Protein Science (1996), 5:1991–1999. Cambridge University Press. Printed in the USA.

Construction and analysis of a profile library characterizing groups of structurally known proteins

ATSUSHI OGIWARA,[1,3] IKUO UCHIYAMA,[1] TOSHIHISA TAKAGI,[1] AND MINORU KANEHISA[2]

[1]Human Genome Center, Institute of Medical Science, The University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108, Japan
[2]Institute for Chemical Research, Kyoto University, Uji, Kyoto 611, Japan

(RECEIVED February 28, 1996; ACCEPTED July 22, 1996)

Abstract

A new sequence motif library StrProf was constructed characterizing the groups of related proteins in the PDB three-dimensional structure database. For a representative member of each protein family, which was identified by cross-referencing the PDB with the PIR superfamily classification, a group of related sequences was collected by the BLAST search against the nonredundant protein sequence database. For every group, the motifs were identified automatically according to the criteria of conservation and uniqueness of pentapeptide patterns and with a dual dynamic programming algorithm. In the StrProf library, motifs are represented by profile matrices rather than consensus patterns to allow more flexible search capabilities. Another dynamic programming algorithm was then developed to search this motif library. When the computationally derived StrProf was compared with PROSITE, which is a manually derived motif library in the best consensus pattern representation, the numbers of identified patterns were comparable. StrProf missed about one third of the PROSITE motifs, but there were also new motifs lacking in PROSITE. The new library was incorporated in SMART (Sequence Motif Analysis and Retrieval Tool), a computer tool designed to help search and annotate biologically important sites in an unknown protein sequence. The client program is available free of charge through the Internet.

Keywords: dynamic programming; integrated database; motif search; profile matrix; sequence interpretation; sequence motif

As DNA sequence determination becomes a fundamental technique for molecular biology, computer analysis of sequence data is now an integral part of deciphering biological functions of nucleic acids and proteins (Griffin & Griffin, 1994). The basic strategy is to find sequence similarities that can be extended to functional similarities, but in practice there are two types of methods. One is the similarity search method (Pearson & Lipman, 1988; Altschul et al., 1990), where a newly determined sequence is compared against the sequence database to see if there are any similar sequences whose functions are already known. The other is the motif search method (Taylor, 1988; Hodgman, 1989), where a sequence is compared against the library of functionally important sequence patterns that are predefined from groups of related sequences. Because the database size is growing continuously, the similarity search method is becoming more problematic because of the cost of computation time and because of duplicate and redundant data that complicate the interpretation of search results. Weak similarities in the so-called twilight-zone are also hard to interpret. The motif approach can in principle cope with these circumstances, for the search is made against better organized data sets.

The degree of similarity in the alignment of two protein sequences is not homogeneous along the sequences. There are specific regions, such as active sites of enzymes, binding sites of DNAs, and modification sites for phosphorylation, that are well conserved in a group of similar sequences. The characteristics of such functionally important sites are represented by consensus sequence patterns and numerical profiles, which are collectively called here as motifs. Examples of known patterns and profiles are organized in motif libraries such as PROSITE (Bairoch, 1992). These libraries are usually constructed manually by collecting and refining experimental observations. In contrast, we have been interested in an automatic procedure (Ogiwara et al., 1992) to extract and update sequence motifs computationally from a vast amount of sequence data already available in computerized databases and other data resources.

