PSST… The Probabilistic Sequence Search Tool

Crispin J. Miller[ϕ] and Teresa K. Attwood[ε]
[ϕ]Kilburn Building, Department of Computer Science,
[ε]Stopford Building, Department of Biochemistry,
The University of Manchester, Oxford Road, Manchester, M13 9PL, United Kingdom.

Abstract

Whole genome comparison and clustering cannot be routinely performed without access to significant resources. If, as expected, repositories continue to grow at the current rate, increasingly large and expensive systems will be required in order to maintain the status quo. The high-proportion of uncharacterised gene-sequences, combined with the fact that the majority of sequence analysis techniques are alignment-based, raises the possibility that alternative approaches might be able to identify relationships that have otherwise been missed. There is a need for alternative ways to predict function.

PSST is an analysis tool with parallels to both pairwise algorithms and multiple motif-based pattern approaches. It is significantly faster than BLAST, and for some families including GPCRs, the tool is more sensitive and selective as well. For others it is worse. This paper describes the algorithm, its implementation, its evaluation against a diverse set of protein families, and discusses the reasons behind its behaviour.

Introduction

Bioinformatics is facing an onslaught of sequence data. Repositories continue to double in size every nine months or so, and the large-scale comparisons associated with genome projects have dramatically increased the complexity of the searches we wish to perform. Databases are already at a size where many experiments, such as clustering and genome-comparison are only possible for those with access to significant IT resources. If, as expected, computers continue to follow Moore's law and double in power every 12-18 months, progressively larger and more expensive systems will be required in order to maintain the status quo. There is a need for faster sequence comparison algorithms.

About 40% of the Open Reading Frames (ORFs) in the draft human genome have yet to be characterised [1,2]. The fact that the majority of sequence analysis techniques are alignment-based raises the possibility that alternative approaches might be able to identify relationships that have otherwise been missed. There is a need for alternative ways to predict function.

Almost all function prediction is done by using tools such as BLAST [3] and FASTA [4] to generate and score local alignments between a query sequence and a repository containing previously annotated entries. High sequence-similarity is used as the basis from which to infer homology (that is, descent from a common ancestor), which in turn is used to infer shared function.

This approach relies on a sufficiently similar entry existing in a reference database of previously characterised sequences, and is limited by the sensitivity of the search tools employed. It can also be compromised by the fact that high similarity does not always arise from homology. Gene duplication events, convergent evolution, and nature's tendency to re-use the same structural elements (modules or domains) in functionally unrelated proteins can all result in strong matches between sequences that are not indicative of common function. The transfer of annotation purely on the basis of statistically significant similarity can be dangerous, and has resulted in many misclassifications that threaten the integrity of our databases [5,6].

Partly in recognition of the need for increased sensitivity and accuracy of function prediction, family-based techniques have been developed. These place sequences in functional groups and then use the combined set to generate a stronger diagnostic signal than a single entry can produce on its own. Two main approaches exist: Those that use profiles (e.g. [7,8,9]) and those that are motif-based (e.g. [10,11,12]). Both rely on multiple sequence alignments to place biologically related residues in correspondence with one another. Profiles represent the entire alignment, and seek to describe which residues are allowed at which positions, which are conserved, and which are degenerate, by, for example, using a Hidden Markov Model (HMM) to encode the alignment as a set of weights in a probabilistic finite-state automaton.

Motif-based techniques identify regions of conservation within a multiple alignment that can be used as a diagnostic family signature. These 'motifs' usually reflect some vital structural or functional role (see, for example, Figure 1), corresponding, as they do, to islands of evolutionary stability in a sea of mutational change.

[ϕ] Corresponding author: crispin@psst.cx

Family-led approaches have resulted in a set of pattern databases, consisting of protein families and associated diagnostic profiles or motif collections. Recently, the most popular of these have been grouped together to form the integrated database, InterPro [14].

One such pattern database, PRINTS, is a collection of protein families and their associated fingerprints – ordered sets of motifs excised from hand-built multiple sequence alignments. PRINTS provided the protein sequences used in the results section of this paper, and is briefly described here.

The current release (version 31) contains 1,550 families, and is the largest manually annotated protein family database in existence. PRINTS entries are placed in a hierarchy such that those at progressively higher levels of the tree correspond to increasingly distant relationships. The topmost level, *clan*, corresponds to sequences for which a common evolutionary origin has been postulated (usually as a consequence of shared structure) but that have no appreciable sequence similarity. A separate class, *domain*, corresponds to sequences that have been grouped because they share a structural motif.

