

Sussman JL, Lin D, Jiang J, Manning NO, Prilusky J, Ritter O, Abola EE (1998) Protein Data Bank (PDB): database of three-dimensional structural information of biological macromolecules. *Acta Crystallogr D Biol Crystallogr* **54**(1.6):1078-84

Swets, J. (1982) *Measuring the Accuracy of Diagnostic Systems.* New York: Academic Press.

Torney, D.C., C. Burks, D. Davison, and K.M. Sirotkin. (1990). Computation of d2 A measure of sequence dissimilarity, 109-125 *In Bell, G. and Marr, T., Eds. Computers and DNA, Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley, New York.*

Venter JC, Adams MD, Sutton GG, Kerlavage AR, Smith HO, Hunkapiller M (1998) Shotgun sequencing of the human genome. *Science* **280**(5369):1540-2

White O, Kerlavage AR (1996) TDB: new databases for biological discovery. *Methods Enzymol* **266**:27-40

Wilbur and Lipman (1983) Rapid similarity searches of nucleic acid and protein data banks *Proc. Natl. Acad. Sci.* (USA) **80;** 726-730.

Wilcox AS, Kahn AS, Hopkins JA and Sikela JM (1991) Use of 3' translated sequences of human cDNAs for rapid chromosome assignment and conversion to STSs: implications for an expression map of the genome. *Nucleic Acids Res* **19**(8):1837-43

Wolfsberg TG, Landsman D (1997) A comparison of expressed sequence tags (ESTs) to human genomic sequences. *Nucleic Acids Res* **25**(8):1626-32

Zhang J, Madden TL (1997) PowerBLAST: a new network BLAST application for interactive or automated sequence analysis and annotation. *Genome Res* 7(6):649-56.

Zhao N, Hashida H, Takahashi N, Misumi Y, Sakaki Y High-density cDNA filter analysis: a novel approach for large-scale, quantitative analysis of gene expression. *Gene* **156**(2):207-13 .