

For "Computer Methods for Macromolecular Sequence Analysis"

R. F. Doolittle, editor

Methods in Enzymology

The Blocks Database and Its Applications

By Jorja G. Henikoff¹ and Steven Henikoff²

¹Fred Hutchinson Cancer Research Center

Seattle, Washington 98104

henikoff@howard.fhcrc.org

²Howard Hughes Medical Institute

Fred Hutchinson Cancer Research Center

Seattle, Washington 98104

henikoff@howard.fhcrc.org

Phone: (206) 667-4515

FAX: (206) 667-5889

Introduction

The detection of homology between a newly determined sequence and a sequence in a databank is often the most important clue to the function of a gene, and so the "homology search" has become a standard tool in the molecular biologists' arsenal. This tool increases in power as databanks expand with representatives from a large percentage of all protein families¹, so that most newly discovered sequences now have recognizable homologs in current databanks². Such successes have fueled large-scale sequencing projects, including those involving single-pass sequencing of cDNAs³ and those of model organisms with high gene densities⁴. As a result of these activities, the sequence databanks have become more complex, and this complexity can complicate the interpretation of homology search results.

Here we discuss a homology searching system based on a database of protein family representations rather than sequences. The representations consist of blocks of aligned segments derived from the most highly conserved regions of proteins. This concentration of information from conserved alignments can improve the detection of relationships in difficult situations. Furthermore, because the number of protein families is increasing slowly relative to the number of sequences, this system does not become substantially more complex with time; rather, the families become more informative.

Since its introduction in 1991⁵, the system has been improved in several ways. The searching program was modified to incorporate a strategy for the detection of multiple blocks representing a group⁶ and to utilize improved sequence weights to reduce redundancy within a protein family⁷. Furthermore, the system has been applied to large DNA sequences, such as those of whole chromosomes⁸, and has been made accessible via electronic mail (e-mail)⁹ and the World Wide Web (WWW)¹⁰. The WWW server also provides sequence logos¹¹ for intuitive graphical display of blocks. In addition to its use in homology searching, the database itself has been utilized in the construction of amino acid substitution matrices¹² and in the evaluation of sequence weighting methods⁷.

Motifs to Blocks

Most protein families can be characterized by sets of locally similar sequence segments, typically referred to as "motifs". Within a single sequence, a motif consists of a contiguous stretch of amino acids shared by a group of proteins and conserved for function. This definition includes motifs found in families of related proteins as well as in collections of unrelated proteins that share a common structural feature. An example of the second kind of motif is the helix-turn-helix DNA-binding motif that is thought to have converged to form a similar 20-residue structure with weak sequence constraints¹³. The terms "signature" and "pattern" are often used as synonyms for motif. Motifs represent local as opposed to global features of a protein, and a protein sequence can contain multiple motifs.

Representations of a motif are based on a local multiple alignment of the group of proteins that share it in the region containing the motif. We call a local multiple alignment made without insertions or deletions (gaps) in any of the sequences a protein "block". Its width is that of the conserved region and its depth is the number of sequences in the group that share the motif. The Blocks Database contains blocks representing motifs found in known families of related proteins. Currently, the families represented in the Blocks Database are those documented in the Prosite Database¹⁴, but the concept of cataloging blocks for other families and for unrelated groups of proteins that share a motif is completely general.

The Blocks Database provides actual multiple alignments, as do the PRINTS¹⁵ and ProDom¹⁶ databases. These differ from some other databases in general use, such as those that provide consensus sequences¹⁷, profiles^{18,14}, patterns¹⁴ and fingerprints¹⁹. An advantage of providing actual alignments is that they contain all of the information needed to derive any of these other representations. The Blocks Database is constructed automatically once the protein groups are identified, whereas some other searchable databases of motif representations have been constructed using a combination of manual and automatic techniques^{20,15}. The ProDom database¹⁶ is similar to the Blocks Database in that it is automated and provides multiple alignment representations; however, the groups represented in ProDom are themselves derived automatically by clustering the database into groups based on sequence similarity¹⁷, whereas the Blocks Database uses protein families identified manually based on multiple

criteria.

Constructing the Blocks Database

The two-step PROTOMAT system constructs the Blocks Database using a fully automated procedure⁵. The only input necessary for PROTOMAT to produce blocks is a group of protein sequences that presumably have one or more motifs in common. Currently, the Blocks Database is based on a Prosite catalog of protein families¹⁴ and its corresponding Swiss-Prot Database of protein sequences²¹. Each entry in Prosite includes a list of known protein sequences in Swiss-Prot for the family. For each Prosite entry, PROTOMAT makes a set of one or more blocks from this list of sequences. Prosite also includes a pattern for each family, but this pattern is not used in any way by PROTOMAT; one of the blocks constructed by PROTOMAT may contain all or part of the Prosite pattern, but this is not guaranteed since the PROTOMAT blocks are found by an entirely different procedure than that used to derive Prosite patterns. The high quality of Prosite and its detailed documentation makes it an outstanding source of related groups on which to base the Blocks Database.

Starting with a list of sequences from related proteins, PROTOMAT first generates a large number of candidate blocks using a motif-finding algorithm. Currently, the MOTIF algorithm of Smith²² is used for this step, but other motif-finders such as the Gibbs sampler²³ could be used, and may be used in the future, to make the Blocks Database. MOTIF scans the sequences exhaustively looking for spaced triplets of amino acids out to a maximum distance in at least a subset of the sequences. Examples of spaced triplets are Ala-Ala-Ala and Val-x-x-x-Ala-x-Cys where x represents any amino acid. A spaced triplet found in enough sequences anchors a local multiple alignment against which sequences lacking the triplet are aligned to maximize a block score. All parameters for MOTIF are determined empirically using the characteristics of the sequences. To maximize the sensitivity of MOTIF, we allow the subset of sequences that must contain a spaced triplet to be so small that a few triplets are found even when the sequences are shuffled by randomly permuting each amino acid.

