

Saturated BLAST: an automated multiple intermediate sequence search used to detect distant homology

Weizhong Li¹, Frederic Pio², Krzysztof Pawłowski³ and Adam Godzik^{3,*}

¹San Diego Supercomputer Center, La Jolla, CA 92093, USA, ²Simon Fraser University, Burnaby, BC V5A 1S6, Canada and ³The Burnham Institute, La Jolla, CA 92037, USA

Received on April 6, 2000; revised on July 25, 2000; accepted on July, 2000

Abstract

Motivation: Two proteins can have a similar 3-dimensional structure and biological function, but have sequences sufficiently different that traditional protein sequence comparison algorithms do not identify their relationship. The desire to identify such relations has led to the development of more sensitive sequence alignment strategies. One such strategy is the Intermediate Sequence Search (ISS), which connects two proteins through one or more intermediate sequences. In its brute-force implementation, ISS is a strategy that repetitively uses the results of the previous query as new search seeds, making it time-consuming and difficult to analyze.

Results: Saturated BLAST is a package that performs ISS in an efficient and automated manner. It was developed using Perl and Perl/Tk and implemented on the LINUX operating system. Starting with a protein sequence, Saturated BLAST runs a BLAST search and identifies representative sequences for the next generation of searches. The procedure is run until convergence or until some pre-defined criteria are met. Saturated BLAST has a friendly graphic user interface, a built-in BLAST result parser, several multiple alignment tools, clustering algorithms and various filters for the elimination of false positives, thereby providing an easy way to edit, visualize, analyze, monitor and control the search. Besides detecting remote homologies, Saturated BLAST can be used to maintain protein family databases and to search for new genes in genomic databases.

Availability: Free from <http://bioinformatics.burnham-inst.org/xblast>

Contact: liwz@spsc.edu or adam@burnham-inst.org

Introduction

The traditional way of comparing two protein sequences is to align them with a dynamic programming algorithm

(Smith and Waterman, 1981). However, alignment programs using sequence information and mutation matrices fail to recognize many proteins that are known to be homologous, but their sequences have diverged. To overcome this problem, people have developed more sensitive sequence alignment programs.

One type of program uses information about allowed mutation patterns at various positions along the sequence; the allowed mutation patterns are deduced from the patterns of substitutions in closely related homologues. Programs that compare a single sequence to a family profile include Profiles (Gribskov *et al.*, 1987), Position Specific Iterated (PSI) BLAST (Altschul *et al.*, 1997), and Hidden Markov Models (HMMs) (Eddy, 1995, 1996; Eddy *et al.*, 1995; Karplus *et al.*, 1998; Krogh *et al.*, 1994). Other programs such as BLOCKS (Henikoff *et al.*, 1998) and FFAS (Rychlewski *et al.*, 2000) compare two profiles to each other. Another approach, often referred to as threading, is to use the structure of one of the proteins being compared to assess whether the second could possibly have a similar structure (Moult, 1999).

Another strategy that explores the sequence diversity of distantly homologous proteins without constructing the explicit multiple sequence alignment or profile is the Intermediate Sequence Search (ISS) (Karplus *et al.*, 1998; Park *et al.*, 1997, 1998; Salamov *et al.*, 1999). The ISS approach is straightforward, and takes advantage of the transitive nature of homologous relationships. If a sequence (I) is homologous to a query sequence (Q) and at the same time target sequence (T) is homologous to (I), the homology between (Q) and (T) can be established. The connection is written as, and can be used to recognize and align distant homologues. ISS was used to recognize remote evolutionarily related sequence pairs derived from SCOP database (Park *et al.*, 1997; Murzin *et al.*, 1995), and the authors claimed that ISS increased the detection rate by 70% compared to FASTA

*To whom correspondence should be addressed.

(Pearson and Lipman, 1988) while maintaining the same error-yield rate. Similar work using BLAST instead of FASTA was described as Double BLAST (Karplus *et al.*, 1998).

