Sequence homology search tools on the world wide web

Ian Holmes
Berkeley Drosophila Genome Project, Berkeley, CA
email: ihh@fruitfly.org

Introduction

Sequence homology search tools may be divided into four groups, illustrated in Figure 1.

1. **Pairwise searches** (*e.g.*, BLAST, Smith-Waterman) These programs compare a single query sequence against each sequence in a (large) database and report significant similarities

2. **Profile searches** (*e.g.*, HMMER) These programs, if supplied with several examples of a family of sequences, will attempt to construct a profile of this family, then search a database for sequences that fit the profile

3. **Automated searches** (*e.g.*, PSI-BLAST) These programs first seek out close relatives of a single query sequence, then use these close relatives to build a profile. In other words, they combine the tasks performed by pairwise and profile search tools

4. **Protein family databases** (*e.g.*, PFAM, PROSITE, BLOCKS) Here, a single query sequence is compared to entries in a database of profiles, each representing a distinct protein family. This approach shares many of the benefits of profile searches, without the duplicated effort of finding members of an already well-characterised family

Each method has its advantages. Automated tools take a lot of the pain out of homology searching, but may return spurious results. Protein family databases bring greater reliability at little extra cost but may miss some homologies. Profile searches are best for investigators who have the time to carefully curate their query alignment. Pairwise searches arguably remain the most transparent (and fastest) of methods.

This article briefly discusses the main examples of each type of search program, outlining relevant issues and giving links to websites where available. Slightly more detail is given for more recent tools such as PSI-BLAST, where commentary is less readily available. Unless otherwise indicated, all programs may be used to search for both DNA-to-DNA and protein-to-protein homologies. At the end of the article, a short section outlining promising developments in automated profiling is included.

This is not intended to be a complete tutorial on sequence homology searching; for that, see *e.g.*, [1] for an introduction or [2] for a more technical treatment. It should also be noted that links can go out of date quickly; often a judiciously worded web search is the easiest way to find the service required.

Note regarding algorithms and implementations. Often, the methodology that a program uses (the underlying algorithm) is shared by more than one program. Recommending one among several implementations of an algorithm can be controversial but it has been attempted here in the interests of clarity.

