# Rapid and sensitive methods for protein sequence comparison and database searching

Torbjørn Rognes

Institute of Medical Microbiology
University of Oslo

July 2000

ISBN 82-7633-154-8

Printed in Norway.

Created with Adobe InDesign, Illustrator, Photoshop and Acrobat, plus Microsoft Word and Excel.

Typeset in Times New Roman and Myriad 11/13.

This thesis is submitted to the Faculty of Medicine at the University of Oslo for evaluation as partial fulfillment of the requirements for the degree of doctor philosophiae.

**Front cover: Visualisation of a Smith-Waterman alignment of two proteins.** The human hXTH protein (518 aa) (see paper IV) (vertical) is aligned with the *E.coli* XTH protein (268 aa) (horisontal). The color of each pixel in the image represent the value of one cell (the *h*-value) in the Smith-Waterman alignment matrix, ranging from black for zero score to white for the highest score. See paper II for details.

# Contents

# List of abbreviations

| | |
|---|---|
| aa | **a**mino **a**cid(s) |
| AP | **ap**urinic/**ap**yrimidinic |
| BLAST | **b**asic **l**ocal **a**lignment **s**earch **t**ool |
| bp | **b**ase **p**air(s) |
| CATH | **c**lass, **a**rchitecture, **t**opology, **h**omology |
| CPU | **c**entral **p**rocessing **u**nit |
| DDBJ | **D**NA **D**ata **B**ank of **J**apan |
| DNA | **d**eoxyribo**n**ucleic acid |
| EBI | **E**uropean **B**ioinformatics **I**nstitute |
| EMBL | **E**uropean **M**olecular **B**iology **L**aboratory |
| EST | **e**xpressed **s**equence **t**ag |
| FPGA | **f**ield **p**rogrammable **g**ate **a**rray |
| FSSP | **f**old classification based on **s**tructure-**s**tructure alignment of **p**roteins |
| GFP | **g**reen **f**luorescent **p**rotein |
| HMM | **h**idden **M**arkov **m**odel |
| HSP | **h**igh-scoring **s**egment **p**air |
| IMAGE | **i**ntegrated **m**olecular **a**nalysis of **g**enomes and their **e**xpression |
| IUPAC | **I**nternational **U**nion of **P**ure and **A**pplied **C**hemistry |
| IUBMB | **I**nternational **U**nion of **B**iochemistry and **M**olecular **B**iology |
| MHz | **m**ega**h**ert**z** |
| MMS | **m**ethyl **m**ethane**s**ulfonate |
| MMX | **m**ulti**m**edia e**x**tensions |
| MSP | **m**aximal-scoring **s**egment **p**air |
| NCBI | **N**ational **C**enter for **B**iotechnology **I**nformation |
| PAM | **p**oint **a**ccepted **m**utation |
| PDB | **p**rotein **d**ata **b**ank |
| PIR | **p**rotein **i**dentification **r**esource |
| RNA | **r**ibo**n**ucleic **a**cid |
| SCOP | **s**tructural **c**lassification **o**f **p**roteins |
| SIMD | **s**ingle-**i**nstruction, **m**ultiple-**d**ata |
| SMP | **s**ymmetric **m**ulti**p**rocessing |
| SSE | **s**treaming SIMD **e**xtensions |
| TIGR | **T**he **I**nstitute for **G**enomic **R**esearch |
| VLSI | **v**ery **l**arge **s**cale **i**ntegration |
| WU | **W**ashington **U**niversity |

# Preface

This project was initiated in 1995 at the Biotechnology Centre of Oslo. The work was carried out at the Institute of Medical Microbiology at the National Hospital in the centre of Oslo, and finished at its new location at Gaustad.

First, I would like to thank my supervisor professor Erling Seeberg for introducing me to the exciting field of bioinformatics, and for his enthusiasm, encouragement and advice through the years. He also taught me some of the basics of genetics and molecular biology and provided excellent working conditions.

I would also like to thank my other co-authors, Luisa Luna, Ann Christin Eikså and Marit Otterlei, for getting some real biology out of my sequence alignments.

I am most grateful to all of my current and former colleagues in the Seeberg group and at the Institute of Medical Microbiology. It has been both great fun and scientifically stimulating to be among this helpful and social group of people. They have also taught me most of what I know about molecular biology, including some of their lab secrets.

I would also like to thank Arne Halaas and Tore Jahnsen for recruiting me into this field while I was still studying in Trondheim, and Rodrigo Lopez for interesting discussions.

Financial support was provided by The Research Council of Norway and The National Hospital.

Finally, I am grateful to my parents for their encouragement and to Rita for her patience.

Oslo, July 2000

Torbjørn Rognes

# Summary

The efforts by the international genome sequencing projects have resulted in huge and exponentially growing databases of public DNA and protein sequence information. The complete genome sequence of many organisms has already been published, and even the human genome passed the phase of sequencing as of writing.

However, a detailed analysis of these genomes, genes, and gene products is necessary  in order to reach a better understanding of their function in the cells of the organism. The major part of the analysis requires experimental biology and biochemistry, however, much information can be obtained by sequence analysis using computational methods.

Fundamental tasks in this analysis are the comparison of two sequences and the searching of databases of amino acid and nucleotide sequences for a similar sequence. This will often reveal valuable information about the possible structure and function of the protein. Several programs exist for performing such searches with varying sensitivity and speed. Accurate database searches may require large computational resources. As the databases are getting larger, longer time is required to search them. In addition, more sensitive tools are required in order to identify less obvious relationships between protein. The aim of this work was hence to develop novel algorithms for database searching with increased sensitivity and speed.

This work presents three new methods for performing both sensitive and rapid database searches. Two of the methods gain speed by taking advantage of 8-way parallel processing technology now available in common computers. By the use of some of these tools, a new family of proteins have also been identified.

# List of papers

**Paper I**
Rognes T. and Seeberg E.
*SALSA: improved protein database searching by a new algorithm for assembly of sequence fragments into gapped alignments.*
**Bioinformatics (1998) 14, 839-45.**

**Paper II**
Rognes T. and Seeberg E.
*Six-fold speed-up of Smith-Waterman sequence database searches with parallel processing on common microprocessors.*
**Bioinformatics (2000)**

**Paper III**
Rognes T.
*ParAlign: a parallel sequence alignment algorithm for rapid and sensitive database searches*
**(submitted for publication)**

**Paper IV**
Luna L., Rognes T., Eikså A.C., Otterlei M. and Seeberg E.
*Identification of a human member of a new family of DNA repair proteins with homology to* E. coli *Exonuclease III*
**(manuscript in prep.)**

# 1 Introduction

Computer tools for comparison of sequences and database searches are essential in the analysis and understanding of genetic sequences. Using these powerful methods, the evolutionary relationships between sequences can often be revealed and may give important clues to the structure and function of the molecules in the cells.

This chapter will start with a short and simplified introduction to some of the basics in molecular biology, followed by some information about sequence databases. Various methods for sequence comparison and database searching will then be presented, followed by a discussion on how these methods can be implemented on computer hardware with parallel processing capability. Finally, some results from comparisons of the different methods will be described.

Introduction to and reviews of methods for genetic sequence analysis are available in several papers (Altschul *et al.* 1994; Vingron and Waterman 1994; Argos 1994; Argos *et al.* 1991) and books (Sankoff and Kruskal 1983; Doolittle 1986, 1990, 1996; Waterman 1995; Gusfield 1997).

## 1.1 Molecular biology basics

With the intent that this thesis should be readable by a larger audience with background in either molecular biology or informatics, a short and simplified introduction to some of the basics in molecular biology is given below.

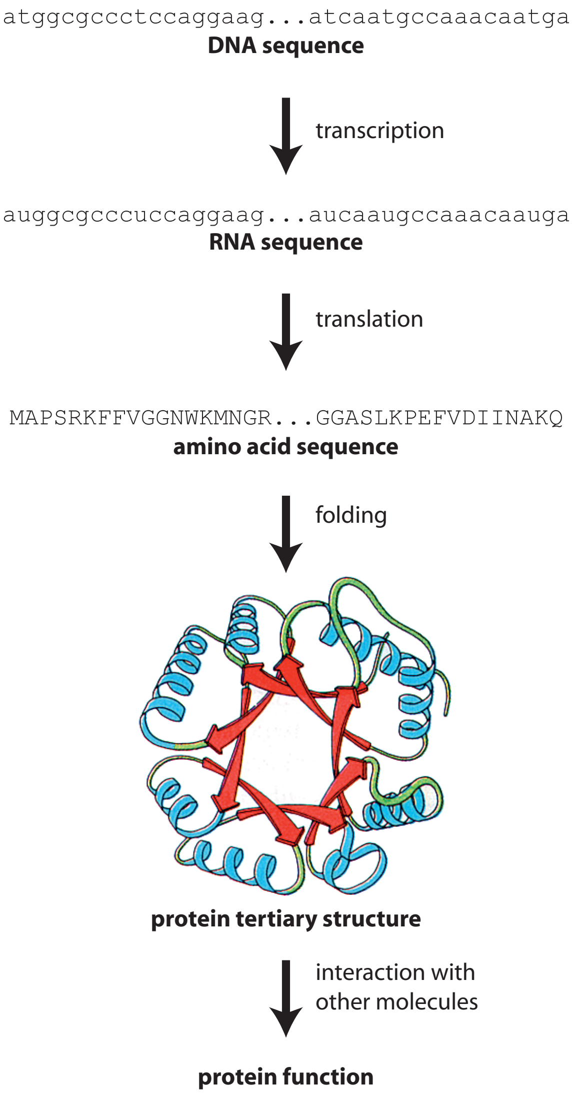
atggcgccctccaggaag...atcaatgccaaacaatga
DNA sequence
transcription
auggcgcccuccaggaag...aucaaugccaaacaauga
RNA sequence
translation
MAPSRKFFVGGNWKMNGR...GGASLKPEFVDIINAKQ
amino acid sequence
folding
protein tertiary structure
interaction with other molecules
protein function

**Figure 1: The flow of genetic information**
The genetic information is encoded in the DNA sequence and transfered to an RNA sequence during transcription and then into an amino acid sequence during translation. The amino acid sequence determines the folding of the protein into a specific structure, which in turn defines the protein function through interaction with other molecules in the cell.

### 1.1.1 DNA

DNA (deoxyribonucleic acid) is the primary medium for permanent storage of genetic information in a biological system, and is responsible for transfering genetic information from parent to progeny. DNA is a linear polymer made up of repeating units of deoxyribonucleotides. Each unit is composed of the sugar 2-deoxyribose, phosphate and a purine (C, T) or pyrimidine (A, G) base. The nucleotides are connected through phophodiester bonds between the phosphate and sugar of consecutive nucleotides. The genetic information is encoded in the sequence of the four possible bases in each nucleotide: adenine (A), cytosine (C), guanine (G) and thymine (T). See table 1 for an overview of the bases and their symbols, including some ambiguous symbols used. The nuclear DNA of complete organisms is found in the form of duplex DNA, which is a pair of two complementary antiparallel DNA strands arranged in a right-handed double helix with hydrogen bonds between complementary bases (base pairs) on opposite strands. The region of DNA that is the basis for a specific molecular cell product, e.g. a protein, including regulatory regions, is called a gene. The entire genetic information encoded in the DNA sequence of an organism is called a genome.

### 1.1.2 RNA

Using specific regions of the DNA as template, the RNA polymerase enzymes synthesise RNA (ribonucleic acid) molecules in a process called transcription. RNA is very similar to DNA, but

**Table 1: Nucleotide symbols**
The nucleotides and bases in DNA and RNA are usually represented by the symbols shown according to the IUPAC-IUBMB standard. Symbols for representing ambiguous positions are also shown. The two rightmost columns show possible encodings of the symbols for computer use.

| Nucleotide | Symbol | Code1 | Code2 |
|---|---|---|---|
| None (gap) | – | 0 | 0 |
| Adenine | A | 1 | 1 |
| Cytosine | C | 2 | 2 |
| Guanine | G | 3 | 4 |
| Thymine / Uracil [†] | T / U | 4 | 8 |
| A or C | M | 5 | 3 |
| A or G | R | 6 | 5 |
| A or T | W | 7 | 9 |
| C or G | S | 8 | 6 |
| C or T | Y | 9 | 10 |
| G or T | K | 10 | 12 |
| A, C or G | V | 11 | 7 |
| A, C or T | H | 12 | 11 |
| A, G or T | D | 13 | 13 |
| C, G or T | B | 14 | 14 |
| A, C, G or T | N | 15 | 15 |

[†] In RNA sequences, thymine is replaced by uracil.

**Table 2: Amino acid symbols**
The amino acids in proteins are usually represented by the three-letter or one-letter symbols shown, according to the IUPAC-IUBMB standard. Some ambigous and other special symbols are also included. A possible encoding for computer use is indicated in the rightmost column.

| Amino Acid | 3-Symbol | 1-Symbol | Code |
|---|---|---|---|
| None (gap) | ––– | – | 0 |
| Alanine | Ala | A | 1 |
| Arginine | Arg | R | 2 |
| Asparginine | Asn | N | 3 |
| Aspartic acid | Asp | D | 4 |
| Cysteine | Cys | C | 5 |
| Glutamine | Gln | Q | 6 |
| Glutamic acid | Glu | E | 7 |
| Glycine | Gly | G | 8 |
| Histidine | His | H | 9 |
| Isoleucine | Ile | I | 10 |
| Leucine | Leu | L | 11 |
| Lysine | Lys | K | 12 |
| Methionine | Met | M | 13 |
| Phenylalanine | Phe | F | 14 |
| Proline | Pro | P | 15 |
| Serine | Ser | S | 16 |
| Threonine | Thr | T | 17 |
| Tryptophan | Trp | W | 18 |
| Tyrosine | Tyr | Y | 19 |
| Valine | Val | V | 20 |
| Aspartic acid or Aspargine | Asx | B | 21 |
| Glutamic acid or Glutamine | Glx | Z | 22 |
| Undetermined amino acid | Xxx | X | 23 |
| Stop | End[†] | * | 24 |
| Selenocysteine | Sec | U | 25 |

[†] Stop is not an amino acid but the symbols and code represent the stop codon during translation.

**Table 3: The universal genetic code**
Codons of triplets from RNA sequences are translated into amino acid sequences by this almost universal code.

| 1st nt | 2nd nt | 3rd nt | | | |
|---|---|---|---|---|---|
| | | **U** | **C** | **A** | **G** |
| **U** | **U** | Phe | Phe | Leu | Leu |
| **U** | **C** | Ser | Ser | Ser | Ser |
| **U** | **A** | Tyr | Tyr | **Stp**[¶] | **Stp**[¶] |
| **U** | **G** | Cys | Cys | **Stp**[¶*] | Trp |
| **C** | **U** | Leu | Leu | Leu | Leu |
| **C** | **C** | Pro | Pro | Pro | Pro |
| **C** | **A** | His | His | Gln | Gln |
| **C** | **G** | Arg | Arg | Arg | Arg |
| **A** | **U** | Ile | Ile | Ile | **Met**[§] |
| **A** | **C** | Thr | Thr | Thr | Thr |
| **A** | **A** | Asn | Asn | Lys | Lys |
| **A** | **G** | Ser | Ser | Arg | Arg |
| **G** | **U** | Val | Val | Val | **Val**[§] |
| **G** | **C** | Ala | Ala | Ala | Ala |
| **G** | **A** | Asp | Asp | Glu | Glu |
| **G** | **G** | Gly | Gly | Gly | Gly |

[¶] Stp indicates the end of the protein coding sequence and is not a real amino acid.
[*] UGA may also code for selenocysteine (Sec).
[§] Both AUG and GUG may serve as initiation codons.

has the deoxyribose sugars replaced by ribose and the thymine (T) bases replaced by uracil (U), as shown in figure 1 and table 1. RNA is a temporary medium of genetic information, but may also have important enzymatic or other functions by itself.

### 1.1.3 Protein sequence

The blueprint of DNA termed mRNA (messenger RNA) is used as a template for protein synthesis. The ribosomes synthesise the proteins by translating codons consisting of three consecutive bases in the RNA into a sequence of amino acids as shown in figure 1. Table 2 shows the 20 different amino acids that can be encoded in DNA. Codons are translated into amino acids by the almost universal genetic code shown in table 3.

The 20 encoded amino acids have different chemical and structural properties, which may be important for the structure and function of the protein. However, some of the amino acids have quite similar properties, like e.g. the small hydrophobic amino acids leucine, isoleucine and valine.

The amino acids are linked by peptide bonds. The simple linear sequence of amino acids in a protein is called the primary structure of a protein.

