Chapter 5
Protein family-based methods for homology detection and analysis

Steven Henikoff and Jorja Henikoff
Howard Hughes Medical Institute, Fred Hutchinson Cancer Research Center, Seattle, USA.

1 Introduction

1.1 Expanding protein families

Most methods for homology detection have traditionally relied upon pairwise comparisons of protein sequences, and in recent years, several improvements in pairwise methods have been introduced (see Chapter 8). But, with sequence data becoming available at an accelerating rate, there is an increasing opportunity to use multiple related sequences for improved homology detection. Even when functional information is lacking for known members of a protein family, these members can be aligned and the alignments used in searches. Protein multiple alignments have been shown to improve performance of secondary structure prediction methods by identifying constraints on positions (1, 2), and so it seems reasonable to expect that improvements will be likewise obtained by using multiple alignments for homology detection and analysis. In this chapter we review some of the numerous methods that are aimed at achievement of this goal.

1.2 Terms used to describe relationships among proteins

Any region of shared similarity between sequences may be referred to as a 'motif'. To call something a motif does not necessarily imply that shared similarity reflects shared ancestry. For example, the well-studied helix-turn-helix DNA binding motif is found in proteins belonging to apparently unrelated families with different origins, and this suggests convergence towards a common structure.

Confusion often arises from the use of the structural term 'domain' to describe regions of sequence similarity. A separately-folded domain may be obvious from looking at the structure of a protein, but, without seeing a structure, it may not be possible to decide from an alignment what is the limit of a domain.

Furthermore, domains need not be contiguous along a sequence, and it is common for proteins to fold starting with one domain, continue on to fold into another domain, then return to the original domain further along the sequence. For sequence analysis applications, a useful concept is that of 'module'. A module can be thought of as a sequence segment that may be found in different contexts in different proteins, the result of mobility during protein evolution. Modules may correspond to separately folded domains, such as the C_2H_2 zinc finger motif, and they may be repeated within a sequence. Unlike domains, modules are necessarily contiguous along a sequence. Nevertheless, readers should be aware that modules identified by sequence similarity are typically referred to as 'domains' without confirming structural evidence, and the term 'multi-domain' is commonly applied to any chimeric protein.

'Family' is a generic term used to describe proteins (or genes) with sufficiently high sequence similarity that common ancestry may be inferred. A multi-domain protein might have modules that belong to several different families. Confusion can arise from the use of terms such as 'superfamily' and 'subfamily', which are not precisely defined. For these terms to be useful, some sense of what is meant by a family is required. Thus, if we refer to the opsins, the beta-adrenergic receptors and the olfactory receptors as separate families, even though they are related to one another, then we would refer to the G-protein coupled receptor superfamily to describe them all. Sometimes, proteins fold similarly, even though no sequence similarity between them is detected. For instance, the TIM barrel fold has been found for dozens of separate superfamilies, and it is not certain as to whether they share common ancestry. Conversely, sequence similarity may be evident, even though common ancestry is doubtful, as in the case of coiled-coil regions of proteins. As a practical matter, the methods described in this chapter are most useful for families and modules, where alignment-based methods can provide profound functional insights.

1.3 Alternative approaches to inferring function from sequence alignment

Opposing views of sequence alignment problems have resulted in two different classes of comparison tools for sequence analysis of protein families. Motif-based tools consider aligned protein sequences to consist of nuggets of alignment information (blocks) separated by regions that have no certain alignment. To proponents of this view ('blockers'), the task is to first find these conserved nuggets. 'Gappers' agree that there are nuggets worth finding, but that these will be best found by determining where to place the gaps in each sequence such that the blocks correctly align. Both blockers and gappers agree that aligning conserved nuggets is worthwhile, but they use different methods for accomplishing this. Blockers favour motif-based methods that first find regions of conservation. Such block-based methods as the BLAST family of searching programs and the BLOSUM amino acid substitution matrices continue to be favoured for many comparative sequence analysis applications (Chapter 8). Gappers favour methods that decide upon gap placement (described in Chapter 3) and use gap-based tools, especially dynamic programming and hidden Markov models (described in Chapter 4), for database probing. As is so often the case, the truth lies somewhere between the extremes. So although we blockers prefer to reduce the protein alignment problem to finding a set of ungapped blocks to represent a protein family or module, we recognize that insertions and deletions occur occasionally within conserved regions, and this is challenging for block-based methods.

Alignment usefulness is the major driving force in developing methodology. Obtaining a correct alignment is more important for some applications than for others. The ability to find corresponding residues and local regions that have similar functions is of unquestionable value, and the better the conservation of a residue or local region in a sequence, the more likely it is that common function can be inferred. Regions of uncertain alignment, such as those that different alignment programs using various score parameters disagree on, have little if any value for drawing functional inferences. However, so much alignment information is present in conserved regions that it might make sense to align beyond what can be done with confidence in order that more nuggets are captured. We suspect that this accounts for the success of many gap-based approaches: gapped alignments may have a high degree of uncertainty, but the proportion that is aligned successfully is sufficient to identify extensive shared regions of sequence similarity in database searches, even to the point of discovering correct folds more successfully than structure-based threading (3).

Practical utility requires ready availability to the general public. Nowadays, this means access via the World Wide Web using a browser, and so nearly all methods highlighted here (*Table* 1) can be performed without any special software, hardware, or computational expertise. Some potentially powerful tools are too computationally intensive to be made available in this way. Additionally, some tools require a specialist's knowledge and are not sufficiently automated for the average biologist to use them wisely. We believe that such tools should be avoided if possible: sequence alignment is fraught with hazards, and erroneous conclusions drawn from naive use of powerful sequence analysis tools abound (4).

2 Displaying protein relationships

2.1 From pairwise to multiple-sequence alignments

Depictions of pairwise sequence alignments are not easily extended to multiple alignments. For displaying pairwise alignments, identities and conservative replacements are typically emphasized using symbols between the aligned sequences. However, adding just a third sequence below the first two leaves open the question of how to represent similarities between the third sequence and the first, and addition of more sequences becomes increasingly complex. Dot matrix representations of pairwise alignments present the same problem.

