2730–2739 *Nucleic Acids Research, 1996, Vol. 24, No. 14*

PairWise and SearchWise: finding the optimal alignment in a simultaneous comparison of a protein profile against all DNA translation frames

Ewan Birney[1,*], Julie D. Thompson and Toby J. Gibson

European Molecular Biology Laboratory, Postfach 102209, Meyerhofstrasse 1, D-69012 Heidelberg, Germany and [1]Balliol College, Oxford OX1 3BJ, UK

Received March 15, 1996; Revised and Accepted May 31, 1996

ABSTRACT

DNA translation frames can be disrupted for several reasons, including: (i) errors in sequence determination; (ii) RNA processing, such as intron removal and guide RNA editing; (iii) less commonly, polymerase frameshifting during transcription or ribosomal frameshifting during translation. Frameshifts frequently confound computational activities involving homologous sequences, such as database searches and inferences on structure, function or phylogeny made from multiple alignments. A dynamic alignment algorithm is reported here which compares a protein profile (a residue scoring matrix for one or more aligned sequences) against the three translation frames of a DNA strand, allowing frameshifting. The algorithm has been incorporated into a new package, WiseTools, for comparison of biological sequences. A protein profile can be compared against either a DNA sequence or a protein sequence. The program PairWise may be used interactively for alignment of any two sequence inputs. SearchWise can perform combinations of searches through DNA or protein databases by a protein profile or DNA sequence. Routine application of the programs has revealed a set of database entries with frameshifts caused by errors in sequence determination.

INTRODUCTION

Comparative analysis of shared characters is undertaken in all biological fields. It is the basic activity which underpins our current understanding of evolutionary processes. As a consequence of the growth of nucleotide and protein sequence databases, a huge expansion has now occurred in the specific area of molecular comparative analysis. This is deployed in the hope of revealing functional residues in protein families, inferring function for new genes by detecting homology with better characterized genes and for establishing phylogenies.

Database searches to extract homologous sequences are at the heart of sequence analysis, hence a variety of methods have been developed and applied in widely available packages or as network servers. In general, there is a trade-off between speed and sensitivity of the algorithms. The quick wordsearch program FASTA (1) and the more recent and even faster BLAST (2) are now the workhorses of database searching. However, because of restrictions on opening gaps, they are found to be less sensitive than the exhaustive but slow Smith–Waterman algorithm (3), which finds a local alignment between two sequences that is mathematically optimal for a given scoring scheme. On current workstations, one can compare a protein sequence against a protein database by Smith–Waterman. However, without a fairly powerful machine, it is impractical to do the most exhaustive search, i.e. against all possible translations of the DNA databases. This is highly desirable, because protein searches are more sensitive than DNA searches, yet the protein databases have consistently under-represented the data in the DNA databases.

Insertion or deletion of one or more bases in a DNA sequence causes shifts between the translation frames (see 4 for a review). The possibility of frameshifts in DNA means that searching all translation frames individually, as do TFASTA and TBLASTN, is suboptimal. The problem is severe for genomic DNA from metazoans, where introns abound: the next exon may be in any frame and an indeterminate distance away. It also arises whenever a base is erroneously inserted or deleted. A number of studies (5–8) have reported that frameshift errors are uncomfortably common. For example, a systematic study of the SWISS-PROT database (9) revealed that at least one in 200 sequences had severe frameshifts (7). The current activity in generating libraries of randomly sequenced short cDNA sequences, known as Expressed Sequenced Tags or ESTs (10), has exacerbated this problem, since single gel readings are unreliable.

Undetected frameshift errors can have dramatic and deleterious consequences for comparative sequence analysis. During multiple alignment, frameshifts cause erroneous INDEL assignment, forcing local misalignment of other sequences. Falsely truncated sequences may lead to domain boundary misassignment. Catalytic residues, normally absolutely conserved, may be erroneously ruled out due to false substitution. Phylogenetic trees may acquire incorrect branching orders and improbable branch lengths.

For all these reasons, we felt that there was a need for a versatile program which could trace segments of amino acid sequence similarity when they were present in more than one translation

* To whom correspondence should be addressed at present address: Sanger Centre, Hinxton Hall, Hinxton, Cambridge CB10 1RQ, UK

frame. Frameshift errors would be revealed early in searches so that they would not degrade subsequent analyses. The application should also be able to use multiple alignment-based protein profiles (reviewed in 11), as well as single sequences.

In this manuscript, we present an algorithm for finding the optimal alignment of a protein sequence or profile against all three DNA translation frames, allowing frameshifts. In addition, the WiseTools program package is described, which allows for the routine application of the algorithm in sequence alignment and database searches. The WiseTools programs perform generalized sequence comparisons among protein sequences, protein profiles and DNA sequences. The use of the programs PairWise and SearchWise for dealing with frameshifts caused either by errors or by introns is then outlined. The utility of these programs is illustrated with a set of newly revealed database entries containing frameshift errors.

MATERIALS AND METHODS

Algorithms

The algorithm for protein–protein comparison is similar to the dynamic programming routines employed by many sequence analysis programs (12,13), being a variant of the Smith–Waterman best local alignment algorithm (3). These algorithms all belong to the class known as minimal string edit algorithms.

To standardize the program operations for comparisons using either a protein sequence or an aligned sequence set, the profile concept (14) is employed. A profile of length *N* is a set of 20 scores, for all possible amino acids for each position 1 to *N*, in a set of one or more aligned protein sequences. Two additional scores per position provide position-specific gap opening (GOP) and gap extension penalties (GEP). Typically, gap penalty reductions are supplied for positions where gaps are already observed (14,15).

The Waterman–Eggert algorithm (16) to extract the top *k* subalignments has also been incorporated into PairWise. This allows the program to report repeated domains.