Although the hierarchy in PRINTS arises from evolutionary relationships, by-and-large it reflects functional relationships as well. Entries near the top of the tree correspond to broad functional categories; those further down to increasingly specific activities. Associated with increasing specificity is an increased amount of sequence conservation. For example, the multiple-alignment in Figure 1a shows a subset of the G protein coupled receptor (GPCR) super-family of proteins. GPCRs mediate the cellular response to a diverse set of signalling molecules and stimuli across the cell wall. Their role in cell signalling makes them an important pharmacological target, and, therefore, of major interest to the pharmaceutical industry. GPCRs also form one of the largest protein families in nature – to-date, over 200 functionally distinct receptors have been cloned, and over 1000 sequences or fragments can be found in the SWISS-PROT database [15].

All GPCRs have a similar structure, characterised by seven hydrophobic transmembrane helices arranged in a cylinder, with loops extending on either side of the membrane (Figure 1c). Residues on the extra-cellular side of the membrane bind to specific ligands, or convey a response to a stimulus such as light. Those on the intra-cellular side interact with members of specific G protein sub-families. These G proteins inhibit or activate various effector enzymes or ion-channels.

When GPCRs are grouped together into multiple alignments, such as that shown in Figure 1, these structural considerations can be seen to result in distinct patterns within the sequences. The seven transmembrane helices that are characteristic of all GPCRs result in seven hydrophobic motifs, indicated by the dark grey boxes in the Figure. Sensitivity to different agonists allows the

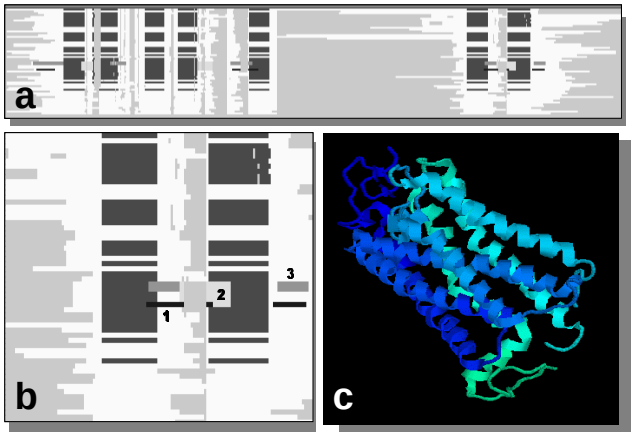

Figure 1. Alignment of the PRINTS GPCR super-family, GPCRRHODOPSN, which includes a diverse set of GPCRS including muscarinic and light sensitive receptors. White horizontal bands correspond to sequences; shaded boxes within the alignment represent PRINTS motifs. (a) The seven dark bands show the diagnostic fingerprint for the entire super-family. Each motif corresponds to one of the seven transmembrane helices in the protein. (b) The last two helices and their neighbouring residues. *1*: Blue sensitive opsin motifs. *2*: Opsin family motifs. *3*: Green sensitive opsin motifs. (c) 3D structure of rhodopsin, taken from PDB [13]. (PDB code 1F88).

GPCRRHODOPSN* super-family to be subdivided, and these families and sub-families can be associated with their own diagnostic motifs (as shown by the boxes in Figure 1c). The motifs labelled with *2* correspond to the OPSIN* family of light-sensitive sequences, *1,3* correspond to blue and green light sensitive sub-families, respectively.

Whilst family based approaches often offer increased sensitivity compared to their pairwise cousins, their coverage is much smaller. This is because in order to use a pairwise technique it is necessary only to have a database of previously characterised sequences. Pattern approaches, by contrast, require these sequences to be further processed – by grouping them into families, generating biologically meaningful alignments and then constructing an appropriate discriminator. Producing alignments is typically the most problematic, because a significant amount of manual 'tweaking' of automatically generated ones is often necessary to make them biologically plausible, and because some relationships are so weak as to defy representation in this way. For example, lipocalins and fatty-acid binding proteins exhibit an almost identical structure, but the only sequence conservation is a `GxW` motif. In order to use patterns as discriminators, it is also necessary to evaluate their performance and to describe the diagnoses they are intended to perform. These validation and annotation steps are also time consuming,

* GPCRRHODOPSN and OPSIN are PRINTS identifier codes.

but are crucial if the patterns are to be used as effective discriminators.

In summary, pairwise algorithms such as BLAST and FASTA can rapidly identify similar sequences within a database, but can only do this with limited sensitivity and selectivity. By contrast, pattern-based approaches offer greater sensitivity and selectivity, but do this at the expense of speed and coverage. An algorithm that could combine the diagnostic power of the pattern-based approaches with the speed and coverage of pairwise techniques would be a useful addition to the sequence analyst's armoury.