Table 1 URLs

1. Displaying alignments	
Boxshade	http://www.ch.embnet.org/software/BOX_form.html
Logos	http://blocks.fhcrc.org/about_logos.html
Trees	http://blocks.fhcrc.org/about_trees.html
2. Finding alignments	
BCM launcher	http://dot.imgen.bcm.tmc.edu:9331/multi-align/multi-align.html
MACAW	ftp:/ncbi.nlm.nih.gov/repository
3. Searching family databases	
Prosite	http://www.expasy.ch/prosite/
Blocks	http://blocks.fhcrc.org/blocks_search.html
Prints	http://www.bioinf.man.ac.uk/fingerPRINTScan/bin/attwood/SearchPrintsForm2.pl http://blocks.fhcrc.org/blocks_search.html
ProDom	http://www.toulouse.inra.fr/prodom/doc/blast_form.html
Pfam	http://pfam.wustl.edu/hmmsearch.shtml http://www.sanger.ac.uk/Pfam/search.shtml
Proclass	http://www-nbrf.georgetown.edu/gfserver/genefind.html
ProfileScan	http://www.isrec.isb-sib.ch/software/PFSCAN_form.html
Identify	http://dna.Stanford.EDU/identify/
Recognize	http://dna.stanford.edu/ematrix/
Prof_Pat	http://wwwmgs.bionet.nsc.ru/mgs/programs/prof_pat/
4. Searching with multiple alignments	
MAST	http://meme.sdsc.edu/meme/website/mast.html
COBBLER	http://blocks.fhcrc.org/
PSI-BLAST	http://www.ncbi.nlm.nih.gov/cgi-bin/blast/psiblast.cgi
LAMA	http://blocks.fhcrc.org/LAMA_search.html

Such displays not only become complex, but also they fail to represent shared similarities. Because of these limitations, multiple alignment representations that emphasize regions of high similarity have been introduced.

Traditional displays of multiple sequence alignments show aligned sequences one above the next, highlighting identical or similar residues in a column using boxes, shading, or colour. These displays can be complex, especially when representing protein families that consist of large numbers of sequences that group into distinct subfamilies. Position-based representations greatly simplify the display of multiple alignments, because relationships between successive amino acids in a sequence are not shown. Indeed, computer programs that utilize multiple sequence alignment information in searches likewise consider all positions in aligned sequences to be independent of one another, and so position-based representations depict approximately what a searching program examines.

2.2 Patterns

The simplest position-based representations of multiple alignments are patterns, which display only key conserved residues. The Prosite database (5) is a compilation of sequence families that provides one or more patterns representing each family. An example of a Prosite pattern is C-x(2,4)-C-x(3)-[LIVMFYWC]-x(8)-H-x(3,5)-H, which is read as: cysteine, followed by 2 to 4 amino acids of any type, followed by cysteine, followed by any 3 amino acids, followed by one of the following—leucine, isoleucine, valine . . . , and so on. Although Prosite patterns are manually derived from multiple sequence alignments, the process of determining patterns from alignments has been automated (6, 7). Pattern-based methods are described in detail in Chapter 7.

2.3 Logos

Sequence logos (8) are vivid graphical displays of multiple sequence alignments consisting of ordered stacks of letters representing amino acids at successive positions (*Figure 1*). The height of a letter in a stack increases with increasing frequency (or probability) of the amino acid, and the height of a stack of letters increases with increasing conservation of the aligned position. Stack heights are displayed in bit units. One bit is the answer to a yes-or-no question, where yes is as likely as no. About 4 bits are required to fully specify a residue at a given position, because the first question narrows the field from 20 residues to 10, the second to 5, etc. The most probable amino acid is at the top of the stack, making it more visible, and below it is the next most probable residue, and so on. Logo colours or shades are chosen to emphasize similar amino acid properties. Logos can be scaled such that the stack height is proportional to the observed frequency of a residue divided by the frequency with which the residue is expected to occur by chance (odds ratio).

2.4 Trees

The most serious drawback of position-based displays is that they show only alignment information in common among the sequences in a family, not the

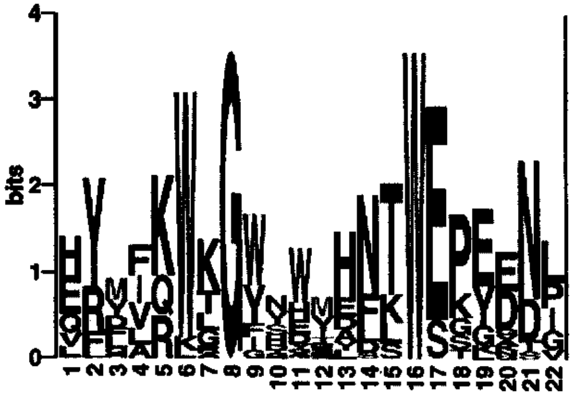


Figure 1 Sequence logo depicting the chromodomain block (BL00598 in Blocks v. 11.0). Each alignment position is represented as a stack of letters, where the height of the stack and the height of each letter is measured in bit units.

differences that distinguish between sequences or groups of sequences. In contrast, trees are designed to discriminate between individual sequences by providing an intuitive diagram of relationships drawn from an alignment. Although trees were introduced as phylogenetic tools, they have become increasingly popular for displaying protein families. Trees can generally be used to distinguish orthologs from paralogs, because orthologs will branch in a manner that is consistent with species phylogeny, whereas paralogs may deviate. Paralogous proteins often have distinct biological activities, and so trees can guide experimental investigations.

For using trees to draw inferences about function and not to infer phylogeny, some aspects of tree construction and interpretation that matter to phylogenetics may be relatively unimportant. Debate continues about how a tree should be constructed from an alignment, whether to use parsimony, distance, or maximum-likelihood methods. However, we are unaware of evidence that the choice of a tree-making program matters much for distinguishing paralogs from orthologs or for deciding whether one branch has a function that is comparable to the function of another branch. That is, we are not using a tree to distinguish whether bakers yeast is closer to fruit flies than to maize, so we can ignore the details at the leaves of a tree and focus on the separation of one group of yeast, fly, and maize proteins from paralogous groups of proteins. In our experience, the quality of the alignment might be important in making such distinctions, but different tree-making programs draw trees that are sufficiently similar for our purposes. Distance methods such as neighbour-joining (9) do have the advantage of being fast and can then be applied to large numbers of proteins and thus are suitable for making trees to analyse protein families.

3 Block-based methods for multiple-sequence alignment

Searching methods that utilize multiple sequence alignment information can be block-based or gap-based. Gap-based methods for finding multiple alignments are described in detail in Chapter 3, and gap-based methods for searching with them, such as hidden Markov models (HMMs), are described in Chapter 4. In this section, we describe block-based strategies for finding multiple-sequence alignments that are then used for database searching by many of the methods described in subsequent sections.