DNA forward frames. For the comparison against DNA, the protein is back-translated. The concept of a *codon profile* for a protein (or alignment) is introduced. This is a set of 64 scores for all possible codons for each position 1 to *N*, plus the gap penalty and gap extension scores. A dynamic programming matrix is then constructed from a DNA sequence against the codon profile. The scheme in Figure 1 illustrates how the algorithm chooses between in-frame and jumped frame paths. The core of the algorithm is the iterative calculation for the cell in the *i*th position down the profile versus the *j*th position along the DNA sequence (Fig. 1). Each matrix cell has a score and a state which can be either MATCH, PROFILEGAP, SEQGAP or FRAMEGAP. The state for each cell is the appropriate state for the max calculation. The first four expressions of the max are the standard in-frame start, match and two gap calculations, but with an offset of three in the DNA dimension. Other features of this algorithm differ from more standard dynamic matrices. First, the $j-2$, and $j-1$ movements cause frameshifting in the alignment. These frameshifts do not count the shifted bases/codons in the overall score. Second, only one score is calculated per cell, rather than a score for each different state for which the max is then taken. This single score regime prevents the fortuitous stringing together of matching segments with the large frame gaps allowed by the low frame extension penalty required to jump introns. The frame jumping behaviour is controlled by a frame opening penalty (FOP) and a frame extension penalty (FEP). These penalties can be customized depending on the particular alignment task.

The algorithm is straightforwardly applied to unusual genetic codes (mitochondria, certain protozoa and so forth) by supplying the appropriate codon table.

DNA reverse frames. The reverse frame alignment is produced while reading the DNA sequence in the forward frame by inverting the profile to read C→N (rather than N→ C) and then mapping the 20 amino acid scores to the appropriate codon number for the reverse strand.

Although classical sequence comparison algorithms are symmetrical, so that either N→C or C→N alignments have identical scores, profile alignments introduce asymmetry due to locally varied penalties. This loss of symmetry has no intrinsic biological significance. The reverse strand implementation (which proceeds 3′→5′ and C→N) may have a slightly different score and alignment than the forward strand comparison of the complemented sequence. With optimal parameter settings, this difference is always minor and may not be seen, but is clearly observable with improperly reduced parameter settings.

New options for profiles

Gap penalties based on observed INDEL length in multiple alignments. The gap penalties are variable at two points: position-specific relative values are provided with each position in the profiles, while the overall gap parameters are set in the menus. By default, profiles prepared with the PairWise build menu supply local gap penalties varied according to the observed tolerance for insertions and deletions in an alignment. These penalties are suggested for use with globular proteins, where INDEL behaviour can be understood in the light of structural and functional restraints: these penalties might not perform well with other classes of protein. At each INDEL (site of insertion and deletion) the sample variance of the INDEL lengths is obtained, correcting for sequence bias by weighting according to sequence divergence.

$$s_{wtd} = \sqrt{\sum_{i=1}^{n} \frac{SeqWeight_i(Length_i - \overline{Length})^2}{\sum SeqWeights - \overline{SeqWeight}}} \qquad \mathbf{1}$$

The weighted sample variance, together with a measure of the total information in the sequences, is used to lower GOP and GEP at INDELS as follows.

GOP and GEP are set at 100 if there is no INDEL. Otherwise,

$$GOP = GEP = \frac{100}{s_{wtd} * Log(1 + \sum SeqWeights)} \qquad \mathbf{2}$$

but if GOP > 50, it is reset to 50 and if GEP > 100, it is reset to 100.

Although heuristic, these gap penalty settings have the following desirable properties. Gap penalties are high for short INDELs, low for long INDELs and are lowered for alignments with many divergent sequences (where it becomes less likely that gaps will open at novel sites). Importantly, the GEP is not lowered at INDELs, where insertions are both rare and short. For example, single residue gaps frequently correspond to a bulged β-strand, an over- or underwound turn of α-helix or a sequencing error: in each of these cases, it would be wrong to lower the GEP.