### 1.1.4 Protein structure

As the protein is synthesised it folds into a complete three-dimensional structure called the ter-

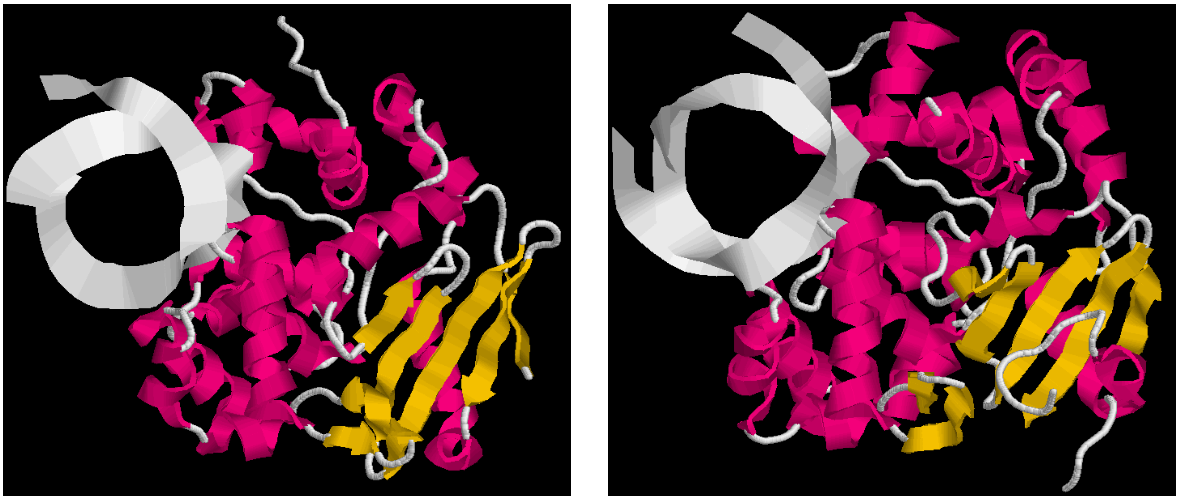

**Figure 2: Two proteins with very similar three-dimensional structure, yet limited sequence similarity**
The structures of two repair proteins in the helix-hairpin-helix superfamily, both complexed with DNA (gray ribbons), are shown.
Left: *E. coli* AlkA (3-methyladenine repair glycosylase) (Hollis *et al.* 2000; PDB: 1DIZ).
Right: Human hOGG1 (8-oxoguanine repair glycosylase) (Bruner *et al.* 2000; PDB: 1EBM).
Red ribbons: α-helices. Yellow arrows: β-sheets. Created with Rasmol (Sayle 1992, Bernstein 1998).

```
E.c. AlkA     72 VKTYIKTIGLYNSKAENIIKTCRILLEQHNG----------EVPEDRAALEALPGVGRKT    121
                 |+ +++ +||   +|  +  + | +||+  |             |   ||  ||||| |
H.s. hOGH1   192 VEAHLRKLGL-GYRARYVSASARAILEEQGGLAWLQQLRESSYEEAHKALCILPGVGTKV    250

E.c. AlkA    122 ANVVLNTAFGWP-TIAVDTHIFRVCNR                                      147
                 |+ +   |   |  + || |++ +  |
H.s. hOGH1   251 ADCICLMALDKPQAVPVDVHMWHIAQR                                      277
```

**Figure 3: Sequence alignment of the proteins above**
The *E. coli* AlkA protein (SWISS-PROT acc.no. P04395; Evensen and Seeberg 1982; Nakabeppu *et al.* 1984) and the human hOGG1 protein (EMBL acc.no. Y11838; Bjoras *et al.* 1997) are optimally aligned using the BLOSUM62 amino acid substitution matrix and affine gap penalties (10, 1). Identical amino acids are indicated by a bar (|), while similar amino acids are indicated by a plus sign (+). Gaps in the sequences are indicated by a dash (–).

tiary structure as indicated in figure 1. Short stretches of amino acids in a protein may form simple and specific secondary structure elements, i.e. α-helices or β-sheets. The protein structure is almost completely determined from the primary sequence of the protein (Anfinsen 1973). In general, a given sequence folds into one specific structure, however, in some cases, e.g. with prions, stable alternate structures are also possible. Occasionally, just a single amino acid change at an important position in the protein, may have a substantial effect on the structure and function of the protein. Other changes may have no effect. The structure of a protein is usually stable, but it may be slightly changed by the interaction with other molecules. It is possible to predict the structure of small peptides from their amino acid sequence by computational methods with reasonable accuracy, however, prediction of the structure of normally sized proteins is still exceedingly difficult (Moult 1999). The protein folding problem is perhaps the most challenging problem within bioinformatics today.

### 1.1.5 Protein function

The function of a protein is determined by the structure and its interactions with other molecules in the cell. Some proteins have catalytic activity and are called enzymes. Other proteins form structural elements in the cells or act as signalling molecules.

### 1.1.6 Evolution

Even though DNA is a relatively stable molecule, it may be damaged by normal cellular metabolism or by environmental agents like radiation and various chemicals (Lindahl and Wood 1999). Replication errors and recombination may also lead to changes in the DNA sequence. Unless being repaired, DNA damage may lead to mutations, which are permanent changes of the DNA sequence. Mutations take the form of base substitutions, deletions or insertions. Due to the redundancy of the genetic code, some mutations in DNA that encode a protein sequence do not lead to changes in the amino acid sequence. Such mutations are called silent mutations. However, a mutant protein will result if the protein sequence is changed.

Due to mutations and recombination, mutant protein forms that have superior properties relative to the original protein may appear. However, in most cases the mutant protein will not be useful for the organism and may have a detrimental effect on the cellular metabolism. The best variants will be selected during evolution. In the course of time many mutations may appear and the protein and DNA sequences will gradually diverge from the originals. The structure and function of the mutant and original protein may still be quite similar.

Later, the mutant and original DNA sequences will have diverged so much that there is no obvious sequence similarity left, however it might still be possible to see sequence similarities between the mutant and original protein sequences, and the protein structures and functions may be quite similar (figures 2 and 3).

After even more mutations, no significant sequence similarity between the mutant and original protein is detectable, even though both structure and function might still be similar.

It is possible that similar protein structures or functions have appeared independently. In this case, the DNA and protein sequences of the proteins are usually completely different.

Two related proteins are said to be ous if they have evolved from a common ancestor. If two protein sequences are similar, the proteins are usually homologous, however absence of sequence similarity does not mean that the proteins are nonhomologous.

### 1.1.7 Use of sequence alignments and database searches

Since the structure and function of a protein is predominantly determined by the amino acid sequence of the protein, there are many important practical uses of the results of sequence alignments and database searches.

*Identification of functionally important residues*

A pairwise alignment of two proteins (or preferably a multiple sequence alignment of many proteins) will indicate which residues are identical, which are conserved and which are not conserved between the sequences. This alignment may indicate the position of important functionally active residues, because important residues often are unchanged or conserved. Conservation of residues in these sites will usually indicate that the functional aspects associated with these sites are conserved between the proteins.

*Prediction of function from sequence*

If, for instance, a bacterial protein has been well characterised and a mammalian homologue is identified in the sequence databases, the mammalian protein might have a similar function as the bacterial protein. This concept has been extensively used, and has often lead to a rapid identification of many important human genes.

*Homology modelling of protein structure*

It is generally impossible to predict the structure of entire proteins computationally ab initio from the sequence alone, but other approaches have been more successful. If a homologous protein with known tertiary structure exists, it may be used as a model. A partial or entire protein may be modelled on the basis of another protein. It has been estimated that the number of essentially different protein structures is limited to about one thousand (Chotia 1992). As the number of proteins with known structure increases, it will be increasingly easier to find another protein with similar sequence and known structure.

*Multiple alignment and phylogenetic analysis*

A pairwise alignment between all pairs of sequences is often an initial step in methods for multiple sequence alignment and phylogenetic analysis. Of particular importance is the sequence alignment score which may be used as a measure of sequence divergence or phylogenetic distance.

*Prediction of protein coding regions in DNA*

If a given DNA sequence is translated in all six possible reading frames and the resulting amino acid sequences are used as query sequences in a search through protein databases, significant matches may indicate that the region in the given reading frame codes for a protein (Gish and States 1993).

## 1.2 Sequence databases

Genetic sequence information is collected in a plethora of databases with interconnections. Some basic information about the largest and most important databases with nucleotide and protein sequence information will be presented.

### 1.2.1 Nucleotide sequence databases

When the nucleotide sequence of a gene or an entire genome has been determined it is often deposited in a public sequence database. Many journals require that sequences are submitted to these databases before manuscripts will be accepted for publication.

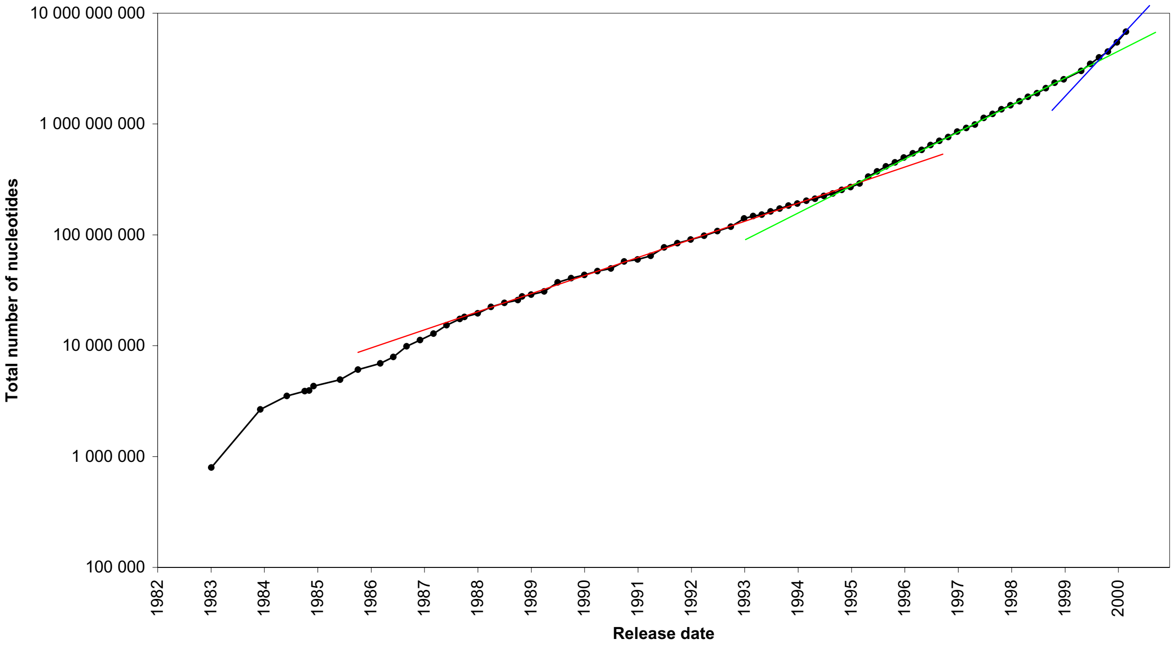

**Figure 4: The growth of the GenBank nucleotide database**
The total number of nucleotides in the database is plotted on a logarithmic scale versus the release date. The trend lines indicate the average growth rate in three different periods, where the red line corresponds to an approximate doubling time of 22 months, the green line to 15 months and the blue line to 7 months.

Nucleotide sequences are sent to one of the three international public nucleotide databases: GenBank, EMBL and DDBJ. GenBank is handled by the National Centre for Biotechnology Information (NCBI) in the United States (Benson *et al.* 2000). The EMBL nucleotide sequence database (Baker *et al.* 2000) is handled by the European Bioinfomatics Institute (EBI), an outstation of the European Molecular Biology Laboratories (EMBL). The DDBJ database (Tateno *et al.* 2000) is handled by the DNA Data Bank of Japan (DDBJ). These three institutions collaborate and exchange information with each other, making the contents of the three databases essentially identical. The information in these databases is continously updated and is available free over the Internet.

The sequence entries are placed in different divisions of the nucleotide databases, which are distributed as several files. Below is a description of the divisions in GenBank. The other two databases have similar divisions.

The main nine divisions are based on the classification of the source organism of the sequences: primate (PRI), rodent (ROD), other mammalian (MAM), other vertebrates (VRT), invertebrates (INV), plant including fungi and algae (PLN), bacterial (BCT), viral (VRL), and phage (PHG). Sequences in these sections are usually of high quality.

Other divisions include the EST division, which contains sequences from Expressed Sequence Tags (ESTs). These are short (usually 300-500bp) single reads of cDNA sequences generated from expressed mRNA sequences from various cells or tissues, most of which are of human and mouse origin. These sequences are a valuable resource in the identification of novel mammalian homologues of proteins previously characterised in other organisms.

The HTG division contains High-Througput Genomic Sequences which are more or less unfinished genomic sequences , mainly of human origin, from various high-throughput sequencing centers. When the sequences are finished they are moved to another relevant division.

The STS division contains Sequence Tagged Sites (STSs), which are short (usually 200-500bp) sequences used as landmarks in a genome. These sequences are unique within their genome and are hence useful in the physical mapping of genes.

The GSS section contains Genome Survey Sequences (GSS), which are similar to EST sequences, but are from genomic DNA, and not mRNA.

In addition, there are divisions for patent (PAT), synthetic (SYN) and unannotated (UNA) sequences.

By July 2000, the public nucleotide databases contain more than 8 000 million nucleotides divided into over 6 millon sequence entries. The databases are growing exponentially, or even faster. Figure 4 shows a graph of the growth of Genbank from December 1982 to February 2000. There has been a few noticable shifts in the growth rate, as indicated by the trend lines in the figure. The size of GenBank was doubling approximately every 22 months in the period from August 1987 to February 1995, and every 15 months in the period from April 1995 to August 1999. However, in the period from October 1999 to February 2000, the growth represented a doubling time of less than 7 months. The first shift probably reflects the publication of several complete bacterial genomes and huge amounts of EST sequences. The most recent shift in growth rate probably reflects the huge amounts of data from the *D.melanogaster* and the human genome sequencing projects.

In addition to the public sequence databases, several commercial and confidential databases exist, which are used by companies in the pharmaceutical and genomics industry.

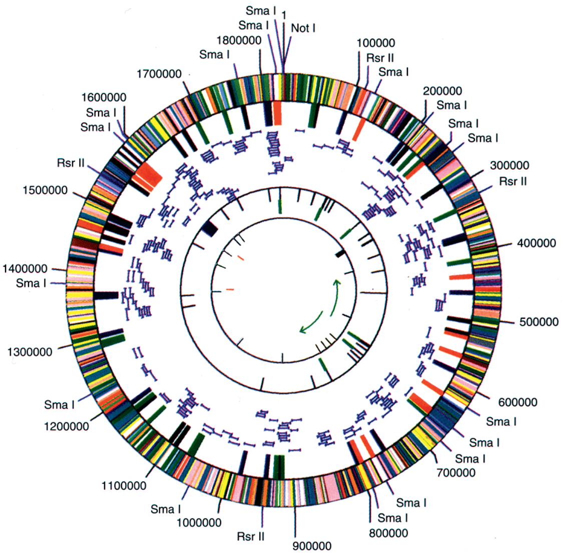

**Figure 5: The circular genome of *Haemophilus influenzae* Rd, the first bacteria to be completely sequenced**
From Fleischmann *et al.* (1995).

### 1.2.2 Genome sequencing projects

Since the publication of the entire genome of the bacteria *Haemophilus influenzae* Rd by Fleischmann *et al.* in 1995 (figure 5), more than 31 other genomes from archaea, bacteria and even eukaryotes have been completely sequenced and published, including the large genomes of baker's yeast (*Saccharomyces cerevisiae*), a roundworm (*Caenorhabditis elegans*), and the fruit fly (*Drosophila melanogaster*). An overview of all published complete genomes of archaea, eubacteria and eukaryotes with references appears in table 4. Many smaller complete genome sequences of plasmids, phage, viruses and eukaryotic organelles have been published previously.

The Human Genome Project (HGP) lead by the Human Genome Organization (HUGO) intends to finish the entire human genome of about 3Gbp in 2003 (Collins *et al.* 1998), and the complete sequence of human chromosome 21 and 22 has already been published (Hattori *et al.* 2000; Dunham *et al.* 1999).

On 26 June 2000, the completion of the initial sequencing of the human genome was announced jointly by HUGO and Celera Genomics. The completed sequence, along with initial annotation will be published later in 2000.

Many other genome sequencing projects are also underway, both in public and private laboratories.

### 1.2.3 Protein databases

Potential protein coding regions in the DNA sequences are translated and collected in protein databases also including proteins indentified by other methods. Some of the proteins are characterised experimentally in great detail, others are given putative functions based on sequence similarity to other proteins, while some are not annotated at all.

SWISS-PROT (Bairoch and Apweiler, 2000) is a protein database containing a subset of these proteins with high quality annotations. SWISS-PROT is curated by the Swiss Institute of Bioinformatics (SIB) and the EBI. Release 38 from July 1999 contained 80 000 sequences with a total of 28 085 965 amino acid residues.

PIR-International (Protein Identification Resource) (Barker *et al.* 2000) is another curated protein database. Release 64 from 31 March 2000 contained a total of 61 827 019 amino acid residues divided into 177 674 sequence entries.

PDB (Protein Data Bank) (Berman *et al.* 2000) is a database of three-dimensional molecular structure information, mainly of proteins, but also of nucleic acids and carbohydrates. The structures have primarily been determined by X-ray crystallography and NMR (Nuclear Magnetic Resonance). By 4 April 2000, PDB contained the structures of 10 703 proteins and 1 369 other molecules.

### 1.2.4 File formats

All sequence databases are distributed in the form of flat ascii text files. Each record in the database contains the sequence and additional information, including source organism, accession numbers, gene or protein name, potential protein coding regions, transcribed regions, regulatory regions, sequence motifs and more. A sample entry from the EMBL database is shown in figure 6. However, the shorter FASTA format where only one line of information in addition to the sequence is included, is often used when a

**Table 4: Overview of completed genomes of archaea, eubacteria and eukaryotes**
Information about the completely sequenced genomes of 6 archaea (A), 23 eubacteria (B) and 3 eukaryotes (E) is summarised. Based on data from The Institute of Genomic Research (TIGR), EBI and NCBI.

| No | Organism | Genome Size | Acccession | Group | Reference |
|---|---|---|---|---|---|
| 1 | *Haemophilus influenzae* Rd, KW20 | 1.83 Mbp | L42023 | B | Fleischmann *et al.* (1995) |
| 2 | *Mycoplasma genitalium*, G-37 | 0.58 Mbp | L43967 | B | Fraser *et al.* (1995) |
| 3 | *Methanococcus jannaschii*, DSM 2661 | 1.66 Mbp | L77117 | A | Bult *et al.* (1996) |
| 4 | *Synechocystis sp.*, PCC 6803 | 3.57 Mbp | AB001339 | B | Kaneko *et al.* (1996) |
| 5 | *Mycoplasma pneumoniae*, M129 | 0.81 Mbp | U00089 | B | Himmelreich *et al.* (1996) |
| 6 | *Saccharomyces cerevisiae*, S288C | 13 Mbp |  | E | Goffeau *et al.* (1996) |
| 7 | *Helicobacter pylori*, 26695 | 1.66 Mbp | AE000511 | B | Tomb *et al.* (1997) |
| 8 | *Escherichia coli*, K-12 | 4.60 Mbp | U00096 | B | Blattner *et al.* (1997) |
| 9 | *Methanobacterium thermoautotrophicum*, delta H | 1.75 Mbp | AE000666 | A | Smith *et al.* (1997) |
| 10 | *Bacillus subtilis*, 168 | 4.20 Mbp | AL009126 | B | Kunst *et al.* (1997) |
| 11 | *Archaeoglobus fulgidus*, DSM4304 | 2.18 Mbp | AE000782 | A | Klenk *et al.* (1997) |
| 12 | *Borrelia burgdorferi*, B31 | 1.44 Mbp | AE000783 | B | Fraser *et al.* (1997)<br>and Casjens *et al.* (2000) |
| 13 | *Aquifex aeolicus*, VF5 | 1.50 Mbp | AE000657 | B | Deckert *et al.* (1998) |
| 14 | *Pyrococcus horikoshii*, OT3 | 1.80 Mbp | BA000001 | A | Kawarabayasi *et al.* (1998) |
| 15 | *Mycobacterium tuberculosis*, H37Rv | 4.40 Mbp | AL123456 | B | Cole *et al.* (1998) |
| 16 | *Treponema pallidum*, Nichols | 1.14 Mbp | AE000520 | B | Fraser *et al.* (1998) |
| 17 | *Chlamydia trachomatis*, serovar D (D/UW-3/Cx) | 1.05 Mbp | AE001273 | B | Stephens *et al.* (1998) |
| 18 | *Rickettsia prowazekii*, Madrid E | 1.10 Mbp | AJ235269 | B | Andersson *et al.* (1998) |
| 19 | *Caenorhabditis elegans* * | 97 Mbp |  | E | The *C. elegans* Sequencing Consortium (1998) |
| 20 | *Helicobacter pylori*, J99 | 1.64 Mbp | AE001439 | B | Alm *et al.* (1999) |
| 21 | *Chlamydia pneumoniae*, CWL029 | 1.23 Mbp | AE001363 | B | Kalman *et al.* (1999) |
| 22 | *Aeropyrum pernix*, K1 | 1.67 Mbp | BA000002 | A | Kawarabayasi *et al.* (1999) |
| 23 | *Thermotoga maritima*, MSB8 | 1.80 Mbp | AE000512 | B | Nelson *et al.* (1999) |
| 24 | *Deinococcus radiodurans*, R1 | 3.28 Mbp | AE000513<br>AE001825 | B | White *et al.* (1999) |
| 25 | *Campylobacter jejuni*, NCTC 11168 | 1.64 Mbp | AL111168 | B | Parkhill *et al.* (2000a) |
| 26 | *Neisseria meningitidis*, MC58, serogroup B | 2.27 Mbp | AE002098 | B | Tettelin *et al.* (2000) |
| 27 | *Chlamydia muridarum*, MoPn | 1.07 Mbp | AE002160 | B | Read *et al.* (2000) |
| 28 | *Chlamydophila pneumoniae*, AR39 | 1.23 Mbp | AE002161 | B | Read *et al.* (2000) |
| 29 | *Drosophila melanogaster* * | 180 Mbp |  | E | Adams *et al.* (2000) |
| 30 | *Neisseria meningitidis*, Z2491, serogroup A | 2.18 Mbp | AL157959 | B | Parkhill *et al.* (2000b) |
| 31 | *Pyrococcus abyssi* | 1.77 Mbp | AL096836 | A | Heilig *et al.* (unpublished) |
| 32 | *Ureaplasma urealyticum*, serovar 3 | 0.75 Mbp | AF222894 | B | Glass *et al.* (unpublished) |

* At the time of publication, these genome sequences still contained a few gaps, but were otherwise essentially complete.

more compact form is required.