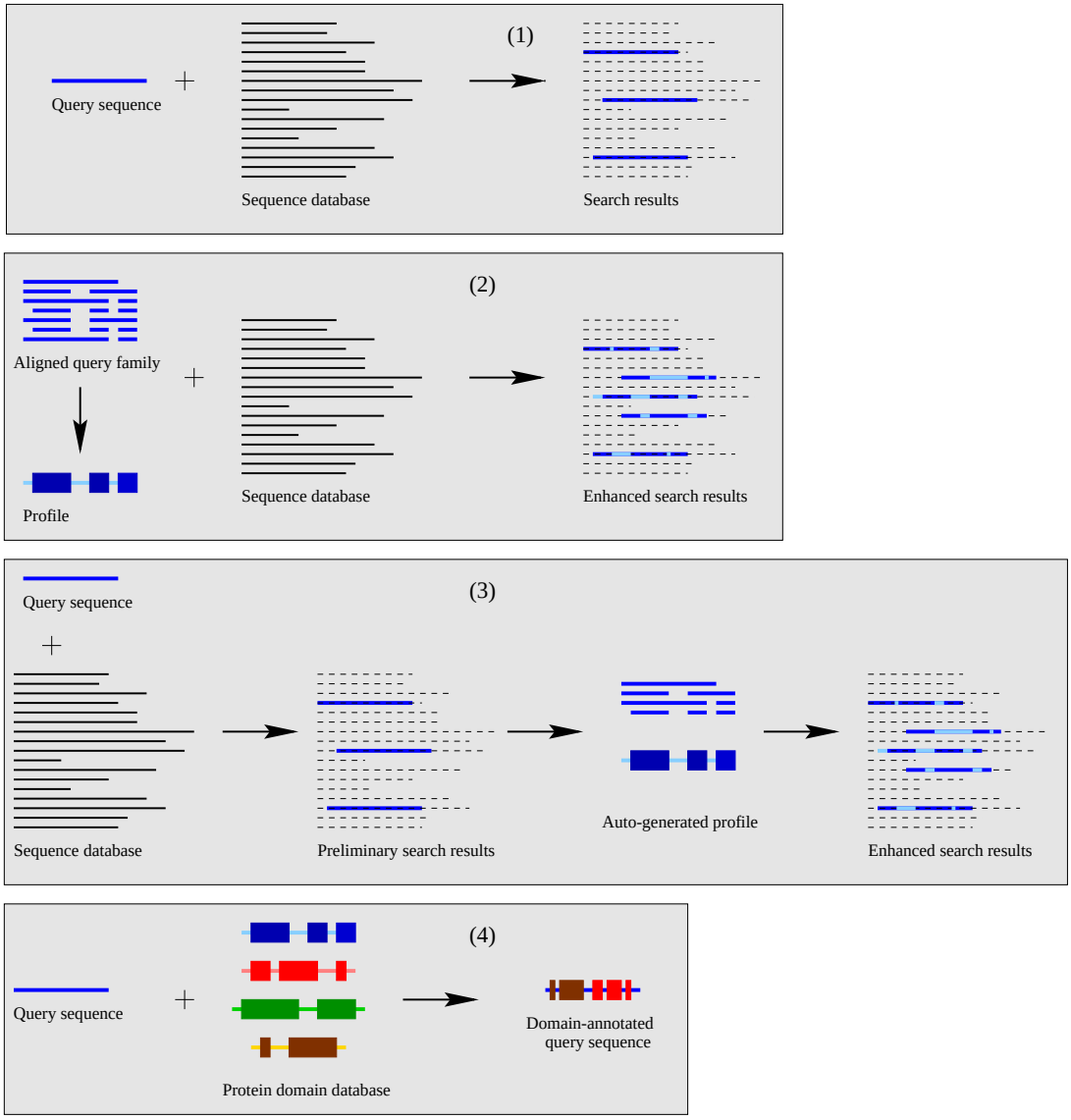

Figure 1. *The four groups of homology search tools.*

Pairwise searches

In a pairwise search, a query sequence is compared to a database sequence, yielding a score that indicates the likelihood of homology. This comparison is repeated for every sequence in the database and high-scoring hits are reported. This basic procedure is shared by all pairwise searches. The various tools available differ most noticeably in speed and sensitivity (*e.g.*, whether, and how often, insertions or deletions [“indels”] in one of the sequences will cause the comparison to fail).

A default scoring scheme is usually provided; this may be overridden by the “power user”. Most often this scoring scheme consists of a substitution matrix, which specifies the likelihoods of all possible point mutations. Other aspects of the scoring scheme can be a gap penalty, specifying the cost of deletions, and a score threshold for reporting hits.

The fastest and most popular pairwise search tool is BLAST. The most sensitive is the Smith-Waterman algorithm, of which the most common implementation is SSEARCH.

BLAST. The BLAST program works on the principle that regions of homology are likely to contain strongly conserved, indel-resistant segments. These areas of strong conservation show up as ungapped blocks in an alignment. Recent versions [3] are also capable of producing gapped alignments, but they still build these gapped alignments up from ungapped segments. This means that BLAST is most likely to fail to detect a homologous sequence where indels are scattered liberally and regularly throughout: in other words, highly divergent sequences may be missed.

Searching for ungapped matches is much faster than searching for gapped matches. BLAST speeds things up even more by looking initially for matches to individual words in the query sequence. This makes BLAST by far the quickest search program on the web today.

Public interfaces to BLAST can be found on the NCBI and EBI websites:

`http://www.ncbi.nlm.nih.gov/BLAST/`

`http://www2.ebi.ac.uk/`

A list of further BLAST servers can be found at:

`http://www.sdsc.edu/ResTools/biotools/biotools1.html`

BLAST databases. The above BLAST servers search query sequences against comprehensive databases such as GENBANK, SWISSPROT, TREMBL or a non-redundant combination of these. In fact, this is true of most search interfaces to be found on the web. Numerous local databases, including the HIV database and many genome sequencing projects, also offer BLAST interfaces to their sequences. It is less common to find such databases offering interfaces to the other kinds of search algorithm described below.