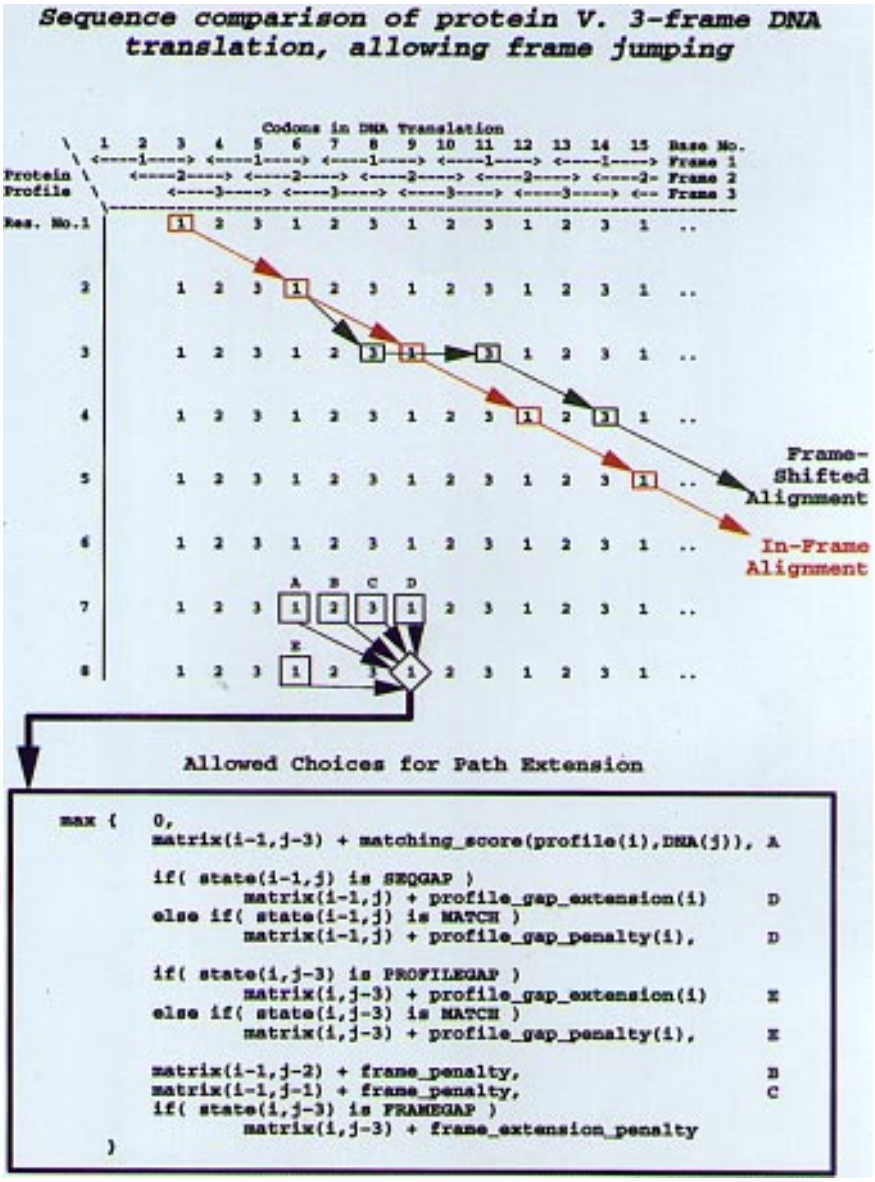

Figure 1. Schematic illustrating the properties of the frameshifting algorithm as applied to a small part of a dynamic matrix. The DNA sequence and the three reading frames are plotted along the horizontal axis and the protein profile down the vertical axis. The matrix positions are shown by their reading frame numbers (the codon comparison scores at the matrix positions are not shown). In red is shown an in-frame path, with no gaps in the alignment, using frame 1. In green is shown a path diverging from the in-frame path by, first, jumping to frame 3 and, second, taking a frame extension step, before resuming the alignment. Blue shows the set of path choices available for selection by the matrix cell marked as a diamond. The blue letters show which choices apply in the algorithm set out below. A and D are in-frame paths, frame jumping occurs at B and C, while E may provide either in-frame gap paths or a frame jump extension. Note that the FEP is only applied every third base, as the frame extension must stay within the reading frame.

The profiles also include a suggested setting for the overall gap penalties that works quite well with the automatic INDEL calculation above (for globular protein sequences). The overall values are estimated from the mean range of the amino acid exchange scores per position. This is helpful, as the profile matrix values drop with increasing alignment divergence, as well as being dependent on the given exchange matrix used. However, these suggestions should be treated as a rough guide and the user should still fine tune the overall gap penalties for a given family, e.g. testing 1.5–2-fold higher and lower values. For non-globular sequences, especially with a strong residue composition bias, the defaults will be a poor guide to optimal penalties.

Modified relative mutabilities. The amino acid exchange matrix used to build the profiles in PairWise may be normalized so that all self-scores are the same. Each position in the matrix is normalized.

$$Normalised\_value_{i,j} = value_{i,j} * \frac{mean\_of\_identity\_scores}{\sqrt{identity\_score_i * identity\_score_j}} \quad \mathbf{3}$$

Differences in the identity scores provide a measure of the relative mutability of each amino acid. However, given a multiple alignment, comparing the columns shows which amino acids are conserved and which not: the column mutabilities are in conflict

with the relative mutabilities. Therefore, profiles for use in alignment and for dotplots with PROPLOT (15) perform better with this normalization. However, the normalization does not improve database searches: in this case the normalization would introduce noise by biasing in favour of those amino acids that are both frequent and mutable, such as asparagine, while penalizing those that are rare and poorly mutable, such as tryptophan. Also, the normalization should not be applied to single sequences, where there is no column mutability information. This normalization is similar to one applied earlier (17) to the Dayhoff PAM 250 matrix (18).

The WiseTools package

The name of the package and programs reflects the concept of generalized *pairwise* comparisons between proteins, alignments and DNA translations. The package currently consists of two components: PairWise for interactive sequence and profile comparisons and SearchWise for database searches. SearchWise is actually two interlinked programs, the SearchWise menus program for parameter set up, which submits the actual database search program SWise to a batch queue. Note that the package provides no new tools for comparing nucleotide sequences against each other and this is not currently supported.

PairWise. PairWise is an interactive, menu-driven program for aligning a protein profile (which may be a single sequence) against a protein or DNA sequence. The program can also find repeats using the Waterman–Eggert algorithm (16). Online help is available in all menus. In the MAIN menu, a sequence and profile are read in and sequence ranges can be set. Moving to the ALIGNMENT menu, there are options for the gap, frameshift and stop codon penalty settings, screen or file output, the number of top alignments to be shown and choice of DNA strand to perform the alignment. In the CONFIG menu, there are options to change the residue substitution matrix and codon table and to choose between default parameter settings for genomic, cDNA or EST, for which different frameshift penalties are appropriate (Table 1).

A BUILD menu allows protein alignments to be used to build new profiles essentially as for the PROFILEWEIGHT program (15), but with two new options. Profiles built in the BUILD menu can be immediately used for alignment. By default, profiles are calculated with the BLOSUM62 matrix (19), sequence weighting and automatic gap penalty reduction based on INDEL variability. Submenus in BUILD allow parameters to be modified. The MATRIX menu allows the amino acid exchange matrix to be varied and, if appropriate, to be normalized for relative mutability. The WEIGHTING menu allows a choice between several weighting schemes or none. The GAP PENALTY menu allows the type of gap penalty at INDELs to be varied, as well as how to treat end gaps in alignments.

PairWise can be linked at compile time to GCG8 (20), whereupon it can extract sequences directly from GCG databases. Where the GCG package is used as the main database management facility, PairWise can be used as a closely integrated tool.

The SearchWise front end. SearchWise provides a menu-driven front end for managing job submission to batch queues (or UNIX background). The menus are aimed at simplifying the use of SearchWise for the occasional computer user. The job parameters, sequence, database, penalties, desired outputs, etc., are set up in menu options, then the batch job is submitted to the specified queue. Defaults for all parameters are read from file, so that the minimum input to initiate a search are a sequence and a database. SearchWise is only appropriate for machines with batch queues or with background operation, e.g. OpenVMS and UNIX. Online help is provided in each menu.

The SearchWise search program SWise. SWise is a command line-driven program to perform database searches which would normally be run on batch queues or in background. It can be submitted from the SearchWise menus or in an edited script or command file. The command line options are listed when the user simply gives the command SWise. SWise allows various permutations of search sequence and database.

Query	Database	
	DNA	PROTEIN
DNA Seq	N	Y
PROTEIN Seq	Y	Y
PROTEIN Profile	Y	Y

Table 1. Penalty set-ups available in the PairWise Config menu

Penalty	Single sequence	Profile from a multiple alignment		
	Default starting set-up	Eukaryotic genomic DNA	Bacterial genomic or cDNA	High error (ESTs etc.)
GOP	1000	1000	1000	1000
GEP	100	100	100	100
FOP	1200	*750*	*850*	*600*
FEP	2	1 (or 0)	*600*	*200*
Stop codon	500	500	500	150

These settings are good starting points for use with single sequences and the BLOSUM62 matrix or with prepared profiles. However, settings may need to be varied for the given sequence or profile. For sequences, settings will not be optimal for other matrices, such as BLOSUM45 or Gonnet PAM250. Optimal settings for profiles will depend on sequence divergence and INDEL gap policy and are best calibrated by trial and error. For profiles, note that in-frame GOP at pre-exisiting INDELs is correctly set lower than the FOP. The FEP will need to be set to 0 for the very large introns which can occur in vertebrates, but this will introduce an element of noise. Therefore, for genomes such as yeast or *C.elegans*, where the introns are shorter, FEP in the range 1–5 should be optimal. The italicized frame penalties are automatically scaled by the profile GOP.