3.1 Pairwise alignment-initiated methods

One general approach to finding motifs involves performing pairwise comparisons between sequences and then asking which high-scoring local regions are in common for most or all of the sequences in the group. Where aligned segment pairs overlap, they are multiply aligned. However, determining which segments truly overlap can be challenging, and different methods have been introduced (10–12). In the MACAW program, overlapping segment pairs that exceed a threshold score are combined into an ungapped block (12). The extent of the block is limited by the requirement that each column have some minimum degree of homogeneity. Blocks that are separated by the same number of residues in all sequences may be fused, and so blocks can contain both conserved and diverged positions. MACAW is an interactive program that allows users to choose a set of blocks from among candidates. The threshold score for block searching can be relaxed by the user in order to find new blocks in regions between blocks that were found in the first pass.

Starting with pairwise alignments presents the same potential drawback as for gap-based hierarchical multiple sequence alignment programs (Chapter 5), which is that information in common for all of the sequences might not be represented in the pairwise alignments. In addition, the number of pairwise comparisons needed is n^2 for n sequences, and this can become somewhat impractical for large protein families and long sequences. Simultaneous methods for finding motifs, described below, can potentially avoid these problems.

3.2 Pattern-initiated methods

The rapidity with which amino acid 'words' can be scanned exhaustively through a set of related sequences has motivated pattern-based motif finders (13–17). An example of this approach, Motif (15), examines all sequences for the presence of spaced triplets of the form aa_1 d_1 aa_2 d_2 aa_3 where d_1 and d_2 are fixed distances between the amino acids. So Ala–Ala–Ala is one triplet, Ala–x–Ala–Ala is another, and Leu–x[16]–Ala–x[7]–Val is another. An exhaustive search is carried out for all such triplets in the full set of related sequences using all combinations of d_1 and d_2 out to a reasonable maximum distance (about 20). The rationale is that true motifs will typically include one or more sets of spaced triplets in all of the sequences in the group. Because some true motifs do not contain aa_1, aa_2, and aa_3 in the full set of sequences, the number of sequences required to contain a triplet (the 'significance level') can be reduced. In such cases, the block containing the triplet is scanned along each of the sequences that lack the triplet to find the best segment based on maximizing an overall score for the block. Each sequence is then rescanned to maximize the score. ASSET generalizes the search for patterns by scanning sequences for shared flexible patterns that occur in multiple sequences at a statistically significant level (17).

3.3 Iterative methods

Other approaches avoid limiting the motif search to a predetermined list without becoming computationally explosive by detecting motif 'seeds' that occur in as few as two sequences, then asking whether any of these seeds can mature to include other sequences in the group (18–21). Both Expectation-maximization (EM) and 'Gibbs sampling' (20) start with a block of specified width, then align random positions within all but one sequence. In EM, this sequence is scanned along the block, and the segment that maximizes a block score is chosen. In Gibbs sampling, the segment is chosen by a random sampling procedure, where the probability of being chosen is proportional to the block score. Other sequences are then sampled in the same way to further improve the significance of the alignment. Successive rounds of EM or Gibbs sampling continue until no further improvement is seen.

3.4 Implementations

Some of these methods are conveniently available over the internet, and sequences in FASTA format may be submitted by either pasting into a window or by file browsing. Because many real motifs can be subtle and as short as a few residues, sensitive methods may return alignments for sequences that are not based upon true relationships. A simple experiment (*Protocol 1*) demonstrates that even sequences chosen at random from a database can be aligned to yield motifs that appear convincing and will easily detect the parent sequences and their homologues from sequence databanks. Furthermore, even gross misalignments can be masked by the existence of significant similarity among just a fraction of sequences, and visual examination is notoriously unreliable (*Figure 2*). One solution is to report a reliability measure for each position (22), and one measure is implemented in Match-Box (23). BlockMaker's solution (24) is to apply two very different motif finders with different scoring systems, Motif and Gibbs sampling. In each case, a block assembly algorithm (25) is used to determine a best set of blocks representing a protein family, and the two sets are compared by the user: blocks with similar alignments obtained by the two methods may be trusted, but those that differ require scrutiny. Both Match-Box and BlockMaker require that the blocks be in order along the sequences, and so repeats might be missed. However, the EM-based program, MEME (21), does not impose an ordering criterion, and MEME finds repeats and displays them within blocks. BlockMaker, MEME and Match-Box are available from the BCM multiple alignment search launcher, which allows successive searches of a single query with several tools, both traditional and motif-based. Performance evaluation of the methods available over the Web show that there are trade-offs between sensitivity and reliability (23), and so it is worthwhile to try several methods on any particular set of sequences and compare the results.

Protocol 1

Finding motifs from unaligned sequences and searching sequence databanks

1 Go to the Swiss-Prot Random-entry retriever (http://www.expasy.ch/sprot/get-random-entry.html) and successively extract 10 Swiss-Prot sequences of length > 300 aa residues in FASTA format (look for the 'FASTA format' link at the bottom of the page).

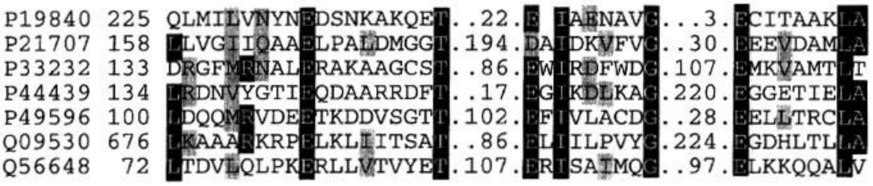
2 Copy and paste these sequences into the large box of the BCM alignment launcher (http://dot.imgen.bcm.tmc.edu:9331/multi-align/multi-align.html). Choose BlockMaker and submit.

Protocol 1 continued

3 From the BlockMaker results page, examine the alignments in both sets of blocks (from Motif above and Gibbs below), and choose the set of blocks that has the most total residues. Click on the MAST direct link (to http://meme.sdsc.edu/meme/website/mast.html) above the chosen set. MAST search results will be returned by e-mail.

4 Compare the names of your submitted sequences to the significant hits from your MAST search. Other significant hits may be homologues of your submitted sequences.