Low-complexity sequence filters. A lot of theoretical work has gone into assigning statistical significance to scores produced by sequence alignment programs. BLAST is the best such studied system. Unfortunately, most statistics can be skewed by repetitive or low-information sequences such as DNA microsatellites or protein coiled-coils. The problem of low-information sequences is so severe that it is unwise to run any search tool without filtering it for low-complexity segments. Programs such as SEG and DUST will handle this. The BLAST server on the NCBI website (see above) is set up to use these filters automatically and most mirror sites will have filtering as a selectable option, if not the default.

A wide selection of tools for predicting and masking features such as coiled-coil sequences can be found at:

`http://www.expasy.ch/tools/`

The Smith-Waterman algorithm. The Smith-Waterman algorithm performs an exhaustive search of all possible gapped alignments between a pair of sequences, given a particular set of scoring parameters. At first this task may appear gargantuan, but the algorithm employs a dynamic programming technique that keeps the search time within manageable bounds. Nonetheless, Smith-Waterman searches do take longer than BLAST searches (as may be expected, since they're more exhaustive).

The most widely web-accessible implementation of Smith-Waterman is SSEARCH [4]. The gain in sensitivity of Smith-Waterman has also prompted some manufacturers (*e.g.*, Paracel, Compugen, Time Logic) to produce hardware accelerators for dynamic programming and some of these boxes also have web front-ends.

Links to web-accessible Smith-Waterman implementations can be found at the following sites, among others:

`http://www.expasy.ch/tools/ similarity`

`http://www2.ebi.ac.uk/`

`http://www.sdsc.edu/ResTools/biotools/biotools1.html`

Smith-Waterman with pre-filtering: FASTA, SCANPS. Somewhere between BLAST and Smith-Waterman in the speed/sensitivity trade-off lie a family of algorithms that initially pre-screen the database for putative hit regions using fast heuristic rules (like BLAST), then focus in on those regions with a full dynamic programming search (like SSEARCH).

Examples of these programs are FASTA [5] and SCANPS; the most commonly encountered of these is FASTA.

Wise2. Programs in the Wise2 suite can compare DNA and protein sequences to one another directly, automatically translating the DNA. For example, the GeneWise program can do a full Smith-Waterman comparison between an amino acid sequence and a stretch of unspliced, untranslated genomic DNA, reporting homologies in spite of intron-exon structure and sequencing errors (including frame-shifts). The Wise2 tools are relatively slow to run, although the package includes "HalfWise" programs that employ BLAST as a pre-filter to speed up gene prediction. The Wise2 suite is written using the dynamic programming language Dynamite [6].

There is a form-based interface to some of the programs on the Wise2 web site:

`http://www.sanger.ac.uk/Software/Wise2/`

Bayesian alignment algorithms. Also worth mentioning are Bayesian probabilistic alignment methods. These handle distant homologies by averaging over all possible evolutionary relationships between a pair of sequences, rather than just picking the most likely one. Notable Bayesian approaches include Lawrence et al's Bayes aligner [7] and Bucher and Hofmann's PSW (probabilistic Smith-Waterman) algorithm [8].

The Bayes aligner is particularly well suited to sequences suspected to contain several conserved ungapped blocks, such as transmembrane proteins. Though not currently accessible through web interfaces, it may be downloaded from:

`http://www.wadsworth.org/resnres/bioinfo/software.html`

The PSW algorithm is implemented in the Wise2 package:

`http://www.sanger.ac.uk/Software/Wise2/`

Profile searches

Profile searches are considerably more sensitive than simple pairwise searches as they make use of position-specific substitution matrices (and sometimes position-specific gap penalties too). Unfortunately they are also even slower than Smith-Waterman; it is possible to use hardware accelerators to speed them up, but interfaces to accelerated profile searches are rarely found on the web.