A variant of ISS that applies more than one intermediate step by making connections such as 'Q-I₁-I₂-I_n-T' is called Multiple Intermediates Sequence Search (MISS) (Salamov *et al.*, 1999). MISS is more powerful than ISS, and can recognize many distantly homologous pairs missed by other methods. When compared to HMM and similar profile-based methods, ISS and MISS provide superior information about evolutionary relationships, and do not rely on sometimes misleading and ambiguous multiple sequence alignments. At the same time, anecdotal evidence suggests that the MISS procedure often diverges, resulting in spurious predictions and a waste of computer time. This makes it difficult to implement fully automated ISS and MISS protocols.

A recipe for MISS could look like that: the first step is to run a database search with the first query. Then new queries are selected from the first search's output and, after validation, used in the subsequent searches. This procedure is repeated as long as new sequences are found. The list of proteins found in this way is filtered to remove those with unwanted properties.

However, MISS can diverge for several reasons. MISS can improperly treat multiple-domain proteins where an intermediate sequence containing two domains could link two unrelated single-domain sequences containing either domain. Also, even a single error in one of the searches can propagate adding many unwanted sequences to the output. MISS can be further misled by the presence of various 'promiscuous' sequences that produce spurious high score alignments. Most of such sequences are removed by low complexity filters, but some still remain. Finally, indiscriminate application of naive MISS may result in multiple, redundant searches that waste time and computer resources.

In this paper, we present a program called Saturated BLAST that provides a series of simple tools designed to deal with these problems and make MISS a reasonable research tool that can be used with minimal effort. At the same time, we introduced several novel features to the MISS procedure, increasing its sensitivity and flexibility. A traditional ISS or MISS method uses sequence versus sequence programs such as FASTA or BLAST to build connections between the intermediate steps. Saturated BLAST uses BLAST and also profile-based PSI-BLAST. This enables Saturated BLAST to combine the benefits of profile and MISS approaches.

To deal with the inherent instability of the MISS procedure, we also introduced tools to filter each search's output and to verify intermediate sequences before they are used as queries. The tools include keywords (both

to search for and to avoid), pre-defined lists of proteins to avoid, criteria for significance threshold, and level of sequence divergence.

Methods, algorithms and implementation

The basic MISS procedure consists of four steps: the initial search, the selection of intermediate queries, the intermediate search loop, and the final result analysis. The initial database search is performed with the original query, yielding some sequences or sequence fragments which are selected as new queries for the subsequent searches. The search process is repeated until it no longer yields new sequences.

We refer to the query sequences as *seeds*. Seeds are selected from sequences found in the database search (we often refer to such sequences as *hits*) and are used as queries in succeeding searches. A seed is the *parent* of the all proteins found in the BLAST search with its sequence. With this nomenclature we can think of the original query as a 'parentless' seed. For each protein, the number of intermediate steps from the original query to itself is this protein's *level*. We define that the original query has level of 0, and all the sequences found by the original query have a level of 1.

In Saturated BLAST we use the BLAST program family as the database search tool. We chose to use BLAST for several reasons: BLAST and PSI-BLAST are the fastest and most sensitive database searching programs; the NCBI offers a web-based BLAST search against the most up-to-date sequence databases. This allows Saturated BLAST to be used on relatively low-end machines because it can submit jobs to the NCBI BLAST server. An additional reason why we chose to use the BLAST suite is that it comprises programs that explore both protein and nucleotide databases.

Saturated BLAST was developed in the Perl script language, and the graphic user interface was implemented in Perl/Tk. The structure of Saturated BLAST is illustrated in Figure 1. The main database stores all of the sequence entries, including the original query and hits, in a format that allows the easy export of data into other databases. The main database is managed by several modules, which also control the BLAST searches. These modules are explained below.