The algorithm PSI-BLAST [16], takes a literal approach to this idea. Initially, it uses a BLAST search to compare a query sequence to a database, in order to generate a set of similar sequences. The algorithm then constructs a profile from this initial set, and performs a search using this profile. By combining pairwise and family techniques in this way, PSI-BLAST is able to offer increased sensitivity, but the generalisation that arises during the construction of profiles can reduce the algorithm's selectivity. For example, a hydrophobic region within a query sequence will find other hydrophobic regions within the database. The resultant hit-set and profile, built from many local alignments of these fragments, will be a very good discriminator for sequences containing hydrophobic regions, rather than a specific functional activity. This may or may not be what was desired.

Like motif-based approaches, word-based algorithms look for short conserved regions between sequences, rather than longer, contiguous weak matches. This parallel suggests a number of reasons why word-based approaches might form the basis for an alternative type of comparison tool. Firstly, as we have seen with GPCRs, proteins are three-dimensional structures, produced by folding a one-dimensional polypeptide chain. Often residues that convey function are close to one another in 3D, but are separated by large distances in primary sequence.

Secondly, whilst some residues are 'functional', others perform a structural role – providing a 'scaffold' to place, for example, residues that form a receptor site in the correct 3-dimensional orientation. These 'structural residues' are less constrained by evolution – substituting one for another is often possible, as long as the general size, shape and gross biochemical properties of its parent element do not change too much. This can be seen for the GPCR alignment in Figure 2.

The membrane region is clearly conserved at the biochemical level – it is possible to swap a leucine for a valine, for example – but there is variability amongst the individual residues. By contrast, although the loop region is much more variable over the entire family, within the functionally distinct sub-families, the level of conservation is much higher – both at the residue and biochemical level.

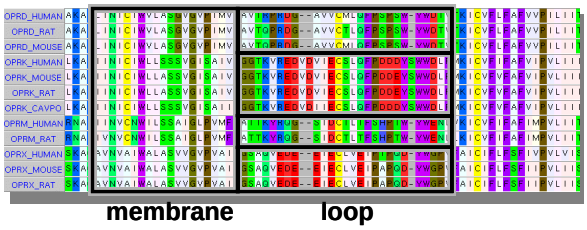

Figure 2. Alignment of opioid-receptors, members of the GPCR super-family. Residues in the transmembrane helix show biochemical conservation, but, even within specific subfamilies, show a certain amount of variation. Residues within the loop region confer specific activity, and are highly conserved within a sub-family – but divergent over the entire class of opioid receptors.

Thirdly, the premise behind *ab initio* structure prediction algorithms, resulting from experiments in ribonuclease folding by Anfinsen [17], is that all the information required by a protein to fold correctly can be found within the sequence. If this is the case, then certain arrangements of amino acids will fold into helices, others into strands, loops and turns. Since proteins consist mainly of helices and strands, the patterns of amino acids that produce these elements must in general be common. This argument, whilst controversial, is supported by the graphs in Figure 3. They were produced by using the DSSP classifications of sequences from PDBFINDER [18] to generate a set of fragments corresponding to loops, strands, helices and turns. Normalised 3-mer frequencies were determined for each of these classes and subtracted from the overall distribution for SWISS-PROT, allowing the relative representation of words in the different classes to be compared. From Figure 3a it can be seen that words over-represented in one class are under-represented in the others (helices and turns are grouped together in the figure because they were found to be correlated). Figure 3b confirms that common words are more likely to be found in helices, strands and turns than in loops. This is an interesting property because it is often the loops that are expressed on the surface of a protein, contain the specific function-giving residues and provide the sequences used for binding and specificity.

Although the arguments outlined here are simplistic – a natural consequence of trying to describe something as complicated as protein function in a way that can be expressed in a sequence comparison algorithm – they do suggest that word-based tools, which find short exact matches between rare sub-sequences, might be good at finding specific functional correspondences between proteins.

Word-based methods are not new. Algorithms such as BLAST and FASTA use an initial word-search to identify promising sequences for alignment, and improve efficiency by using these matches to constrain the alignment algorithm. FLASH [19] generates *k*-tuples that

are used as the input to a geometric hashing algorithm, and others such as STACK_PACK [20], Miropeats [21], EMBLSCAN [22] and RAPID [23] forgo the alignment step altogether, generating a similarity score directly from the *k*-tuple matches.

This paper describes a novel sequence analysis tool, the Probabilistic Sequence Search Tool (PSST) that uses a word-based algorithm to compute similarity. By avoiding alignments PSST is able to compute a similarity score extremely efficiently. It is a natural development of RAPID, generalised for both protein and DNA sequences, and, like RAPID, adds knowledge to the scoring system by using word frequencies determined empirically from sequence data. The hypothesis is that sequences that share rare words are more likely to be biologically related than sequences that share common ones. This is analogous to web search engines and information retrieval tools. If two pages contain rare words such as 'sequence', 'analysis', 'algorithm' and 'k-mer' they are likely to be referring to the same thing, but if they only share common ones such as 'because', 'and', and 'the', they probably aren't [24,25].