5 Now that you have done the necessary control, you are ready to use your own sequences. Return to the BCM alignment launcher, copy and paste your own sequences into the box and successively click on the various choices of multiple aligners.



```
P19840 225 QLMILVNYNEDSNKAKQET..22.E IAENAVG...3.ECITAAKLA
P21707 158 LLVGIIQAAELPALDMGGT.194.DAIDKVFVG..30.EEEVDAMLA
P33232 133 DRGFMRNALERAKAAGCST..86.EWIRDFWDG.107.EMKVAMTLT
P44439 134 LRDNVYGTIEQDAARRDFT..17.EGIKDLKAG.220.EGGETIELA
P49596 100 LDQQMRVDEETKDDVSGTT.102.EFIVLACDG..28.EELLTRCLA
Q09530 676 LKAAARKRPELKLIITSAT..86.ELIILPVYG.224.EGDHLTLLA
Q56648  72 LTDVLQLPKERLLVTVYET.107.ERISAIMQG..97.ELKKQQALV
```

Figure 2 A typical example of a set of MOTIF-generated blocks obtained using Protocol 1. Boxshade was used to highlight 'conserved' positions among the randomly chosen sequences. Using these blocks as input to MAST for searching Swiss-Prot, each of the sequences represented in the block was detected with an E-value between 1.6×10^{-16} and 7.4×10^{-6}. The first unrelated sequence was detected E = 0.212.

The interactive MACAW program provides an excellent alternative to automated web-based multiple aligners. The program allows a user to choose either MACAW or Gibbs sampling for making blocks, and to make parameter choices at different stages in the alignment process. The program is available to run under popular computer operating systems.

4 Position-specific scoring matrices (PSSMs)

Alignments, patterns, logos, and trees provide useful visual displays, but for searching databases, score-based representations are most widely used. These were introduced by McLachlan (26) and popularized by Gribskov *et al.* (27) who coined the term 'position-specific scoring matrix' (or PSSM, pronounced 'possum'). A PSSM consists of columns of weights for each amino acid derived from corresponding columns of a multiple sequence alignment. Other terms have been used to describe this basic idea, including weight matrix, profile and HMM. Profiles are PSSMs constructed using the average score method (27), although the term has also been used to describe matrices representing a string of local environments for successive residues in a structure (28). Profile HMMs, described in Chapter 4, are PSSMs that are constructed using an iterative probabilistic algorithm for determination of position-specific gap penalties (29–31). A simple PSSM has as many columns as there are positions in the alignment, and 20 rows, one for each amino acid. In some applications, a PSSM consists of rows that correspond to successive positions in the alignment (27), rather than columns, and in some, there are position-specific gap scores.

Because they consist of numbers, PSSMs are useful for computer-based alignment and database searching methods but not for visual display. However, logos are computed from PSSMs, and rules can be applied to convert PSSMs to patterns (6) or consensus sequences (32). The construction of PSSMs from multiple alignments has improved over the years, and as a result, we are better able to detect weak similarities in searches (33). To construct effective PSSMs, two major issues, described below, must be addressed.

4.1 Sequence weights

PSSM performance can be improved by differentially weighting sequences to reduce redundancy resulting from non-representative sampling of sequences (34–37). Very similar sequences get low weights and more diverged sequences get higher weights in order to make a PSSM more representative of the family as a whole. Several different strategies have been used to arrive at sequence weights. Some methods start with a tree and find a root, where the weight of a sequence is proportional to its distance from the root (e.g. 35). Pairwise distance methods calculate a weight from the average distance of a sequence to all other sequences, either to the observed sequences or to imaginary sequences derived by sampling residues from the observed sequences (e.g. 36). Position-based sequence weights are calculated by determining the weight of a residue within its column in an alignment and adding residue weights for all positions (37). In the maximum discrimination method, weights are chosen to best discriminate between true positives and background (31). Comprehensive empirical evaluation of sequence weighting methods has revealed that weighting sequences is much better than not weighting at all (37). However, no single method stands out, and at least one variant of each of these strategies provided excellent results.

4.2 PSSM column scores

Pairwise alignment methods utilize amino acid substitution matrices to provide a set of scores for each aligned residue. Current applications utilize log-odds scores computed from alignment data, such as PAM (38), JTT (39), or BLOSUM (40) substitution matrices, and theory advocates the suitability of log-odds scores for pairwise alignments (41). However, scoring a multiple alignment against a sequence is more complex, requiring a scoring scheme that is able to utilize the observed occurrences of residues in a column corresponding to an alignment position. That is, the column of an alignment should be modelled in a way that, when aligned with each of the 20 amino acids, a meaningful score can be obtained. The original average score method (27) simply extends the use of pairwise scores by averaging them. For instance, when aligned with a serine, a position represented by an alanine and 3 cysteines would get a score equal to the serine–alanine score plus 3 times the serine–cysteine score divided by 4 (ignoring sequence weights which would alter the relative contribution of each occurrence to the sum).

Unfortunately the average score is insensitive to the number of sequences in the multiple alignment. The average score for a serine aligned with an alanine and 3 cysteines is identical to that for a serine aligned with 10 alanines and 30 cysteines. The problem with this situation is that a serine might be expected to occur frequently given only 4 observations of such similar residues, but after 40 observations without seeing a serine, we would expect to see one only rarely. Therefore, the average score method becomes less and less realistic as the number of different sequences in an alignment increases. An effective way of dealing with this problem is to add 'pseudocounts' to the observed counts of residue occurrences (42–44). Intuitively, this is equivalent to adding hypothetical sequences to those that have been observed, and for each sequence, the choice of residues at each aligned position is governed by what might be expected for real related sequences not yet seen. So if we have already observed an alanine and 3 cysteines, we might expect to see more cysteines and alanines, but also occasional serines but maybe not arginines. Hypothetical occurrences can be added to real occurrences as fractional pseudocounts. Notice that if we add the same pseudocounts when 10 alanines and 30 cysteines have been seen, then the relative proportion of real observations to pseudocounts increases 10-fold; this conforms with our intuition that we are much more certain that the observed occurrences adequately model future occurrences when we have a large number of independent observations. Comprehensive evaluations demonstrate that using pseudocounts modeled on alignment data, much better overall performance is obtained than using the average score method (33, 44).