In order to avoid wasting unnecessary time on disk reading and to perform effective database searches, the database text files should be parsed and stored in a more efficient format prior to searching. Additionally, nucleotide data contain special codes in cases where the sequencing has not been able to identify unambigously the correct nucleotide in a given position. A total of 15 different symbols representing all possibilities is used according to the IUPAC-IUBMB standard, as shown in table 1.

Usually, nucleotide data is compressed to 2 bits per nucleotide by randomly replacing the ambigous symbols by one of the possible symbols. Four nucleotide positions can hence be represented in a byte. This means that the entire 8Gbp of nucleotide data available at present can be stored in about 2GB of memory. The information about the ambigous positions can be stored separately, if necessary.

NCBI has defined two standard database file formats that are used by BLAST version 1 and 2 (Altschul *et al.* 1990; 1997), respectively. In this format, all sequence data is stored in one file, while additional information about each sequence, e.g. organism, accession numbers, gene or protein name etc, is stored in a second file. A third file contains indicies into the first two files with the positions of each sequence and additional information. Summary information (size, name, date etc) about the database is also stored in the third file.

### 1.3 Protein sequence comparison and database searches

To assess the amount of similarity or differences between sequences, various sequence comparison methods are employed. The amount of similarity is expressed by a score or a statistical parameter. In addition, a form of visualisation of the similarity is often given.

The degree of similarity can be defined in different ways, however it is usually based on an alignment of the two sequences. Alternative measures of protein similarity can be based on amino acid composition or oligopeptide frequency (Solovyev and Makarova 1993) or in

```
ID   HSA011311  standard; RNA; HUM; 1956 BP.
XX
AC   AJ011311;
XX
SV   AJ011311.1
XX
DT   15-JUN-1999 (Rel. 60, Created)
DT   15-JUN-1999 (Rel. 60, Last updated, Version 1)
XX
DE   Homo sapiens mRNA for AP endonuclease XTH2, putative
XX
KW   AP endonuclease XTH2; XTH2 gene.
XX
OS   Homo sapiens (human)
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Teleostomi;
OC   Euteleostomi; Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
XX
RN   [1]
RP   1-1956
RA   Rognes T.;
RT   ;
RL   Submitted (16-SEP-1998) to the EMBL/GenBank/DDBJ databases.
RL   Rognes T., Institute of Medical Microbiology, University of Oslo, The
RL   National Hospital, N-0027, NORWAY.
XX
RN   [2]
RA   Luna L., Rognes T., Henriksen A.C., Bjoras M., Seeberg E.;
RT   ”Putative human AP endonuclease XTH2”;
RL   Unpublished.
XX
CC   Related sequences: AL020991, Z83821, AF068624, AA554484, C01178,
CC   N59497,
CC   AL020991, N59092, N59517
XX
FH   Key             Location/Qualifiers
FH
FT   source          1..1956
FT                   /chromosome=”X”
FT                   /db_xref=”taxon:9606”
FT                   /organism=”Homo sapiens”
FT                   /tissue_type=”lung tumor”
FT                   /map=”Xp11.21”
FT                   /clone=”IMAGE:978889”
FT   5’UTR           <1..66
FT                   /gene=”XTH2”
FT   CDS             67..1623
FT                   /gene=”XTH2”
FT                   /product=”AP endonuclease XTH2, putative”
FT                   /protein_id=”CAB45242.1”
FT                   /translation=”MLRVVSWNINGIRRPLQGVANQEPSNCAAVAVGRILDELDADIVC
FT                   LQETKVTRDALTEPLAIVEGYNSYFSFSRNRSGYSGVATFCKDNATPVAAEEGLSGLFA
FT                   TQNGDVGCYGNMDEFTQEELRALDSEGRALLTQHKIRTWEGKEKTLTLINVYCPHADPG
FT                   RPERLVFKMRFYRLLQIRAEALLAAGSHVIILGDLNTAHRPIDHWDAVNLECFEEDPGR
FT                   KWMDSLLSNLGCQSASHVGPFIDSYRCFQPKQEGAFTCWSAVTGARHLNYGSRLDYVLG
FT                   DRTLVIDTFQASFLLPEVMGSDHCPVGAVLSVSSVPAKQCPPLCTRFLPEFAGTQLKIL
FT                   RFLVPLEQSPVLEQSTLQHNNQTRVQTCQNKAQVRSTRPQPSQVGSSRGQKNLKSYFQP
FT                   SPSCPQASPDIELPSLPLMSALMTPKTPEEKAVAKVVKGQAKTSEAKDEKELRTSFWKS
FT                   VLAGPLRTPLCGGHREPCVMRTVKKPGPNLGRRFYMCARPRGPPTDPSSRCNFFLWSRP
FT                   S”
FT   3’UTR           1624..1956
FT                   /gene=”XTH2”
FT   polyA_signal    1936..1941
FT                   /gene=”XTH2”
XX
SQ   Sequence 1956 BP; 409 A; 582 C; 522 G; 443 T; 0 other;
     ctgaacagga agcagttcgc tcgcgcctag gttggcgcgg gctgggaggt gttccagccc        60
     tttaagatgt tgcgcgtggt gagctggaac atcaatggga ttcggagacc cctgcaaggg       120
     gtggcaaatc aggaacccag caactgtgcc gccgtggccg tggggcgcat tttggacgag       180
     ctggatgcgg atatcgtctg tctccaggaa accaaagtga ccagggatgc actgacagag       240
     cccctggcta tcgttgaggg ttataactcc tatttcagct tcagccgcaa ccgtagcggc       300
     tattctggtg tagccacctt ctgtaaggac aatgctaccc cagtggctgc tgaagaaggc       360
     ctgagtggcc tgtttgccac ccagaatggg gatgttggtt gctatggaaa catggatgag       420
     tttacccaag aggaactccg ggctctggat agtgagggca gggccctcct cacacagcat       480
     aagatccgca catgggaagg taaggagaag accttgaccc taatcaacgt gtactgcccc       540
     catgcggacc ctgggaggcc tgagcggcta gtctttaaga tgcgcttcta tcgtttgctg       600
     caaatccgag cagaagccct cctggcggca ggcagccatg tgatcattct gggtgacctg       660
     aatacagccc accgccccat tgaccactgg gatgcagtca acctggaatg ctttgaagag       720
     gacccagggc gcaagtggat ggacagcttg ctcagtaact tggggtgcca gtctgcctct       780
     catgtagggc ccttcatcga tagctaccgc tgcttccaac caaagcagga gggggccttc       840
     acctgctggt cagcagtcac tggcgcccgc catctcaact atggctcccg gcttgactat       900
     gtgctggggg acaggaccct ggtcatagac acctttcagg cctctttcct gctgcctgag       960
     gtgatgggct ctgaccactg ccctgtgggt gcagtcttga gtgtgtcctc tgtgcctgca      1020
     aaacagtgcc cacctctgtg cacccgcttc ctccctgagt ttgcaggcac ccagctcaag      1080
     atccttcgct tcctagttcc tctcgaacaa agtcctgtgt tggagcagtc gacgctgcag      1140
     cacaacaatc aaacccgggt acagacatgc caaaacaaag cccaagtgcg ctcaaccagg      1200
     cctcagccca gtcaggttgg ctctagcaga ggccagaaaa acctgaagag ctactttcag      1260
     ccctccccta gctgtcccca agcctctcct gacatagagc tgcctagcct accactgatg      1320
     agcgccctca tgaccccgaa gactccagaa gagaaggcag tggccaaagt ggtgaagggg      1380
     caggccaaga cttcagaagc caaagatgag aaggagttac ggacctcatt ctggaagtct      1440
     gtgctggcgg ggcccttgcg cacacccctc tgtgggggcc acagggagcc atgtgtgatg      1500
     cgtactgtga agaagccagg acccaacttg ggccgccgct tctacatgtg tgccaggccc      1560
     cggggtcctc ccactgaccc ctcctcccgg tgcaacttct tcctctggag caggcccagc      1620
     tgaaccaatg gaggcctggg gacatctggc atggtcaccc ctgcacatga tctgaggcca      1680
     gctccccttc cctgagctgc ctcctgcttc tccctcaaag tctcctaccc ttctcttcct      1740
     cttttaagcc ctctcttcct cgctttcctt cctacctagc tccttgttgg tgagcttctt      1800
     gtgccttaat cctgtgaccc agccccttac accactttcc accttcctgt ccgaagtaca      1860
     cggacactag ctgccccagg aagttgtgtg attttaaatc acttctgtct ttgctggaaa      1920
     gtgtatttgt gcataaataa agtctgtgta tttgtt                                1956
//
```

**Figure 6: Sample EMBL sequence database entry**

The sequence shown is that of the novel human hXTH gene described in paper IV.

molecular weight and isoelectric point (Hobohm and Sander 1995).

One of the first ways to compare sequences was using a dot matrix plot (Gibbs and McIntyre 1970; Maizel and Lenk 1981). However, they required visual inspection to evaluate the amount of similarity. Sequence alignments as shown in figure 3 was quickly adopted and gave an alignment score that could be evaluated computationally.

Pairwise sequence alignment involves only two sequences, while multiple sequence alignment may be considered as a generalisation of a pairwise alignment to three or more sequences. However, a number of additional factors complicate multiple alignments, both theoretically and computationally. Multiple alignments are useful for examining the evolutionary relationships within a group of sequences, and are often constructed in connection with phylogenetic trees. Many multiple sequence methods involve the initial computation of the alignments of all pairs. Different programs for multiple sequence alignment have recently been reviewed (Thompson *et al.* 1999).

Sequence database searches can be performed to find the sequences most similar to a given query sequence. The query sequence is then compared to every sequence in the database and a similarity score is computed which is used to rank the database sequences. Some widely used database search programs are listed in table 5.

The most accurate sequence alignment algorithms are based on brute-force dynamic programming, which is very time consuming in general. Hence, heuristic alignment algorithms are often used in database searches, where the query sequence shall be aligned to thousands or millons of database sequences within reasonable computation time.

This thesis deals primarily with pairwise protein sequence alignments and how they can be efficiently implemented in database searches.

### 1.3.1 Dynamic programming alignment algorithms

Sequence alignment is a form of sequence comparison where the differences between two sequences are expressed by three basic operations, namely replacement, deletion or insertion of symbols. Pairwise alignments can be visualised as in figure 3. When sequence alignments are visualised, deletions and insertions are represented by gaps (‘–‘) in the sequences. Based on the alignment, a measure of sequence similarity or difference can be calculated and expressed as a score or cost. An excellent review of methods for sequence alignment based on dynamic programming has been published by Pearson and Miller (1990).

**Table 5: The most widely used sequence similarity database search programs**

The program name, latest available version and literature reference for the programs are shown. There are major differences between version 1 and 2 of NCBI BLAST, and the versions are hence listed seperately.

| Program | Version | Reference |
|---|---|---|
| FASTA | 3.3.t05 | Pearson and Lipman (1988) † |
| SSEARCH | 3.3.t05 | Pearson (1991) * |
| NCBI BLAST | 1.4.9 | Altschul *et al.* (1990) |
| NCBI BLAST | 2.0.11 | Altschul *et al.* (1997) |
| WU-BLAST | 2.0a19 | Gish (1996) |

† FASTA runs in two major modes, with the ktup parameters set to 1 or 2.

* SSEARCH is based on the algorithm of Smith and Waterman (1981) and Gotoh (1982).

As a basis for the following discussion of sequence alignments, consider a query sequence **A** of length $m$ with symbols $a_1$, $a_2$, ..., $a_i$, ..., $a_{m-1}$, $a_m$ and a database sequence **B** of length $n$ with symbols $b_1$, $b_2$, ..., $b_j$, ..., $b_{n-1}$, $b_n$. A so-called edit distance is associated with a sequence alignment. The edit distance is computed by a process where the first sequence (A) is transformed into the second sequence (B) in a series of steps, where each step is one of three simple operations. The three operations are (1) replacement of one symbol by another, (2) deletion of one or more symbols from sequence A, or (3) insertion of one or more score symbols from sequence B. There is a certain cost associated with each of these operations, and the total cost is calculated by adding these costs together. When alignments are used to look at similarities, instead of differences, equal symbols in the two sequences are given a high score. The optimal alignment of two sequences is the alignment that has the lowest cost or highest score.

When the entire lengths of the sequences are aligned, it is called a *global sequence alignment*. When only a subsequence from each of the sequences is aligned, it is called a *local sequence alignment*. Only the most similar subsequence of the sequences is included in the optimal local alignment.

When comparing two quite similar sequences in their entire length and looking for the differences between them, a global alignment is computed and an alignment cost or distance is calculated. However, when comparing two rather different sequences and focusing on their similarities, a local alignment and a similarity score is computed. The latter is usually the most approriate for database searches.

For each of the possible amino acid replacements, an amino acid substitution score is retrieved from a matrix. A positive score is assigned to similar amino acids, and a negative score is assigned to dissimilar amino acids. These amino acid substituition matrices are discussed in detail in section 1.3.3.

Deletion of symbols from the first sequence or insertion of symbols from the second sequence is usually represented by gaps in the opposite sequence. Such gaps are introduced at the expense of a gap penalty, usually dependent on the length of the gap. Various gap penalty schemes are described in detail in section 1.3.4.

In some cases, as in database searches, the alignment score is of primary interest. In other cases the explicit alignment is needed. It is computationally easier to compute just the score. There are different methods for these two tasks, but they have much in common.

*Global sequence alignment*

The first dynamic programming algorithm for global sequence alignment was described by Needleman and Wunsch (1970). It was a method for maximising the amount of similarity between two sequences. Sellers (1974) described another global alignment method which minimised the differences between the sequences and computed a distance measure. This method was generalised for gaps of any size by Waterman *et al.* (1976). There is a duality between these two methods to approach essentially the same problem. Smith *et al.* (1981) proved that these two methods were equivalent with appropriate substitution scores and gap penalties.

Goad and Kanehisa (1982) also made some refinements to the Needleman-Wunsch algorithm.

*Local sequence alignment*

When looking for similarities between subsequences of two sequences, as is usually the goal in the methods used to find homologues by database searches, a local alignment method is more appropriate than a global. The simple dynamic programming algorithm described by Smith and Waterman (1981) is the basis for this type of alignments. This algorithm can be used both to compute the optimal alignment score and for creating the actual alignment. It uses memory space proportional to the product of the lengths of the two sequences, *mn*, and computing time proportional to *mn*(*m*+*n*). The recursion relations used in the original Smith-Waterman algorithm are the following:

$$e_{i,j} = \max_{0<k<i} \{ h_{i-k,j} - g(k) \}$$

$$f_{i,j} = \max_{0<l<j} \{ h_{i,j-l} - g(l) \}$$

$$h_{i,j} = \max \{ h_{i-1,j-1} + \mathbf{Z}[a_i , b_j] , e_{i,j} , f_{i,j} , 0 \}$$

Here, $h_{i,j}$ is the score of the optimal alignment ending at position (*i*,*j*) in the matrix, while $e_{i,j}$ and $f_{i,j}$ are the scores of optimal alignments that ends at the same position but with a gap in sequence **A** or **B**, respectively. **Z** is the amino acid substitution score matrix, while *g*(*k*) is the gap penalty function. The computations should be started with $e_{i,j} = f_{i,j} = h_{i,j} = 0$ for all $i = 0$ or $j = 0$, and proceeded with *i* going from 1 to *m* and *j* going from 1 to *n*. The order of computation is strict, because the value of *h* in any cell in the alignment matrix cannot be computed before all cells to the left or above it has been computed.The overall optimal alignment score is equal to the maximum value of $h_{i,j}$.

Gotoh (1982) reduced the time needed by the algorithm to be proportional to *mn* when affine gap penalties of the form *g*(*k*)=*q*+*kr* are used, where *q* is the gap opening penalty and *r* is the gap extension penalty. When only the actual optimal local alignment score is required, the space requirements were reduced to be proportional to the smallest of *m* and *n*. The new recursion relations for $e_{i,j}$ and $f_{i,j}$ are as follows:

$$e_{i,j} = \max \{ e_{i,j-1} , h_{i-1,j} - q \} - r$$

$$f_{i,j} = \max \{ f_{i-1,j} , h_{i,j-1} - q \} - r$$

This local sequence algorithm has been implemented in the SSEARCH program (Pearson 1991).

In the SWAT program (Green 1993), further improvements to the algorithm were introduced with an improvement in computing time of a factor of about two (dependent on the computer architecture). These so-called SWAT-optimisations are based on the fact that most of the values of *e* and *f* in the matrix are zero and hence do not contribute to *h*. Only when *h* is larger than the penalty of a single symbol gap (*q*+*r*) will *e* and *f* get a positive value. Along a row or column, *e* and *f* will remain zero until such a large value of *h* is encountered. This observation saves a lot of computation and has also been incorporated in the most recent versions of SSEARCH.