PSST, is about two orders of magnitude faster than BLAST, and, for some protein families, including GPCRs, it is significantly more sensitive and selective. For others, it is worse – PSST is not a replacement for existing algorithms. It is, however, a potential new tool in the toolbox.

Algorithm

The algorithm considers similarity to be proportional to the number of words shared between a pair of sequences, weighted by their rarity.

Let *a* be a sequence, of length *l* built from an alphabet of *n* symbols (n=*4* for DNA sequences, *20* for proteins).

A binary vector w^a, of length n^k, represents the presence or absence of each of the possible *k*-mers in *a*, such that if word *i* is present in *a*, $w_i^a=1$, otherwise $w_i^a=0$.

Another vector *p* represents the normalised frequency of each word occurring, such that they sum to 1. This allows the score for a match between two sequences *q* and *t* to be computed:

$$S = \sum_i \frac{1}{p_i} \times w_i^q \times w_i^t$$

The lack of positional information means that the algorithm is unable to distinguish between a cluster of matches in one region of a sequence and the same set of matches distributed over its entire length. Whilst, for short sequences this is unimportant, for large ones it significantly reduces the algorithm's selectivity by allowing chance matches summed over the entire length of the sequence to mask regions of local similarity. For this reason, large sequences are broken into smaller fragments and each fragment treated individually.

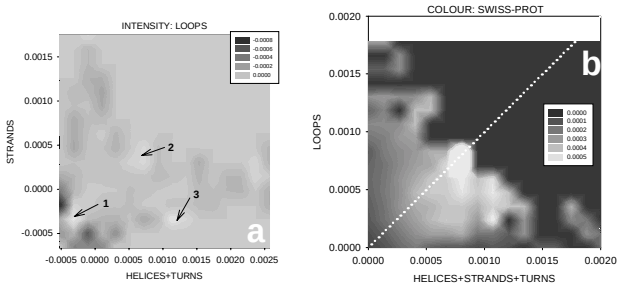

Figure 3. 3mer distributions for helices + turns, strands and loops were computed using data from DSSP and compared to the overall distribution in SWISS-PROT (39). The difference between these distributions is shown using contour plots; the axes show the deviation in the probability distributions from that of SWISS-PROT: **(a) Strands, loops and helices + turns.** The DSSP distributions were subtracted from that of SWISS-PROT in order to give deviations from the overall distribution. Words over-represented in loops tend to be under-represented in strands, helices and turns (peak 1). There is some correlation between loops and turns: this results in peak 3. Peak 2 corresponds to a large set of words that are generally common in all three classes. **(b) 'Scaffolding', loops and SWISS-PROT.** Words that occur in helices, strands or turns tend not to occur in loops, and these words tend to be common in SWISS-PROT. Intensity is inversely proportional to rarity in SWISS-PROT: common words are light, rare ones, dark.

Typically fragments, are 300bp long for proteins, 1000bp for DNA.

Implementation

The design of the algorithm allows an extremely efficient implementation that achieves speed at the expense of memory. This compromise was chosen because memory is cheap compared to processing capacity, and the coarse-grained nature of the task makes it well suited to low-cost cluster computers, built from standard PCs and networking components. The procedure uses two distinct steps:

In the first, the target database is pre-indexed to create an efficient data-structure for searching. This is performed in a single off-line computation. The data structure lists, for each possible *k*-mer, the sequences that contain that word. At the same time, word frequency statistics are also produced.

The second (search) step scans a window across the query sequence to generate its constituent overlapping *k*-mers. These are treated as base 4 or 20 numbers (corresponding to DNA or protein sequences) and used to index into the word lists. This allows all the sequences that contain a given word to be found in constant time. Each word list is scanned, and for each matching sequence, a bin incremented by the appropriate frequency-weighted score. Finally, the bins are iterated over, and the scores outputted. The process is shown diagrammatically in Figure 4.

For single sequence searches, only a subset of the word lists are required. Loading the entire database into memory, in advance, provides a significant start-up cost.

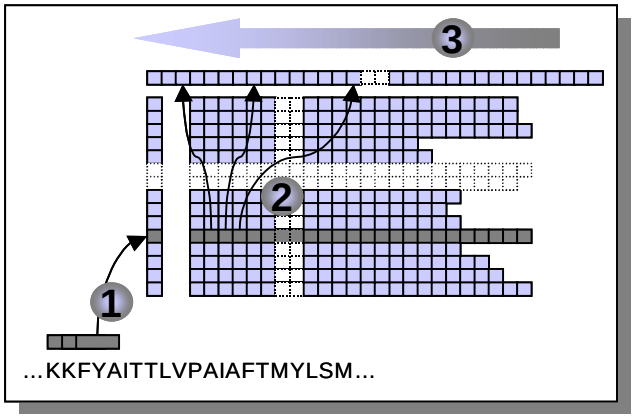

Figure 4. **(1)** Sequences are scanned to produce a set of overlapping words. These are treated as base 4, or base 20, numbers (for DNA or protein sequences, respectively), and used to index into a vector listing each sequence that contains a given word. **(2)** The list is scanned and the presence or absence of a word recorded by indexing a bin. **(3)** once the sequence has been scanned, the bins are read and their score outputted.