5 Searching family databases with sequence queries

For any protein sequence of interest, a search of the latest databanks is the first and often the most important step toward understanding function, and the identification of homologues in this way has been a major driving force in both academic biology and in the growing genomics industry. A second step should involve searching protein family databases. There are several reasons for this: Making sense of dozens or hundreds of hits in the sequence databanks can be challenging, whereas hits in protein family databases provide immediate classification and entries to the literature. The different regions of multi-domain proteins are readily classified using family databases, whereas in searches of sequence databanks, modules can be missed if hits to family members containing them are low on the list. Searches of family databases can be more sensitive than searches of sequence databanks because multiple alignment information is utilized. The much smaller size of family databases, typically only ~1% the size of sequence databanks, reduces noise.

Currently, there are several choices of family databases and searching options available over the internet (*Table 1*). An illustrative example is depicted in *Figure 3*, which shows how well the different methods detected key features of a protein that we recently described, a cytosine-5 DNA methyltransferase homologue with an embedded chromodomain module, called a 'chromomethylase', which is encoded by the *Arabidopsis thaliana* CMT1 locus (45). In addition to being the subject of current experimental work in our group, we chose this sequence because chromomethylases are not yet present in any of the family databases, although both the cytosine-5 DNA methyltransferases and the chromodomains are represented in all of them, and because this example reveals strengths and weaknesses of the different methods especially well. Both the DNA methyltransferase and the chromodomain represent novel subfamilies of their respective families, and so detection in their entirety can be challenging for a protein family classification method that does not generalize well from known examples. This is an anecdotal example, and overall performance can only be judged using

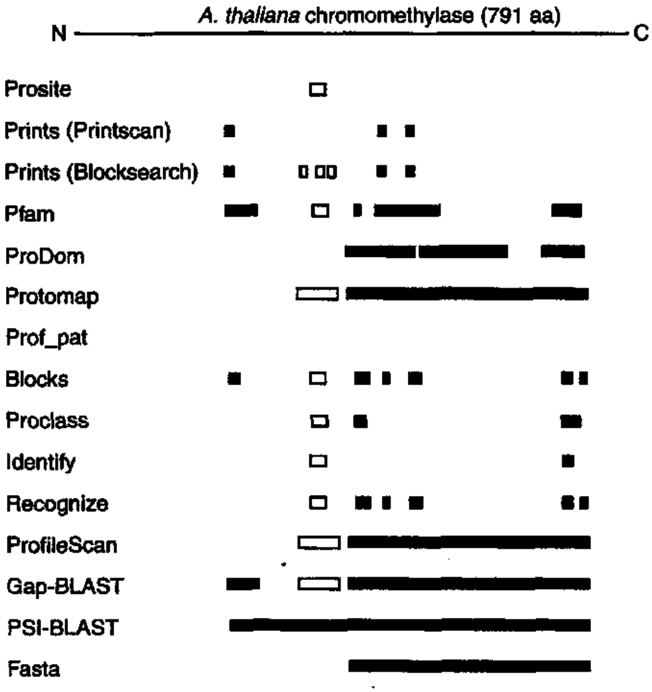


Figure 3 Classification of the 791 aa *A. thaliana* chromomethylase by family databases. The horizontal line indicates the length of the protein from the amino (N) to the carboxyl (C) end, the closed boxes show the extent of cytosine-5 DNA methyltransferase regions detected and the open boxes show chromodomain regions. For methods that report E- or p=values, a 0.05 level of significance was considered to be the threshold for detection, and this exceeded the level of the highest-scoring false positives. For methods that do not report E- or p-value statistics (Prosite, Printscan, Prof_pat, Proclass) or for those that report multiple levels of stringency (Identify and Recognize), the threshold level of detection was considered to be just above the first false positive hit.

comprehensive empirical evaluations. However, because coverage of different databases varies widely, such rigorous direct comparisons have not been carried out.

5.1 Curated family databases: Prosite, Prints, and Pfam

Prosite is the original family database, introduced in 1989. Prosite provides excellent documentation and carefully crafted patterns for searching (*Section 2.2*). In cases where patterns are difficult to find, Prosite provides a profile PSSM (*Section* 4). Prosite 15 (July, 1998) has 1020 documentation entries, mostly representing families, and 1358 patterns based on sequences in Swiss-Prot. Searching a query sequence against Prosite patterns is strictly a hit-or-miss affair, and no statistics are provided. The chromomethylase example illustrates this vividly. Prosite reported the chromodomain, which is a highly diverged module that is relatively difficult to detect. However, Prosite failed to detect the DNA methyltransferase, even though some of the conserved regions are very easily detected by standard searching programs (*Figure* 3) and this family is represented by two patterns in the database. Indeed, a comprehensive empirical evaluation showed that even standard BLAST searching outperforms searching of Prosite patterns (46).

Prints, introduced in 1993, is similar to Prosite in providing excellent documentation. Rather than patterns, Prints provides carefully crafted 'fingerprint' multiple alignments (ordered sets of blocks), that can be searched using pattern or PSSM methods. Prints 20 (October, 1998) has 990 fingerprint entries for 5701 blocks based on sequences in the OWL protein database. Printscan detected all 3 blocks in the fingerprint representing the upstream and central conserved regions of the DNA methyltransferase, but did not detect the chromodomain.

Maintaining curated databases and crafting patterns or fingerprints is made especially difficult because of the rapid expansion of protein families in recent years. Pfam (47), introduced in 1996, addresses this problem by using seed alignments that are manually constructed, and HMM (hidden Markov model) PSSMs from the seeds are then used to automatically extract and align new sequences from databanks. Unlike Prosite and Prints, Pfam does not provide documentation beyond a family name and links to source databases and does not delineate conserved regions within entries. Pfam 3.2 (October, 1998) has 1344 entries representing families and modules based on sequences in Swiss-Prot/TrEMBL. HMM PSSMs are used to search Pfam. For the chromomethylase, all of the conserved regions of the DNA methyltransferase and the chromodomain were detected.

5.2 Clustering databases: ProDom, DOMO, Protomap, and Prof_pat

An alternative to curation is to search a database against itself, then cluster similar sequences into families automatically. Although the procedure sounds simple, in practice it is fraught with difficulties owing to the complexity of proteins and protein families and to the need to avoid chance similarities when comparisons are carried out on such a large scale. The first public database of this type was introduced in 1990 (48), and several have been introduced over the years, only some of which are extant. ProDom, which was introduced in 1994 (49), has been continually maintained and enhanced (50); version 36 (August, 1998) contains 17 777 entries from Swiss-Prot with more than 2 sequences. ProDom entries vary from short single motifs to longer stretches of similarity that might encompass nearly entire sequences. ProDom is searched with multiple alignments or consensus sequences. Using either option, ProDom detected the central and downstream conserved regions of the DNA methyltransferase, missing the upstream region and the chromodomain.