Database manager

When we begin a new MISS task, we add the original query to the database as the first entry. The hits from the first and subsequent BLAST searches are filtered and appended to the database. Each database entry also contains the properties and data obtained or calculated from the BLAST search. These properties and data include: NCBI-gi identifiers, some annotations, the range of sequence identified in the search, the alignment to the

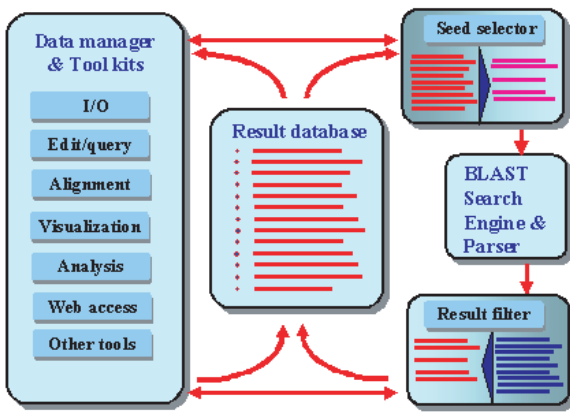


Fig. 1. The flowchart of the Saturated BLAST program. All the modules are described in detail in the text.

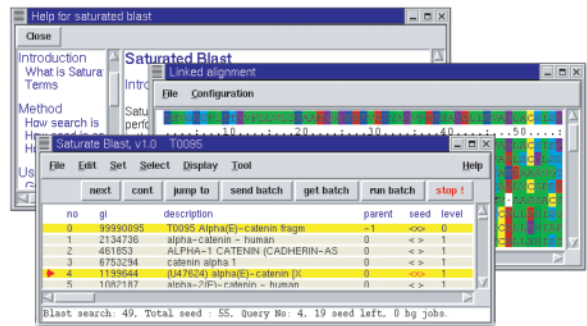


Fig. 2. Sample screen snapshots of the Saturated BLAST GUI. The main window (center), the multiple alignment (right-hand side), the Help window (left-hand side).

(a)

SCOP	gi	Domain	1NWP:A	1A8Z:_	1PLC:_	1AAC:_
2.5.1.1.19	461600	1NWP:A	-	>1000, 77	>1000, >1000	>1000, >1000
2.5.1.1.20	3318937	1A8Z:_	2,7, 5,2	-	>1000, 0.003	>1000, >1000
2.5.1.1.2	230657	1PLC:_	3,0, 4,4	3,0, 4,6	-	0,19, 9e-32
2.5.1.1.1	442593	1AAC:_	2,9, 4,1	3,6, 3,7	2,1, 5,2	-

(c)

```

1NWP:A  eckvtvdstddssnkdiadkscckfvelthsgslpkvnmfllskeadmpp
1A8Z:_  tvhwaav---lpgfpfsvwhdkkntldpaga-ivdvtntkng---fcfsfd
1PLC:_  -dillgaddg-slavyp-sfsfsp---sekivknag---pplih
1AAC:_  -vvvria-----kkyk---pnhvkv---gdvfwream---pnhv

1NWP:A  ---skea---dmgpiatdglsagidkqylkdqdarvih---kviig---ekd
1A8Z:_  -tqkt---ppfavmpv-----ipivag---gfspvpkdkfyln
1PLC:_  -dedsipsgv-----askismseedllnk---lelf
1AAC:_  -vagvl-----geaalkg---pmmkke---gay

1NWP:A  vdvsklasekkgfcdpfi-skkktvllk
1A8Z:_  fwhp--tag--yyvcgripaaatcxfkivk
1PLC:_  evals--nk--helsyca-hgg-aavkvtyn
1AAC:_  lute---atvdhctb--fp-fargkvtle
    
```