In order to avoid this, the pre-indexed data-structure is created as a memory image that is stored on disk. Rather than the software explicitly loading it, it is mapped it into the program's virtual address space. This hands over the task of data loading to the operating system. The first time a word list is accessed, it is not in physical memory. A page-fault is generated and the operating system loads the data for that page directly off disk. This has two consequences. Firstly, the process is performed at the kernel level without additional buffering, making it more efficient. Secondly, only pages that are relevant to the search get loaded at all. As a search progresses, more and more of the database ends up in physical memory.

Parallelisation.

The coarse-grained nature of the task makes it 'embarrassingly parallel'. A fragment of the database can be placed on each node, query sequences distributed, searched, and results returned. This leads to a producer-consumer model, with one node generating word lists and sorting the results from a previous search, whilst the other nodes look for hits against the next sequence.

An interesting issue that arises is load balancing. Sequence data-files contain a significant amount of local structure, resulting in similar sequences – such as a batch-submitted EST library – occurring close to each other in the file.

Placing the first *n* sequences on the first node, the next *n* on the next one, and so on, results in search 'hotspots' occurring. In such situations, one node does most of the work, whilst the others sit idle. To avoid this, adjacent sequences are placed on different nodes.

The program has been implemented in C using MPI. Binaries exist for Linux-Intel and SGI Origin 2000. Efficient operation on cluster computers relies on reducing the amount of inter-process communication to a minimum and on dealing with the high latency associated with cheap interconnects. Asynchronous communication and judicious hacking of data-structures have been used to improve network-efficiency.

A note on *p*-scores

Alignment techniques generally return their similarity score as an *E*- or *p*-value, produced by estimating the likelihood that the given score did not occur by chance. Underpinning any such calculation is the null hypothesis used to describe random patterns. For sequences, the majority of models are derived from the BLAST algorithm [3]. Unfortunately, no such standard models exist for pure word-based techniques, and as a consequence, no well-understood scoring system.

The problems discussed in the introduction also demand caution when assessing the results of similarity searching. The statistical likelihood of a score provides a lower bound beyond which matches cannot be distinguished from noise, and an ordering of matches that can be used to evaluate relative significance. This might be used, with the aforementioned caveats, to infer common function, but success is dependent on whether the underlying similarity metric is appropriate.

The community-wide experience derived from generating millions of BLAST searches has resulted in accepted score-thresholds that are many orders of magnitude more stringent than the statistical model suggests – demonstrating that even with a well-understood scoring system, the biological performance of a search tool may well be different from the model-based predictions. For these reasons, diverse hand-classified sequences were clustered in order to test sensitivity and selectivity, to determine score thresholds and to compare PSST's performance to that of BLAST. Results of this analysis are presented below.

Family identifier	Description	Hierarchy
ALPHAHAEM	Alpha haemoglobin	Family
BETAAMYLOID	Beta-amyloid peptide (beta-APP)	Family
CYTOCHROMEF	Cytochrome F	Family
DPTHRIATOXIN	Diphtheria toxin	Family
EUMOPTERIN	Eukaryotic molybdopterin domain	Family
FANCONICGENE	Fanconi anaemia group C protein	Family
FNTYPEIII	Fibronectin type III repeat	Domain
GLHYDRLASE3	Glycosyl hydrolase family 1	Family
GPCRRHODOPSN	Rhodopsin-like GPCR superfamily	Super-family
HEATSHOCK90	90Kd heat shock protein	Family
KINESINLIGHT	Kinesin light chain	Family
KRINGLE	Kringle domain	Domain
LIPOCALIN	Lipocalin	Super-family
NIHGNASESMLL	Ni-Fe hydrogenase small subunit	Family
OPSIN	Opsin	Family
PHOTOSYSPSAAB	Plant photosystem I psaA and psaB	Family
PRION	Prion protein	Family
RHODOPSIN	Rhodopsin	Sub-family
URICASE	Uricase	Family
ZINCFINGER	C2H2-type zinc finger	Domain

Table 1. Families represented in the PRINTS subset, miniPRINTS.