*Alignment in linear space*

When the explicit alignment is required and not just the optimal score, the methods of Smith and Waterman (1981) and Gotoh (1982) still require space proportional to *mn*(*m*+*n*). This is because a matrix of size *mn* must be held in memory in order to be able to do a traceback to find the optimal alignment. Several authors have reduced the space requirements by constant factors (Altschul and Erickson 1986; Gotoh 1987).

However, Hirschberg (1975) described a linear space algorithm for computing maximal common subsequences. This recursive divide-and-conquer algorithm was adapted by Myers and Miller (1988) for local sequence alignment with affine gap penalties.

*Suboptimal alignments*

In addition to the optimal alignment, there are usually many suboptimal alignments that might be of interest. Most of these are variations of the optimal alignment providing additional support for the possible homology. Completely independent suboptimal alignments, defined as those that do not involve the same pairs of amino acids, are also important. Waterman and Eggert (1987) presented an algorithm to find the *k* best non-intersecting similar subsequences with a minimum score. Their method used time proportional to *kmn* and space proportional to *mn*. Huang *et al.* (1990) improved this algorithm to require only linear space. Huang and Miller (1991) further improved the time usage of this algorithm. Barton (1993) also described an algorithm for locating suboptimal alignments.

*Constrained alignments*

Chao *et al.* (1992) presented an algorithm for computing an optimal sequence alignment restricted to a diagonal band of width *w* in the matrix using time proportional to *nw* and space proportional to *n*. This method was utilized in the FASTA programs (Pearson and Lipman 1988) for computing the final 'optimized' scores.

A generalisation of this algorithm where the alignment was constrained to an arbitrarily bounded region with area *X* was later described by Chao *et al.* (1993). It used time proportional to *X* and space proportional to (*m*+*n*).

Zhang *et al.* (1998) presented another restricted alignment algorithm which was used in BLAST 2.0.x (Altschul *et al.* 1997). This method finds alignments without low-scoring regions, but poses no *apriori* bounds on the alignment region. It uses time proportional to the area of the region examined. It works by extending an alignment from a starting point (seed) until a score is reached that is a certain level below the maximum score found.

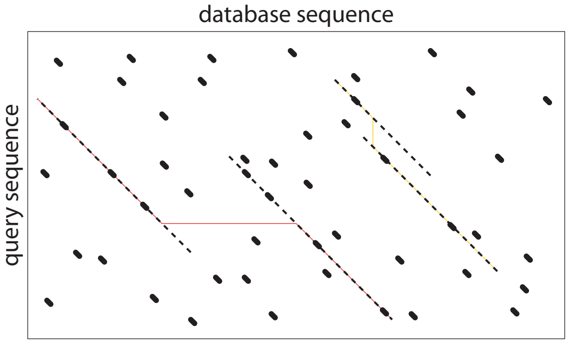

**Figure 7: Graphical illustration of sequence alignment**
K-tuples (black slabs), high-scoring ungapped alignment regions (dotted lines), gapped alignments (yellow and red lines), which are involved in different stages of heuristic alignments methods.

### 1.3.2 Heuristic alignment algorithms

When alignments are computed in the context of database searching, the brute-force dynamic programming algorithms described by Smith and Waterman (1981) and Gotoh (1982) are in general too time-consuming to be practical. Hence, several heuristic methods have been employed to obtain faster searches at the expense of reduced sensitivity.

*FASTA and related programs*

Wilbur and Lipman (1983; 1984) described an algorithm called NUCALN for fast identification of similarities between sequences based on an initial identification of tuples of *k* nucleotides (or amino acid residues) that are identical in the two sequences (see figure 7). These tuples are also called *k*-tuples or just ktups. Based on a precomputed table of the query sequence positions of all the $4^k$ (for nucleotides) or $20^k$ (for amino acids) possible tuples, it could rapidly find the position of all the *k*-tuples while scanning the database sequences. This table-lookup method was initially described by Dumas and Ninio (1982). Subsequently, the program identified the diagonals in the alignment matrix that contained a significantly higher number of *k*-tuples than expected by chance with random sequences. A diagonal *d* is defined as the cells in the alignment matrix having a position (*i*,*j*) where *j*-*i*=*d*. The assembly of bands of width *w* surrounding each of the significant diagonals was termed window space, and a Needleman-Wunsch type of alignment was perfomed within this space.

Lipman and Pearson (1985) developed these ideas further in the FASTP program for protein sequence database searches. When the five most interesting diagonals have been identified as described in the previous paragraph, these regions are rescored using an amino acid substitution matrix, and the highest score of these regions is called the initial score, which is used to rank the matching sequences. In addition, an optimised score for the highest ranking sequences is performed by a Needleman-Wunsch type of optimal alignment.

The FASTA program (Pearson and Lipman 1988) was an improved version of the FASTP program generalised for both nucleotide (FASTN) and protein (FASTP) sequences. The most detailed description of the FASTA algorithm, including several examples of its use, is given by Pearson (1990). The FASTA program proceeds in five steps:

1) First, the most interesting diagonal regions are found based on the number of *k*-tuples identified on the diagonal and their distance along the diagonal using a simple scoring formula. The ten best diagonal regions are processed further. This step finds ungapped alignments based on identical residues.

2) These regions are then rescored using a amino acid substitution score matrix, which also take conservative substitutions into account and not just identities. The regions are trimmed, and the initial subregion with the highest score is recorded for each of the ten regions. These are partial ungapped sequence alignments. Only initial regions with a score above a cutoff value are considered further. The highest of these scores is called the *init1* score. This step also finds ungapped alignments, but does consider conservative substitutions.

3) FASTA subsequently calculates an estimated gapped alignment score by joining together a combination of the compatible initial regions using a joining penalty. The resulting score, called the *initn* score, is reported and is used to rank the sequences. This step finds approximate gapped alignments.

4) FASTA also computes an optimal local alignment (Smith and Waterman 1981) constrained to a band (32 cells wide) centered around the highest scoring initial region, as described later by Chao *et al.* (1992). The score, *opt*, of this optimal alignment is also reported. This step results in an optimal gapped alignment.

5) Finally, some statistical computations are performed. FASTA plots a histogram of the distribution of scores and determines the standard deviation of the distribution of initial scores.

### *BLAST and related programs*

The first version of the NCBI BLAST programs was described by Altschul *et al.* (1990). The algorithm proceeds in much the same way as the FASTA algorithm, but there are some important differences.

In the first step, BLAST uses *words* of length *w* instead of *k*-tuples. These words include also conservative substitutions. A similar scheme was also suggested by Brutlag *et al.* (1990). The words used in BLAST contain all *w*-tuples that receive a score, *T*, above a certain level when compared using the amino acid substitution matrix. By default, BLAST uses *w*=3 and *T*=11. A given triplet in the query sequence will then match the triplets in the database sequence that has a score of 11 or more when the three pairs of amino acids are compared. This change gives increased sensitivity compared to FASTA.

In the second step, BLAST extends the initial words into so-called High-scoring Segment Pairs (HSPs) using the amino acid substitution matrix.This extension is performed in both directions along the diagonal from the initial word and is stopped when the potential score falls a level *X* below the currently found maximum score of the HSP.

The first version of BLAST does not consider gapped alignments at all, but computes a statistical measure of significance based on the highest scoring HSPs using sum-statistics (Karlin and Altschul 1990).

Later, WU-BLAST 2, which is a variant of NCBI BLAST that also takes gapped alignments into account was made available by Gish (1996), but this algorithm has never been published and the source code is not available, hence the algorithm cannot be analysed. The statistics used in WU-BLAST and a possible precursor called BLASTGP is mentioned by Altschul and Gish (1996). Comparison of the results of different methods have shown that WU-BLAST is quite sensitive.

Altschul *et al.* (1997) describe version 2 of NCBI BLAST which includes a few improvements that increases both the speed and the sensitivity of the program.

In the first step, BLAST 2 requires two matching words within a distance of about 40 on the same diagonal. This double-hit method reduces the number of hits substantially, but also reduces sensitivity relative to the first version of BLAST.

The extension of HSPs in the second step is performed in the same manner as with the previous version although with far fewer HSPs, and hence much faster.

Using midpoints on the HSPs as seeds,

**Table 6: The BLOSUM62 amino acid substitution score matrix**
The table shows the log-odds score associated with the replacement of one amino acid with another.
From Henikoff and Henikoff (1992).

| | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V | B | Z | X | * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 4 | -1 | -2 | -2 | 0 | -1 | -1 | 0 | -2 | -1 | -1 | -1 | -1 | -2 | -1 | 1 | 0 | -3 | -2 | 0 | -2 | -1 | 0 | -4 |
| R | -1 | 5 | 0 | -2 | -3 | 1 | 0 | -2 | 0 | -3 | -2 | 2 | -1 | -3 | -2 | -1 | -1 | -3 | -2 | -3 | -1 | 0 | -1 | -4 |
| N | -2 | 0 | 6 | 1 | -3 | 0 | 0 | 0 | 1 | -3 | -3 | 0 | -2 | -3 | -2 | 1 | 0 | -4 | -2 | -3 | 3 | 0 | -1 | -4 |
| D | -2 | -2 | 1 | 6 | -3 | 0 | 2 | -1 | -1 | -3 | -4 | -1 | -3 | -3 | -1 | 0 | -1 | -4 | -3 | -3 | 4 | 1 | -1 | -4 |
| C | 0 | -3 | -3 | -3 | 9 | -3 | -4 | -3 | -3 | -1 | -1 | -3 | -1 | -2 | -3 | -1 | -1 | -2 | -2 | -1 | -3 | -3 | -2 | -4 |
| Q | -1 | 1 | 0 | 0 | -3 | 5 | 2 | -2 | 0 | -3 | -2 | 1 | 0 | -3 | -1 | 0 | -1 | -2 | -1 | -2 | 0 | 3 | -1 | -4 |
| E | -1 | 0 | 0 | 2 | -4 | 2 | 5 | -2 | 0 | -3 | -3 | 1 | -2 | -3 | -1 | 0 | -1 | -3 | -2 | -2 | 1 | 4 | -1 | -4 |
| G | 0 | -2 | 0 | -1 | -3 | -2 | -2 | 6 | -2 | -4 | -4 | -2 | -3 | -3 | -2 | 0 | -2 | -2 | -3 | -3 | -1 | -2 | -1 | -4 |
| H | -2 | 0 | 1 | -1 | -3 | 0 | 0 | -2 | 8 | -3 | -3 | -1 | -2 | -1 | -2 | -1 | -2 | -2 | 2 | -3 | 0 | 0 | -1 | -4 |
| I | -1 | -3 | -3 | -3 | -1 | -3 | -3 | -4 | -3 | 4 | 2 | -3 | 1 | 0 | -3 | -2 | -1 | -3 | -1 | 3 | -3 | -3 | -1 | -4 |
| L | -1 | -2 | -3 | -4 | -1 | -2 | -3 | -4 | -3 | 2 | 4 | -2 | 2 | 0 | -3 | -2 | -1 | -2 | -1 | 1 | -4 | -3 | -1 | -4 |
| K | -1 | 2 | 0 | -1 | -3 | 1 | 1 | -2 | -1 | -3 | -2 | 5 | -1 | -3 | -1 | 0 | -1 | -3 | -2 | -2 | 0 | 1 | -1 | -4 |
| M | -1 | -1 | -2 | -3 | -1 | 0 | -2 | -3 | -2 | 1 | 2 | -1 | 5 | 0 | -2 | -1 | -1 | -1 | -1 | 1 | -3 | -1 | -1 | -4 |
| F | -2 | -3 | -3 | -3 | -2 | -3 | -3 | -3 | -1 | 0 | 0 | -3 | 0 | 6 | -4 | -2 | -2 | 1 | 3 | -1 | -3 | -3 | -1 | -4 |
| P | -1 | -2 | -2 | -1 | -3 | -1 | -1 | -2 | -2 | -3 | -3 | -1 | -2 | -4 | 7 | -1 | -1 | -4 | -3 | -2 | -2 | -1 | -2 | -4 |
| S | 1 | -1 | 1 | 0 | -1 | 0 | 0 | 0 | -1 | -2 | -2 | 0 | -1 | -2 | -1 | 4 | 1 | -3 | -2 | -2 | 0 | 0 | 0 | -4 |
| T | 0 | -1 | 0 | -1 | -1 | -1 | -1 | -2 | -2 | -1 | -1 | -1 | -1 | -2 | -1 | 1 | 5 | -2 | -2 | 0 | -1 | -1 | 0 | -4 |
| W | -3 | -3 | -4 | -4 | -2 | -2 | -3 | -2 | -2 | -3 | -2 | -3 | -1 | 1 | -4 | -3 | -2 | 11 | 2 | -3 | -4 | -3 | -2 | -4 |
| Y | -2 | -2 | -2 | -3 | -2 | -1 | -2 | -3 | 2 | -1 | -1 | -2 | -1 | 3 | -3 | -2 | -2 | 2 | 7 | -1 | -3 | -2 | -1 | -4 |
| V | 0 | -3 | -3 | -3 | -1 | -2 | -2 | -3 | -3 | 3 | 1 | -2 | 1 | -1 | -2 | -2 | 0 | -3 | -1 | 4 | -3 | -2 | -1 | -4 |
| B | -2 | -1 | 3 | 4 | -3 | 0 | 1 | -1 | 0 | -3 | -4 | 0 | -3 | -3 | -2 | 0 | -1 | -4 | -3 | -3 | 4 | 1 | -1 | -4 |
| Z | -1 | 0 | 0 | 1 | -3 | 3 | 4 | -2 | 0 | -3 | -3 | 1 | -1 | -3 | -1 | 0 | -1 | -3 | -2 | -2 | 1 | 4 | -1 | -4 |
| X | 0 | -1 | -1 | -1 | -2 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -2 | 0 | 0 | -2 | -1 | -1 | -1 | -1 | -1 | -4 |
| * | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | -4 | 1 |

BLAST 2 performs an accurate gapped alignment constrained not to contain any low-scoring regions, as described by Zhang *et al.* (1998). This gapped alignment leads to much increased sensitivity over the original BLAST program. The alignments take a lot of time and is hence only performed for the HSPs scoring above about 40, representing only about 2% of the database sequences.

Finally, NCBI BLAST 2 uses the new statistics for gapped alignments described by Altschul and Gish (1996) to compute an *E*-value expressing the expected number of random matches in the database having a given score.

*Other heuristic alignment algorithms*

Chao and Miller (1995) and Chao *et al.* (1995) have described algorithms for gapped local alignments of very long DNA sequences built on identified fragments of *k*-tuples with similarity between the sequences.

### 1.3.3 Substitution score matrices

All alignment methods involve the use of an amino acid substitution score matrix that indicates the score associated with replacing one amino acid with another. An example of such a matrix is shown in table 6.

Replacing an amino acid with an identical amino acid always gives a high score, but it varies according to the amino acid. Tryptophanes and cysteins give the highest score because of the low abundance. Substitutions with similar amino acid, e.g leucine by valine or isoleucine, gives a small positive score. Neutral substitutions give a zero score, and dissimilar amino acids give a negative score.

The scores in the matrices are based on the so-called log odds score (lods), which is by convention defined as the logarithm of the ratio between the probability that the amino acids are aligned in related sequences, and the probability that they are aligned by chance. This logarithm is usually multiplied by a scale factor and rounded to the nearest integer.

The first matrices in general use were the PAM (point accepted mutations per 100 residues, also known as percent accepted mutations) series of matrices compiled by Dayhoff *et al.* (1978). This series of matrices was based on estimated mutation rates in closely related (at least 85% identical) sequences. Matrices for less similar proteins were estimated by extrapolation. Different matrices were compiled and should be used dependent on the evolutionary distance between the proteins compared. Accordingly, a series of matrices were presented. A matrix designed for evolutionary distances of about 100 pam is called PAM100, while a matrix for 250 pam is called PAM250.

Several other matrices based on similarity in genetic code or chemical properties of the amino acids have been proposed. Gonnet *et al.* (1992) proposed a matrix based on their exhaustive matching of the entire protein database.

The most widely used modern matrices are the BLOSUM series (Henikoff and Henikoff 1992). These matrices are based on a database of blocks of aligned protein fragments. The blocks are grouped according to a minimum percentage of identical amino acids between the sequences. The BLOSUM62 matrix shown in table 6 is the default matrix in many alignment and database search programs. Recently, Henikoff *et al.* (1999) computed an updated version of the BLOSUM62 matrix based on updated protein sequence data, but it did not show significant increase in performance over the original.

Modern versions of the PAM matrices have been described by Jones *et al.* (1992), and a matrix based on structural alignments of homologous proteins have been described by Johnson and Overington (1993).

In alignments and database searches, the query sequence and amino acid substituition matrix can be replaced by a position-specific scoring profile (Gribskov *et al.* 1987; Thompson *et al.* 1994). The sequence  profile is created from a multiple alignment of closely related protein sequences, and gives a score for each possible amino acid in each position of the query profile. Profiles can be used in sensitive database searches for identification of sequences that are distantly related to a protein family. Profiles can be considered as an alternative to motifs.

### 1.3.4 Gap penalty functions

There is limited theoretical and empirical background for the treatment of gaps in alignments. However, it is common practice to deduct a gap penalty from the alignment score for each gap. The gap penalty is usually a function $g(k)$ dependent on the length of the gap, $k$. Figure 8 illustrates six different types of gap penalty functions.

To be biologically meaningful and intuitive the function should preferably adhere to the following rules: (1) It must be defined for all positive integers of $k$. (2) It must be positive for all values of $k$. (3) There should be a large penalty for opening a gap at all because gaps are relatively rare compared to simple substitutions. (4) The penalty for two or more gaps with total length $k$ should be larger than one gap of length $k$. (5) The penalty should increase with the length of the gap (monotonic). (6) The increase in penalty from $k$ to $k$+1 should be smaller than from $k$-1 to $k$ (concave).

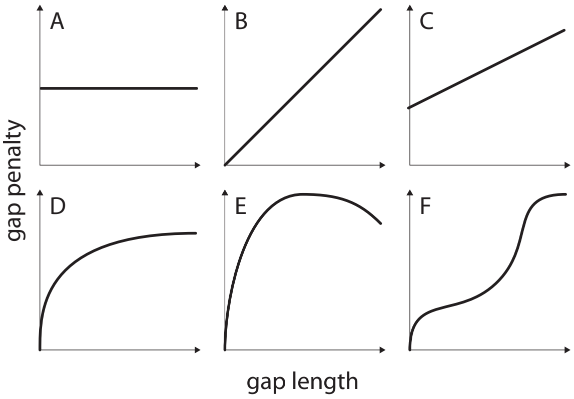

**Figure 8: Examples of gap penalty functions**
Graphs of constant (A), proportional (B), affine (C), logarithmic (D), concave (E) and monotonic (F) gap penalty functions are shown.