Recently, three new clustering databases have been introduced. DOMO (51), which is based on Swiss-Prot and PIR, is similar to ProDom, although it uses different methodology to generate the database. DOMO clusters tend to be longer and fewer in number than ProDom clusters. At present, DOMO does not allow user-supplied sequences to be searched for classification. Protomap (52), which is based on Swiss-Prot, does not yield multiple alignments as do ProDom and DOMO, but rather provides a graphical tree-like view of the clustering. To classify a protein sequence with Protomap, a Smith–Waterman search of Swiss-Prot is performed, and each individual cluster that contains a sequence hit is reported. For the chromomethylase, Protomap detected the chromodomain and the central and downstream conserved regions of the DNA methyltransferase, missing the upstream region of conservation. Prof_pat (53) extracts patterns from clustering Swiss-Prot/TrEMBL, and these can be searched. Prof_pat did not detect either the DNA methyltransferase or the chromodomain above false positives.

5.3 Derived family databases: Blocks and Proclass

Intermediate between the curated and automated databases are those that utilize protein family groupings provided by other resources. The Blocks Database, which was introduced in 1991, uses the automated Protomat system for finding blocks (ungapped regions of local conservation) representing a protein family. Starting with Swiss-Prot sequences listed in Prosite family entries, alignment blocks are found (patterns or profiles provided with Prosite are not used) and concatenated into a database. Blocks 11.0 (August, 1998) contains 994 families and 4034 blocks based on Swiss-Prot and is searched using the PSSM-based BlockSearch method that reports single and multiple block hits along a sequence. Whereas other protein family searchers on the internet require a protein sequence, Blocks can be searched with a DNA sequence query, in which case hits from all three frames on each strand are assembled. For searching, the current default database is Blocks+, a superset of families from Blocks, Prints, Pfam, ProDom and DOMO. Blocks+ (Nov. 1998) includes 8388 blocks representing 1922 families. Except for Prints, where fingerprint blocks are utilized directly for searching, Protomat is used to make blocks for entries from each database, and families that have block regions in common are removed to avoid redundancy. BlockSearch detected the chromodomain and all of the conserved regions of the DNA methyltransferase. When the Prints database was searched with BlockSearch, all 3 upstream and central DNA methyltransferase motifs and all 3 chromodomain motifs in Prints were now detected at highly significant levels.

Proclass (54), introduced in 1997, also combines families from different sources: Prosite, PIR superfamilies and families automatically discovered using the GenFind program (55). Proclass v. 3 (March, 1998) contains 1275 Prosite groups and 3979 PIR superfamilies and is searched using a neural network-based system. To our knowledge, Proclass searching is the only system that detects sequence similarity using methodology that is not alignment-based. When the chromomethylase sequence was searched, Proclass reported the Prosite chromodomain pattern and both of the Prosite DNA methyltransferase patterns, which were missed by the Prosite scanner.

5.4 Other tools for searching family databases

Identify (7) searches sequences versus pattern-based representations of individual blocks and fingerprints derived from the Blocks and Prints databases. Because patterns can be searched much more rapidly than scored-based representations of multiple alignments (see Chapter 7), Identify search results are returned within a second or so. Identify detected the chromodomain and only one downstream DNA methyltransferase blocks above all false positives. Using Recognize, which is a score-based version of Identify, the central and other downstream regions were detected as well.

A collection of profile PSSMs from Prosite, Pfam and other sources is available for searching using generalized HMM-like profile PSSMs at the ProfileScan site (56). ProfileScan reported the chromodomain and the central and downstream DNA methyltransferase conserved regions but missed the conserved upstream region.

In summary, there are numerous protein family searching tools available for sequence classification. None is perfect, and as illustrated by the chromomethylase example, it is worthwhile to try several of them for analysing a sequence of interest. Pairwise sequence tools also varied in their ability to confidently detect features of the chromomethylase. GAP-BLAST detected the chromodomain and all DNA methyltransferase conserved regions, although subsequent iterations of PSI-BLAST caused the chromodomain to be lost at the expense of the DNA methyltransferase that surrounds it. FASTA failed to detect the upstream conserved region of the DNA methyltransferase, and the chromodomain was reported, but at a non-significant level. As a practical matter, the chromodomain would have gone unnoticed or assumed to be a chance hit because it is preceded by ~100 higher-scoring DNA methyltransferase sequences in databanks, and indeed its presence was not noted in the original sequence entry (GenBank/EMBL U53501). By using family databases for classification, this potential problem can be minimized.

6 Searching with family-based queries

Finding homologues in sequence databanks underlies much of the recent progress in functional genomics, both in academia and industry, and pairwise methods, such as BLAST searching, currently dominate. However, as more and more sequences fall into families, opportunities increase for using family information for identifying modules and new family members. Progress in making better PSSMs described in *Section 3.3* has resulted in improvements in searching performance, and practical tools have become available for taking advantage of protein family information in searching sequence databanks.

6.1 Searching with embedded queries

A potential drawback to block-based approaches is that regions of uncertain alignment are not scored, and the loss of this alignment information can potentially reduce searching sensitivity. This problem arises because even with effective motif-finding systems, the 'edges' of blocks are often uncertain, and they might be chosen differently for different subsets of proteins in a family (51, 57, 58). This problem has been addressed by implementation of a simple 'embedding' strategy: a consensus is determined for a set of related sequences, the sequence that is closest to the consensus is chosen, and blocks are embedded into that sequence (46). Because interblock regions of uncertain alignment are represented as a single sequence, they cannot be misaligned (this would reduce the specificity of a PSSM), while multiple alignment information in block regions is retained. Embedding of PSSMs using this system has not been implemented on the internet for general database searching, although the basic idea has been incorporated into PSI-BLAST (described below). As an approximation, using the COBBLER (COnsensus Biasing By Locally Embedding Residues) system, a consensus residue is determined for each position of all the blocks. A single sequence is chosen as the one closest to the consensus over all block positions, and these consensus residues are then substituted for the real residues in the chosen sequence. This consensus-biased sequence can then be used to search sequence databanks using available single sequence querying tools, such as BLAST and PSI-BLAST. The improved overall performance that results is especially useful for identifying known modules in unexpected places: For instance, the chromodomain in the *A. thaliana* chromomethylase (*Figure 3*) was initially identified using a COBBLER-embedded sequence to search the nr protein databank with BLAST (45).