In this section, tools to work with probabilistic profiles (the most successful type of profile) are described.

Hidden Markov models. The majority of profiling tools currently available make use of hidden Markov models (HMMs). The solid basis of HMMs in Bayesian machine learning theory has helped this field considerably. Chief amongst HMM tools is the HMMER package [9]; predating HMMER, but less widely used, is SAM [10].

To train an HMM profile from a set of sequences, it is generally required that the sequences be aligned. Version 1 of HMMER attempted to do this alignment itself, but good multiple sequence alignment is a non-trivial challenge and so this feature has been dropped from HMMER version 2. The most popular of the multiple alignment packages available is CLUSTAL [11]. A recent benchmark of multiple alignment programs may be found in [12].

A comprehensive list of links to HMM-related software can be found on the HMMER web site:

`http://hmmer.wustl.edu/`

SAM may be accessed via a web interface:

`http://www.cse.ucsc.edu/research/compbio/HMM-apps/HMM-applications.html`

The previously mentioned Wise2 package is capable of comparing protein HMMs to DNA, as well as protein sequences.

The Wise2 web site is located at:

`http://www.sanger.ac.uk/Software/Wise2/`

Stochastic context-free grammars. A generalisation of hidden Markov models, capable of modelling the nested correlations between base pairs that are characteristic of RNA structures, are "stochastic context-free grammars" or SCFGs [2]. While their increased modelling power makes them

potentially more sensitive tools, SCFGs are considerably trickier to use than HMMs and demand more computational resources. Software for training SCFGs and using them to search sequence databases can be downloaded from

`http://www.genetics.wustl.edu/eddy/software/`

Automated searches

PSI-BLAST. A major improvement on the original release of BLAST, PSI-BLAST automates the process of profile construction and database searching. Given a query sequence, PSI-BLAST will search a sequence database for close relatives of the query, then construct a profile using these relatives and search the database again using the profile. This process can be repeated several times.

While it has been possible for some time to construct a fully automated search-and-profiling tool from the modular components described above, PSI-BLAST is the first such integrated system to have gained wide appeal. Since its release, PSI-BLAST has proved a highly popular and useful tool, due perhaps to its ease of use as well as its increased speed and sensitivity compared to BLAST.

For reference, the successive steps performed in a single PSI-BLAST iteration are summarised below:

Candidate match "seeds" are picked from the database using an algorithm similar to BLAST (but incorporating a two-word-hit rule that is more stringent than BLAST's single-hit rule, thereby saving time later). Seeds are extended by exploring only high-scoring cells in the dynamic programming matrix. An appropriate score cutoff for these high-scoring alignment extensions is found by regression. The parameters were estimated in simulations performed by the program authors. Sufficiently high-scoring pairwise alignments between query and database sequences are combined to give a multiple alignment with no gaps in the query. Virtually identical sequences in this multiple alignment are thrown out. Closely related sequences in the multiple alignment are downweighted, using a sequence weighting scheme that works best for small families. A substitution/deletion profile is created using a pseudocount method to incorporate prior knowledge into sparse datasets. This profile is used as a query and the whole process begins again, performing a pre-specified number of iterations before terminating. enumerate

Although the relative contribution of each step to the speed and sensitivity gains of PSI-BLAST are not entirely clear from the published data, it seems that step 1 yields a significant speed gain over BLAST, steps 2-3 improve on the limited sensitivity of BLAST's ungapped scoring scheme and steps 4-8 are profiling steps that focus the next iteration of the search on probable family members.

An obvious problem with PSI-BLAST is the very same problem that would be encountered were a profile being constructed manually: if a chance similarity is mistakenly included in the profile training set at an early stage, the error may become "fixed" if the next iteration of the search algorithm picks up relatives of the imposter sequence rather than members of the query family. This potential amplification of false positives is an inherent feature of any iterative search algorithm; the best way to guard against it is to manually check that the sequences reported by the program appear relevant when compared to the query, not just to one another.