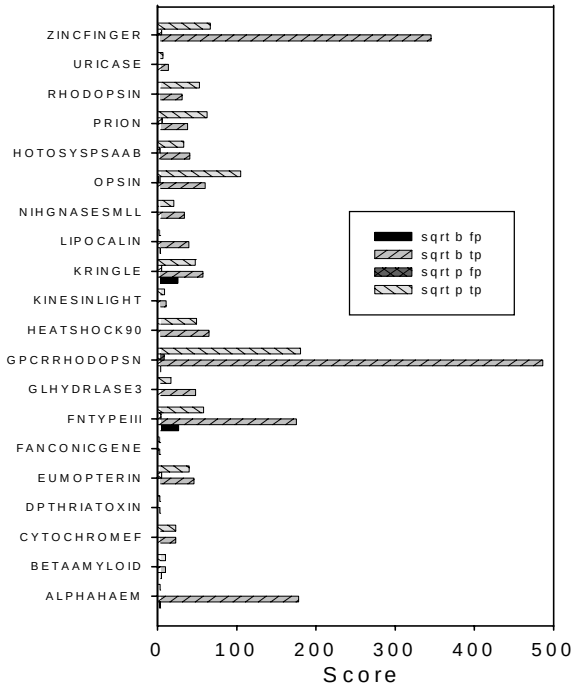

Figure 5. Scores for each of the families in miniPRINTS. The score threshold that minimised the number of misclassifications were selected – for BLAST 10^{-4}, for PSST, 12.8. The graph shows the number of true and false positives arising for each family at the given threshold. Since the comparison was a clustering of N sequences, yielding a possible N^2 hits, the figure plots the square-root of the recorded true and false positives.

Results

Sensitivity and selectivity

The PRINTS database contains a diverse set of hand-classified sequences arranged into a family hierarchy, with separate classes corresponding to domains. These annotations make PRINTS a useful resource with which to evaluate sequence comparison algorithms, and form the basis for the results described here.

Although PRINTS is small in comparison to a repository such as SWISS-PROT, it still contains over 55,000 sequences, making it cumbersome – and prohibitively large for all against all clustering using BLAST on the hardware available. For this reason, a subset of PRINTS, miniPRINTS, has been created (Table 1). miniPRINTS contains 20 families, intended to provide a representative cross-section of sequence-space, in order to highlight the different issues that arise in sequence classification. To this end, miniPRINTS contains representatives from highly divergent super-families, domain/modular families and sequences with repeats. miniPRINTS also contains members of small, well-defined families, intended to provide a baseline set of sequences that are relatively easy to deal with.

Figure 5 shows the results of clustering miniPRINTS using BLAST and PSST. The bars show the number of true and false positives occurring with a score threshold selected to minimise the number of misclassifications. For BLAST, this was an *E*-value of 10^{-4}, for PSST, a score threshold of 12.8.

For most of the entries in miniPRINTS, PSST performs similarly to BLAST.

For super-family and domain-based relationships (ZINCFINGER, FNTYPEIII, ALPHAHAEM and GPCRRHODOPSN), BLAST is significantly more sensitive. These are precisely the types of matches that a local-alignment based tool would be expected to perform well with. Sequences related by domain, or super-family share similar structural folds, but different activities are produced by decorating this 'scaffolding' with different functional groups. Assigning proteins to super- or domain-families requires spotting the kind of biochemical conservation seen with the transmembrane helices in the GPCR alignment in Figure 2.

By contrast, successfully placing a protein into a more precise functional category requires matches between short, highly conserved sets of residues. This is the kind of relationship seen with the loop regions in Figure 2. PSST is better than BLAST for opsins and rhodopsins: families that require this kind of relationship to be identified if their members are going to be correctly classified.

Figure 6 shows the distribution of scores resulting from clustering the Opsin subset of miniPRINTS, from which it can be seen that the choice of 12.8 as a score threshold results in many true positives being eliminated from the search. This suggests that PSST might form the basis of a pre-filtering step that reduces a database to a manageable size for processing by a more complex tool

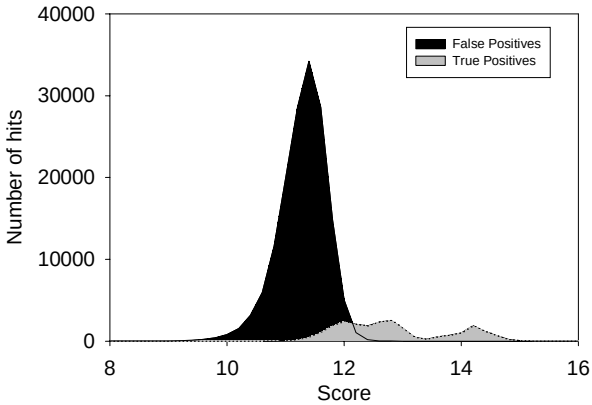

Figure 6. Scores distribution for the opsin family clustered using PSST. The high number of matches is due to the fact that the search is all-against-all – with a potential N^2 matches.