6.2 Searching with PSSMs

Using PSSMs to search sequence databanks is computationally demanding, and the availability of services is relatively limited. The Multiple Alignment Searching Tool (MAST) program (57) searches block-based multiple alignments against the standard sequence database sets, which are updated daily. MAST output provides excellent statistics for both individual and multiple block hits with block maps for intelligent interpretation of search results. MAST accepts PSSMs directly from MEME and BlockMaker. Additionally, the Blocks server provides a processor that can be used to convert other multiple alignments into efficient PSSMs for sending directly to the MAST server.

6.3 Iterated PSSM searching

Several of the concepts highlighted above have been incorporated into the Gap/PSI-BLAST searcher, an elegant extension of the popular BLAST database searching program (59). The first round of searching employs Gap-BLAST, a new pairwise method for detecting family members in the traditional way. From the significant hits detected in the first round, a PSSM is constructed and this is used to search the databank again, a process that can be repeated multiple times until no further hits are reported above a chosen level of significance.

Gap-BLAST is especially notable because it represents a successful block-based approach to the pairwise searching problem. When databanks are searched, computational speed is an important factor, because finding an 'optimal' alignment using traditional methods for placing gaps is too slow to be practical on a large scale with standard hardware. Speedy methods, such as FASTA and BLAST, begin by searching exhaustively for matches or short motifs shared by two sequences, extending these and stringing them together to find high scoring alignments. However, the gain in speed is accomplished at the expense of reduced searching performance (60). Gap-BLAST combines speed with near-optimal searching performance by starting with short motifs, but accepting only those that define opposite ends of a high scoring ungapped alignment. This alignment is extended, and only if it exceeds a threshold score is a gapped alignment sought, that is, gapping is employed to optimize alignment of highly similar regions. Searching performance of Gap-BLAST is nearly indistinguishable from that of an optimal gap placement method (Smith–Waterman dynamic programming) when the same scoring parameters are used. This is one inroad of blocker concepts into the gapper realm for pairwise alignment; another is the realization that pairwise alignment can be generally improved by allowing highly dissimilar regions to be skipped over (61, 62).

PSSM construction in PSI-BLAST is similar to that described in *Section 4*, employing position-based sequence weights (37) and pseudocounts that are modelled upon amino acid substitution probabilities (33, 44). The embedding concept described above is generalized in PSI-BLAST to deal with the complication that for any position in the query sequences, there may be a variable number of database sequences that align. Thus, the final PSSM provides position-specific scores that represent as few as one (the query sequence alone) and as many as all of the sequences detected in the previous round of the search. The high sensitivity to distant relationships provided by PSI-BLAST, and the enjoyment that a user may get by iteratively searching for homologues in real time, can lead to its overenthusiastic use, and serious errors may result. This is because any chance hit that is included in the developing PSI-BLAST PSSM will almost inevitably pull out its neighbours in subsequent rounds, and this can lead to erroneous inferences of homology. A defence against this type of error is to use conservative levels of statistical significance for addition of sequences to the PSSM. However, because proteins are not comprised of random sequences of residues, the random statistical model that underlies the BLAST programs can be unreliable (63), and so novel conclusions drawn from iterative searches should be viewed with appropriate caution.

6.4 Multiple alignment-based searching of protein family databases

The effectiveness of utilizing family-based information to search databases encourages the use of multiple alignments for searching multiple alignment databases. LAMA (for local alignment of multiple alignments) is a program that searches ungapped blocks versus family databases (64). In LAMA, PSSM columns are scored against one another by calculating a correlation coefficient, and a high scoring alignment is one in which the sequence-weighted distribution of residues is highly similar overall between aligned columns. The high sensitivity of LAMA for locally aligned regions has led to its use in discovering subtle similarities, such as those shared by helix-turn-helix DNA binding motifs found in unrelated modules.

Tools such as LAMA, which thrive on abundant alignment data, are likely to become more widely used as protein families expand in size. Because of its high sensitivity, LAMA or its descendants should become increasing valuable for modelling 3-D structures of sequences by facilitating local alignment to family members of known structure. As the percentage of unclassified proteins dwindles, a major alignment-based problem facing biologists will be to determine which subfamily a protein belongs to, and from this, more precise structural and functional inferences may be made. We anticipate the development of a next generation of computational tools to deal with this problem.

References

1. Rost, B. (1996). In *Methods in enzymology* (ed. R. F. Doolittle), Vol. 266, p. 525. Academic Press.
2. Garnier, J., Gibrat, J.-F., and Robson, B. (1996). In *Methods in enzymology*, Vol. 266, p. 540.
3. Moult, J., Hubbard, T., Bryant, S. H., Fidelis, K., and Pedersen, J. T. (1997). *Proteins: Struct. Funct. Genet.*, **Suppl. 1**, 2.
4. Henikoff, S. (1991). *New Biol.*, **3**, 1148.
5. Bairoch, A., Bucher, P., and Hofmann, K. (1997). *Nucleic Acids Res.*, **25**, 217.
6. Jonassen, I., Collins, J. F., and Higgins, D. G. (1995). *Protein Sci.*, **4**, 1587.
7. Nevill-Manning, C. G., Wu, T. D., and Brutlag, D. L. (1998). *Proc. Natl. Acad. Sci. USA*, **95**, 5865.
8. Schneider, T. D. and Stephens, R. M. (1990). *Nucleic Acids Res.*, **18**, 6097.
9. Saitou, N. and Nei, M. (1987). *Mol. Biol. Evol.*, **4**, 406.
10. Vingron, M. and Argos, P. (1991). *J. Mol. Biol.*, **218**, 33.