The NCBI has a web interface to PSI-BLAST with an attractive results display (Figure 2):

`http://www.ncbi.nlm.nih.gov/cgi-bin/BLAST/nph-psi`

Arguably, there is considerable room for improvement on PSI-BLAST. The previously-described tools specialising in multiple alignment and profile training are much better at these jobs than the corresponding parts of PSI-BLAST. This is particularly evident in the way that gaps are handled. Packages such as MEME [13] and work on probabilistic models of protein evolution [14] suggest that more consistent approaches to integrated systems can be found. At the time of writing, however, PSI-BLAST is the best (perhaps the only) freely available reliable automated system for homologous sequence discovery and its popularity signals a clear challenge for computational biology to shift up a gear.

MEME. MEME is a program for identifying motifs from a training set of unaligned sequences [13]. Particularly suited to finding short motifs such as nucleotide binding sites, it proceeds by a "greedy"

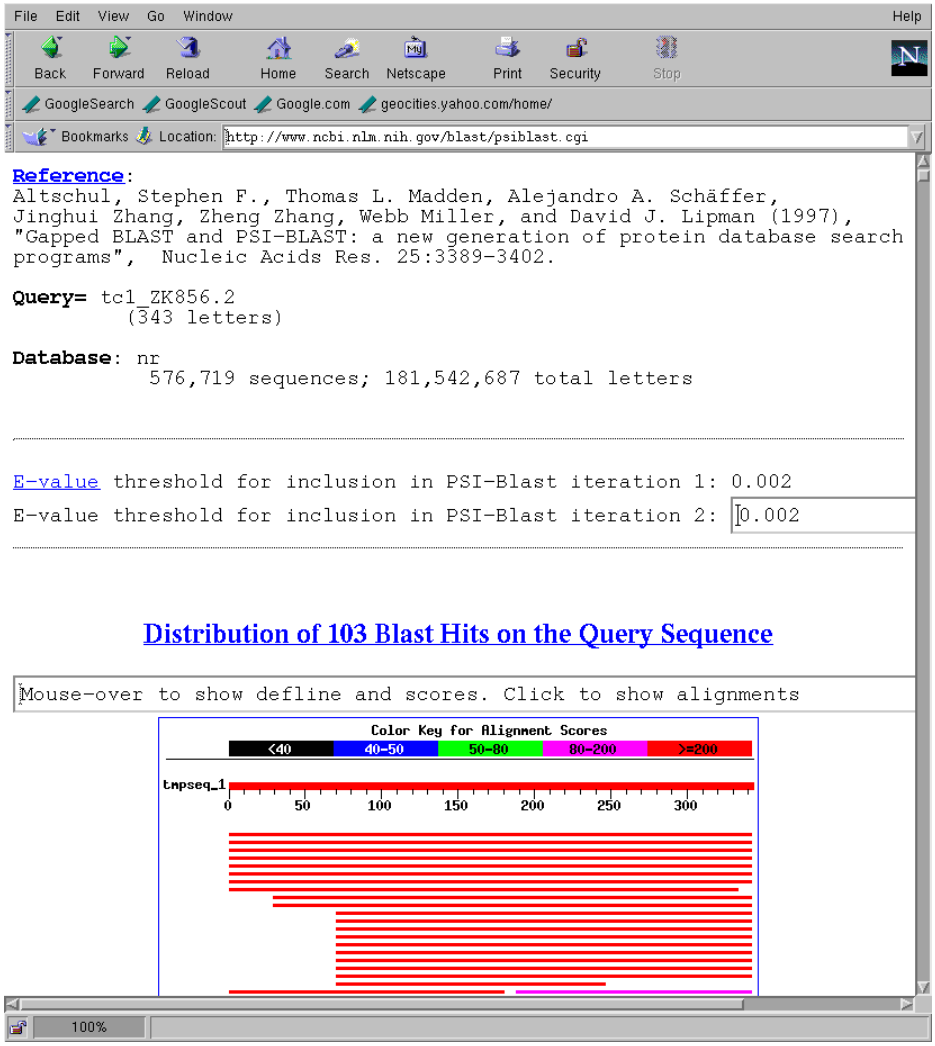

Figure 2. *The NCBI web interface to PSI-BLAST.*

strategy: beginning with a hidden Markov model seeded on a single subsequence from the training set, it iteratively refines this model until no more improvements to the overall score can be found. This algorithm is a reasonably fast and highly effective way of locating repeated motifs. MEME is web-accessible:

`http://meme.sdsc.edu/meme/website/`

Gibbs sampling. The Gibbs sampling algorithm for multiple sequence alignment has become a popular tool for discovery of short ungapped motifs [15]. The algorithm performs a “random walk” through the space of multiple alignments, weighted by the likelihood of those alignments (so that, over long enough times, high-scoring alignments should be visited proportionally more often than low-scoring ones).

Gibbs samplers are theoretically less prone to getting “stuck” than greedy algorithms like MEME, as they are capable of making choices that are unfavourable in the short term. The downside of this is that convergence is unpredictable. The algorithm can be VERY slow to run. Perhaps for this reason it is hard to find interfaces to Gibbs samplers on the web; however, one can download software from the following URL:

`http://stl.wustl.edu/~ecr/GIBBS/`

Protein family databases

There are a number of efforts underway to attempt to organise the protein database into clusters, corresponding to motifs that are conserved throughout nature. Searching these databases rather than the "raw" protein databases brings several advantages:

- the search is more sensitive, as the query sequence is compared to a profile of each family, domain or cluster rather than individual members of the cluster alone. This also makes the search process faster;
- the results of the search are more succinct than a typical "raw" database search;
- and the database may have links to other online resources, such as structure or functional annotation. itemize

The main drawback is that the protein family database may be inaccurate or incomplete

Unfortunately there is no good competitive evaluation of the various protein family databases accessible on the web. It is probably best to try more than one; most of them now contain cross-references to one another anyway, as well as links to structure and literature databases.

There follows a brief description of the most widely used databases.

Pfam. Pfam is a database of strictly non-overlapping protein domains [16]. Each domain entry consists of a seed alignment from which an HMM profile has been trained. Pfam is the only protein family database to be built using HMM-profiles from the very beginning and its development is inextricably linked with that of the HMMER package.

In addition to the curated database (Pfam-A) there exists an automatically generated database of putative families (Pfam-B), from which more families are "upgraded" into Pfam-A with each release.

Pfam can be accessed at the following URLs:

`http://www.cgr.ki.se/Pfam/`
`http://www.sanger.ac.uk/Software/Pfam/`
`http://pfam.wustl.edu/`

PROSITE. PROSITE is a collection of protein families and domains with comprehensive and thorough annotation that includes links to scientific literature and even email addresses of contactable experts [17]. The profiles for each domain are manually constructed and less quantitatively flexible than the statistical HMM methods used by Pfam.

PROSITE is located at:

`http://www.expasy.ch/prosite/`

PRINTS. PRINTS is a well-annotated database of protein fingerprints, each of which may be compounded of several distinct motifs [18]. This approach differs from those described above in that a single motif may be used by multiple fingerprints. This is intended to reflect the observation that structural patterns may be decomposed into smaller elements; thus the curators of PRINTS suggest it to be a useful tool for recognising larger structural patterns.

PRINTS is located at:

`http://www.biochem.ucl.ac.uk/bsm/dbbrowser/PRINTS/PRINTS.html`

ProDom. The ProDom database is automatically generated from a clustering of the protein sequence database which currently uses the PSI-BLAST algorithm [19]. This automatic generation procedure makes ProDom less tidy than some of the other databases, but its coverage should be more complete.

`http://www.toulouse.inra.fr/prodom/doc/prodom.html`

InterPro. The above four databases (Pfam, PROSITE, PRINTS and ProDom) have been gathered together as the composite database InterPro. Sequences can be queried simultaneously against all four databases and the results displayed in parallel. InterPro is searchable online at the following URL:

`http://www.ebi.ac.uk/interpro/`

Blocks. Blocks is a database of ungapped multiple alignments [20]. The exclusion of gaps has two immediate implications: firstly, some motifs present in the other databases will be missing from Blocks;

and secondly, those motifs that are present will tend to be more strongly conserved.