(b)

```

1NWP:A  eCkvvdvtgdfsnkdiadkCktvelthsgslpkvnmfllskeadmpp
114738  eCetvtgvtgdtysrflvPACaeenckhmPktGCMNVAASAA-gdv
97549   -aaalagatllPntv---rlidvngll---vqfNWIvnggldaaa
4757375 -agypkdiasev-ff-vdk-ffPpPpdeada-
1AAC:_  -mkypetpelnkvdv-w-ire-ampPdh

1NWP:A  tdlshvdkqyldkdvahkvhka-axadtdvdkaaagkkgfCrfP
114738  kkephhannvtpdkRL-afplog-ktkkykSaSkdaitfCayP
97549   vntagmualyppctanaqamama-agspdrPab---lllctgp
4757375 -EACktalnhdylagnpffilkR-eyfCebq
1AAC:_  -vcllgaaal-kgpmmkqaylltea-ldyhCtP-

1NWP:A  imkvtlks Search history, expect & identity in BLAST
114738  fknrtfke 1NWP:A -> 114738 4e-50, 49%
97549   tpiR- 114738 -> 97549 2e-26, 32%
4757375 gagRv- 97549 -> 4757375 3e-4, 31%
1AAC:_  -pfrkvt- 4757375 -> 1AAC: 3e-19, 31%
    
```

(d)

```

1NWP:A  eckvtvdstddssnkdiadkscckfvelthsgslpkvnmfllskeadmpp
1A8Z:_  tvhwaav---lpgfpfsvwhdkkntldpaga-ivdvtntkng---fcfsfd
1PLC:_  -dillgaddg-slavyp-sfsfsp---sekivknag---pplih
1AAC:_  -vvvria-----kkyk---pnhvkv---gdvfwream---pnhv

1NWP:A  iatdgl sagidkqylkdqdarvih---kviig---ekd
1A8Z:_  -p f avmpvlpivag---gfspvpkdkfyln
1PLC:_  -p daskismseedllnk---lelf
1AAC:_  -v avlgeal-l-kgpmmkqaylltea-ldyhCtP-

1NWP:A  ekgfcdpfi-skkktvllk
1A8Z:_  yyvcgripaaatcxfkivk
1PLC:_  -elsyca-hgg-aavkvtyn
1AAC:_  -tldyhctp-pfrkvt-
    
```

Fig. 3. (a) The pair-wise sequence and structure similarity between the four proteins from the example discussed in the text. Structural similarity as RMSD and CE Z-score (upper/lower number, lower triangle). Sequence similarity as BLAST and PSI BLAST expect value (upper/lower number, upper triangle). (b) The multiple alignment of the two of the proteins from the example discussed in the text (1NWP and 1AAC), together with three intermediate sequences identified in the search. The intermediate sequences aligned with 1NWP and 1AAC are identified by their NCBI-gi identifiers. The alignment was obtained as explained in the text. The expect value and sequence identity in every intermediate step is displayed. Positions that are identical for a parent sequence and its child sequence are highlighted and shown in uppercase. Other positions are in lowercase. (c) The multiple structural alignment of the four proteins from the example discussed in the text, obtained from the CE server (<http://cl.sdsc.edu/ce.html>) (Shindyalov and Bourne, 1998). Positions that are at least partly conserved in the multiple alignment are highlighted and shown in uppercase. Other positions are in lowercase. (d) The multiple alignment of the four proteins from the example discussed in the text as built by Saturated BLAST. The alignment was obtained from pairwise alignments following the MISS chain, the intermediate sequences are not shown. On the top line, the position is marked as '+' or '-' if the alignment of this residue is fully or partly in agreement with CE alignment. The highlighting scheme is identical to that in Figure 3c.

query, the parent, the search level, the alignment score, the expect value, and the sequence identity.

The database manager provides methods to query and to select the sequences according to various properties and logical patterns. The database manager is also used to delete entries, to import sequences and to export data in several formats.

Interface

Saturated BLAST's graphic user interface (GUI) is shown in Figure 2. Entries are tabulated in a main display window according to user-defined display parameters. The GUI provides various windows to set parameters and thresholds, to define BLAST search options, to display messages, to edit the database, to open web links, and to use other common GUI-requiring applications.