Candidate blocks are passed on to the MOTOMAT block assembly program. At this stage the candidates may overlap one another in all or

some of the sequences. MOTOMAT refines them by merging those that overlap consistently in all of the sequences and extending them in both directions until similarity falls off, out to a maximum width. After refinement, MOTOMAT uses graph theory techniques to assemble a best set of blocks that occur in the same order within a sequence, without overlapping, in a significant number of sequences. Sequences that do not conform are dropped from the set of blocks.

After a block is made, weights are computed for each sequence segment in it. These weights are intended to compensate for over-representation of some of the sequences. Low weights are given to redundant sequences, and high weights to diverged sequences. There are many methods for computing sequence weights²⁴, but the Blocks Database currently uses position-based sequence weights which are easy to compute and have been shown to perform well in searching applications⁷ (see Fig. 6). These are simple weights derived from the number of different residues and their frequency in each position of the block.

No gaps are inserted in the block alignments by PROTOMAT, and some of the sequences may be imperfectly aligned, especially near the edges of the blocks. PROTOMAT was designed to make blocks for database searching applications and occasional misalignments can be tolerated so long as the block is correctly aligned for the large majority of sequences. In such cases a contribution from misaligned segments will be diluted out.

Blocks 8.0 was derived from Prosite 12 and Swiss-Prot 29 and contains 2,884 blocks representing 770 different protein families, an average of 3.75 blocks per family. These blocks contain from 2 to 507 sequences, and range in width from 4 to a maximum of 55 amino acids. Fig. 1 contains a sample entry from the Blocks 8.0.

Users can make blocks from their own group of protein sequences using PROTOMAT implemented on the Blockmaker server. Blockmaker-generated blocks are appropriate for searching sequence databanks for new family members²⁵ and for PCR primer design²⁶.

Searching the Blocks Database

The Blocks Database is used primarily to aid in the detection and

interpretation of protein sequence homology. A protein or DNA sequence is compared to the protein blocks to see if the sequence belongs to any family represented in the Blocks Database. The rationale behind searching a database of blocks is that information from multiply aligned sequences is present in a concentrated form, reducing background and increasing sensitivity to distant relationships. If a hit is found the Prosite documentation for the family may provide insights as to the function of the sequence.

A block can be searched either as a query compared with a target database of sequences, or as an entry in a target database which is compared with a query sequence. In either case, the block must be converted into a representation that can be scored against a sequence, such as a pattern, consensus sequence, or profile. In this representation, the block is aligned with a sequence at every possible position and its score computed.

The Blocksearcher system uses the BLIMPS (BLocks IMProved Searcher) program followed by the BLOCKSORT analysis program to perform a search of a query sequence against the Blocks Database⁶, although other methods have also been used²⁷. Each block in the database is converted to a position-specific scoring matrix (PSSM), which is similar to a profile¹⁸. A PSSM has as many columns as there are positions in the block, and 20 rows, one for each amino acid. It also contains additional rows for other characters that may be encountered in a protein or translated DNA sequence: B (D or N), Z (E or Q), X (unknown residue), - (gap), and * (stop codon). Each PSSM entry consists of numeric scores based on the ratio of the observed frequency of an amino acid in a block column to its expected overall frequency in Swiss-Prot (odds ratio). The observed frequencies are weighted for sequence redundancy using the sequence weights in the block, and pseudo-counts (described below) are added to compensate for non-observed amino acids. Currently, Blocksearcher uses a variation of the data dependent pseudo-count method²⁸, our unpublished results). PSSM entries are normalized so that each entry is an integer between 0 and 99 (Fig. 2).

During a search, the query sequence is aligned with each block at all possible positions, and an alignment score is computed by adding the scores

for each position. Each block is scored individually, including multiple blocks for the same family, and the top scores are saved. If the sequence is DNA, BLIMPS translates it in all 6 possible frames and scores all 6 translated sequences against the block. BLIMPS scores all possible alignments and saves all high-scoring ones, so that repeated motifs can be detected.

Before scores resulting from a search of a query sequence against the Blocks Database can be used to rank the blocks relative to one another, the blocks must be calibrated to provide standardized scores that are comparable between blocks. This is because the blocks in the database vary both in width and in the number of sequences they contain. For example, the raw score for a wide block is likely to be larger than that for a narrow block just because the wide block has more columns to add up. Each block in the Blocks Database has two standard scores for comparison (Fig. 1). They are obtained by searching each block against the version of Swiss-Prot from which the sequences making the block were extracted and then analyzing the raw scores. Using Prosite's list of known members of the block's family as the set of "true positive" sequences for the block and assuming all other sequences are "true negatives", we compare the score distributions of these two sets of sequences. Perfect separation of the two distributions, in which the highest-scoring true negative sequence scores lower than the lowest scoring true positive sequence, is not usually obtained, sometimes because of the presence of uncatalogued true positives. To allow for errors such as this, the 99.5th percentile of the true negative sequence scores is chosen as a "lower" calibration score. Scores above this are likely to be interesting, and those below are likely to be true negatives. Blocksearcher divides the raw score by this lower calibration score, multiplies the result by 1000 and ranks the search results by this calibrated score. Calibrated scores above 1000 can therefore be considered higher than 99.5% of scores for known true negative sequences.