SWise output consists of one obligatory file, the high score list, and two optional files, the corresponding top alignments and a list of HTML links to the ENTREZ WWW facility (21). The latter allows the user to peruse the entries and take advantage of further links in exploring the hits.

WiseTools programming, distribution and information. WiseTools is written in ANSI C as a series of modules linked by front end programs (also ANSI C), hence it is in principle portable to any suitable hardware platform. WiseTools programs have been run on the following platforms: DEC alpha and OpenVMS v. 6.1; DEC Vax 3000 and OpenVMS v. 6.1; DEC alpha and OSF/1, SGI and IRIX 5.2; Sun and Solaris 2.3. Binaries are provided for these platforms. For the alignments, at least 32 MB of main memory should be available. Mac and PC versions are considered for the future, but are not currently supported (multitasking is highly desirable). SearchWise can read the following databases files: GCG binary .seq files, GCG ASCII .seq files, Fasta format and EMBL/SWISS-PROT .dat format.

The C files for WiseTools v. 1.5 are available via anonymous ftp to nmrz.ocms.ox.ac.uk in the directories /pub/wise. A comprehensive help (including installation instructions) are provided on the WWW at http://www.sanger.ac.uk/~birney/wise/topwise.html.

Profile preparation and application

Profiles for the PH domain, PHD finger and RNP domain were prepared with the PairWise build menus using alignments based on those reported (22–24). The Gonnet PAM250 matrix (25) was used together with sequence weighting (15) and default INDEL penalties. The recommended gap penalty settings were then used for preparing the alignment figures. See Bork and Gibson (11) for some guidelines on residue exchange matrix choice and parameter set-up in profile searches.

RESULTS

Detection of amino acid similarity in DNA sequence databases

Genomic DNA from eukaryotes. Most genes from multicellular organisms are interrupted by several, often many, introns. These are less common in simpler eukaryotes, such as yeast (although here they have been consistently under-reported). Some 20 million bases of genomic DNA from *Candida elegans* alone have now been deposited in the EMBL database and most genes are heavily spliced (26). The optimal settings for penalties are FOP > GOP, but FEP < GEP (if necessary, set FEP = ZERO for organisms with long introns) as illustrated in Table 1. With these settings, the alignment stays in-frame for moderate in-frame INDELS, yet it can jump frame and extend an arbitrary distance to the next exon, almost regardless of intron length. Figure 2 shows a profile, made with the build menu from the collection of RNP domains (24), aligned by PairWise to the first RNP domain of the human hn*RNPA1* gene, which is split by an intron. The score for the individual exons are 4590 and 4969, compared with the score for the whole alignment, 8491. The latter score, but not the subscores, allows the sequence to be detected in a database search. Note that the algorithm jumps near to, but not exactly at, the splice junctions in the RNP domain. The program currently has no intrinsic knowledge of splice junctions. Therefore, the splice sites should be verified by reference to splice consensi for the relevant organism. This can be done by inspection or algorithmically, e.g. with the Staden analysis package (27), GRAIL (28), GeneParser (29), etc.

Genomic DNA from prokaryotes and cDNAs. Both cDNA and prokaryotic genomic DNA are normally expected to lack RNA-spliced introns and to have lowish frameshift error rates. Therefore, the FOP should be set higher than the GOP, while the FEP should be substantial, as shown in Table 1, since long frame extensions are meaningless in this context. Examples of error detection in cDNAs are given below.

Expressed sequence tags. EST databases (10) are proving very useful for researchers interested in gene families: for example over half of all known SM proteins were first detected as ESTs (30). ESTs are generally short (<300–400 bases), while the error rate is very high (if variable between projects) and can include double ligations and bacterial or yeast contamination. In a test search with the PH profile (22), 42% of ESTs were frameshifted at least once. EST databases should be searched separately, since the scores are usually only for fragments of sequences, and hence low, while the parameter settings should be set to high error, as in Table 1. FOP is set low, while FEP is raised, favouring genuine frameshift errors.

Detection of frameshifted sequences by homologous comparison

Frameshift errors are usually detected by comparing homologous sequences with each other. With PairWise, a DNA sequence is compared to the protein sequences of homologues. If one sequence consistently jumps frame in a particular region when compared to the related sequences, majority rule assigns it to be the guilty sequence. This verdict should be safe when aligning proteins with >50% identity, but should be issued with caution when comparing highly divergent homologues (e.g. <25% identity): in such cases, short random matches in other frames may occasionally score higher than correct but low scoring sequences. Therefore, it is particularly important not to set the FOP too softly. For highly divergent proteins, a profile prepared from the rest of the family is a more reliable probe for frameshifting than a straight sequence comparison.

The routine application of SearchWise in database screens for domains and proteins of interest at EMBL has resulted in the detection *en passant* of a number of frameshifted entries, almost all of which can be ascribed to sequencing error. Figure 3 shows a PairWise comparison of two closely related G2-specific cyclins B from starfish species. For much of the cyclin box, the sequences are >90% identical. However, the C-terminus in particular makes multiple frameshifts and there are at least 13 sites of nucleotide insertion/deletion, probably rather more. By comparing both sequences to other cyclins B, the EMBL entry APCYCLI (31) has all the sequencing errors. Before we noticed the frameshifts, this entry caused us severe problems. Not only did it hinder our attempts to align the cyclin family, but, until its removal from the search set, the introduced noise precluded profile searches from revealing the cyclin/TFIIB/RB multidomain family (32).