Filters

Filters in Saturated BLAST are the most important means to confine a MISS to a desired direction and to keep it from diverging. There are four filter types: a redundancy filter, a low significance filter, a keywords filter, and a smart filter.

The redundancy filter is used to prevent the addition of redundant sequences to the database. Different seeds in MISS commonly find the same sequence segments. So, for a new sequence B from BLAST search, if another sequence A with same identifier exists in the database, and if the overlap between A and B is greater than a threshold, B is treated as the same as A and simply removed.

The low significance similarity filter eliminates sequences with high expect values, short length or low sequence identity.

The keywords filter is designed to accept or remove proteins according to the expert judgment of the user, rather than the expect value. There are classes of proteins that often appear with high scores in PSI BLAST searches even though they are unrelated to the seed, and their inclusion as seeds leads to divergence in MISS. Here, biological annotation from BLAST output can be considered as a more solid argument than alignment score.

The smart filter remembers user-deleted sequences and stores them as a list. The list can be edited and input separately. This filter automatically removes from the BLAST output those proteins that are in the list.

Seed selector

The output of MISS can be intermediate sequences that are themselves closely related, and it is unnecessary and undesirable to use each of these sequences as seeds. Searches with redundant seeds will not contribute any new information and will waste time and resources. Saturated BLAST clusters output sequences with a given threshold of identity, and selects one representative of each cluster to become a seed for future searches. The selection of seeds

is also based on the expected value, sequence identity, sequence length, and keywords pattern. Alternatively, the user can assign the representative seeds through GUI.

BLAST engine and parser

This module is located between the seed selector and the filters. It takes the seeds, runs the specified BLAST program, then parses the results and sends them back to the filters. The engine can run BLAST on a local computer or submit jobs to NCBI's BLAST server. In Saturated BLAST, the user can simultaneously specify different databases and BLAST programs, allowing for multiple BLAST searches with a single seed.

Input, output and restart

Saturated BLAST automatically saves all parameters and current results in a 'restart' file before each search. This file is a backup of everything in case the program crashes, and is also the main output file.

The output file can be exported into several other formats: plain FASTA, HTML, a tab-delimited table, and plain text. The FASTA format can be imported into and analyzed with other software packages, such as ClustalX (Jeanmougin *et al.*, 1998). HTML is platform-independent, so it can be opened across any operating system, and hyperlinks and dynamic-display features provided by JavaScript make it convenient for viewing. The table file can be imported and edited in software like Microsoft Excel or Access.

Analysis and visualization

Alignments are crucial for sequence analysis, and Saturated BLAST provides four kinds of alignments, which are derived directly or indirectly from the BLAST output. The multiple alignment of a parent query and all hits can be directly derived from the output of a single BLAST search. A protein and its ancestor from any level can be aligned via the alignments of intermediate connections, recall that the presence of intermediate steps is a unique feature of MISS strategies. Saturated BLAST can also align any pair of sequences by aligning them to their common parent or grandparent or by using the built-in Smith–Waterman dynamic programming subroutine (Smith and Waterman, 1981). Finally, groups of sequences can be selected by the user and aligned via intermediate sequences and common ancestors.

In Saturated BLAST, users can open an unlimited number of windows containing color-coded alignments. The program has gapped and ungapped viewing options because multiple alignments built from intermediate sequence alignments are often distorted by gaps introduced at different levels.

Another tool in Saturated BLAST is cluster analysis. It is often very informative to classify large protein

families into sub-groups. All or a selection of sequences can be clustered using the standard average linkage cluster method. The all-against-all similarity matrix that is needed for this purpose can be calculated from pairwise alignments available within the Saturated BLAST (see above). Given the alignment, the normalized score S (Altschul and Gish, 1996; Karlin and Altschul, 1990) used in the similarity matrix is calculated as

$$S = s - [\ln(Kmn)/\lambda]$$

where s is the raw score, and n and m are the lengths of the sequences. In the default BLAST matrix *blosum62*, λ is set to 0.216 and K set to 0.014. When an alignment is derived indirectly, only the score of the highest scoring segment of the alignment is applied. It should be noted that this procedure does not create optimal clustering for the purposes of phylogenetic analysis. Other programs, such as ClustalX or Phylip are better suited for this purpose. Clustering provided by Saturated BLAST is designed solely for the purpose of choosing optimal seeds.