The Blocks database may be searched at the URL below. The search engine can also search ungapped subsets of all the other protein family databases mentioned above.

`http://blocks.fhcrc.org/`

SMART. The SMART database, like Pfam, contains HMM profiles of protein families [21]. Although it covers a smaller fraction of the protein universe than Pfam (it contains around 400 families as opposed to Pfam's 2290), it compensates by providing significantly enriched structural, functional and phyletic annotation for each domain. SMART may be queried online at:

`http://smart.embl-heidelberg.de/`

Developments in integrated profiling

Multiple alignment, phylogenetic tree construction and sequence profiling are all attempts to handle the observed statistical nature of relationships between protein sequences. For maximum sensitivity and precision, one would ideally like to be able to model all these aspects of protein evolution using a single integrated tool.

PSI-BLAST may be viewed as one attempt at this. However, as has been mentioned, it makes little use of statistical modelling theory. More promising are probabilistic models of sequence evolution emanating from the phylogenetics camp [22, 23, 24, 14, 25] where debates on flavours of statistical method constitute a familiar theme.

The common theme of these methods is to define a scoring function for a set of sequences aligned to a profile and related by a phylogenetic tree. The profile, alignment and tree that optimise this (probabilistic) scoring function are then to be simultaneously optimised. By taking phylogenetic correlations into account, these methods are theoretically capable of detecting signals that phylogenetically naive programs (read: most of the other algorithms described in this article) will miss. It is the large number of parameters and unknowns in this problem that favour approaches with a solid mathematical foundation such as probabilistic modeling.

Perhaps the simplest of these models is the links model [22], in some ways the probabilistic multiple-alignment analogue of the Smith-Waterman algorithm. This model treats residue insertion and deletions as independent events that occur at a constant rate along the length of the sequence, using a birth-death process familiar from probability theory. The links model has been implemented as a multiple alignment algorithm [26] which may feasibly be extended to a non-homogenous profiling system by concatenating multiple links models.

For profiles without gaps, the RIND program is an interesting and effective way of modelling site-to-site heterogeneity in conservation patterns and substitution rates for sets of pre-aligned protein sequences [24]. RIND attempts to fit the best substitution matrix to each separate column of an alignment, using a variant of the EM algorithm from statistical theory that allows all the entries in the rate matrices to vary.

In contrast, "tree HMMs" [23, 14] permit only a finite number of rate matrices for each column, but are otherwise similar to hidden Markov models in that they allow for large deletions (but not insertions) relative to the profile. Tree HMMs are also capable of aligning sequences de novo using methods akin to Gibbs sampling. (Such methods tend to be less effective than dedicated multiple alignment programs such as CLUSTAL [11] but are, arguably, easier for independent parties to improve on.) A fusion of tree HMMs with the links model would allow insertions, bringing tree HMMs closer to the profile HMMs used for database searching by HMMER and SAM, but this is speculation.

Probabilistic methods have many advantages over more heuristic approaches. They are well-defined, facilitating both collaborative and competitive development by multiple groups. They are also capable of sophisticated refinement and great sensitivity, as has been demonstrated in the case of profile HMMs. However they require considerably more effort to develop than heuristic algorithms that can be bolted together relatively quickly. It may be hoped that the success of HMMs in modelling patterns in sequences will encourage this effort to be spent in developing further probabilistic models in sequence analysis and indeed bioinformatics in general.

Acknowledgments

This article has benefited from conversations with Bill Bruno, Ewan Birney and Roger Sayle. The author is in receipt of the Fulbright-Zeneca 1998-1999 Fellowship for Research in Bioinformatics and has also received support from Los Alamos National Laboratory.

References

[1] A. Baxevanis and B. F. Francis Ouellette, editors. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. John Wiley Sons, Inc., 1998.

[2] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, UK, 1998.

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

[4] W. R. Pearson. Effective protein sequence comparison. *Methods in Enzymology*, **266**:227–258, 1996.