Table 2 summarizes a set of 28 frameshifted entries which were found during routine searches with protein families of interest to us or our colleagues. These frameshifts are now annotated in the appropriate SWISS-PROT entries. In the case of the human *VAV*

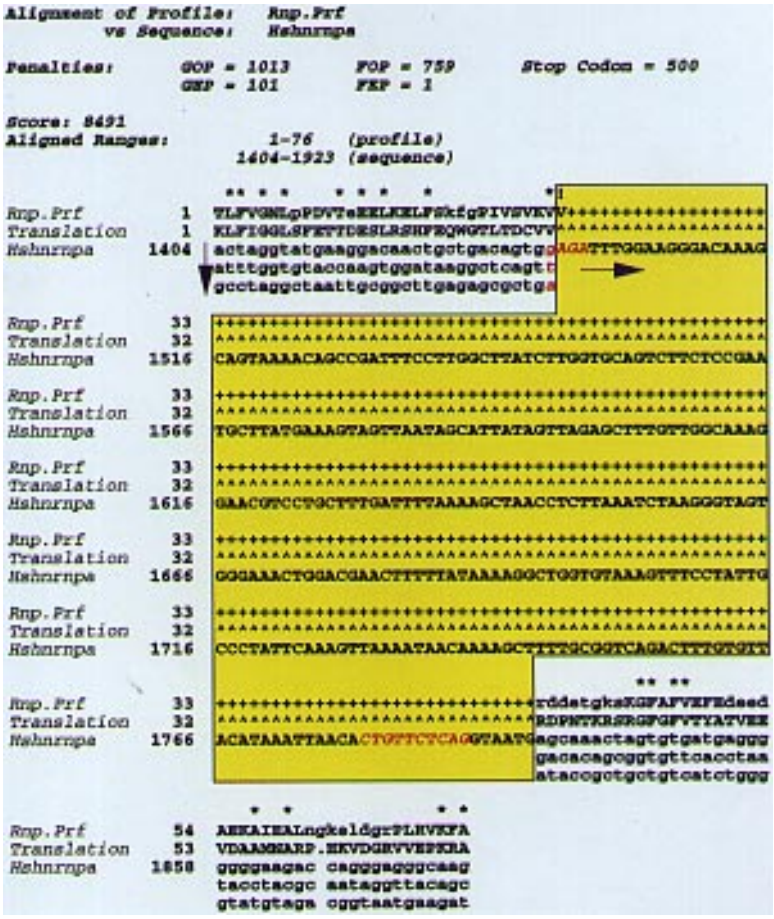

Figure 2. PairWise alignment of the RNP domain profile against human hnRNP_A1 genomic DNA (accession no. X12671). Using the recommended gap penalties configured with the genomic DNA frameshift settings, the profile has successfully jumped the intron which splits the RNP domain. A yellow box encloses the untranslated DNA spanned by the frame jump. The actual splice junction motifs are in red italic letters. The ends of the frame jump are within three and six residues of the splice points. WiseTools output alignments are formatted to clearly distinguish between translated and jumped segments. Untranslated nucleotides within the frame jump are in upper case and capped by & symbols. The site where the frame jump opens is denoted by the symbols. The translated DNA is presented beneath inferred amino acids as lower case codons which are read from top to bottom. Blue arrows indicate the reading direction for translated and jumped DNA. The RNP profile is represented by consensus residues, using lower case at INDELs in the RNP alignment.

oncogene (33), a frameshift had earlier been reported in the DAG domain (34) and was subsequently corrected. The additional frameshift reported here, in the cdc24-like domain, has been independently identified and corrected. The human VAV oncoprotein now agrees well with the mouse sequence (35). These frameshifts were largely responsible for the delay in recognizing VAV as an intracellular signalling protein.

Detection of repeats using profiles

The Waterman–Eggert algorithm (16) returns the top *k* non-overlapping alignments in a sequence comparison. In a self-comparison of a highly repeated sequence, the algorithm does not yield alignments of the individual repeats, but returns instead alignments between subsets of the repeats. Thus the second best alignment is the set consisting of repeat 1 to repeat $n-1$ aligned with repeats 2 to n, and so on. Therefore, the algorithm reveals the existence of the repeats, but does not return them.

In a comparison of a domain profile against a sequence containing multiple domains, the algorithm works well. In this case, the top *k* non-overlapping alignments should correspond to the top *k* individual repeats (except in extremely awkward cases, such as very long insertions). This facility in PairWise was used extensively in the analysis of the PHD finger, which often occurs multiply in a protein sequence (23). Figure 4 illustrates the four PHD fingers in *Drosophila* Trx protein as detected by PairWise comparison with a PHD finger profile. PairWise can detect these repeats equally well in the Trx DNA.

DISCUSSION

We have outlined the development and application of a general purpose sequence comparison package, WiseTools, that is suitable for a range of comparative analyses using amino acid sequence information, whether this is in the form of protein sequence, protein multiple alignment or encoded in DNA. We now discuss some general issues arising from this work.

Improved search sensitivity in DNA database searches

SearchWise has been applied in profile searches with several protein domain families (22,23,36–38). The ability to detect

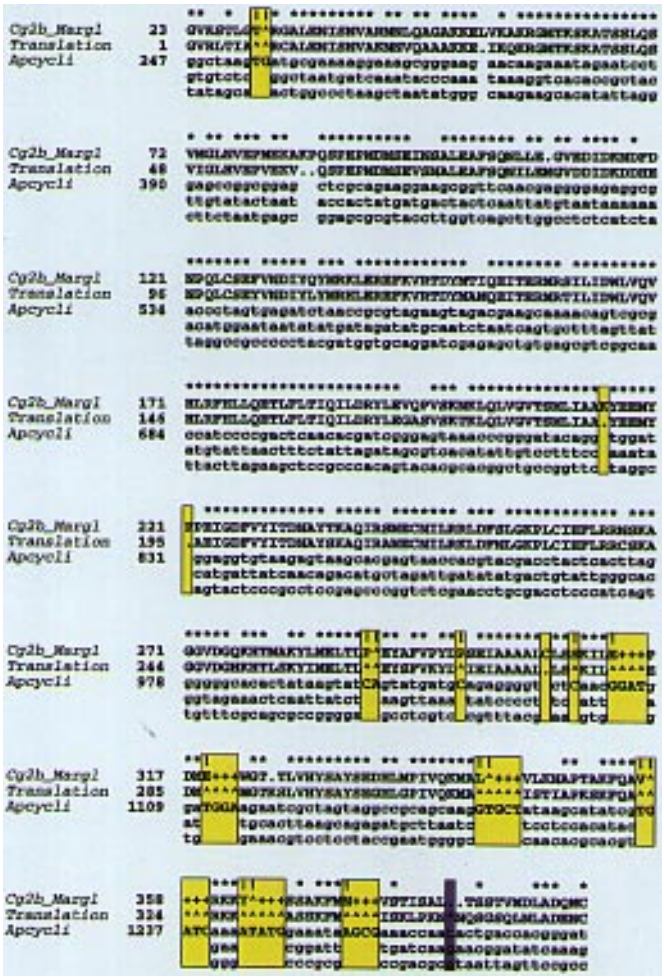

Figure 3. Example of multiple frameshift errors in a database entry. PairWise output for two starfish cyclin sequences. The SWISS-PROT entry CG2B_MARGL is compared with three frames of the EMBL database entry APCYCLI (31). Yellow boxes mark 13 sites of base insertion or deletion, including some 3 bp deletions which maintain the reading frame: due to the high sequence similarity; the frame jumps are mostly at the exact sites of error. A further error causing a premature stop codon is indicated by the purple box. Penalty settings were: GOP = 1000, GEP = 100, FOP = 800, FEP = 100, Stop_codon = 500. The low FOP was necessary to pick up the frameshifted tripeptide with the sequence KIL.