Enhancements and flexibility

The Saturated BLAST package is designed to be flexible and suitable for different types of searches. Besides the identification of distant homologues, Saturated BLAST can be used to define a protein family and monitor for the appearance of new genes in genomic databases. To this end, we have designed the program to give users a lot of control over searches. The user can run MISS automatically, manually step-by-step, or the search can be stopped at predefined break points. The user can stop the search at any moment and modify all parameters and thresholds, manually reset and input one or more seeds, and delete unwanted protein sequences and groups of sequences. The user is also able to enable, disable and modify filters, and change the BLAST search's database, programs and parameters.

Saturated BLAST search allows for active user participation throughout the entire iterative procedure.

Applications and examples

Previously, ISS methods have been studied and compared to other search tools (Karplus *et al.*, 1998; Park *et al.*, 1997, 1998; Salamov *et al.*, 1999). In this paper, we introduced the particular realization of a MISS procedure using the BLAST family of programs, and will now validate Saturated BLAST as a tool to efficiently detect remote homologues. We will present a full evaluation of this tool as applied to fold-recognition in a separate publication.

To test Saturated BLAST for remote homology recognition, we chose several single-domain proteins: 1NWP:A, 1A8Z, 1PLC and 1AAC. They all belong to

the 'Plastocyanin/azurin-like' protein family according to the SCOP classification, and the close similarity of their biological functions strongly argues for their close evolutionary relationship. Their structural similarity has been verified by the combinatorial extension (CE) algorithm using the protein-fold comparison server (<http://cl.sdsc.edu/ce.html>) (Shindyalov and Bourne, 1998). Their pairwise sequence similarities were obtained by BLAST and PSI BLAST searches at NCBI's BLAST server against the non-redundant protein (NRP) database. PSI BLAST searches detected homology with a significant expect value only between 1PLC and 1AAC ($9e^{-32}$). One more pair was found with marginal significance, but all other pairs were not found (see Figure 3a).

We started Saturated BLAST using 1NWP:A as the query sequence, with the aim of finding the other three proteins. The initial PSI BLAST search found 50 new sequences and 11 seeds surpassed Saturated BLAST's default parameters. The program then performed three levels of automated searches, and all three target proteins were found with significant expect values (better than $3e^{-4}$).

The supporting tools were then used to analyze the results. The alignment of the search path from 1NWP:A to 1AAC was generated by Saturated BLAST and is shown in Figure 3b. The multiple alignment of these four domains as calculated by Saturated BLAST is compared to the alignment obtained from structure comparison by the CE server (Figure 3c and d). It is clear that the Saturated BLAST alignment is consistent with the CE results in the most important segments.

Other applications of the Saturated BLAST package

Besides detecting remotely homologous sequences, the Saturated BLAST package can be used in several other types of projects including maintaining and updating protein family databases and mining genomic databases for new genes that are distantly homologous to a protein family of interest.

A protein family database can be set up by performing an interactive Saturated BLAST search on the NRP database. Once the search parameter, threshold and filter settings have been optimized; members of a certain protein family can be maintained in a Saturated BLAST database. The optimized settings identified in the interactive search can be saved and used to periodically run Saturated BLAST against monthly updates of the NRP database. This will automatically keep the protein family up-to-date.

We can mine genomic databases using these protein family databases and the settings optimized for searching the NRP database. The key is to use the clustering tool to select divergent representatives as seeds, and then to perform a new generation of searches using *tblastn* against genomic databases such as EST, GSS and HTGS, instead

of using blastp or blastpgp against the NRP database. The output should be parsed, clustered and submitted against the NRP database for verification. This procedure has been used to discover several new potential genes, with the experimental verification of the predictions now in progress.