[5] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the USA*, **4**:2444–2448, 1988.

[6] E. Birney and R. Durbin. Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. In T. Gaasterland, P. Karp, K. Karplus, C. Ouzounis, C.Sander, and A. Valencia, editors, Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology, pages 56–64, Menlo Park, CA, 1997. AAAI Press.

[7] J. Zhu, J. S. Liu, and C. E. Lawrence. Bayesian adaptive sequence alignment algorithms. *Bioinformatics*, **14**:25–39, 1998.

[8] P. Bucher and K. Hofmann. A sequence similarity search algorithm based on a probabilistic interpretation of an alignment scoring system. In D. J. States, P. Agarwal, T. Gaasterland, L. Hunter, and R. F. Smith, editors, Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology, pages 44-51,Menlo Park, CA, 1996. AAAI Press.

[9] S. R. Eddy. Hidden Markov models. *Current Opinion in Structural Biology*, **6**:361–365, 1996.

[10] A. Krogh, M. Brown, I. S. Mian, K. Sj lander, and D. Haussler. Hidden Markov models in computational biology: applications to protein modeling. *Journal of Molecular Biology*, **235**:1501–1531, Feb.1994.

[11] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Research*, **22**:4673–4680, 1994.

[12] J. D. Thompson, F. Plewniak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*, **27**(13):2682–2690, 1999.

[13] W. N. Grundy, T. L. Bailey, C. P. Elkan, and Michael E. Baker. Meta-MEME: Motif-based hidden M arkov models of protein families. *Computer Applications in the Biosciences*, **13**:397–406, 1997.

[14] G. J. Mitchison. A probabilistic treatment of phylogeny and sequence alignment. *Journal of Molecular Evolution*, **49**(1):11–22,1999.

[15] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, **262**(5131):208–214, 1993.

[16] A. Bateman, E. Birney, R. Durbin, S. R. Eddy, R. D. Finn, and E.L. Sonnhammer. Pfam 3.1: 1313 multiple alignments match the majority of proteins. *Nucleic Acids Research*, **27**(1):260–262, 1999.

[17] K. Hofmann, P. Bucher, L. Falquet, and A. Bairoch. The PROSITE database, its status in 1999. *Nucleic Acids Research*, **27**(1):215–219, 1999.

[18] T. K. Attwood, D. R. Flower, A. P. Lewis, J. E. Mabey, S. R. Morgan, P. Scordis, J. Selley, and W. Wright. PRINTS prepares for the new millennium. *Nucleic Acids Research*, **27**(1):220–225, 1999.

[19] F. Corpet, J. Gouzy, and D. Kahn. Recent improvements of the Pro Dom database of protein domain families. *Nucleic Acids Research*, **27**(1):263–267, 1999.

[20] J. G. Henikoff, S. Henikoff, and S. Pietrokovski. New feature sof the Blocks Database servers. *Nucleic Acids Research*, **27**(1):226–228, 1999.

[21] C.P. Ponting, J.Schultz, F.Milpetz, andP.Bork. SMART: identification and annotation of domains from signalling and extracellular protein sequences. *Nucleic Acids Research*, (**27**):229–232, 1999.

[22] J. L. Thorne, H. Kishino, and J. Felsenstein. Inching toward reality: an improved likelihood model of sequence evolution. *Journal of Molecular Evolution*, **34**:3–16, 1992.

[23] G. J. Mitchison and R. Durbin. Tree-based maximal likelihood substitution matrices and hidden Markov models. *Journal of Molecular Evolution*, **41**:1139–1151, 1995.

[24] W. J. Bruno. Modelling residue usage in aligned protein sequences via maximum likelihood. *Molecular Biology and Evolution*,**13**(10):1368–1374, 1996.

[25] A.L.Halpern and W.J.Bruno. Evolutionary distances forprotein-coding sequences: modeling site-specific residue frequencies. *Molecular Biology and Evolution*, **15**(7):910–917, 1998.

[26] I. Holmes and W.J.Bruno. Evolutionary HMMs: a Bayesian approach to multiple alignment. Submitted.