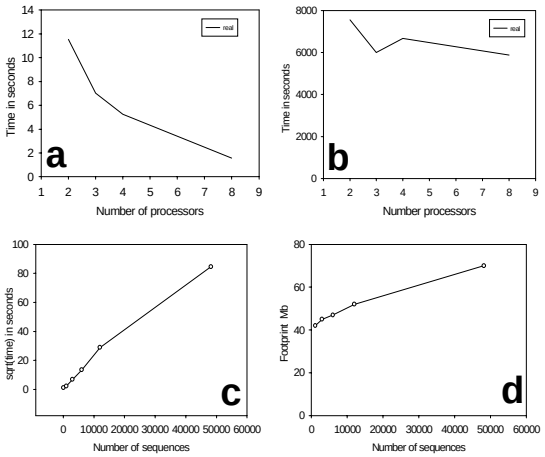

Figure 7. Relative performance for PSST and BLAST **(a,b)** Time vs. number of processors for PSST and BLAST respectively, clustering 1093 GPCRs (total 438,065 residues). **(c,d)** Speed and memory usage vs. clustered database size. Since the timings are for clustering experiments (N^2 comparisons) the square root of the timings usage are plotted.

designed specifically to find GPCRs.

Time and Space Complexity

In order to measure PSST's time and space performance, sequence sets of varying size were clustered. All experiments were conducted on a 16 processor Origin 2000: each processor being a MIPS 195MHz R10000. The L2 cache was 4M and the total memory 6144M. BLAST, version 2.0.6 was used with default parameters throughout. All times stated are wall-clock times as reported by the UNIX `'time'` command.

PSST is between 2 and 3 orders of magnitude faster than BLAST. Figures 7a and 7b show how the running time for clustering 1,093 sequences varies with number of processors. On 4, PSST takes 5.25s compared to 7,737s for BLAST – 1,473 times faster. It can also be seen from the figure that PSST scales much better than BLAST with the number of processors.

Since the number of words to look up is proportional to the size of the query sequence, PSST should scale linearly with the query. Similarly, the size of the pre-computed word-lists is simply proportional to the size of the database to be searched – PSST should scale linearly in both time and space with the database size. Figures 7c,d shows that this is indeed the case.

Discussion

PSST is a novel algorithm that is significantly faster than alignment based techniques. The nature of the algorithm makes it coarse-grained and embarrassingly parallel – reflected by the fact that it scales well with the available processors.

PSST does not achieve this speed and efficiency without some cost, from both an implementational and a biological point-of-view. Memory usage is higher than computationally-intensive alignment algorithms: a compromise chosen with multiple-node cluster computers in mind, which typically have at least 256Mb of memory per node. From a biological perspective, the tool is not good at spotting the kind of biochemically-conserved matches associated with super-families and domains. These relationships are characterised by relatively weak local alignments, and tools already exist to find this kind of similarity. It is also the case that as more sequences become characterised, and the sequence landscape becomes better annotated, the need to identify distant relationships will become less pressing as closer entries in the databases can be used for classification and analysis.

Even when the only clue to a sequence's function is a distant, putative, homologue, finding such a match is only the first step towards a more detailed characterisation: identifying a GPCR is less interesting than determining that the same protein is, for example, a beta adrenergic receptor. PSST is more sensitive for these pharmaceutically important proteins, and tools such as PSST have a role in helping make these classifications.

Sequence analysis software is not always able to make correct assignments: human intervention is required in order to assure a high level of reliability. When the annotator or biologist is recognised as being a vital part of the whole system, it can be seen that bioinformatics tools perform two roles: filtering and presentation.

Sequence analysis algorithms provide mechanisms by which manageable subsets of a database can be extracted and presented to a human being in an appropriate form. Alignments not only produce a similarity score, they also offer a rich metaphor for representing the relationships between a set of DNA or protein sequences. A tool such as PSST is able to perform the filtering step very quickly, but, because it forgoes alignments, it is not able to present its results as anything more than a number denoting similarity.

Generating alignments on the fly, as part of a user-interface, is one potential solution to this problem; using PSST as a pre-filtering step before searching with a slower, more complex tool, such as FPSCAN [27] is another. This latter application is particularly appealing because, as Figure 6. shows, it is possible to increase the number of true positives found by PSST by decreasing the score threshold, but opening the valve in this way also increases the number of false positives. A post-processing step, performed using a discriminator designed for the family of interest, has the potential to offer speed, sensitivity and selectivity.

Attempts to improve the process of gene-identification, characterisation and annotation have

typically focused on the gene-sequences rather than the annotations themselves. One consequence of this is that different relationships (scaffolding vs. decoration, for example) are often elided into the ill-defined catch-all term, 'function' [28].