The median calibrated score for the known true positive sequences is reported as the "strength" of the block. Strength is a measure of how highly true positive sequences score against the block for comparison with the query sequence score. A low value for strength indicates that the block is weak and may fail to exclude chance alignments. Alternatively, a block

that is too strong may be so specific that it will exclude distant relatives.

The BLOCKSORT analysis program provides additional aids for evaluating the results of a search against the Blocks Database⁶. BLOCKSORT processes the BLIMPS search results and groups together all blocks from the same family. The calibrated BLIMPS score along with strength are useful for evaluating the local similarity between a query sequence and a single block, but it is important to interpret them in terms of a reasonable model of chance. In order to assess both the effectiveness of the calibrated score and its significance, we took 7,082 sequences from Swiss-Prot that were not represented in the Blocks Database, shuffled each one by random permutation of each amino acid, and searched each shuffled sequence against the Blocks Database. The resulting distribution of calibrated scores for the first-ranking hits showed that all blocks were about equally likely to score at or above a given level, demonstrating that the calibration step is effective. It also provided BLOCKSORT with a percentile value to assign to the calibrated score for a real search.

If a query sequence obtains a good score for more than one block from the same family in the Blocks Database, and if these blocks are in the same order and similar distances apart as they are in the sequences in the blocks, then this provides strong confirmatory evidence that the query may belong to the family²⁵. A multiple block hit contains information about global similarity between the query and members of the family. It is very difficult to construct a realistic theoretical model for scoring multiple block hits because different protein groups include different numbers of members, because groups are represented in the Blocks Database by different numbers of blocks with diverse properties, and because not necessarily all of the blocks for a group may score high in a search. Therefore, BLOCKSORT computes an "expectant value" based on an intuitive model that has been verified empirically to quantify the degree of global similarity seen in a multiple block hit⁶.

A nomogram showing the frequency of first-ranking hits with various expectant values observed in the 7,082 searches of shuffled sequences is provided with the BLOCKSORT output to assist evaluation (Fig. 3). The nomogram presents a level of confidence given a combination of a local score (or percentile) and global expectant value. Although the

nomogram is based on shuffled sequences, searches carried out with the same 7,082 sequences, but not shuffled, provided a very similar distribution of scores, once the uncatalogued true positive sequences were pruned from the results list. This confirms that confidence levels based on search results for randomized sequences are realistic and can be applied to the evaluation of search results for real sequences.

An example illustrates how the nomogram can be used to judge whether or not a hit reflects significant similarity (Fig. 4). The protein encoded by the *S. cerevisiae* *INO2* gene, which is involved in the regulation of phospholipid metabolism, has been shown by extensive genetic and biochemical evidence to contain a helix-loop-helix DNA-binding motif²⁹. Nevertheless, the level of similarity to other proteins that contain this motif is so low that BLAST³⁰ and Smith-Waterman³¹ searches with the *INO2* protein sequence fail to detect any of the 103 members of this family catalogued in Prosite 8.0. Consistent with such a low level of similarity, Blocksearcher ranks the helix-loop-helix block A (the "anchor" block) in second place at only the 80th percentile (Fig. 4), high enough to be reported, but not high enough to be considered interesting. However, the 23rd-ranking segment in the search aligns with Block B at a compatible distance (10 amino acids) downstream of Block A, providing an expectant value (an approximation of a P-value) of $P < 0.0021$ for Block B in support of Block A. The combined probability of obtaining one block at the 80th percentile and other blocks in support at $P < 0.0021$ ($= 10^{-2.7}$) is seen from the nomogram (Fig. 3) to occur on the average in only 1 search in 1000, consistent with the experimental evidence that this region is truly a helix-loop-helix DNA-binding domain.

This example also illustrates a potential complication of interpreting hits in which the anchor block is not the first ranking block. To allow for the possibility that a sequence might belong to more than one family, the BLOCKSORT does not explicitly penalize hits that do not rank first, even though the nomogram is based on only first ranking blocks. Therefore, confidence that a hit is real must be reduced whenever the anchor block is outranked by a presumed false positive hit, such as in the example. Here, the helix-loop-helix Block A ranks a close second (the first ranking block is found at the 86th percentile), so our confidence in the result remains high.

In general, confidence in a hit decreases with decreasing rank, to an extent that depends upon the available evidence.

For each multiple block hit representing a family in the Blocks Database, BLOCKSORT prints a map of the blocks from the database and compares it with a similar map of the blocks in the query sequence. For each block included in the hit, the alignment of the query with the sequence from the block in the Blocks Database with which it shares the most identical residues is shown to further assist evaluation (Fig. 4). If a family is documented in Prosite as having a maximum of r repeated motifs within a sequence, then BLOCKSORT will map alignments from the search up to r repeats found in a single sequence.

The Blocksearcher e-mail server was inaugurated in the summer of 1992 and currently averages about 700 searches per month. Over 5,000 different users have contacted this server since its inception. In the summer of 1994 Blocksearcher was also made available via the World Wide Web (WWW), and subsequently a sequence logo¹¹ option was added.