The simplest gap penalty function is just a **constant** penalty for each gap independent of its length where $g(k)=a$ as shown in figure 8A. Because of its simplicity of calculation, this scheme was used in some of the early alignment algorithms, and also in some hardware implementations. However, not taking the gap length into account seems oversimplified and makes alignments less accurate.

Another simple scheme is to deduct a gap penalty that is directly **proportional** to the length, k, of the gap, using a gap penalty function $g(k)=bk$ as in figure 8B. It is also simple to calculate, but not very accurate. It has also been used in early algorithms and in hardware implementations.

The most widely used scheme is the **affine** gap penalty function of the form $g(k)=a+bk$ shown in figure 8C, where $a$ is the gap opening penalty and $b$ is the gap extension penalty. This form of gap penalty was used in some of the early alignment methods, e.g. by Gotoh (1982). It is relatively easy to implement and has produced good alignment results. Note that in some contexts the gap open penalty is understood as the total gap penalty of a single residue gap ($a$+$b$). A gap opening penalty of 11 and a gap extension penalty of 1 is commonly used in combination with the BLOSUM62 matrix.

A **logarithmic** gap penalty function as illustrated in figure 8D of the form $g(k)=a+b\log(k)$ was suggested by Gonnet *et al.* (1992), and Benner *et al.* (1993) based on experiments with empirical data. The logarithmic gap penalty function seems to be the function best founded by experimental data, but is unfortunately much more complicated to calculate and implement in alignment programs than an affine function.

The **concave (**also known as convex) gap penalty functions shown in figure 8E are a class of general functions where $g(k) - g(k\text{-}1) \geq g(k+1) - g(k)$. It includes the affine and logarithmic forms described above, but is not monotonic (described below) in general. Concave functions are also harder to calculate and more complicated to implement in a rapid program than affine functions. Algorithms for optimal sequence alignment with concave gap penalty functions have been described by Waterman (1984) and Miller and Myers (1988).

The **monotonic** gap penalty functions depicted in figure 8F where $g(k) \geq g(k\text{-}1)$ are a general class of gap penalty functions that include the affine and logarithmic schemes described above, but are different from the concave funtions. Mott (1999) proposed an algorithm for optimal sequence alignment based on these gap functions.

The actual choice of constants in the functions should be based on empirical data or experiments and is dependent on the evolutionary distance between the sequences compared, and should be adapted to the choice of substitution score matrix.

Altschul (1998) suggested the use of **generalised affine** gap penalties of the form $g(k,l) = a + bk + cl$, where $k$ is the gap length and $l$ is the number of unaligned residues included. Based on multiple alignments and structural data, it was observed that the similarity between proteins is concentrated in segments of aligned residues separated by regions of unaligned residues and gaps. Instead of forcing a meaningless alignment on all residues, it seems better to penalise these unaligned residues in combination with a gap by a constant $l$ per residue. Alignments using this scheme were shown to be better than aligments based on an affine scheme, but the alignments took longer to compute.

### 1.3.5 Locally biased sequence composition

The existence of regions of locally biased sequence composition is a common source of mistaken homology between sequences (Lipman *et al.* 1984; Wootton and Federhen 1996). If the two sequences compared both contain regions where the frequency of certain amino acids or nucleotides is higher than normal, the two sequences might seem more similar than they really are. Several protein families, especially structural proteins, contain long regions dominated by e.g. alanine, glycine and serine. Some regions of DNA sequences may be dominated by very CG-rich or AT-rich regions. These similarities often represent statistically insignificant similarities. A common technique to avoid some of the problems, is to remove or mask these regions from the query sequence before a search is performed. A number of tools exist for masking these regions, e.g. XNU (Claverie and States 1993) and SEG (Wootton and Federhen 1993).

### 1.3.6 Significance of alignments

When is a sequence alignment significant? What alignment score is needed to assume that two sequences are related? Even an alignment of two random or unrelated sequences may reach a high score, and it can often be hard to draw the line in this twilight zone between the significant and insignificant alignments on a list of matches. A review on this subject have been published by Altschul and Gish (1996).

Simple metrics for assessing the significance of an alignment are based on the the percentage of identical or similar amino acid residues, possibly combined with the length of the alignment. However, a statistical measure based on the alignment score is usually required to assess the similarity between sequences.

Based on experiments with real DNA sequences, Smith *et al.* (1985) found that the optimal local sequence alignment score was proportional to the logarithm of the product of the sequence lengths. Collins *et al.* (1988) made a similar observation with unrelated or randomly shuffled protein sequences.

An idea of the significance of a real alignment can be obtained by comparing the raw score from a Smith-Waterman alignment of two real sequences with the expected score for an alignment of two random sequences of the same length. In addition to the alignment score and the length of the sequences compared, the amino acid composition in the sequences and the existence of locally biased regions should also be taken into account. The choice of substitution score matrix and gap penalties are of course also important.

The FASTA program (Pearson and Lipman 1988) displays a histogram of the score distribution, that visualises how the significant similarities stand out from the rest of insignificant alignments. In addition to the raw score, various values (Z-scores, ln()-scaled scores and P-values) have been used to measure the significance of an alignment.

Rigourous statistical treatment of the subject has resulted in a good quantitative measurement of the significance of alignments. The distribution of optimal alignment scores for random sequences was found to follow an extreme value distribution. Karlin and Altschul (1990) and Karlin and Brendel (1992) described the theory for the distribution of ungapped alignment scores, while the so-called sum-statistics for the score of multiple HSPs used in BLAST version 1.4 (Altschul *et al.* 1990) was described by Karlin and Altschul (1993). It was later shown that the theory for ungapped alignments could be generalised to gapped alignments by applying an edge-correcting factor (Mott 1992; Arratia and Waterman 1994; Waterman and Vingron 1994a, 1994b; Altschul and Gish 1996; Mott and Tribe 1999).

The final conclusion of the statistical theory is that the expected number, $E$, of alignments of random sequences with an optimal score equal to or above $S$ is expressed by the following formula:

$$E = \mathbf{K}mn\ \mathrm{e}^{-\lambda S}$$

Here, $m$ and $n$ are the lengths of the query and database sequences, respectively. In the context of a database search, the total length of all database sequences should be used for $n$. $\mathbf{K}$ and $\lambda$ are parameters that depend on the scoring scheme used. A length-correcting factor $z$ must be subtracted from the sequence lengths for the above formula to be precise in the case of gapped alignments.

$$z = \ln(\mathbf{K}mn) / \mathbf{H}$$

Appropriate values of the Karlin-Altschul parameters $\mathbf{K}$, $\lambda$ and $\mathbf{H}$ for a range of commonly used substitution score matrices and affine gap penalties have been estimated by large scale numerical simulations with random sequences (Altschul and Gish 1996).

### 1.3.7 Alignment with translated DNA sequences

A protein sequence can be compared to a DNA sequence even if the positions of potential protein coding regions are unknown. This is useful for searching the EST databases or other unannotated nucleotide sequence databases. This kind of searches may also be used to identify protein coding regions (Gish and States 1993). The DNA sequence can be translated into six different amino acid sequences using the universal genetic code shown in table 3 starting at the three first positions on each strand. The query protein sequence can then be aligned to each of these six amino acid sequences.

A procedure similar to the one indicated above was first implemented in the program "Translated Search" (Peltola *et al.* 1986) using a Smith-Waterman alignment, and later in the BLASTX, TBLASTX and TBLASTN programs using the heuristic BLAST algorithm (Altschul *et al.* 1990; Gish and States 1993) and in the TFASTA program using the heuristic FASTA algorithm (Pearson and Lipman 1988).

The early methods did not take into account the abundance of sequencing errors in the DNA sequences and other sources of so-called frameshifts that results in distribution of the similarity on more than one frame of translation. The EST and GSS databases have relative high rates of sequencing errors as they are based on single sequencing reads.

More recent programs like GenAl (Hein and Støvlbæk 1994; 1996), LAP (Huang and Zhang 1996), Grail (Guan and Uberbacher 1996), FASTX, FASTY, TFASTX, TFASTY (Zhang *et al.* 1997; Pearson *et al.* 1997) and FrameSearch Plus (Halperin *et al.* 1999) model frameshifts (and some also introns) in great detail. When a single nucleotide is deleted or inserted a frameshift penality is usually applied.

The programs with names ending with an X or Y usually translate the given query nucleotide sequence and compare it to the sequences in the database, while the programs with names starting with a T usually translate the nucleotide sequences in the database before comparing them to the query. A combination where both sequences are translated is also applied.

## 1.4 Database searching on parallel computer architectures

The algorithms for database searching can be implemented to run efficiently on various types of hardware with the ability to perform several operations simultaneously. There is a wide range of different hardware available on which the algorithms can be implemented. Hughey (1996) has reviewed various types of hardware that can be used and their performance. The hardware can be divided into a group of general-purpose computers which can be used for many different kinds of computations, and a group of hardware specifically designed for performing sequence alignments and database searches.

### 1.4.1 General-purpose parallel computers

General purpose computers with parallel processing capabilities usually contain a number of connected processors, ranging from dual-CPU workstations to supercomputers. The well-known dynamic programming or heuristic algorithms must be rewritten to run on such computers. The algorithms can be parallelised on different scales, from a simple coarse-grained parallelisation where e.g. the database sequences are divided on two or more processors each comparing the database sequence to the query sequence, to a complicated fine-grained parallelisation where the comparison of the query sequence against one database sequence is parallelised. The speed gained varies according to the type of algorithm and computer architecture.

The degree of connection between the processors in a parallel computer is a very important factor. A cluster of workstations connected by an Ethernet network is an example of loosely connected processors, while a workstation based on multiple CPUs with symmetric multiprocessing (SMP) is an example of tightly connected processors. Clusters are generally the least expensive, because of the advanced technology required for connecting CPUs together with high communication capacity. The degree of connection required depends on the type of computation that is performed. Clusters are very interesting for sequence database searches, because of the independence between the different sequences in the database.

Parallel computers are classified into SIMD (Single-Instruction, Multiple-Data) and MIMD (Multiple-Instruction, Multiple-Data) types according to whether the processing units perform the same or different operations on their data. The MasPar computer is an example of the former, while Paragon and SMP workstations are examples of the latter. Database searches with the dynamic programming algorithms involve many repetitions of the same simple operations, and a SIMD computer is hence well suited to this task. The Smith-Waterman algorithm has been implemented on the MasPar computer in the MPsrch (Sturrock and Collins 1993) and BLAZE (Brutlag *et al.* 1993) programs. The heuristic algorithms generally perform more complex computations and are hence probably not so easy to implement efficiently on SIMD computers.

The SSEARCH (Pearson 1991), FASTA (Pearson and Lipman 1988) and BLAST (Altschul *et al.* 1990; 1997) programs have all been implemented for SMP computers using individual threads that handle different database sequences. Other parallelisations of alignment and database search algorithms for various computer architectures include Deshpande (1991), Miller *et al.* (1991), Huang *et al.* 1992), Vogt and Argos (1992) and Julich (1995).

Microparallelism is an interesting form of SIMD, where a wide (e.g. 64 bits) integer register of a CPU is divided into many (e.g. eight) smaller units (e.g. 8 bits), and where the same arithmetic or logical operation can be performed simultaneously and independently on the data in each of the individual units. This technique can be performed on ordinary CPUs using normal instructions combined with a technique involving masking of the high order bits in each unit. However, it has become much easier recently with the introduction of MMX (Intel 1999) and related technology. Alpern *et al.* (1995) and Wozniak (1997) have presented implementations of the Smith-Waterman algorithm and database search programs using this type of parallelism.

### 1.4.2 Special-purpose parallel hardware

A number of different designs for special-purpose hardware for performing sequence alignments and database searching have been proposed and implemented. Their advantage over general-purpose computers is that they can be tailored specifically to perform sequence comparisons at a high speed, while the disadvantage is high cost.

Special-purpose hardware is usually built using either Xilinx FPGA (Field-Programmable Gate Arrays) or custom VLSI (Very Large Scale Integration) technology. The advantage of FPGA is that they are reprogrammable and can be built to work in a given function, and hence can be changed to remove bugs or to work with different algorithms, while VLSI is customly designed to a very specific purpose and cannot be changed. The advantage of VLSI is a lower cost per unit (at least in large volumes) and a higher processing speed. However, the design and initial costs for VLSI systems are higher than for FPGA.

Timelogic's DeCypher (TimeLogic Inc. 2000) and Compugen's Bioccelerator and BioXL (Compugen Inc. 2000) are commercial systems based on FPGA, while Paracel's Fast Data Finder and GeneMatcher systems (Paracel Inc. 2000) are based on VLSI. These commercial systems have been quite successful and are installed in many research centers around the world. All of these systems perform Smith-Waterman and other searches at high speed. BioScan (Singh *et al.* 1996) is another VLSI-based system that uses a simplified approach similar to NCBI BLAST version 1.4.

## 1.5 Comparison of methods for protein sequence similarity searches

The sensitivity, speed and cost of the various methods for sequence database comparisons have been evaluated and compared by several authors.

The performance of various score matrices, gap penalty schemes and statistical evaluation methods have also been assessed. See table 5 for an overview of many of the programs compared and references.

Pearson (1991) compared the sensitivity and selectivity of the Smith-Waterman algorithm to the FASTA program. He extracted 67 protein superfamilies from a superset of the PIR database (Barker *et al.* 1996; 2000), and examined the ability of the programs to identify distantly related sequences belonging to the classified superfamilies. The conclusion was that Smith-Waterman performed best in general, followed closely by FASTA with ktup=1, while FASTA with ktup=2 had the worst performance.

Pearson (1995) compared the Smith-Waterman algorithm to FASTA and NCBI BLAST version 1 using various matrices, gap penalties and score scaling. He used the same data set as earlier (Pearson 1991), and also introduced a criterion for performance called the equivalence number (EN). He found that the BLOSUM62 matrix was better than the PAM250 matrix. Using the BLOSUM62 matrix, FASTA with optimized score ranking and ktup=2 was only slightly better than BLAST, but FASTA with ktup=1 was significantly better than BLAST. The best performer was Smith-Waterman with ln()-scaling of scores.

Shpaer *et al.* (1996) compared implementations of the Smith-Waterman algorithm in software (SSEARCH) and on the Fast Data Finder (FDF) hardware (Paracel, 2000) to FASTA (ktup 1) and NCBI BLAST 1. They used a database and a performance measure (EN) similar to Pearson (1995). They found that SSEARCH performed equal to the Smith-Waterman implementation on the FDF, with some variations dependent on the choice of matrix and gap penalties, followed by FASTA and finally BLAST. They also found that a structural amino acid substitution matrix (Johnson and Overington 1993) performed slightly better than the BLOSUM matrices, and that ln()-scaling of scores performed better than raw scores or Karlin-Altschul statistics.

Thanaraj and Flores (1997) compared the fast implementations of the Smith-Waterman algorithm on the FDF (Paracel Inc. 2000), Bioccelerator (Compugen Inc. 2000) and MasPar (Sturrock and Collins 1993), and evaluated their sensitivity, speed and cost. Even though the gap penalty schemes used in these implementations varied, the ranking of the database sequences were found to be remarkably similar, and the sensitivity to be essentially equal. In the configurations tested, the FDF was fastest followed by Bioccelerator and MasPar. The FDF was also found to be the most cost-effective when taking the cost of the hardware into account, however the MasPar is a more general computer than the other architectures and may also be used for other purposes.

Agarwal and States (1998) compared SSEARCH, NCBI BLAST 1, WU-BLAST 2, FASTA with ktup=2 and PSW (Probabilistic Smith-Waterman) (Bucher and Hoffman 1996) using the same data set as Pearson (1991). They found that PSW performed best in general with full-length sequences closely followed by SSEARCH. When partial sequences were used WU-BLAST 2 and NCBI BLAST 1 performed best followed by SSEARCH.

In order for the comparison of the methods to be objective it is important to have a good definition of which proteins should be considered homologues. The classification of superfamilies in PIR may be biased in this respect as noted by Brenner *et al.* (1998) who instead used information from the SCOP database (Structural classification of proteins; Murzin *et al.* 1995). They assessed the performance of NCBI BLAST version 1, WU-BLAST version 2, FASTA and SSEARCH. They found that SSEARCH, FASTA with ktup=1 and WU-BLAST 2 performed best, followed by BLAST and FASTA with ktup=2. Interestingly, even the best programs were only able to identify about 18% of the structurally homologous protein pairs in their database.

# 2 Aims of the study

The general, overall goal of the study was to develop new computational methods for extracting as much as possible of biochemical and biological knowledge from available genomic sequence information. Because database searching based on sequence similarity is one of the most fundamental and computationally demanding tasks in the analysis of genomic sequence information, and because we had some background in this field, we aimed at developing better tools for performing such searches.

The software tools should be designed to take into account four major criteria for optimized performance and usability.

**Sensitivity:** In order for researchers to be able to discover new biological relationships in the sequence data using the tools, they should be sufficiently sensitive to detect even remote homologues in the database.

**Rapidity:** It is also important that these tools are rapid to enable database searches to be completed within a reasonable amount of time. Quick searches will also enable researchers to be more productive as they can perform several searches in rapid succession in order to explore relationships, instead of waiting for overnight searches.

**Affordability:** The tools should preferably use standard hardware technology available at low cost, in order to be available to researchers in general.

**Availability:** The tools should be readily available potential users, both in the sense that they should be easily distributed to where the users are working, and also be easy to use. It was hence one of our goals to make the tools developed available through a well-designed Internet service.

The criteria are to some extent mutually conflicting, but a reasonable balance should be achieved.

# 3 Summary of results

Papers I, II and III present three new methods for sequence alignment and database searching. The algorithms in both paper II and III are based on the use of parallel processing technology that has recently been introduced in modern general-purpose microprocessors. The method described in paper II is used in the final stage of the algorithm presented in paper III.

Services for performing online searches in a wide range of public nucleotide and protein databases using the tools described in papers I-III have been established at http://dna.uio.no/salsa/ and http://dna.uio.no/search/ on the Internet.

Paper IV presents an example of the use of various computer tools, including some of those presented in the first three papers, to identify a novel DNA repair gene from "raw" sequence data in the databases. The computer predictions have been confirmed by experimental approaches using techniques of molecular biology.

## 3.1 Paper I - SALSA: improved protein database searching by a new algorithm for assembly of sequence fragments into gapped alignments

This paper describes a new heuristic algorithm for creating gapped sequence alignments by assembling a set of initially identified fragments of similarity. The algorithm was implemented as part of a protein sequence database search tool, and the sensitivity was shown to be better than both FASTA (ktup=2) (Pearson and Lipman 1988) and BLAST (Altschul *et al.* 1990, 1997), and comparable to the unpublished WU-BLAST algorithm (Gish 1996). The speed was similar to FASTA and WU-BLAST.