11. Boguski, M. S., Hardison, R. C., Schwartz, S., and Miller, W. (1991). *New Biol.*, **4**, 247.
12. Schuler, G. D., Altschul, S. F., and Lipman, D. J. (1991). *Proteins: Struct. Funct. Genet.*, **9**, 180.
13. Sobel, E. and Martinez, H. M. (1986). *Nucleic Acids Res.*, **14**, 363.
14. Posfai, J., Bhagwat, A. S., Posfai, G., and Roberts, R. J. (1989). *Nucleic Acids Res.*, **17**, 2421.
15. Smith, H. O., Annau, T. M., and Chandrasegaran, S. (1990). *Proc. Natl. Acad. Sci. USA*, **87**, 826.
16. Depiereux, E. and Feytmans, E. (1992). *CABIOS*, **8**, 501.
17. Neuwald, A. F. and Green, P. (1994). *J. Mol. Biol.*, **239**, 698.
18. Bacon, D. J. and Anderson, W. F. (1986). *J. Mol. Biol.*, **191**, 153.
19. Stormo, G. D. and Hartzell, G. W. 3rd (1989). *Proc. Natl. Acad. Sci. USA*, **86**, 1183.
20. Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., and Wootton, J. C. (1993). *Science*, **262**, 208.
21. Bailey, T. and Elkan, C. (1994). In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pp. 28–36. AAAI Press, Menlo Park, CA.
22. Notredame, C., Holm, L., and Higgins, D. G. (1998). *Bioinformatics*, **14**, 407.
23. Briffeuil, P., Baudoux, G., Lambert, C., De Bolle, X., Vinals, C., Feytmans, E., *et al.* (1998). *Bioinformatics*, **14**, 357.
24. Henikoff, S., Henikoff, J. G., Alford, W. J., and Pietrokovski, S. (1995). *Gene*, **163**, GC17.
25. Henikoff, S. and Henikoff, J. G. (1991). *Nucleic Acids Res.*, **19**, 6565.
26. McLachlan, A. D. (1983). *J. Mol. Biol.*, **169**, 15.
27. Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987). *Proc. Natl. Acad. Sci. USA*, **84**, 4355.
28. Bowie, J. U., Luthy, R., and Eisenberg, D. (1991). *Science*, **253**, 164.
29. Krogh, A., Brown, M., Mian, I. S., Sjolander, K., and Haussler, D. (1994). *J. Mol. Biol.*, **235**, 1501.
30. Baldi, P., Chauvin, Y., Hunkapiller, T., and McClure, M. A. (1994). *Proc. Natl. Acad. Sci. USA*, **91**, 1059.
31. Eddy, S. R., Mitchison, G., and Durbin, R. (1995). *J. Comput. Biol.*, **2**, 9.
32. Patthy, L. (1987). *J. Mol. Biol.*, **198**, 567.
33. Henikoff, J. G. and Henikoff, S. (1996). *CABIOS*, **12**, 135.
34. Luthy, R., Xenarios, I., and Bucher, P. (1994). *Protein Sci.*, **3**, 139.
35. Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). *CABIOS*, **10**, 19.
36. Sibbald, P. R. and Argos, P. (1990). *J. Mol. Biol.*, **216**, 813.
37. Henikoff, S. and Henikoff, J. G. (1994). *J. Mol. Biol.*, **243**, 574.
38. Dayhoff, M. (1978). *Atlas of protein sequence and structure*, Vol. 5, suppl. 3, pp. 345–58. National Biomedical Research Foundation, Washington, DC.
39. Jones, D. T., Taylor, W. R., and Thornton, J. M. (1992). *CABIOS*, **8**, 275.
40. Henikoff, S. and Henikoff, J. G. (1992). *Proc. Natl. Acad. Sci. USA*, **89**, 10915.
41. Altschul, S. F. (1991). *J. Mol. Biol.*, **219**, 555.
42. Dodd, I. B. and Egan, J. B. (1987). *J. Mol. Biol.*, **194**, 557.
43. Brown, M. P., Hughey, R., Krogh, A., Mian, I. S., Sjolander, K., and Haussler, D. (1993). In *Proc. First Int. Conf. on Intelligent Systems for Molecular Biology* (ed. L. Hunter, D. Searls, and J. Shavlik), pp. 47–55. AAAI Press, Washington DC.
44. Tatusov, R. L., Altschul, S. F., and Koonin, E. V. (1994). *Proc. Natl. Acad. Sci. USA*, **91**, 12091.
45. Henikoff, S. and Comai, L. (1998). *Genetics*, **149**, 307.
46. Henikoff, S. and Henikoff, J. G. (1997). *Protein Sci.*, **6**, 698.
47. Sonnhammer, E. L., Eddy, S. R., and Durbin, R. (1997). *Proteins: Struct. Funct. Genet.*, **28**, 405.

48. Smith, R. F. and Smith, T. F. (1990). *Proc. Natl. Acad. Sci. USA*, **87**, 118.
49. Sonnhammer, E. L. L. and Kahn, D. (1994). *Protein Sci.*, **3**, 482.
50. Corpet, F., Gouzy, J., and Kahn, D. (1998). *Nucleic Acids Res.*, **26**, 323.
51. Gracy, J. and Argos, P. (1998). *Bioinformatics*, **14**, 174.
52. Gona, G., LInial, N., Tishby, N., and Linial, M. (1998). *ISMB*, **6**, 212.
53. Bachinsky, A. G., Yarigin, A. A., Guseva, E. H., Kulichkov, V. A., and Nizolenko, L. P. (1997). *CABIOS*, **13**, 115.
54. Wu, C. H., Zhao, S., and Chen, H. L. (1996). *J. Comput. Biol.*, **3**, 547.
55. Wu, C. H., Shivakumar, S., Shivakumar, C. V., and Chen, S. C. (1998). *Bioinformatics*, **14**, 223.
56. Bucher, P., Karplus, K., Moeri, N., and Hofmann, K. (1996). *Comput. Chem.*, **20**, 3.
57. Bailey, T. L. and Gribskov, M. (1997). *J. Comput. Biol.*, **4**, 45.
58. Nicodeme, P. (1998). *Bioinformatics*, **14**, 508.
59. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Aneng, Z., Miller, W., *et al.* (1997). *Nucleic Acids Res.*, **25**, 3389.
60. Pearson, W. R. (1995). *Protein Sci.*, **4**, 1145.
61. Altschul, S. F. (1998). *Proteins: Struct. Funct. Genet.*, **32**, 88.
62. Alexandrov, M. M. and Luethy, R. (1998). *Protein Sci.*, **7**, 254.
63. Brenner, S. E., Chothia, C., and Hubbard, T. J. (1998). *Proc. Natl. Acad. Sci. USA*, **95**, 6073.
64. Pietrokovski, S. (1996). *Nucleic Acids Res.*, **24**, 3836.