Other Applications of the Blocks Database

Amino Acid Substitution Matrices

Methods for analysis of sequence similarity typically use scores from an amino acid substitution matrix. A substitution matrix is a 20 by 20 symmetric array of numeric values that provide scores that are applied to each position in an alignment. These scores can be derived from presumed true positive alignments, an approach pioneered by Margaret Dayhoff³². In such cases, scores are based on the odds that one amino acid may be substituted by another. Large positive values in the matrix indicate frequent substitution and large negative values indicate that the substitution is rare. Although other scoring schemes have been proposed, Dayhoff's mutation data matrices³³ were for many years considered the standard for pairwise alignment and searching programs. In the Dayhoff model, substitution rates are derived from global alignments of pairs of proteins sequences that are at least 85% identical and then extrapolated to more distant sequences using a Markov model. As pointed out by Altschul³⁴, the basis of a substitution matrix is its "target frequencies", which are the observed frequencies of substitution for each pair of amino acids. The aligned segments in the

Block Database provide ample data for estimating the target frequencies in regions of local similarity, which we used to construct a series of amino acid substitution matrices from Blocks 5.0¹².

Rather than estimate substitution rates and extrapolate as in the Dayhoff approach, target frequencies were obtained directly from alignments of segments in blocks by counting pairs of aligned amino acids at each position within a block. The total counts for all of the 210 possible amino acid pairs obtained for every position of every block in the Blocks Database were converted to target frequencies and odds ratios were calculated by dividing them by the corresponding expected frequencies for each pair.

To make a series of substitution matrices using this approach, sequences were clustered within each block and each cluster was weighted as a single sequence in counting aligned amino acid pairs. A clustering percentage was specified in which sequence segments that are identical for at least that percentage of amino acids were grouped together. For example, if the percentage was set at 35% and sequence segment A is identical to sequence segment B in at least 35% of their aligned positions, then A and B were clustered. If C is identical to either A or B in at least 35% of aligned positions, it was also clustered with them and the contributions of A, B and C were averaged in counting aligned amino acid pairs. Pairs were only counted between clusters, not within clusters. If all of the segments in a block clump into one cluster, then the block provides no counts. We refer to the series of matrices constructed in this way as the BLOSUM series (for BLOcks SUbstitution Matrix). BLOSUM 35, therefore, is based on observed substitutions between segments that are less than 35% identical, BLOSUM 62 on segments that are less than 62% identical, BLOSUM 100 on segments that are less than 100% identical, and so forth. BLOSUM 100 is based on over 11 million observed substitutions from 2,106 blocks, BLOSUM 62 on 1.3 million aligned amino acid pairs from 1,572 blocks and BLOSUM 35 on 0.1 million pairs from 439 blocks. The pairs counted for BLOSUM 35 are from segments less similar than those from BLOSUM 100, so that BLOSUM 35 has correspondingly lower relative entropy³⁴ (0.21 bit) than BLOSUM 100 (1.45 bits).

Extensive testing using the BLAST searching algorithm, which

depends on ungapped local alignments³⁰, has shown that the Blosum series outperforms other matrices for this task^{12,35}. BLOSUM 62 (Fig. 5) with relative entropy of 0.7 bit performs the best for database searching, and has been adopted as the default for BLAST and other similar applications. Information theory predicts that other matrices in the series might be better suited for other applications³⁴. For example, BLOSUM 45 was found to perform especially well in profile searches³⁶. It is likely that gap penalties employed by many programs will have an important effect. There is currently no accepted theory of gap penalties, so that determination of the best matrix to use for an application that employs gap penalties should be guided by empirical testing.

Dirichlet Mixtures

PSSM representations of multiple alignments are based on the observed frequencies of amino acids in each position of the alignment. It is typical that for any position, most amino acids do not appear at all, and so it must be decided how to deal with these non-occurrences. One approach is to assume that, since a residue is not observed in a position, it should never occur there; this is unrealistic if the alignment contains only a few sequences. Another is to use a substitution matrix to arrive at a weighted average score for the unobserved amino acid¹⁸; this will reduce specificity of the PSSM if the alignment is well-represented.

A statistical approach to the problem of non-occurrences adds imaginary pseudo-counts to the observed counts of each amino acid at a position, based on some belief about amino acids expected to occur there. Brown, *et al.*³⁷ have utilized Dirichlet mixtures to calculate pseudo-counts for a hidden Markov model application. They noted that each position in a multiple alignment can be thought of as a sample from a multinomial distribution and that the Dirichlet is the natural conjugate family of prior distributions. Once the many parameters are estimated, the Dirichlet priors provide elegant pseudo-counts that take into consideration amino acid interrelationships. Estimation of the parameters is laborious, however, and requires many sample aligned positions. Initially, Brown *et al.* estimated parameters from the HSSP database of multiple alignments based on structure³⁸, but this group has also found alignments from the Blocks

Database to be useful for this purpose (D. Haussler, personal communication).

PSSM evaluations

The Blocks Database together with the lists of known true positive sequences for the families in it (from Prosite) has been used to evaluate alternative ways of computing PSSMs. The general approach is to compute a PSSM from each block in the database using two different methods, then search the PSSMs against Swiss-Prot, and finally compare the two distributions of known true positive and true negative sequences. This procedure is similar to the calibration searches described above, but when testing searching methods it is more effective to search a sequence database that contains some true positive sequences that were not used to make the blocks. Since Prosite and Swiss-Prot are maintained in tandem, we accomplish this by searching blocks made from an older version of Swiss-Prot against a newer version of Swiss-Prot that contains more sequences, and using the corresponding newer version of Prosite to provide the lists of true positive sequences.

One study using this approach compared several different sequence weighting schemes⁷. Three methods were found to perform best among those tested: position-based weights⁷, Voronoi weights³⁹ and branch proportional weights⁴⁰. An update of this study is shown in Fig. 6. Currently, we are comparing different methods of computing pseudo-counts, extending the analysis of Tatusov *et al.*²⁸ to large numbers of protein families.