SALSA initially identifies a set of fragments of ungapped sequence alignments, much in the same way as the BLAST 1.4 program (Altschul *et al.* 1990) does. However, SALSA subsequently performs an accurate assembly of these fragments to construct a complete gapped alignment. It examines which fragments are compatible and can be joined by inserting gaps after trimming or extension of the fragments. An accurate estimate of an optimal gapped alignment score is calculated by summing substitution scores and penalising for gaps.

The estimated score is often equal to or near the optimal alignment score, and is used for deciding which database sequences should be considered in detail by a brute-force Smith-Waterman alignment (1981). Making this decision using an estimated score instead of the raw score of the highest-scoring initial fragment, is probably the reason for the increased sensitivity in SALSA relative to BLAST.

SALSA has also been implemented to run efficiently on an SMP computer with multiple microprocessors.

## 3.2 Paper II - Six-fold speed-up of Smith-Waterman sequence database searches with parallel processing on common microprocessors

The Smith-Waterman algorithm (1981) is generally considered to be one of the most sensitive algorithms for performing sequence comparisons. However, it is very slow on ordinary computers. Special-purpose hardware has been designed to increase the speed, but is available only at a high cost (Hughey 1996).

We present an implementation of the Smith-Waterman algorithm using the parallel processing capabilities of the MMX/SSE technology in the Intel Pentium MMX, II and III microprocessors (Intel 1999) which are used in ordinary PCs and are available at low cost. Similar technology is embedded in most modern microprocessors. A sixfold increase in speed was obtained relative to the SSEARCH (Pearson 1991) program which is a fast non-parallel Smith-Waterman (1981) implementation. A speed of more than 150 million cell updates per second was obtained on an Intel Pentium III microprocessor running at 500MHz, which probably makes this implementation of the algorithm the fastest described to date.

The MMX/SSE technology enables eight independent arithmetic or logical operations on byte values to be performed simultaneously. The implementation was based on vectors corresponding to eight cells in the alignment matrix. In contrast to previous attempts at making a parallel version of the Smith-Waterman algorithm, we used vectors parallel to the query sequence, rather than vectors parallel to the minor diagonal in the matrix. The advantage of this arrangement is the simplified loading of values from memory, while the disadvantage is the loss of independence between the eight elements of the vector in the calculations. However, due to the relatively rare occurence of gaps in optimal alignments, and an optimized implementation, the loss of independence does not have a major impact on speed.

### 3.3 Paper III - ParAlign: a rapid and sensitive sequence database search algorithm using parallel processing on modern microprocessors

This paper describes a sequence alignment and database searching algorithm called ParAlign that is specifically designed to take advantage of the features of the Intel MMX technology (Intel 1999) . The algorithm is shown to be very close to the Smith-Waterman algorithm (Smith and Waterman 1981) in terms of sensitivity and close to the NCBI BLAST 2 algorithm (Altschul *et al.* 1997) in terms of speed.

Initially, the algorithm calculates the exact optimal ungapped alignment score for each diagonal in the alignment matrix. These maximum partial sums of substitution scores are computed very efficiently by the MMX-based implementation. Secondly, using a novel approach, the algorithm estimates the gapped alignment score by combining the scores of several neighbouring diagonals. Finally, the fraction of database sequences with the highest estimated score is subject to an optimal alignment with the query sequence performed by the procedure described in paper II.

The sensitivity and speed of ParAlign was evaluated using a set of 11 query sequences and compared to several other programs. Of the total 2 578 database sequences found to be statistically significant matches by the Smith-Waterman algorithm, ParAlign missed only 2 (0.1%). WU-BLAST (Gish 1996) was found to be the most sensitive of the other programs and missed 1.1%, but was 1.7 times slower than ParAlign. Only the NCBI BLAST 2 program was found to be faster than ParAlign, by a factor of 1.6, but missed 2.4%.

### 3.4 Paper IV - Identification of a human member of a new family of DNA repair proteins with homology to *E. coli* Exonuclease III

We have identified a novel familiy of proteins in *H.sapiens*, *M.musculus*, *A.thaliana, S.pombe* and *S.cerevisiae* with significant sequence similarity to DNA endonucleases. In this family, the *S.cerevisiae* APN2 protein has been experimentally suggested to be involved in the repair of abasic sites in DNA and to confer resistance to methyl methanesulfonate (MMS) in complementation experiments (Johnson *et al.* 1998; Bennett 1999; Morland, Seeberg and Bjørås, *in prep.*).

We identified the human gene in genomic (acc.no. Z83821) and EST sequence databases by similarity to the *APN2* gene (Johnson *et al.* 1998) using sequence similarity search database tools (paper I; Altschul *et al.* 1990; 1997). The human genomic sequence was located at position Xp11.21. The gene structure was predicted to consist of 6 exons encoding a 518aa 57.4kDa protein using the tools GeneMark (Borodovsky and McIninch 1993), Grail (Uberbacher and Mural 1991) and FEXH (Solovyev *et al.* 1994) and by taking into account information from the genomic and EST sequences and similarities to homologous proteins. The 300aa N-terminal region of the protein shows extensive sequence similarity to members of the AP endonuclease family (Gorman *et al.* 1997) and to L1 endonuclease (Feng *et al.* 1996), but the 200aa C-terminal region does not show any significant resemblance to any protein, except for a potential $Zn^{2+}$-binding motif also found in eukaryotic topoisomerase III enzymes (Hanai *et al.* 1996). The sequence was submitted to the EMBL database (acc.no. AJ011311).

An IMAGE clone (Auffray *et al.* 1995; Lennon *et al.* 1996) containing a full-length cDNA sequence was obtained and the predicted sequence was confirmed. By northern blot hybridisation analysis the protein was shown to be ubiquitously expressed in normal human tissues, with elevated levels in heart, kidney, liver and placenta. Constructs of green fluorescent protein (GFP) fused to either end of the protein were expressed, and the protein was shown to be transported to the cell nucleus, even though no known nuclear localisation signals (NLS) were found in the protein sequence by the PSORT II program (Nakai and Kanehisa 1992).

# 4 Discussion and conclusion

## 4.1 General discussion

The SALSA program presented in paper I, employs a dynamic-programming algorithm for assembling a set of ungapped alignments (fragments) into fully gapped alignments. It is based on an initial identification of words and extension of these into HSPs in a manner similar to BLAST (Altschul *et al.* 1990). The sensitivity and speed of the program was shown to be comparable to the unpublised WU-BLAST algorithm (Gish 1996).

In Paper II we presented a database search program implementing the Smith-Waterman algorithm using microparallelism. It seems to be the currently most rapid implementation of Smith-Waterman-based searches on ordinary hardware. It works on the least expensive and most commonly available hardware, and it attains a speedup of 6 over the best previously known implementation on the same hardware and a speed of over 150 million cell updates per second on an Intel Pentium III 500 MHz microprocessor. It should be easy to extend the algorithm to run on a SMP computer or a cluster, to perform high-performance Smith-Waterman alignments in the most cost-effective way. This solution should be competetive with the special-purpose hardware designed to perform Smith-Waterman searches.

In paper III, the ParAlign algorithm for database searches was presented. This algorithm also exploited microparallelism on common hardware. In the case of ungapped alignments, this algorithm computes exactly the same score as the algorithm of Smith and Waterman (1981) using affine gap penalties (Gotoh 1982). For gapped alignments the algorithm makes use of a heuristic that nevertheless results in a sensitivity that is very close to the optimal. In the standard tests, it misses only 0.1% of the significant alignments, compared to 1.1% for WU-BLAST 2 (Gish 1996), which is the second most sensitive heuristic algorithm, and 2.4% for NCBI BLAST 2 (Altschul *et al.* 1997). ParAlign is faster than all other heuristic programs, except for NCBI BLAST 2.0, which is 1.6 times faster. WU-BLAST is 1.7 times slower than ParAlign.

ParAlign provides an optimal combination of speed and sensitivity, and is better than NCBI BLAST 2 with respect to sensitivity. Even when compared to the Smith-Waterman implementation presented in paper II, ParAlign is a good alternative, since the small difference in sensitivity may be negligible and ParAlign runs much faster. It should also be easy to implement the ParAlign algorithm on other computer architectures.

We have previously used the tools presented to identify several enzymes involved in oxidative DNA repair (Luna *et al.* 1999), e.g. the human hOGH1 enzyme (Bjørås *et al.* 1997), the yeast NTG1 and NTG2 enzymes (Alseth *et al.* 1999) and the human hNTH enzyme (Luna *et al.* 2000). In paper IV, we have shown that the presented algorithms are useful for identifying a new family of proteins that share similarity with various DNA endonucleases. However, we have so far not been able to determine the exact enymatic activity of the human protein. Perhaps it only works in concert with yet another factor, or we have not determined the precise substrate for the enzyme.

## 4. 2 Improvements of the algorithms developed

The sensitivity of the algorithms presented have been compared with the Smith-Waterman algorithm as a reference. However, a more extensive assessment of the algorithms' ability to recognise structural homologous proteins, as described by Brenner *et al.* (1998) would be of great interest. This type of assesment should probably be included in the evaluation of new algorithms for sequence database searches.

The algorithms presented in the paper I is already designed to run on SMP computers using threads. The algorithms presented in papers II and III can easily be adapted to run on such computers. All three algorithms can also easily be modified to take a sequence profile as the query.

The tools described currently report only the highest-scoring alignment for each database sequence. They should be extended to report all significant non-overlapping suboptimal alignments using some of the existing techniques (Waterman and Eggert 1987; Huang *et al.* 1990; Huang and Miller 1991).

The implementations of the three algorithms are able to search a DNA database by translating each sequence into the six possible reading frames and comparing each of them to a protein query sequence. However, this does not take possible frameshifts into account. An obvious improvement would be to also accept DNA query sequences and implement one of the more advanced models described in section 1.3.7.

An extension of the algorithm to direct DNA sequence comparisons should also be considered, although it is dubious whether the high sensitivity of these algorithms is necessary in this type of comparisons. A faster and possibly less sensitive algorithm should rather be designed specifically to exploit microparallelism.

Searches with the tools are available on the Internet, and the results are presented with alignments and links to the sequence database entries. However, there are many possibilities for improvements in the presentation of the results, for instance with multiple alignments of matching sequences.

### 4.3 Issues for further studies

Future research will hopefully result in increased sensitivity and speed of sequence comparison and database searching methods. There are many different possibilities for improvement over the current techniques.

*Improving sensitivity*

Even the best of the current sequence alignment methods are only able to identify a fraction (less than half) of the homologous proteins in the structural databases (Brenner *et al.* 1998; Rost 1999). They quickly run into problems when the fraction of identical amino acids falls below 25%. It would be interesting to know how far it is theoretically possible to reach with techniques based solely on the primary amino acid sequence.

Improvements of the current algorithms or variations of these may be achieved by enhanced amino acid substitution score matrices, sequence profiles, or other ways to better model the changes in amino acids sequences between homologous proteins. Some kind of *k*-peptide substitution matrix or scoring schemes that better model the changes in proteins in more than single a amino acid position may lead to progress in this area. Hidden Markov model (HMM) methods (Krogh *et al.* 1994) have received increasing attention recently and are able to identify many structural relationships (Eddy 1998). HMMs may be considered as a position-dependent score profile where substitutions, insertions and deletions are modelled in detail with different probabilities at each position.

The gap penalty scheme of the classical alignment model may also be improved. Altschul (1998) showed how the use of generalised affine gap penalties could improve the sensitivity of alignments of proteins in several families. Mott (1999) described how the use of a logarithmic gap penalty function increases the sensitivity for detecting alignments with long gaps. Both of these methods unfortunately increase the computation time compared to alignments with ordinary affine gap penalties, but they might be worthwhile in many cases. Other treatments of gaps may also lead to increased sensitivity.

Alternative alignment algorithms may also lead to better results. The probabilistic Smith-Waterman (PSW) algorithm (Bucher and Hoffman 1996), which uses a HMM and also takes into account the importance of suboptimal alignments in addition to just the optimal alignment, was shown to be better than other algorithms in many cases (Agarwal and States 1998).

If a protein A is homologous to protein B, and B is homologous to protein C, then A must also be homologous to C. This transitivity property of homology (Pearson 1996) can be exploited in various ways. PSI-BLAST (Altschul *et al.* 1997; Altschul and Koonin 1998) uses this to gradually build up a sequence profile by iterative searches. However, this concept requires that the query protein has at least one homologous protein with significant sequence similarity in the database. Park *et al.* (1997; 1998) showed that sequence database searches using intermediate sequences could detect two to three times the number of homologues in the SCOP database as ordinary pairwise sequence alignments. Salamov *et al.* (1999) used a multiple intermediate sequence search (MISS) to detect many homologues in the CATH (Orengo *et al.* 1997) structural database that were not detected by simple pairwise sequence alignments. Karplus *et al.* (1999) used an iterative HMM method to construct protein family profiles and predict protein structures.

*Improving speed*

The amount of computing power available for a constant cost has been increasing exponentially for many years, and is now doubling approximately every 18 months, according to Moore's law (Intel 2000). However, if the rate of growth in GenBank over the last months continue, it is clear that increasingly longer time or increasingly more expensive computers will be needed in the future to search the entire database of DNA.

There has been numerous attempts at building special-purpose hardware solutions to the speed problem (Hughey 1996). However, because of the rapid increase in speed of general-purpose computers, such solutions are often advantageous over software implementations only over a short time period. The use of general-purpose computers is also favored by shifts in the algorithms used. However, parallel computers in the form of clusters or in the form of SIMD microparallelism seem to be a general concept that is not likely to become outdated in the near future. Rather, the use of such parallelism will probably be increasingly widespread in many fields of computing. Further advances in the SIMD technology and other new advances in hardware will probably improve the speed of the algorithms in the future.

However, to exploit microparallelism efficiently, new implementations of classical algorithms are needed and often require considerable effort. It is probably worthwhile in many cases, as shown in paper II. SIMD implementations of HMM algorithms, as briefly mentioned by Eddy (1998), or other advanced algorithms will probably be attempted in the near future.

Novel algorithms designed specifically to exploit the advantages of the parallel architecture at hand, is perhaps a better way to go. For instance, a new heuristic algorithm using microparallelism for protein database searches that is much faster than NCBI BLAST 2 would be very useful, even if some sensitivity would have to be sacrificed.

*Improved evaluation of methods*

Standardised and objective measures of structural similarity should be used for evaluating the performance of sequence similarity methods. The use of structural databases like SCOP (Murzin *et al.* 1995), CATH (Orengo *et al.* 1997) or FSSP (Holm and Sander 1996) for evaluation of the sensitivity of the methods available is clearly an advance over the evaluation methods based on the possibly subjective superfamily classification in PIR (Brenner *et al.* 1998; Barker *et al.* 1996; 2000).

### 4.4 Other future problems

The enormous size and growth of the databases may be a problem for some file systems that are limited to files of size less than 2GB, due to the use of signed 32-bit integers. Also some computers and operating systems have problems with memory sizes of this order. Additionally, the indices in the NCBI database formats are limited to 32 bits, and cannot be used for nucleotide databases in a single file above 4GB, corresponding to 16Gbp. With the present rate of growth of the databases, this limit will be reached by the end of 2000. A new database format should be designed with this in mind.

The huge size of the databases also limits the ability for individual users to have the entire databases available on their local harddisk. The use of centralised Internet servers with the databases and search services will hence probably become even more widespread than today.

### 4.5 Conclusion

Novel algorithms for sequence database searching has been described that are superior to the existing software with respect to a combination of sensitivity and speed. The programs take advantage of the parallel processing technology available in common microprocessors and include novel approaches to the computation of gapped alignment scores. The algorithms developed can be developed further and it will be of major interest to a establish a computing cluster that can fully exploit the capability of the programs.

# 5 References

Adams, M.D., Celniker, S.E., Holt, R.A., Evans, C.A., Gocayne, J.D., Amanatides, P.G., Scherer, S.E., Li, P.W., Hoskins, R.A., Galle, R.F. *et al.* (2000) The genome sequence of *Drosophila melanogaster*. Science, 287, 2185-2196.

Agarwal, P. and States, D.J. (1998) Comparative accuracy of methods for protein sequence similarity search. Bioinformatics, 14, 40-47.

Aho, A.V. (1986) Compilers: principles, techniques, and tools. Addison-Wesley. Reading, MA, USA.

Alm, R.A., Ling, L.S., Moir, D.T., King, B.L., Brown, E.D., Doig, P.C., Smith, D.R., Noonan, B., Guild, B.C., deJonge, B.L., Carmel, G., Tummino, P.J., Caruso, A. , Uria-Nickelsen, M., Mills, D.M., Ives, C., Gibson, R., Merberg, D., Mills, S.D., Jiang, Q., Taylor, D.E., Vovis, G.F. and Trust, T.J. (1999) Genomic-sequence comparison of two unrelated isolates of the human gastric pathogen *Helicobacter pylori*. Nature, 397, 176-180.

Alpern, B., Carter, L., and Gatlin, K. S. Microparallelism and High Performance Protein Matching. Proceedings of the 1995 ACM/IEEE Supercomputing Conference: San Diego, California, Dec 3-8, 1995. http://www.supercomp.org/sc95/proceedings/549_LCAR/SC95.HTM

Alseth, I., Eide, L., Pirovano, M., Rognes, T., Seeberg, E. and Bjørås, M. (1999) The Saccharomyces cerevisiae homologues of endonuclease III from Escherichia coli, Ntg1 and Ntg2, are both required for efficient repair of spontaneous and induced oxidative DNA damage in yeast. Mol. Cell Biol., 19, 3779-3787.

Altschul, S.F. and Erickson, B.W. (1986) Optimal sequence alignment using affine gap costs. Bull. Math. Biol., 48, 603-616.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool. J. Mol. Biol., 215, 403-410.

Altschul, S.F., Boguski, M.S., Gish, W. and Wootton, J.C. (1994) Issues in searching molecular sequence databases. Nat. Genet., 6, 119-129.

Altschul, S.F. and Gish, W. (1996) Local alignment statistics. Methods Enzymol., 266, 460-480.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids. Res., 25, 3389-3402.

Altschul, S.F. and Koonin, E.V. (1998) Iterated profile searches with PSI-BLAST - a tool for discovery in protein databases. Trends Biochem. Sci., 23, 444-447.

Altschul, S.F. (1998) Generalized affine gap costs for protein sequence alignment. Proteins, 32, 88-96.

Andersson, S.G., Zomorodipour, A., Andersson, J.O., Sicheritz-Ponten, T., Alsmark, U.C., Podowski, R.M., Naslund, A.K., Eriksson, A.S., Winkler, H.H. and Kurland, C.G. (1998) The genome sequence of *Rickettsia prowazekii* and the origin of mitochondria. Nature, 396, 133-140.