domains in newly submitted DNA entries helped to provide up-to-date lists of the domains, including domains detected in unannotated regions of DNA sequences. The frameshifting capability allowed a number of frameshifted ESTs to be detected, while PairWise comparisons of problematic sequences alerted us to a number of frameshifted cDNA entries (Table 2).

Searching protein databases with DNA sequences

SearchWise allows a DNA query to be compared to a protein database. This facility should be useful for elucidating the coding contents of cDNAs (including ESTs) and short regions of genomic DNA. As well as revealing similarities to encoded proteins, we anticipate that routine application with newly generated sequence data would be useful in revealing frameshifts before the sequences reach the databases. Nevertheless, this is a last line of defence, not a panacea: it cannot replace proper and conscientious sequence determination, most particularly the full determination of a sequence on both DNA strands.

Other frameshift detection methods

The aim of the WiseTools package is to provide a general purpose and sensitive approach to routine sequence comparison, which will reveal frameshifts as they occur in matching sequences. Other methods are available which may do some of these tasks comparably or do them faster but with less sensitivity. For example, standard database searches with both TBLASTN and (to a lesser extent) TFASTA are capable of detecting obvious high scoring frameshifts. In a complementary approach to error detection by homology, there are several programs that use coding preference statistics to assign likely reading frames.

Claverie (7) has developed special substitution matrices representing frameshifted codons for protein–protein comparison using BLAST. This is a very quick method to detect relatively long frameshifts in protein databases. The speed of BLASTP enabled this approach to be used in an exhaustive search for frameshifts in the SWISS-PROT database.

The DETECT program of Posfai and Roberts (6) does a fast pattern search over all reading frames separately. The method is sensitive to fairly short frameshifts with high identity or longer frameshifts with lower identity and is applicable with introns. New DNA sequence can be submitted as the query against a protein database.

States and Botstein (5) have developed a method for pairwise comparison in which a frameshifting Smith–Waterman algorithm uses probability tables for bases in each codon position. Recently, Guan and Uberbacher (39) have implemented a jumping, three frame Smith–Waterman comparison. These methods should be comparably sensitive to PairWise in detection of short frameshifted regions, as in the severe multiple frameshifts of the cyclin in Figure 3. They are not designed for intron-containing genomic sequences, having no frame extension penalty.

Several methods have been developed which use codon preferences to predict translation frames, providing independent means of detecting frameshifts. The Staden package has for some time provided a useful graphical representation of reading frame codon preference (40) which we have used for frameshift error analysis (41). GRAIL II (42) applies coding statistics in a dynamic programming routine to find frameshifts. Codon frequency information has also been applied in a wordsearch algorithm (43). Due to the relatively weak statistical signal, these methods are likely to be less sensitive than WiseTools comparisons under many conditions, but do not require a homologous sequence. Therefore, they provide independent ways to analyse frameshifts in situations where WiseTools cannot be applied or provides ambiguous output.

What is the frequency of frameshift errors in databases?

Two studies have come up with estimates of the frameshift error frequencies in database entries by comparison of homologous sequences. Claverie (7) reported a frequency of ≥0.5% frameshifts in SWISS-PROT v. 24 (28 154 sequences). Posfai and Roberts (6) found 156 problems in a set of 6000 bacterial unidentified ORFs which were thought likely to be of below average reliability.

We intend to apply WiseTools in a systematic analysis of frameshifts. So far we have found that virtually every protein

Table 2. Frameshifted database entries

SWISS-PROT		EMBL		Notes on	Revealed by
Entry	Accession	Entry	Accession	Regions affected[&]	Comparison with
LEU2_SALTY*	P15717	STLEUC	X51476	66-72,232-238	LEU2_ECOLI
LEU2_RHIRA	P17279	MCIPIA	M33166	1-3, 582-712	LEUD_ECOLI
LEU2_PHYBL	P18250	PBLEU1	X53090	C-TER 671-709	LEUD_ECOLI
GAL1_STRLI*	P13277	SLGALO	M18953	49-63,181-244,289-330	GAL1_ECOLI
GAL7_STRLI*	P13217	"	"	59-75,307-354	GAL7_ECOLI
GALE_STRLI*	P13226	"	"	41-64,178-184,218-243,317-329	GALE_STRTR
-		ML22181	U22181	DNA:310-340, 620-780	END3_ECOLI
URE1_UREUR	P17272	UUUREASE	X51315	C-TER 553-572	URE1_BACPA
URE2_UREUR	P17273	"	"	C-TER 113-125	URE2_BACPA
VAV_HUMAN*	P15948	HSVAVPO	X16316	322-355	VAV_MOUSE
CG2B_ASTPE*	P18063	APCYCLI	M33880	>13 SHIFTS	CG2B_MARGL
-		SLMA52	X73879	4 shifts in DNA: 2730-2830	CG2B_HUMAN
SWH1_YEAST*	P35845	-		Extensive entry problems	EM:SCSWH1
IL6A_RAT*	P22273	RRIL6R	M58587	227-261	IL6A_MOUSE
TRFE_XENLA*	P20223	XLTRSFER	X54530	66-72,232-238	TRFE_HUMAN
SPCA_HUMAN*	P02549	HSSPTA01	M61877	2407-2418	SPCA_DROME
JAK1_CYPCA*	N/A	CCFJAK	L24895	66-72,232-238	JAK1_HUMAN
-		MMJAK3H	L33768	7 SHIFTS & partially spliced?	EM:RNJAK3
STA5_SHEEP	P42231	OAMGP	X784428	92-106,259-270,770-780	STA5_MOUSE
COLE_LEPMA*	P98085	LM17431	U17431	391-418	CA1A_CHICK
HRX_HUMAN*	Q03164	HSHRX	L04284	317-379	EM:MMALL1A
-		MITHMF5	X75329	DNA:50-90	EM:CS3KCT
-		RSLAACT	X78116	DNA:<1-70,190-270,1030-1080	THIL_HUMAN
YHL6_YEAST[$]	P38781	SCH8025[$]	U00061	Gal4 domain precedes YHR056C	Gal4 Profile
-		EGANTA	M96564	C-TER SH3 domain	EM:MMH74GENE
-		EMANTB	M96565	C-TER SH3 domain	"
-		GG111	U00111	760-770 & codon drop	EM:GGP50B
-		HSSPRMTK	L08961	N-ter missing, 2 internal shifts, partially spliced?	EM:HS08023