Access

The Blocks Database, the Blosum series of matrices and associated software is available by anonymous ftp from the NCBI repository:

```
ftp ncbi.nlm.nih.gov
cd blocks
```

However, most users will find e-mail and WWW servers to be preferable to installation of the software. E-mail server addresses are:

```
blocks@blocks.fhcrc.org
```

for searching the Blocks Database or retrieving a set of blocks and

corresponding Prosite files, and:

`blockmaker@blocks.fhcrc.org`

for making blocks from user-defined protein sequences. Send the word 'help' in the subject line of a blank message or as the only word in the body of a message to obtain help files for either server. Both servers are able to interpret FASTA, GenBank, EMBL, PIR, GCG and Genepro sequence formats. In addition, the Blocks searcher will decide whether the submitted sequence is protein or DNA, and in the case of DNA, will search it in all 6 reading frames, putting together multiple hits from the same strand⁶. Blockmaker accepts up to 250 protein sequences in the same format concatenated into a single message. DNA sequences or messages with fewer than 3 sequences should not be sent to Blockmaker.

The WWW is an especially efficient method for utilizing tools described in this chapter. The Uniform Resource Locator for the Blocks home page is:

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

The WWW server allows protein and DNA queries to be searched against the current Blocks Database by placing a sequence in the text box provided on the Block search page. Results are returned with hypertext links to Prosite and Swiss-Prot, and from there to other databases, including EMBL/GenBank and Medline. Because of the hypertext capability of the WWW, results can be evaluated rapidly with minimum effort on the part of the user. The WWW server also allows retrieval and browsing, returning the full set of blocks and Prosite documentation. An added feature of the WWW server is the provision of a sequence logo¹¹ for each block in the Blocks Database, further aiding the evaluation of search results. Logos are also provided with the WWW Blockmaker server¹⁰.

Summary

Protein blocks consist of multiply aligned sequence segments without gaps that represent the most highly conserved regions of protein families. A database of blocks has been constructed by successive application of the fully automated PROTOMAT system to lists of protein family members obtained from Prosite documentation. Currently, Blocks v. 8.0 based on protein families documented in Prosite 12 consists of 2,884

blocks representing 770 families. Searches of the Blocks Database are carried out using protein or DNA sequence queries, and results are returned with measures of significance for both single and multiple block hits. The database has also proven useful for derivation of amino acid substitution matrices (the Blosum series) and other sets of parameters. WWW and e-mail servers provide access to the database and associated functions, including a PROTOMAT-based blockmaker for sequences provided by the user.

References

1. P. Green *Curr. Opin. Struct. Biol.* **4**, 404 (1994).
2. E. V. Koonin, P. Bork & C. Sander *EMBO J.* **13**, 493 (1994).
3. M. D. Adams, J. M. Kelley, J. D. Gocayne, M. Dubnick, M. H. Polymeropoulos, H. Xiao, C. R. Merrill, A. Wu, B. Olde, R. F. Moreno, A. R. Kerlavage, W. R. McCombie & J. C. Venter *Science* **252**, 1651 (1991).
4. S. G. Oliver, Q. J. M. van der Aart, M. L. Agostoni-Carbone, M. Aigle, L. Alberghina, D. Alexandraki, G. Antoine, R. Anwar & J. P. G. Ballesta *Nature* **357**, 38 (1992).
5. S. Henikoff & J. G. Henikoff *Nucleic Acids Res.* **19**, 6565 (1991).
6. S. Henikoff & J. G. Henikoff *Genomics* **19**, 97 (1994).
7. S. Henikoff & J. G. Henikoff *J. Mol. Biol.* **243**, 574 (1994).
8. S. Henikoff & J. G. Henikoff *Proc. 27th Hawaii Int. Symp. Systems Sci.* 265 (1994).
9. S. Henikoff, J. G. Henikoff, S. Agus & J. C. Wallace, in "Automated DNA sequencing and analysis techniques" vol., (ed.,

- ed.) . Academic Press, 1993.
10. S. Henikoff, J. G. Henikoff, W. J. Alford & S. Pietrokovski (1995)
Submitted for publication.
 11. T. D. Schneider & R. M. Stephens *Nucleic Acids Res.* **18**, 6097
(1990).
 12. S. Henikoff & J. G. Henikoff *Proc. Natl. Acad. Sci. USA* **89**, 10915
(1992).
 13. I. B. Dodd & J. B. Egan *Nucleic Acids Res.* **18**, 5019 (1990).
 14. A. Bairoch *Nucleic Acids Res.* **20**, 2013 (1992).
 15. T. K. Attwood & M. E. Beck *Protein Engineering* **7**, 841 (1994).
 16. E. L. L. Sonnhammer & D. Kahn *Prot. Sci.* **3**, 482 (1994).
 17. R. F. Smith & T. F. Smith *Proc. Natl. Acad. Sci. USA* **87**, 118
(1990).
 18. M. Gribskov, A. D. McLachlan & D. Eisenberg *Proc. Natl. Acad.
Sci. USA* **84**, 4355 (1987).
 19. J. T. L. Wang, T. G. Marr, D. Shasha, B. A. Shapiro & G.-W. Chirn
Nucleic Acids Res. **22**, 2767 (1994).
 20. S. Pongor, V. Skerl, M. Cserzo, Z. Hatsagi, G. Simon & V.
Bevilacqua *Nucleic Acids Res.* **21**, 3111 (1993).
 21. A. Bairoch & B. Boeckmann *Nucleic Acids Res.* **20**, 2019 (1992).
 22. H. O. Smith, T. M. Annau & S. Chandrasegaran *Proc. Natl. Acad.
Sci. USA* **87**, 826 (1990).