Anfinsen, C.B. (1973) Principles that govern the folding of protein chains. Science, 181, 223-230.

Argos, P., Vingron, M. and Vogt, G. (1991) Protein sequence comparison: methods and significance. Protein Eng., 4, 375-383.

Argos, P. (1994) Sensitive methods for determining the relatedness of proteins with limited sequence homology. Curr. Opin. Biotechnol., 5, 361-371.

Arratia, R. and Waterman M.S. (1994) A phase transition for the score in matching random sequences allowing deletions. Ann. Appl. Prob., 4, 200-225.

Auffray, C., Behar, G., Bois, F., Bouchier, C., Da Silva, C., Devignes, M.D., Duprat, S., Houlgatte, R., Jumeau, M.N., Lamy, B., Lorenzo, F., Mitchell, H., Mariage-Samson, R., Pietu, G., Pouliot, Y., Sebastiani-Kabaktchis, C. and Tessier, A. (1995) IMAGE: molecular integration of the analysis of the human genome and its expression. C.R.Acad.Sci.III., 318, 263-272.

Bairoch, A. and Apweiler, R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids. Res., 28, 45-48.

Baker, W., van den Broek, A., Camon, E., Hingamp, P., Sterk, P., Stoesser, G. and Tuli, M.A. (2000) The EMBL nucleotide sequence database. Nucleic Acids. Res., 28, 19-23.

Barker, W.C., Pfeiffer, F. and George, D.G. (1996) Superfamily classification in PIR-

International Protein Sequence Database. Methods Enzymol., 266, 59-71.

Barker, W.C., Garavelli, J.S., Huang, H., McGarvey, P.B., Orcutt, B.C., Srinivasarao, G.Y., Xiao, C., Yeh, L.S., Ledley, R.S., Janda, J.F., Pfeiffer, F., Mewes, H.W., Tsugita, A. and Wu, C. (2000) The protein information resource (PIR). Nucleic Acids.Res., 28, 41-44.

Barton, G.J. (1993) An efficient algorithm to locate all locally optimal alignments between two sequences allowing for gaps. Comput. Appl. Biosci., 9, 729-734.

Benner, S.A., Cohen, M.A. and Gonnet, G.H. (1993) Empirical and structural models for insertions and deletions in the divergent evolution of proteins. J. Mol. Biol., 229, 1065-1082.

Bennett, R.A. (1999) The *Saccharomyces cerevisiae* ETH1 gene, an inducible homolog of exonuclease III that provides resistance to DNA-damaging agents and limits spontaneous mutagenesis. Mol. Cell Biol., 19, 1800-1809.

Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Rapp, B.A. and Wheeler, D.L. (2000) GenBank. Nucleic Acids. Res., 28, 15-18.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. (2000) The Protein Data Bank. Nucleic Acids. Res., 28, 235-242.

Bernstein, H. (1999) RasMol version 2.7.1.

Bjørås, M., Luna, L., Johnsen, B., Hoff, E., Haug, T., Rognes, T. and Seeberg, E. (1997) Opposite base-dependent reactions of a human base excision repair enzyme on DNA containing 7,8-dihydro-8-oxoguanine and abasic sites. EMBO J., 16, 6314-6322.

Blattner, F.R., Plunkett, G., Bloch, C.A., Perna, N.T., Burland, V., Riley, M., Collado-Vides, J., Glasner, J.D., Rode, C.K., Mayhew, G.F., Gregor, J., Davis, N.W., Kirkpatrick, H.A., Goeden, M.A., Rose, D.J., Mau, B. and Shao, Y. (1997) The complete genome sequence of *Escherichia coli* K-12. Science, 277, 1453-1474.

Borodovsky, M. and McIninch, J.D. (1993) GeneMark: Parallel gene recognition for both DNA strands. Computers & Chemistry, 17, 123-133.

Brenner, S.E., Chothia, C. and Hubbard, T.J. (1998) Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. Proc. Natl. Acad. Sci. U.S.A., 95, 6073-6078.

Bruner, S.D., Norman, D.P. and Verdine, G.L. (2000) Structural basis for recognition and repair of the endogenous mutagen 8-oxoguanine in DNA. Nature, 403, 859-866.

Brutlag, D.L., Dautricourt, J.P., Maulik, S. and Relph, J. (1990) Improved sensitivity of biological sequence database searches. Comput. Appl. Biosci., 6, 237-245.

Brutlag, D.L., Dautricourt, J.P., Diaz, R., Fier, J., Moxon, B. and Stamm, R. (1993) BLAZE - An implementation of the Smith-Waterman sequence comparison algorithm on a massively-parallel computer. Computers & Chemistry, 17, 203-207.

Bucher, P. and Hofmann, K. (1996) A sequence similarity search algorithm based on a probabilistic interpretation of an alignment scoring system. Intellingent Systems for Molecular Biology, 4, 44-51.

Bult, C.J., White, O., Olsen, G.J., Zhou, L., Fleischmann, R.D., Sutton, G.G., Blake, J.A., FitzGerald, L.M., Clayton, R.A., Gocayne, J.D., Kerlavage, A.R., Dougherty, B.A., Tomb, J.F., Adams, M.D., Reich, C.I., Overbeek, R., Kirkness, E.F., Weinstock, K.G., Merrick, J.M., Glodek, A., Scott, J.L., Geoghagen, N.M. and Venter, J.C. (1996) Complete genome sequence of the methanogenic archaeon, *Methanococcus jannaschii*. Science, 273, 1058-1073.

Casjens, S., Palmer, N., van Vugt, R., Mun, H.W., Stevenson, B., Rosa, P., Lathigra, R., Sutton, G., Peterson, J., Dodson, R.J., Haft, D., Hickey, E., Gwinn, M., White, O. and Fraser, M. (2000) A bacterial genome in flux: the twelve linear and nine circular extrachromosomal DNAs in an infectious isolate of the lyme disease spirochete *Borrelia burgdorferi*. Mol. Microbiol., 35, 490-516.

Chao, K.M., Pearson, W.R. and Miller, W. (1992) Aligning two sequences within a specified diagonal band. Comput. Appl. Biosci., 8, 481-487.

Chao, K.M., Hardison, R.C. and Miller, W. (1993) Constrained sequence alignment. Bull. Math. Biol., 55, 503-524.

Chao, K.M., Zhang, J., Ostell, J. and Miller, W. (1995) A local alignment tool for very long DNA sequences. Comput.Appl.Biosci., 11, 147-153.

Chao, K.M. and Miller, W. (1995) Linear-space algorithms that build local alignments from fragments. Algorithmica, 13, 106-134.

Chothia, C. (1992) Proteins. One thousand families for the molecular biologist. Nature, 357, 543-544.

Cole, S.T., Brosch, R., Parkhill, J., Garnier, T., Churcher, C., Harris, D., Gordon, S.V., Eiglmeier, K., Gas, S., Barry, C.E., Tekaia, F., Badcock, K., Basham, D., Brown, D., Chillingworth, T., Connor, R., Davies, R., Devlin, K., Feltwell, T., Gentles, S., Hamlin, N., Holroyd, S., Hornsby, T., Jagels, K. and Barrell, B.G. (1998) Deciphering the biology of *Mycobacterium tuberculosis* from the complete genome sequence. Nature, 393, 537-544.

Collins, F.S., Patrinos, A., Jordan, E., Chakravarti, A., Gesteland, R. and Walters, L. (1998) New goals for the U.S. Human Genome Project: 1998-2003. Science, 282, 682-689.

Collins, J.F., Coulson, A.F. and Lyall, A. (1988) The significance of protein sequence similarities. Comput. Appl. Biosci., 4, 67-71.

Compugen Inc. (2000) Bioccelerator. http://www.cgen.com/

Dayhoff, M.O., Schwartz, R.M., and Orcutt B.C. (1978) A model of evolutionary change in proteins. Matrices for detecting distant relationships. In Atlas of Protein Sequence and Structure. Dayhoff, M.O. (ed.) National Biomedical Research Foundation, Washington DC. 5, Suppl. 3, 345-358.

Deckert, G., Warren, P.V., Gaasterland, T., Young, W.G., Lenox, A.L., Graham, D.E., Overbeek, R., Snead, M.A., Keller, M., Aujay, M., Huber, R., Feldman, R.A., Short, J.M., Olsen, G.J. and Swanson, R.V. (1998) The complete genome of the hyperthermophilic bacterium *Aquifex aeolicus*. Nature, 392, 353-358.

Deshpande, A.S., Richards, D.S. and Pearson, W.R. (1991) A platform for biological sequence comparison on parallel computers. Comput. Appl. Biosci., 7, 237-247.

Doolittle, R.F. (1986) Of URFS and ORFS : a primer on how to analyze derived amino acid sequences. University Science Books. Mill Valley, CA, USA.

Doolittle, R.F. (1990) Molecular evolution: computer analysis of protein and nucleic acid sequences. Academic Press. San Diego, CA, USA.

Doolittle, R.F. (1996) Computer methods for macromolecular sequence analysis. Academic Press. San Diego, CA, USA.

Dumas, J.P. and Ninio, J. (1982) Efficient algorithms for folding and comparing nucleic acid sequences. Nucleic Acids. Res., 10, 197-206.

Dunham, I., Shimizu, N., Roe, B.A., Chissoe, S., Hunt, A.R., Collins, J.E., Bruskiewich, R., Beare, D.M., Clamp, M., Smink, L.J., Ainscough, R., Almeida, J.P., Babbage, A., Bagguley, C., Bailey, J., Barlow, K., Bates, K.N., Beasley, O., Bird, C.P., Blakey, S., Bridgeman, A.M., Buck, D., Burgess, J., Burrill, W.D. and O'Brien, K.P. (1999) The DNA sequence of human chromosome 22. Nature, 402, 489-495.

Eddy, S.R. (1998) Profile hidden Markov models. Bioinformatics, 14, 755-763.

Evensen, G. and Seeberg, E. (1982) Adaptation to alkylation resistance involves the induction of a DNA glycosylase. Nature, 296, 773-775.

Feng, Q., Moran, J.V., Kazazian, H.H.J. and Boeke, J.D. (1996) Human L1 retrotransposon encodes a conserved endonuclease required for retrotransposition. Cell, 87, 905-916.

Fleischmann, R.D., Adams, M.D., White, O., Clayton, R.A., Kirkness, E.F., Kerlavage, A.R., Bult, C.J., Tomb, J.F., Dougherty, B.A. and Merrick, J.M. (1995) Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. Science, 269, 496-512.

Fraser, C.M., Gocayne, J.D., White, O., Adams, M.D., Clayton, R.A., Fleischmann, R.D., Bult, C.J., Kerlavage, A.R., Sutton, G. and Kelley, J.M. (1995) The minimal gene complement of *Mycoplasma genitalium*. Science, 270, 397-403.

Fraser, C.M., Casjens, S., Huang, W.M., Sutton, G.G., Clayton, R., Lathigra, R., White, O., Ketchum, K.A., Dodson, R., Hickey, E.K., Gwinn, M., Dougherty, B., Tomb, J.F. , Fleischmann, R.D., Richardson, D., Peterson, J., Kerlavage, A.R., Quackenbush, J., Salzberg, S., Hanson, M., van Vugt, R., Palmer, N., Adams, M.D., Gocayne, J. and Venter, J.C. (1997) Genomic sequence of a Lyme disease spirochaete, *Borrelia burgdorferi*. Nature, 390, 580-586.

Fraser, C.M., Norris, S.J., Weinstock, G.M., White, O., Sutton, G.G., Dodson, R., Gwinn, M., Hickey, E.K., Clayton, R., Ketchum, K.A., Sodergren, E., Hardham, J.M., McLeod, M.P., Salzberg, S., Peterson, J., Khalak, H., Richardson, D., Howell, J.K., Chidambaram, M., Utterback, T., McDonald, L., Artiach, P., Bowman, C., Cotton, M.D. and Venter, J.C. (1998) Complete genome sequence of *Treponema pallidum*, the syphilis spirochete. Science, 281, 375-388.

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

Gish, W. and States, D.J. (1993) Identification of protein coding regions by database similarity search. Nat. Genet., 3, 266-272.

Gish, W. (1996) WU-BLAST. http://blast.wustl.edu/

Goad, W.B. and Kanehisa, M.I. (1982) Pattern recognition in nucleic acid sequences. I. A general method for finding local homologies and symmetries. Nucleic Acids. Res., 10, 247-263.

Goffeau, A., Barrell, B.G., Bussey, H., Davis, R.W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J.D., Jacq, C., Johnston, M., Louis, E.J., Mewes, H.W., Murakami, Y., Philippsen, P., Tettelin, H. and Oliver, S.G. (1996) Life with 6000 genes. Science, 274, 546-567.

Gonnet, G.H., Cohen, M.A. and Benner, S.A. (1992) Exhaustive matching of the entire protein sequence database. Science , 256, 1443-1445.

Gorman, M.A., Morera, S., Rothwell, D.G., de La Fortelle, E., Mol, C.D., Tainer, J.A., Hickson, I.D. and Freemont, P.S. (1997) The crystal structure of the human DNA repair endonuclease HAP1 suggests the recognition of extra-helical deoxyribose at DNA abasic sites. EMBO J., 16, 6548-6558.

Gotoh, O. (1982) An improved algorithm for matching biological sequences. J. Mol. Biol., 162, 705-708.

Gotoh, O. (1987) Pattern matching of biological sequences with limited storage. Comput. Appl. Biosci., 3, 17-20.

Green, P. (1993) SWAT. http://www.genome.washington.edu/uwgc/analysistools/swat.htm

Gribskov, M., McLachlan, A.D. and Eisenberg, D. (1987) Profile analysis: detection of distantly related proteins. Proc. Natl. Acad. Sci. U.S.A., 84, 4355-4358.

Guan, X. and Uberbacher, E.C. (1996) Alignments of DNA and protein sequences containing frameshift errors. Comput. Appl. Biosci., 12, 31-40.

Gusfield, D. (1997) Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press.

Halperin, E., Faigler, S. and Gill-More, R. (1999) FramePlus: aligning DNA to protein sequences. Bioinformatics, 15, 867-873.

Hanai, R., Caron, P.R. and Wang, J.C. (1996) Human TOP3: a single-copy gene encoding DNA topoisomerase III. Proc. Natl. Acad. Sci U.S.A., 93, 3653-3657.

Hattori, M., Fujiyama, A., Taylor T.D., Watanabe H., Yada T., Park H.-S., Toyoda, A., Ishii, K., Totoki, Y., Choi, D.-K. *et al.* (2000) The DNA sequence of human chromosome 21. Nature, 405, 311-319.

Hein, J. and Støvlbæk, J. (1994) Genomic alignment. J. Mol. Evol., 38, 310-316.

Hein, J. and Støvlbæk, J. (1996) Combined DNA and protein alignment. Methods Enzymol., 266, 402-418.

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

Henikoff, S., Henikoff, J.G. and Pietrokovski, S. (1999) Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. Bioinformatics, 15, 471-479.

Himmelreich, R., Hilbert, H., Plagens, H., Pirkl, E., Li, B.C. and Herrmann, R. (1996) Complete sequence analysis of the genome of the bacterium *Mycoplasma pneumoniae*. Nucleic Acids. Res., 24, 4420-4449.

Hirschberg, D.S. (1975) A linear space algorithm for computing longest common subsequences. Commun. Assoc. Comput. Mach., 18, 341-343.

Hirschberg, J.D., Dahle, D.M., Karplus, K., Speck, D. and Hughey, R. (1998) Kestrel: A programmable array for sequence analysis. Journal of VLSI signal processing systems for signal image and video technology, 19, 115-126.

Hobohm, U. and Sander, C. (1995) A sequence property approach to searching protein databases. J. Mol. Biol., 251, 390-399.

Hollis, T., Ichikawa, Y. and Ellenberger, T. (2000) DNA bending and a flip-out mechaanism for base excision by the helix-hairpin-helix DNA glycosylase, *Escherichia coli* AlkA. EMBO J., 19, 758-766.

Holm, L. and Sander, C. (1996) The FSSP database: fold classification based on structure-structure alignment of proteins. Nucleic Acids. Res., 24, 206-209.

Huang, X., Miller, W., Schwartz, S. and Hardison, R.C. (1992) Parallelization of a local similarity algorithm. Comput. Appl. Biosci., 8, 155-165.

Huang, X. and Zhang, J. (1996) Methods for comparing a DNA sequence with a protein sequence. Comput. Appl. Biosci., 12, 497-506.

Huang, X.Q., Hardison, R.C. and Miller, W. (1990) A space-efficient algorithm for local similarities. Comput. Appl. Biosci., 6, 373-381.

Huang, X.Q. and Miller, W. (1991) A time-efficient, linear-space local similarity algorithm. Adv. Appl. Math., 12, 337-357.

Hughey, R. (1996) Parallel hardware for sequence comparison and alignment. Comput. Appl. Biosci., 12, 473-479.

Intel (1999) Intel Architecture Software Developer's manual; Volume 2: Instruction Set Reference. http://developer.intel.com/design/pentiumii/manuals/243191.htm

Intel (2000) Moore's law. http://www.intel.com/intel/museum/25anniv/hof/moore.htm

Johnson, M.S. and Overington, J.P. (1993) A structural basis for sequence comparisons. An evaluation of scoring methodologies. J. Mol. Biol., 233, 716-738.

Johnson, R.E., Torres-Ramos, C.A., Izumi, T., Mitra, S., Prakash, S. and Prakash, L. (1998) Identification of APN2, the *Saccharomyces cerevisiae* homolog of the major human AP endonuclease HAP1, and its role in the repair of abasic sites. Genes Dev., 12, 3137-3143.

Jones, D.T., Taylor, W.R. and Thornton, J.M. (1992) The rapid generation of mutation data matrices from protein sequences. Comput. Appl. Biosci., 8, 275-282.

Julich, A. (1995) Implementations of BLAST for parallel computers. Comput. Appl. Biosci., 11, 3-6.

Kalman, S., Mitchell, W., Marathe, R., Lammel, C., Fan, J., Hyman, R.W., Olinger, L., Grimwood, J., Davis, R.W. and Stephens, R.S. (1999) Comparative genomes of *Chlamydia pneumoniae* and *C. trachomatis*. Nat. Genet., 21, 385-389.

Kaneko, T., Sato, S., Kotani, H., Tanaka, A., Asamizu, E., Nakamura, Y., Miyajima, N., Hirosawa, M., Sugiura, M., Sasamoto, S., Kimura, T., Hosouchi, T., Matsuno, A. , Muraki, A., Nakazaki, N., Naruo, K., Okumura, S., Shimpo, S., Takeuchi, C., Wada, T., Watanabe, A., Yamada, M., Yasuda, M. and Tabata, S. (1996) Sequence analysis of the genome of the unicellular cyanobacterium *Synechocystis* sp. strain PCC6803. II. Sequence determination of the entire genome and assignment of potential protein-coding regions. DNA Res., 3, 109-136.