*Annotated or corrected in later SWISS-PROT release.
[&]Where appropriate, numbered according to revised protein sequence in future SWISS-PROT release.
[$]DNA corrected in future release.

family we look at throws up problem sequences, for example, two frameshifts in Jak kinases, one in axl-like RPTKs (44), one in spectrins and one in each of the galactose utilization enzymes, which are all small database groupings. From the average size of the various families in Table 2, it appears that problems are occurring in at least one in 20 protein sequences, which implies that SWISS-PROT v. 32 has ~2000 frameshifted entries. While this is hopefully an upper limit due to sampling bias in Table 2, it highlights the severity of the problem in protein databases.

The majority of the sequences in SWISS-PROT come from small scale, targeted gene sequencing. However, some of the large scale sequencing projects have been shown to have high error rates. In particular, substantial problems were found for the first sequenced yeast chromosome, the partially sequenced *Escherichia coli* chromosome and recently for the *Haemophilus influenzae* genome (45). Therefore, the error rate in some genomic DNA sequence projects may be even higher than for the derived protein databases.

Future directions

There are several clear improvements which could enhance the usefulness of the software as described here. Performance improvements would be gained by incorporating memory-efficient alignment into PairWise and by parallelization of SearchWise.

A frequent suggestion is to lower the FOP at splice recognition sequences. So far, we have avoided taking this step for the

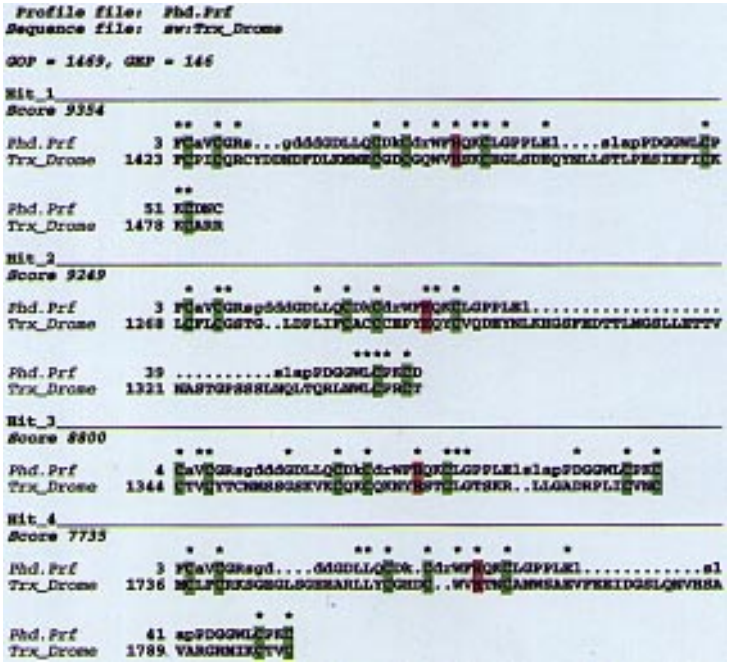

Figure 4. Example of repeat detection using the Waterman–Eggert algorithm in PairWise in conjunction with a domain profile. The four top scoring, non-overlapping alignments when comparing the PHD finger profile against the SWISS-PROT entry TRX_Drome (accession no. P20659) are all PHD fingers. Putative Zn^{2+} ligand residues critical to the domain are coloured green for Cys and pink for His.

following reasons. Since splice site consensi vary between organisms, and even between cellular differentiation states, it is impossible to conduct comprehensive database searches with accurate frame opening costs at splice sites. Furthermore, in organisms such as yeast, the splice junction validity is contingent upon a third motif within the intron, at the branch site. The latter motif, based on the hexamer ACTAAC, is much more highly conserved than the splice junctions themselves. Clearly, the logic needed to accurately pre-screen sequences for splice junctions would add a considerable computational load. Nevertheless, penalty reduction with appropriate splice consensi would be very useful for genomic sequence analysis with PairWise.

A desirable upgrade for SearchWise would be a capability to search libraries of protein family profiles. Searching calibrated profile databases provides a fast and extremely sensitive way to describe the protein and domain classes in a sequence. (For example, see the experimental ProfileScan server at ISREC - http://ulrec3.unil.ch/.) Adding this facility would be timely, since future releases of the PRO-SITE database (46) will supply profile matrices.

ACKNOWLEDGEMENTS

The ideas and programs presented here benefited greatly from discussions and tests involving many colleagues. We particularly wish to thank Adrian Krainer, Des Higgins, Rein Aasland, Kay Hoffman, Liz Cowe, Jasper Rees, Thure Etzold and everyone on the WiseTools mailing list. We also thank Rolf Apweiler and Amos Bairoch for prompt annotation of frameshifted SWISS-PROT entries. Finally we thank Kevin Leonard for supporting this project.