23. C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald & J. C. Wootton *Science* **262**, 208 (1993).
24. M. Vingron & P. R. Sibbald *Proc. Natl. Acad. Sci. USA* **90**, 8777 (1993).
25. S. Henikoff *New Biol.* **4**, 382 (1992).
26. M. D'Esposito, G. Pilia & D. Schlessinger *Hum. Mol. Genet.* **3**, 735 (1994).
27. R. Fuchs *CABIOS* **9**, (1993) In press.
28. R. L. Tatusov, S. F. Altschul & E. V. Koonin *Proc. Natl. Acad. Sci. USA* **91**, 12091 (1994).
29. D. M. Nikoloff & S. A. Henry *J. Biol. Chem.* **269**, 7402 (1994).
30. S. F. Altschul, W. Gish, W. Miller, E. W. Myers & D. J. Lipman *J. Mol. Biol.* **215**, 403 (1990).
31. T. F. Smith & M. S. Waterman *J. Mol. Biol.* **147**, 195 (1981).
32. M. O. Dayhoff & R. V. Eck, "Atlas of protein sequence and structure." National Biomedical Research Foundation, Silver Spring, Maryland, 1968 p. 33.
33. M. Dayhoff, "Atlas of protein sequence and structure." National Biomedical Research Foundation, Washington, D. C., 1978 pp. 345-358.
34. S. F. Altschul *J. Mol. Biol.* **219**, 555 (1991).
35. S. Henikoff & J. G. Henikoff *Proteins: Struct. Funct. Genet.* **17**, 49 (1993).

36. R. Luthy, I. Xenarios & P. Bucher *Prot. Sci.* **3**, 139 (1994).
37. M. P. Brown, R. Hughey, A. Krogh, I. S. Mian, K. Sjolander & D. Haussler, *in* "Proc. First Int. Conf. on Intelligent Systems for Molecular Biology" vol., (eds., ed.) 47. AAAI Press, 1993.
38. C. Sander & R. Schneider *Proteins: Struct. Funct. Genet.* **9**, 56 (1991).
39. P. R. Sibbald & P. Argos *J. Mol. Biol.* **216**, 813 (1990).
40. J. D. Thompson, D. G. Higgins & T. J. Gibson *CABIOS* **10**, 19 (1994).
41. C. O. Pabo & R. T. Sauer *Ann. Rev. Biochem.* **61**, 1053 (1992).
42. S. R. Eddy, G. Mitchison & R. Durbin *J. Comput. Biol.* (1995) In press.
43. M. Vingron & P. Argos *CABIOS* **5**, 115 (1989).
44. S. F. Altschul, R. J. Carroll & D. J. Lipman *J. Mol. Biol.* **207**, 647 (1989).
45. W. R. Pearson *Prot. Sci* In press.

Fig. 1. A sample entry from Blocks 8.0 for the Glutaredoxin family of proteins as documented in Prosite 12.0. The protein sequences used to make the block were extracted from Swiss-Prot 29. The ID, AC and DE and lines are adapted from Prosite; the ID and DE lines contain a short and longer description of the family respectively. The AC line contains the Blocks Database accession number, BL00195C, which is adapted from the Prosite AC, PS00195. The "C" indicates that this is the third block for the family in the blocks database, the preceding blocks will end in "A" and "B". The "distance from previous block=(4,14)" means that among the 11 sequences included in BL00195C, the minimum distance from the end of BL00195B to the beginning of BL00195C was 4 residues and the maximum was 14. The BL line contains the name of the initial motif from the MOTIF program around which this block was constructed, the width of the block (31), the number of sequences in the block (11), the 99.5 percentile score of presumed true negative sequences when this block was searched against Swiss-Prot 29 (581), and the median calibrated score of known true positive sequences, called strength, from the same search (2254). During a search of a sequence against the Blocks Database, the raw score is divided by the 99.5% score so that diverse blocks can be compared, and the strength score is reported as an aid to evaluating a possible hit. Following the BL line are the aligned segments of the 11 sequences included in the block. The Swiss-Prot sequence name is followed by the location of the first residue in parentheses, the segment, and the position-based sequence weight. The sequence weights are normalized so that the most distant sequence receives a weight of 100. The segments are clustered into groups separated by a blank line. Segments appear in the same cluster if any two of them have at least 80% identical residues.

Fig. 2. Position-specific scoring matrix (PSSM) computed for the block in Fig. 1 using position-based weights and odds ratios. The PSSM is rotated for display with the 31 columns of the block as rows and one column for each possible amino acid, plus values for B, Z, X, - and *. The conserved P in the 16th column and G in the 27th receive the maximum score of 99. During a search of a sequence against the Blocks Database, the block is aligned with the query sequence and the alignment scored by adding values

from a PSSM such as this.

Fig. 3. Nomogram that is returned with search results that is used to evaluate how likely any particular hit could have occurred by chance. The "anchor block" is the single highest-ranking block in the set of blocks from a single family, and it is provided with a percentile score. A nomogram of this type is general for any sequence query and any version of the Blocks Database, but is specific for the methods used to construct PSSMs representing blocks in the database (Fig. 2). This nomogram is for PSSMs based on position-based weights and pseudo-counts derived from BLOSUM 62 (unpublished results), which is currently implemented in Blocksearcher; it differs from the nomogram reported previously for PSSMs based on 80% clustering weights and odds ratios⁶. The solid symbol shows the intersection of the 80th percentile and $P=0.002$ found for hit 2 using INO2_YEAST as query (see Fig. 4).