Karlin, S. and Altschul, S.F. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc. Natl. Acad. Sci. U.S.A., 87, 2264-2268.

Karlin, S. and Brendel, V. (1992) Chance and statistical significance in protein and DNA sequence analysis. Science, 257, 39-49.

Karlin, S. and Altschul, S.F. (1993) Applications and statistics for multiple high-scoring segments in molecular sequences. Proc. Natl. Acad. Sci. U.S.A., 90, 5873-5877.

Karplus, K., Barrett, C., Cline, M., Diekhans, M., Grate, L. and Hughey, R. (1999) Predicting protein structure using only sequence information. Proteins, 37, 121-125.

Kawarabayasi, Y., Sawada, M., Horikawa, H., Haikawa, Y., Hino, Y., Yamamoto, S., Sekine, M., Baba, S., Kosugi, H., Hosoyama, A., Nagai, Y., Sakai, M., Ogura, K., Otsuka, R., Nakazawa, H., Takamiya, M., Ohfuku, Y., Funahashi, T., Tanaka, T., Kudoh, Y., Yamazaki, J., Kushida, N., Oguchi, A., Aoki, K. and Kikuchi, H. (1998) Complete sequence and gene organization of the genome of a hyper-thermophilic archaebacterium, *Pyrococcus horikoshii* OT3. DNA Res., 5, 55-76.

Kawarabayasi, Y., Hino, Y., Horikawa, H., Yamazaki, S., Haikawa, Y., Jin-no, K., Takahashi, M., Sekine, M., Baba, S., Ankai, A., Kosugi, H., Hosoyama, A., Fukui, S., Nagai, Y., Nishijima, K., Nakazawa, H., Takamiya, M., Masuda, S., Funahashi, T., Tanaka, T., Kudoh, Y., Yamazaki, J., Kushida, N., Oguchi, A. and Kikuchi, H. (1999) Complete genome sequence of an aerobic hyper-thermophilic crenarchaeon, *Aeropyrum pernix* K1. DNA Res., 6, 83-52.

Klenk, H.P., Clayton, R.A., Tomb, J.F., White, O., Nelson, K.E., Ketchum, K.A., Dodson, R.J., Gwinn, M., Hickey, E.K., Peterson, J.D., Richardson, D.L., Kerlavage, A.R., Graham, D.E., Kyrpides, N.C., Fleischmann, R.D., Quackenbush, J., Lee, N.H., Sutton, G.G., Gill, S., Kirkness, E.F., Dougherty, B.A., McKenney, K., Adams, M.D., Loftus, B. and Venter, J.C. (1997) The complete genome sequence of the hyperthermophilic, sulphate-reducing archaeon *Archaeoglobus fulgidus*. Nature, 390, 364-370.

Krogh, A., Brown, M., Mian, I.S., Sjolander, K. and Haussler, D. (1994) Hidden Markov models in computational biology. Applications to protein modeling. J. Mol. Biol., 235, 1501-1531.

Kunst, F., Ogasawara, N., Moszer, I., Albertini, A.M., Alloni, G., Azevedo, V., Bertero, M.G., Bessieres, P., Bolotin, A., Borchert, S., Borriss, R., Boursier, L., Brans, A., Braun, M., Brignell, S.C., Bron, S., Brouillet, S., Bruschi, C.V., Caldwell, B., Capuano, V., Carter, N.M., Choi, S.K., Codani, J.J., Connerton, I.F. and Danchin, A. (1997) The complete genome sequence of the gram-positive bacterium *Bacillus subtilis*. Nature, 390, 249-256.

Lennon, G., Auffray, C., Polymeropoulos, M. and Soares, M.B. (1996) The I.M.A.G.E. Consortium: an integrated molecular analysis of genomes and their expression. Genomics, 33, 151-152.

Lindahl, T. and Wood, R.D. (1999) Quality control by DNA repair. Science, 286, 1897-1905.

Lipman, D.J., Wilbur, W.J., Smith, T.F. and Waterman, M.S. (1984) On the statistical significance of nucleic acid similarities. Nucleic Acids. Res., 12, 215-226.

Lipman, D.J. and Pearson, W.R. (1985) Rapid and sensitive protein similarity searches. Science, 227, 1435-1441.

Luna L., Bjørås M., Berdal K.G., Alseth I., Eide L., Rognes T., and Seeberg E. (1999) Genes and enzymes for the removal of oxidative DNA base damage. In Molecular biology of aging. Bohr, V. A., Clark, B. F. C., and Stevnsner, T. (eds.) Alfred Benzon Symposium, Copenhagen, 44, 272-283.

Luna, L., Bjørås, M., Hoff, E., Rognes, T. and Seeberg, E. (2000) Cell-cycle regulation, intracellular sorting and induced overexpression of the human NTH1 DNA glycosylase involved in removal of formamidopyrimidine residues from DNA. Mut. Res. (in press)

Maizel, J.V.J. and Lenk, R.P. (1981) Enhanced graphic matrix analysis of nucleic acid and protein sequences. Proc. Natl. Acad. Sci U.S.A., 78, 7665-7669.

Miller, W. and Myers, E.W. (1988) Sequence comparison with concave weighting functions. Bull. Math. Biol., 50, 97-120.

Mott, R. (1992) Maximum-likelihood-estimation of the statistical distribution of Smith-Waterman local sequence similarity scores. Bull. Math. Biol., 54, 59-75.

Mott, R. (1999) Local sequence alignments with monotonic gap penalties. Bioinformatics, 15, 455-462.

Mott, R. and Tribe, R. (1999) Approximate statistics of gapped alignments. J. Comput. Biol., 6, 91-112.

Moult, J. (1999) Predicting protein three-dimensional structure. Curr. Opin. Biotechnol., 10, 583-588.

Murzin, A.G., Brenner, S.E., Hubbard, T. and Chothia, C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol., 247, 536-540.

Myers, E.W. and Miller, W. (1988) Optimal alignments in linear space. Comput Appl. Biosci., 4, 11-17.

Nakabeppu et al (1984). Structure and expression of the alkA gene of Escherichia coli involved in adaptive response to alkylating agents. J. Biol. Chem. 259, 13730-13736.

Nakai, K. and Kanehisa, M. (1992) A knowledge base for predicting protein localization sites in eukaryotic cells. Genomics, 14, 897-911.

Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol., 48, 443-453.

Nelson, K.E., Clayton, R.A., Gill, S.R., Gwinn, M.L., Dodson, R.J., Haft, D.H., Hickey, E.K., Peterson, J.D., Nelson, W.C., Ketchum, K.A., McDonald, L., Utterback, T.R., Malek, J.A., Linher, K.D., Garrett, M.M., Stewart, A.M., Cotton, M.D., Pratt, M.S., Phillips, C.A., Richardson, D., Heidelberg, J., Sutton, G.G., Fleischmann, R.D., Eisen, J.A. and Fraser, C.M. (1999) Evidence for lateral gene transfer between Archaea and bacteria from genome sequence of *Thermotoga maritima*. Nature, 399, 323-329.

Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B. and Thornton, J.M. (1997) CATH - a hierarchic classification of protein domain structures. Structure, 5, 1093-1108.

Paracel Inc. (2000) Fast Data Finder (FDF) and GeneMatcher. http://www.paracel.com/

Park, J., Teichmann, S.A., Hubbard, T. and Chothia, C. (1997) Intermediate sequences increase the detection of homology between sequences. J. Mol. Biol., 273, 349-354.

Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T. and Chothia, C. (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. J. Mol. Biol., 284, 1201-1210.

Parkhill, J., Wren, B.W., Mungall, K., Ketley, J.M., Churcher, C., Basham, D., Chillingworth, T., Davies, R.M., Feltwell, T., Holroyd, S., Jagels, K., Karlyshev, A.V., Moule, S., Pallen, M.J., Penn, C.W., Quail, M.A., Rajandream, M.A., Rutherford, K.M., van Vliet, A.H., Whitehead, S. and Barrell, B.G. (2000) The genome sequence of the food-borne pathogen *Campylobacter jejuni* reveals hypervariable sequences. Nature, 403, 665-668.

Parkhill, J., Achtman, M., James, K.D., Bentley, S.D., Churcher, C., Klee, S.R., Morelli, G., Basham, D., Brown, D., Chillingworth, T., Davies, R.M., Davis, P., Devlin, K., Feltwell, T., Hamlin, N., Holroyd, S., Jagels, K., Leather, S., Moule, S., Mungall, K., Quail, M.A., Rajandream, M.A., Rutherford, K.M., Simmonds, M., Skelton, J. and Whitehead, S. (2000) Complete DNA sequence of a serogroup A strain of *Neisseria meningitidis* Z2491. Nature, 404, 502-506.

Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. U.S.A., 85, 2444-2448.

Pearson, W.R. (1990) Rapid and sensitive sequence comparison with FASTP and FASTA. Methods Enzymol., 183:63-98, 63-98.

Pearson, W.R. (1991) Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. Genomics, 11, 635-650.

Pearson, W.R. (1995) Comparison of methods for searching protein sequence databases. Protein Sci., 4, 1145-1160.

Pearson, W.R. (1996) Effective protein sequence comparison. Methods Enzymol., 266, 227-258.

Pearson, W.R., Wood, T., Zhang, Z. and Miller, W. (1997) Comparison of DNA sequences with protein sequences. Genomics, 46, 24-36.

Peltola, H., Soderlund, H. and Ukkonen, E. (1986) Algorithms for the search of amino acid patterns in nucleic acid sequences. Nucleic Acids. Res., 14, 99-107.

Read, T.D., Brunham, R.C., Shen, C., Gill, S.R., Heidelberg, J.F., White, O., Hickey, E.K., Peterson, J., Utterback, T., Berry, K., Bass, S., Linher, K., Weidman, J., Khouri, H., Craven, B., Bowman, C., Dodson, R., Gwinn, M., Nelson, W., DeBoy, R., Kolonay, J., McClarty, G., Salzberg, S.L., Eisen, J. and Fraser, C.M. (2000) Genome sequences of *Chlamydia trachomatis* MoPn and *Chlamydia pneumoniae* AR39. Nucleic Acids. Res., 28, 1397-1406.

Rost, B. (1999) Twilight zone of protein sequence alignments. Protein Eng., 12, 85-94.

Salamov, A.A., Suwa, M., Orengo, C.A. and Swindells, M.B. (1999) Combining sensitive database searches with multiple intermediates to detect distant homologues. Protein Eng., 12, 95-100.

Sankoff, D. and Kruskal J.B. (1983) Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Addison-Wesley. Reading, MA, USA.

Sayle, R. (1995) RasMol version 2.6.8.

Sellers P.H. (1974) On the theory and computation of evolutionary distances. SIAM J. Appl. Math., 26, 787-793.

Shpaer, E.G., Robinson, M., Yee, D., Candlin, J.D., Mines, R. and Hunkapiller, T. (1996) Sensitivity and selectivity in protein similarity searches: a comparison of Smith-Waterman in hardware to BLAST and FASTA. Genomics, 38, 179-191.

Singh, R.K., Hoffman, D.L., Tell, S.G. and White, C.T. (1996) BioSCAN: a network sharable computational resource for searching biosequence databases. Comput Appl. Biosci., 12, 191-196.

Smith, D.R., Doucette-Stamm, L.A., Deloughery, C., Lee, H., Dubois, J., Aldredge, T., Bashirzadeh, R., Blakely, D., Cook, R., Gilbert, K., Harrison, D., Hoang, L., Keagle, P., Lumm, W., Pothier, B., Qiu, D., Spadafora, R., Vicaire, R., Wang, Y., Wierzbowski, J., Gibson, R., Jiwani, N., Caruso, A., Bush, D. and Reeve, J.N. (1997) Complete genome sequence of *Methanobacterium thermoautotrophicum* deltaH: functional analysis and comparative genomics. J. Bacteriol., 179, 7135-7155.

Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences. J. Mol. Biol., 147, 195-197.

Smith, T.F., Waterman, M.S. and Fitch, W.M. (1981) Comparative biosequence metrics. J. Mol. Evol., 18, 38-46.

Smith, T.F., Waterman, M.S. and Burks, C. (1985) The statistical distribution of nucleic acid similarities. Nucleic Acids. Res., 13, 645-656.

Solovyev, V.V. and Makarova, K.S. (1993) A novel method of protein sequence classification based on oligopeptide frequency analysis and its application to search for functional sites and to domain localization. Comput. Appl. Biosci., 9, 17-24.

Solovyev, V.V., Salamov, A.A. and Lawrence, C.B. (1994) Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. Nucleic Acids. Res., 22, 5156-5163.

States, D.J. (1993) Information enhancement methods for large-scale sequence analysis. Computers & Chemistry, 17, 191-201.

Stephen, G.A. (1994) String Searching Algorithms. World Scientific Publishing.

Stephens, R.S., Kalman, S., Lammel, C., Fan, J., Marathe, R., Aravind, L., Mitchell, W., Olinger, L., Tatusov, R.L., Zhao, Q., Koonin, E.V. and Davis, R.W. (1998) Genome sequence of an obligate intracellular pathogen of humans: *Chlamydia trachomatis*. Science, 282, 754-759.

Sturrock, S.S. and Collins, J.F. (1993) MPsrch V 1.3 User Guide. Biocomputing Research Unit, University of Edinburgh, UK.

Tateno, Y., Miyazaki, S., Ota, M., Sugawara, H. and Gojobori, T. (2000) DNA data bank of Japan (DDBJ) in collaboration with mass sequencing teams. Nucleic Acids. Res., 28, 24-26.

Tettelin, H., Saunders, N.J., Heidelberg, J., Jeffries, A.C., Nelson, K.E., Eisen, J.A., Ketchum, K.A., Hood, D.W., Peden, J.F., Dodson, R.J., Nelson, W.C., Gwinn, M.L., DeBoy, R., Peterson, J.D., Hickey, E.K., Haft, D.H., Salzberg, S.L., White, O., Fleischmann, R.D., Dougherty, B.A., Mason, T., Ciecko, A., Parksey, D.S. , Blair, E., Cittone, H. and Clark, E.B. (2000) Complete genome sequence of *Neisseria meningitidis* serogroup B strain MC58. Science, 287, 1809-1815.

Thanaraj, T. A. and Flores, T. (1997) Assessment of Smith-Waterman sequence search tools implemented in Bioccelerator, FDF and MasPar. Biostandards Report. European Bioinformatics Institute (Industry Support Programme), Hinxton, Cambridge, UK.

The C.elegans sequencing consortium (1998) Genome sequence of the nematode *C. elegans*: a platform for investigating biology. Science, 282, 2012-2018.

Thompson, J.D., Higgins, D.G. and Gibson, T.J. (1994) Improved sensitivity of profile searches through the use of sequence weights and gap excision. Comput. Appl. Biosci., 10, 19-29.

Thompson, J.D., Plewniak, F. and Poch, O. (1999) A comprehensive comparison of multiple sequence alignment programs. Nucleic.Acids.Res., 27, 2682-2690.

Timelogic Inc. (2000) DeCypher. http://www.timelogic.com/

Tomb, J.F., White, O., Kerlavage, A.R., Clayton, R.A., Sutton, G.G., Fleischmann, R.D., Ketchum, K.A., Klenk, H.P., Gill, S., Dougherty, B.A., Nelson, K., Quackenbush, J., Zhou, L., Kirkness, E.F., Peterson, S., Loftus, B., Richardson, D., Dodson, R., Khalak, H.G., Glodek, A., McKenney, K., Fitzegerald, L.M., Lee, N., Adams, M.D. and Venter, J.C. (1997) The complete genome sequence of the gastric pathogen Helicobacter pylori. Nature, 388, 539-547.

Uberbacher, E.C. and Mural, R.J. (1991) Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. Proc. Natl. Acad. Sci. U.S.A., 88, 11261-11265.

Vingron, M. and Waterman, M.S. (1994) Sequence alignment and penalty choice. Review of concepts, case studies and implications. J. Mol. Biol., 235, 1-12.

Vogt, G. and Argos, P. (1992) Searching for distantly related protein sequences in large databases by parallel processing on a transputer machine. Comput. Appl. Biosci., 8, 49-55.

Waterman, M.S., Smith, T.F. and Beyer, W.A. (1976) Some biological sequence metrics. Adv. Appl. Math., 20, 367-387.

Waterman, M.S. (1984) Efficient sequence alignment algorithms. J. Theor. Biol., 108, 333-337.

Waterman, M.S. and Eggert, M. (1987) A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. J. Mol. Biol., 197, 723-728.

Waterman, M.S. and Vingron, M. (1994) Rapid and accurate estimates of statistical significance for sequence data base searches. Proc. Natl. Acad. Sci. U.S.A., 91, 4625-4628.

Waterman, M.S. and Vingron, M. (1994) Sequence comparison significance and poisson approximation. Stat. Sci., 9, 367-381.

Waterman, M.S. (1995) Introduction to Computational Biology: Maps, sequences and genomes. Interdisciplinary Statistics. Chapman & Hall, London, UK.

White, O., Eisen, J.A., Heidelberg, J.F., Hickey, E.K., Peterson, J.D., Dodson, R.J., Haft, D.H., Gwinn, M.L., Nelson, W.C., Richardson, D.L., Moffat, K.S., Qin, H., Jiang, L., Pamphile, W., Crosby, M., Shen, M., Vamathevan, J.J., Lam, P., McDonald, L., Utterback, T., Zalewski, C., Makarova, K.S., Aravind, L., Daly, M.J. and Fraser, C.M. (1999) Genome sequence of the radioresistant bacterium *Deinococcus radiodurans* R1. Science, 286, 1571-1577.

Wilbur, W.J. and Lipman, D.J. (1983) Rapid similarity searches of nucleic acid and protein data banks. Proc. Natl. Acad. Sci. U.S.A., 80, 726-730.

Wilbur, W.J. and Lipman, D. (1984) The context dependent comparison of biological sequences. SIAM J.Appl. Math., 44, 557-567.

Wootton, J.C. and Federhen, S. (1993) Statistics of local complexity in amino-acid sequences and sequence databases. Computers & Chemistry, 17, 149-163.

Wootton, J.C. and Federhen, S. (1996) Analysis of compositionally biased regions in sequence databases. Methods Enzymol., 266, 554-571.

Wozniak, A. (1997) Using video-oriented instructions to speed up sequence comparison. Comput. Appl. Biosci., 13, 145-150.

Zhang, Z., Pearson, W.R. and Miller, W. (1997) Aligning a DNA sequence with a protein sequence. J. Comput. Biol., 4, 339-349.

Zhang, Z., Berman, P. and Miller, W. (1998) Alignments without low-scoring regions. J. Comput. Biol., 5, 197-210.