Conclusions

Intermediate sequence search methods effectively improve the sensitivity of sequence comparisons, but in the past have been difficult to automate. Our Saturated BLAST package executes automated intermediate sequence searches with minimal user input while providing great flexibility, a graphic user interface and many supporting tools.

We illustrated the successful use of Saturated BLAST in the identification of distant homologues. We also mentioned two other applications for Saturated BLAST: mining genomic databases for novel proteins from important protein families, and maintaining and updating protein family databases.

Saturated BLAST's performance depends on the diversity of the protein family, the databases used, and the search parameters. Saturated BLAST cannot be applied if a simple BLAST search doesn't identify any homologues. The default parameters provided with the program work well in several different examples, but they may not work in all cases.

The Saturated BLAST package's source code, as implemented on LINUX, is available from <http://bioinformatics.burnham-inst.org/xblast> under the open source license agreement.

Acknowledgements

We thank Dr Kutbuddin S. Doctor for the suggestions and testing of this software. This research was partly supported by the NIH grant GM60049.

References

- Altschul,S.F. and Gish,W. (1996) Local alignment statistics. *Meth. Enzymol.*, **266**, 460–480.
- Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
- Eddy,S.R. (1995) Multiple alignment using hidden Markov models. *Ismb*, **3**, 114–120.
- Eddy,S.R. (1996) Hidden Markov models. *Curr. Opin. Struct. Biol.*, **6**, 361–365.
- Eddy,S.R., Mitchison,G. and Durbin,R. (1995) Maximum discrimination hidden Markov models of sequence consensus. *J. Comput. Biol.*, **2**, 9–23.
- Gerstein,M. (1998) Measurement of the effectiveness of transitive sequence comparison, through a third 'intermediate' sequence. *Bioinformatics*, **14**, 707–714.
- Gribskov,M., McLachlan,A.D. and Eisenberg,D. (1987) Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA*, **84**, 4355–4358.
- Henikoff,S., Pietrovski,S. and Henikoff,J.G. (1998) Superior performance in protein homology detection with the blocks database servers. *Nucleic Acids Res.*, **26**, 309–312.
- Jeanmougin,F., Thompson,J.D., Gouy,M., Higgins,D.G. and Gibson,T.J. (1998) Multiple sequence alignment with Clustal X. *Trends Biochem. Sci.*, **23**, 403–405.
- Karlin,S. and Altschul,S.F. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA*, **87**, 2264–2268.
- Karplus,K., Barrett,C. and Hughey,R. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.
- Krogh,A., Brown,M., Mian,I.S., Sjolander,K. and Haussler,D. (1994) Hidden Markov models in computational biology. Applications to protein modeling. *J. Mol. Biol.*, **235**, 1501–1531.
- Moult,J. (1999) Predicting protein three-dimensional structure. *Curr. Opin. Biotechnol.*, **10**, 583–588.
- Murzin,A.G., Brenner,S.E., Hubbard,T. and Chothia,C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.
- Park,J., Teichmann,S.A., Hubbard,T. and Chothia,C. (1997) Intermediate sequences increase the detection of homology between sequences. *J. Mol. Biol.*, **273**, 349–354.
- Park,J., Karplus,K., Barrett,C., Hughey,R., Haussler,D., Hubbard,T. and Chothia,C. (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol.*, **284**, 1201–1210.
- Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, **85**, 2444–2448.
- Rychlewski,L., Jaroszewski,L., Li,W. and Godzik,A. (2000) Comparison of sequence profiles. Strategies for structural predictions using sequence information. *Protein Sci.*, **9**, 232–241.
- Salamov,A.A., Suwa,M., Orengo,C.A. and Swindells,M.B. (1999) Combining sensitive database searches with multiple intermediates to detect distant homologues. *Protein Eng.*, **12**, 95–100.
- Shindyalov,I.N. and Bourne,P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*, **11**, 739–747.
- Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.