Fig. 4. Actual search results returned by the e-mail (or WWW server) when the sequence of INO2_YEAST was submitted, showing the top 2 (of 7) hits reported. The second hit consists of an anchor block (BL00038) and one supporting block (BL00038B) representing the helix-loop-helix family of DNA-binding proteins. Note that this hit falls near the 1/1000 border of the nomogram (middle line in Fig. 3), although because it is the second-ranking hit, confidence that it is a true positive must be less than if it were a first ranking hit. Each numbered result consists of one or more blocks from a Prosite group detected by the query sequence. For each anchor block, BLOCKSORT analyses the highest-scoring set of supporting blocks that are in the correct order and are separated by distances along the query sequence comparable to those along the sequences comprising the blocks. If such supporting blocks exist, then the probability that they support the anchor block is reported. Note that no supporting blocks were detected for the first hit, and so a probability was not calculated. Maps of the database blocks and query sequence are shown, where "AAA" represents the first block roughly in proportion to its width, colons represent the minimum distance between blocks in the database, periods represent the maximum distance between blocks in the database and "< >" indicate the sequence has been

truncated to fit the page. The query map is aligned on the highest scoring block. Multiple block hits that are consistent with the highest scoring block are separated by colons. Block hits that are not consistent are mapped below. The alignment of the query sequence with the sequence closest to it in the BLOCKS database is depicted below the map. The distance between detected blocks is listed as "(min, max):" for the database entry followed by the distance in the query. The distance to the first detected block is also shown. Upper case in the query indicates at least one occurrence of the residue in that column of the block. This search was returned in 2 minutes after receipt; longer protein sequences and DNA sequences, which must be translated in all 6 frames, will require proportionally longer times.

Fig. 5. The BLOSUM 62 amino acid substitution matrix in half-bit units. The cluster percentage of 62 provides a good general purpose matrix for local alignment applications, with relative entropy 0.7 bit and an expected value of -0.5 bit. Matrices in Blosum series are available in various formats and scales by anonymous ftp.

Fig. 6. Evaluation of sequence weighting methods. A set of 1679 blocks from Blocks v. 5.0 (based on Prosite v. 8.0 keyed to Swiss-Prot 22) were converted to PSSMs using various sequence weighting methods and odds ratio scoring. Each block was then used to search Swiss-Prot v. 29 which corresponds to Prosite 12.0, from which lists of true positive sequences were obtained. All 1679 blocks had been updated with new sequences in Prosite v. 12.0, so that these new sequences were necessarily absent from the block queries. The sequence weighting methods tested were: PB, position-based⁷; VOR, modified Voronoi^{24 41}; BP, branch proportional⁴⁰; MD, maximum discrimination⁴²; VA, Vingron-Argos⁴³; ACL, Altschul, Carroll and Lipman⁴⁴; 80%, 80% clustering¹². All test PSSMs are compared against a PSSM made with equal sequence weights and SWISS-PROT amino acid frequency weights. The solid bars represent the number of test PSSMs for which performance was better than performance of the corresponding equal-weighted PSSMs. The hatched bars represent the number of equal-weighted PSSMs for which performance was better than for test PSSMs. Here, performance of a PSSM was measured as the

"equivalence number"⁴⁵, which is the point at which the number of true positive sequences equals the number of true negative sequences, so that a lower equivalence number reflects better separation of true positives from true negatives in the vicinity of the "twilight zone". These results are very similar to our results reported previously using different search and evaluation criteria⁷, except that maximum discrimination weights⁴² were not available at that time.

```

ID   GLUTAREDOXIN; BLOCK
AC   BL00195C; distance from previous block=(4,14)
DE   Glutaredoxin proteins.
BL   VIG motif; width=31; seqs=11; 99.5%=581; strength=2254
GLRX_ECOLI ( 45) KEDLQQKAGKPVETVPQIFVDQQHIGGYTDF 74

    THIO_BPT4 ( 51) LLTKLGRDTQIGLTMQVFAPDGS HIGGFDQ 100

YRUB_CLOPA ( 37) KEREEMRSLSKQSGVPVINIDGNIIVGFNKA 94

GLRX_BOVIN ( 55) EIQDYLQQLTGARTVPRVFIGQECIGGCTDL 27
GLRX_HUMAN ( 55) EIQDYLQQLTGARTVPRVFIGKDCIGGCSDL 25
    GLRX_PIG ( 55) EIQDYLQQLTGARTVPRVFIGKECIGGCTDL 26
GLRX_RABIT ( 55) EIQDYLQQLTGARTVPRVFLGKDCIGGCSDL 28

GLRX_VACCC ( 56) ELRDYFEQITGGRTVPRIFFGKTSIGGYSDL 31
    GLRX_VARV ( 56) KLHDYFEQITGGRTVPRIFFGKTSIGGYSDL 34

GLRX_YEAST ( 60) EIQDALEEISGQKTVPNVYINGKHIGGNSDL 39
YCD5_YEAST ( 61) DIQAALYEINGQRTVPNIYINGKHIGGNDDL 50