This is unfortunate because different types of analysis tools are good at spotting different types of functional relationship: developing an insight into why tools behave the way they do can hopefully help to improve the quality of the annotations at our disposal. It is, after all, the annotation that forms the interface between the biologist and the algorithm, and the biologist that adds meaning to the whole undertaking.

Acknowledgements

The multiple alignments were produced with the help of Philip Lord, using CINEMA. Crispin Miller is grateful to the MRC for a Bioinformatics Fellowship, Terri Attwood is a Royal Society University Research Fellow.

References

1. Venter JC et al. (2001) The sequence of the human genome. *Science* 291(5507), 1304-51.
2. The Genome International Sequencing Consortium (2001) Initial sequencing and analysis of the human genome *Nature* 409, 860-921.
3. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. (1990) Basic local alignment search tool. *J Mol Biol.* 215(3),403-10.5.
4. Pearson WR (1990) Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol* 183,63-98.
5. Brenner, SE. (1999) Errors in genome annotation. *Trends in Genetics*, 15(4),132-133.
6. Karp, P. (1998) What we do not know about sequence analysis and sequence databases? *Bioinformatics* 14(9): 753-754.
7. Luthy R, Xenarios I, Bucher P. (1994) Improving the sensitivity of the sequence profile method. Protein Sci 3(1),139-46.
8. Eddy, SR. (1998) Profile hidden Markov models *Bioinformatics* 14: 755.
9. Bateman A, Birney E, Durbin R, Eddy SR, Howe KL, Sonnhammer EL. (2000) The Pfam protein families database. *Nucleic Acids Res.* 28(1),263-6.
10. Attwood TK, Croning MD, Flower DR, Lewis AP, Mabey JE, Scordis P, Selley JN,Wright W.(2000) PRINTS-S: the database formerly known as PRINTS. *Nucleic Acids Res.* 28(1),225-7.
11. Henikoff JG, Greene EA, Pietrokovski S, Henikoff S. (2000) Increased coverage of protein families with the blocks database servers. *Nucleic Acids Res.* 28(1),228-30.
12. Hofmann K, Bucher P, Falquet L, Bairoch A. (1999) The PROSITE database, its status in 1999. *Nucleic Acids Res* 27(1),215-9.
13. Berman, M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, TN., Weissig, H., Shindyalov, IN., Bourne, PR. (2000) The Protein Data Bank. *Nucleic Acids Research*, 28 235-242.
14. Apweiler R, et al. (2001) The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Res.* 29(1),37-40.
15. Bairoch A, Apweiler R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res 28(1),45-8.
16. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D.J. (1997) "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." Nucleic Acids Res. 25:3389-3402.
17. Anfinsen CB., Haber, E., Sela, M. and White, FH. Jr. (1961) The Kinetics of Formation of Native Ribonuclease During Oxidation of the Reduced Polypeptide Chain *PNAS*, 47, 1309-1314.
18. Hooft RW, Sander C, Scharf M, Vriend G. (1996) The PDBFINDER database: a summary of PDB, DSSP and HSSP information with added value. *Comput Appl Biosci.* 12(6),525-9.
19. Califano, A. and Rigoutsos I. (1993) FLASH: a fast look-up algorithm for string homology. *Proc Int Conf Intell Syst Mol Biol.* 1, 56-64.
20. Miller RT, Christoffels AG, Gopalakrishnan C, Burke J, Ptitsyn AA, Broveak TR, Hide WA. (1999) A comprehensive approach to clustering of expressed human gene sequence: the sequence tag alignment and consensus knowledge base. *Genome Res.* 9(11),1143-55.
21. Parsons JD (1995) Miropeats: graphical DNA sequence comparisons. *Comput Appl Biosci.* 11(6),615-9.
22. Bishop, M., and Thompson, E. (1984) Fast Computer Search for Similar DNA Sequences, *Nucleic Acids Res.* 12, 5471-5474.
23. Miller, C., Gurd, J and Brass, A (1999) A RAPID algorithm for sequence database comparisons: application to the identification of vector contamination in the EMBL databases, *Bioinformatics*, 15, 111-121.
24. Salton G. (1971) The SMART Retrieval System – Experiments in automatic Document Processing. *Prentice Hall Inc., Englewood NJ.*
25. Salton G. and Lesk ME. (1968) Computer evaluation of indexing and text processing. *Journal of the ACM,* 15(1),8-36.
26. Karlin, S. & Altschul, S.F. (1990) "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes." Proc. Natl. Acad. Sci. USA 87:2264-2268.
27. Scordis P, Flower DR, Attwood TK. (1999) FingerPRINTScan: intelligent searching of the PRINTS motif database. *Bioinformatics.* 15(10),799-806.
28. Attwood, TK. and Miller, CJ. (2001) Which craft is best in bioinformatics? *Computers and Chemistry* 25, 329-339.