REFERENCES

1 Pearson,W.R. and Lipman,D.J. (1988). *Proc. Natl. Acad. Sci. USA*, **85**, 2444–2448.
2 Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) *J. Mol. Biol.*, **215**, 403–410.
3 Smith,T.F. and Waterman,M.S. (1981) *Adv. Appl. Math.*, **2**, 482–289.
4 States,D.J. (1992) *Trends Genet.*, **8**, 52–55.
5 States,D.J. and Botstein,D. (1991) *Proc. Natl. Acad. Sci. USA*, **88**, 5518–5522.
6 Posfai,J. and Roberts,R.J. (1992) *Proc. Natl. Acad. Sci. USA*, **89**, 4698–4702.
7 Claverie,J.-M. (1993) *J. Mol. Biol.* **234**, 1140–1157.
8 Beck,S., (1993) *DNA Sequence*, **4**, 215–217.
9 Bairoch,A. and Bsckmann,B. (1994) *Nucleic Acids Res.*, **22**, 3578–3560.
10 Adams,M.D., Kelley,J.M., Gocayne,J.D., Dubnick,M., Polymeropoulos,M.H., Xiao,H., Merril,C.R., Wu,A., Olde,B., Moreno,R.F., Kerlavage,A.R., McCombie,W.R. and Venter,J.C. (1991) *Science*, **252**, 1651–1656.
11 Bork,P. and Gibson,T.J. (1996) *Methods Enzymol.*, **266**, 162–184.
12 Pearson,W.R. and Miller,W. (1992) *Methods Enzymol.*, **210**, 575–601.
13 Searls,D.B. and Murphy,K.P. (1995) *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology.* pp. 341–350, AAAI Press.
14 Gribskov,M., McLachlan,A.D. and Eisenberg,D. (1987) *Proc. Natl. Acad. Sci. USA*, **84**, 4355–4358.
15 Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) *Comput. Applicat. Biosci.*, **10**, 19–29.
16 Waterman,M.S. and Eggert,M. (1987) *J. Mol. Biol.*, **197**, 723–728.
17 Gribskov,M. and Burgess,R.R. (1986) *Nucleic Acids Res.*, **14**, 6745–6763.
18 Dayhoff,M.O., Schwartz,R.M. and Orcutt,B.C. (1978) In Dayhoff,M.O. (ed.), *Atlas of Protein Sequence and Structure.* NBRF, Washington, DC, Vol. 5, Suppl. 3, pp. 345–352.
19 Henikoff,S. and Henikoff,J.G. (1992). *Proc. Natl. Acad. Sci. USA*, **89**, 10915–10919.
20 Devereux,J., Haeberli,P. and Smithies,O. (1984). *Nucleic Acids Res.*, **12**, 387–395.
21 Benson,D.A., Boguski,M., Lipman,D.J. and Ostell,J. (1994) *Nucleic Acids Res.*, **22**, 3441–3444.
22 Gibson,T.J., Birney,E., Hyvönen,M., Musacchio,A. and Saraste,M. (1994) *Trends Biochem. Sci.*, **19**, 349–353.

23 Aasland,R., Gibson,T.J. and Stewart,A.F. (1995) *Trends Biochem. Sci.*, **20**, 56–59.
24 Birney,E., Kumar,S. and Krainer,A.R. (1993) *Nucleic Acids Res.*, **21**, 5803–5816.
25 Benner,S.A., Cohen,M.A. and Gonnet,G. H. (1994) *Protein Engng*, **7**, 1323–1332.
26 Wilson,R., Ainscough,R., Anderson,K., Baynes,C., Berks,M., Bonfield,J., Burton,J., Connell,M., Copsey,T., Cooper,J., Coulson,A., Craxton,M., Dear,S., Du,Z., Durbin,R., Favello,A., Fulton,L., Gardner,A., Green,P., Hawkins,T., Hillier,L., Jier,M., Johnston,L., Jones,M., Kershaw,J., Kirsten,J., Laister,N., Latreille,P., Lightning,J., Lloyd,C., McMurray,A., Mortimore,B., O'Callaghan,M., Parsons,J., Percy,C., Rifken,L., Roopra,A., Saunders,D., Shownkeen,R., Smaldon,N., Smith,A., Sonnhammer,E., Staden,R., Sulston,J., Thierry-Mieg,J., Thomas,K., Vaudin,M., Vaughan,K., Waterston,R., Watson,A., Weinstock,L., Wilkinson-Sproat,J. and Wohldman,P. (1994) *Nature*, **368**, 32–38.
27 Staden,R. (1990) *Methods Enzymol.*, **183**, 193–211.
28 Uberbacher,E.C. and Mural,R.J. (1991) *Proc. Natl. Acad. Sci. USA.*, **88**, 11261–11265.
29 Snyder,E.E. and Stormo,G.D. (1995) *J. Mol. Biol.*, **248**, 1–18.
30 Séraphin,B. (1995) *EMBO J.*, **14**, 2089–2098.
31 Tachibana,K., Ishiura,M., Uchida,T. and Kishimoto,T. (1990) *Dev. Biol.*, **140**, 241–252.
32 Gibson.T.J., Thompson,J.D., Blocker,A. and Kouzarides,T. (1994) *Nucleic Acids Res.*, **22**, 946–952.
33 Katzav,S., Martin-Zanca,D. and Barbacid,M. (1989) *EMBO J.*, **8**, 2283–2290.
34 Boguski,M.S., Bairoch,A., Attwood,T.K. and Michaels,G.S. (1992) *Nature*, **358**, 113.
35 Adams,J.M., Houston,H., Allen,J., Lints,T. and Harvey,R. (1992) *Oncogene*, **7**, 611–618.
36 Aasland,R. and Stewart,A.F. (1995) *Nucleic Acids Res.*, **23**, 3168–3173.
37 Kharrat,A., Macias,M.J., Gibson,T.J., Nilges,M. and Pastore,A. (1995) *EMBO J.*, **14**, 3572–3584.
38 Aasland,R., Gibson,T.J. and Stewart,A.F. (1996) *Trends Biochem. Sci.*, **21**, 87–88.
39 Guan,X. and Uberbacher,E.C. (1996) *Comput. Applicat. Biosci.*, **12**, 31–40.
40 Staden,R. (1984) *Nucleic Acids Res.*, **12**, 551–567.
41 Gibson,T.J. and Higgins,D.G. (1993) *DNA Sequence*, **3**, 333–335.
42 Xu,Y., Mural,R.J. and Uberbacher,E.C. (1995) *Comput. Applicat. Biosci.*, **11**, 117–124.
43 Fichant,G.A. and Quentin,Y. (1995) *Nucleic Acids Res.*, **23**, 2900–2908.
44 Bork,P. (1996) *Science*, **271**, 1431–1432.
45 Tatusov,R.L., Mushegian,A.R., Bork,P., Brown,N.P., Hayes,W.S., Borodovski,M., Rudd,K.E. and Koonin,E.V. (1996) *Curr. Biol.*, **6**, 279–291.
46 Bairoch.A., Bucher,P. and Hofmann,K. (1996) *Nucleic Acids Res.*, **24**, 189–196.