```

Figure 1

```

ID    GLUTAREDOXIN; MATRIX
AC    BL00195C; distance from previous block=(4,14)
DE    Glutaredoxin proteins.
MA    VIG motif; width=31; seqs=11; 99.5%=581; strength=2254
A    B    C    D    E    F    G    H    I    K    L    M    N    P    Q    R    S    T    V    W    X    Y    Z    *    -
0    6    0    11   33    0    0    0    0    41   13    0    0    0    0    0    0    0    0    0    0    0    20    0    0
0    0    0    0    33    0    0    0    44    0   22    0    0    0    0    0    0    0    0    0    0    0    20    0    0
0    7    0    12    0    0    0    13    0    0    0    0    0    0    40   20    0   14    0    0    0    0    16    0    0
7   25    0    46   17    0    0    0    0    20    9    0    0    0    0    0    0    0    0    0    0    0    10    0    0
11  0    0    0    14    0    0    0    0    0   10    0    0    0    17    0    0    0    0    0    0    48   15    0    0
0    0    0    0    0   15   13    0    0    0   19   36    0    0    17    0    0    0    0    0    0    0    7    0    0
0    0    0    0    15    0    0    0    0   12    0    0    0    0   24   34    0    0    0    0    0    14   19    0    0
10  10    0    19   14    0    0    0    0    0    0    0    0    0   43    0   13    0    0    0    0    0   25    0    0
0    0    0    0    0    0   13    0   36    0   28    0    0    0    0    0    22    0    0    0    0    0    0    0
0    5    0    0    0    0    0    0    0   13    0    0   12    0   25    0   19   30    0    0    0    0   10    0    0
0    0    0    0    0    0   42    0   21   19    0    0    0   17    0    0    0    0    0    0    0    0    0    0
15  0    0    0    0    0   25    0    0    0    0    0    0    0   48    0    0    0   12    0    0    0   19    0    0
0    0    0    0    14    0    0    0    0    8   13    0    0    0    0   49   16    0    0    0    0    0    9    0    0
0    0    0    0    0    0   15    0    0    0    0    0    0    0    0    0    84    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0   39    0    0    0    0    0    60    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0   99    0    0    0    0    0    0    0    0    0    0
0    8    0    0    0    0    0    0    0    0    0    0   18    0   39   29    0    0   13    0    0    0   15    0    0
0    0    0    0    0    0    0    0    0   57    0    0    0    0    0    0    0    42    0    0    0    0    0    0
0    7    0    0    0   63    0    0    0    0    0    0   16    0    0    0    0    0    0    0    0    20    0    0
14  0    0    0    0    18    0    0   51    0    3    0    0    0    0    0    0    12    0    0    0    0    0    0
0   28    0   33    0    0   25    0    0    0    0    0   21   20    0    0    0    0    0    0    0    0    0    0
0   11    0   20    0    0   27    0    0   26    0    0    0    0   26    0    0    0    0    0    0    0   10    0    0
0   15    0   10    8    0   14    0    0   15    0    0   21    0   18    0    0   11    0    0    0    0   12    0    0
0    0   34    0    0    0    0   42   10    0    0    0    0    0    0    13    0    0    0    0    0    0    0
0    0    0    0    0    0    0   36   63    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0   59    0   23    0    0    0    0    0    0    0    0    18    0    0    0    0    0
0    0    0    0    0    0   99    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    5   37    0    0   15    9    0    0    0    0    0   12    0    0    0    0    0    0    0    0    27    0    0
0   14    0    9    0   25    0    0    0    0    0    0   21    0    0    0    22   22    0    0    0    0    0    0
0   45    0   83    0    0    0    0    0   16    0    0    0    0    0    0    0    0    0    0    0    0    0    0
14  0    0    0    0   22    0    0    0    0   33    0    0    0   29    0    0    0    0    0    0    0    11    0    0

```

Figure 2

Query=INO2_YEAST ,
 Size=304 Amino Acids
 Database=mats.dat, Blocks Searched=2884

```
1.-----
Block      Rank Frame Score Strength  Location (aa) Description
BL00482A   1    0  1084  1419      149-    161 Dihydroorotase proteins.
```

1084=86.22th percentile of anchor block scores for shuffled queries
 P not calculated for single block BL00482A

```
      |--- 107 amino acids---|
BL00482 AAA:::B:::.....CC
INO2_YEAST AAA
```

```
BL00482A   <->A   (11,1486):148
PYR1_DICDI 1437   DVHVHLREPGATH
           |||||
INO2_YEAST 149   esHLHiRSPKkqH
```

```
2.-----
Block      Rank Frame Score Strength  Location (aa) Description
BL00038A   2    0  1076  1277      237-    260 Myc-type, 'helix-loop-he
BL00038A   389   0   922  1277      248-    271 Myc-type, 'helix-loop-he
BL00038B   23    0   979  1264      271-    291 Myc-type, 'helix-loop-he
```

1076=80.80th percentile of anchor block scores for shuffled queries
 P<0.0021 for BL00038B in support of BL00038A

```
      |--- 38 amino acids---|
BL00038 AAAAAAAAAAAAAAAAAA:::.....BBBBBBBBBBBBBBB
INO2_YEAST AAAAAAAAAAAAAAAAAA:::BBBBBBBBBBBBBBB
INO2_YEAST <      AAAAAAAAAAAAAAAAAA
```

```
BL00038A   <->A   (0,667):236
CBF1_YEAST 223   RKDSHKEVERRRRENINTAINVLS
           || | | | | | | | | |
INO2_YEAST 237   RKwKHVqMEKIRRiNtKEAFERLi
```

```
BL00038B   A<->B   (4,46):10
AST4_DROME 143   HKKISKVDTLRIAVEYIRSLQ
           | | | | | | | | |
INO2_YEAST 271   GKRIpKhILLTcVMNdIKSIR
```

Figure 4

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

Figure 5

Figures 3 and 6 follow ...