To define a sequence motif, it is customary to perform the multiple sequence alignment of functionally related proteins and identify blocks of conserved residues. Alternatively, the conserved

Reprint requests to: Atsushi Ogiwara, National Institute for Basic Biology, 38 Nishigounaka, Myoudaiji-cho, Okazaki 444, Japan; e-mail: ogi@nibb.ac.jp.
[3]Present address: National Institute for Basic Biology, 38 Nishigounaka, Myoudaiji-cho, Okazaki 444, Japan.

segments may first be identified in each sequence and then multiply aligned (Galas et al., 1985; Schneider et al., 1986). Somewhat similar in spirit to such local segment alignment algorithms, we developed a new method to automatically extract sequence motif patterns from a protein superfamily, a group of evolutionarily related proteins (Ogiwara et al., 1992). The method adopted two simple criteria to screen candidates of sequence motifs: a motif must be conserved in the group (conservation), and it must appear exclusively within the group (uniqueness). This simple idea is derived from the concept that, if a motif is a feature of a certain biological function, such a function is expected to be conserved within related proteins, and it must be unique and distinguishable from other protein sequences. We constructed a motif library named MotifDic from the PIR superfamilies, which has since been made available through the Japanese GenomeNet database service by an e-mail server (motif@genome.ad.jp) and a World Wide Web (WWW) server (http://www.genome.ad.jp/SIT/MOTIF.html).

The explosion of protein sequence information is now followed by the explosion of protein structural information. The number of entries in the Protein Data Bank (PDB) shows a dramatic increase in the past few years. In view of this growing number of structurally resolved proteins, we think it useful to construct and maintain a new motif library, now called StrProf, that characterizes protein groups derived from the structural data. We adopt a numerical representation of profile matrices (Gribskov et al., 1987; Bucher & Bairoch, 1994), rather than a pattern representation of sequence motifs to allow more flexible searching. At the same time, we have developed a new computer tool called SMART (Sequence Motif Analysis and Retrieval Tool) for making use of StrProf as well as PROSITE in real biological situations. SMART runs under the client/server mode; the user needs only the client program, which automatically requests all necessary data and computations at a server machine on GenomeNet.

Results

Profile library

We have constructed a profile library named StrProf, which characterizes groups of structurally known proteins as explained in the following. The protein sequences in the April 1994 version of the PDB were classified into 372 groups according to the superfamily classification of PIR protein sequence database. Utilizing our PDBSTR database, which is a reorganized PDB database for sequence analysis, conserved and unique pentapeptide segments were collected for each group. We call these peptide segments unique peptide words (UPWs). The two parameters for conservation (C) and uniqueness (U) controlled the screening of UPWs (Ogiwara et al., 1992). The degree of conservation is determined by the fraction of sequences containing a specific pattern in a group, whereas the degree of uniqueness is determined by the fraction of sequences outside the group that have a specified pattern. The locations of UPWs in each sequence were then examined, overlapping UPWs were merged, and a consensus form of the UPW patterns was defined in each group. The consensus form was represented as a sequence of profile matrices, each corresponding to a block of conserved residues. The collection of such consensus forms is the StrProf profile library (see Materials and methods).

As for the parameters controlling the UPW screening, we report here the following two cases, although we examined several other parameter sets:

1. Strict condition ($C = 0.8$ and $U = 0.8$).
2. Loose condition ($C = 0.7$ and $U = 0.3$).

The parameters $C = 0.7$ and $U = 0.3$ mean, for example, that a pentapeptide is a UPW if it appears in more than 70% of sequences in the group, and if it also appears in less than 70% of sequences outside the group. Under this loose condition, UPWs were found in 797 entries of 220 groups, which covered about 50% of the whole PDBSTR sequences and about 60% of the superfamilies in PDBSTR. When the condition becomes looser, it is expected to cover more PDBSTR entries. In fact, with $C = 0.5$ and $U = 0.5$, about three quarters of the PDBSTR superfamilies were covered, whereas under the condition of $C = 0.8$ and $U = 0.8$, only about one third were covered. Table 1 shows the statistics of extracted motifs in the two cases shown above with the minimum group size parameter $M = 5$.

The profile library named StrProf was constructed as a collection of profile entries for each protein group. A profile entry has a chain of profile matrices connected by spacers. It also contains an identifier record and descriptive information, such as the name of the PIR superfamily into which this profile is classified. Also included are cross-reference records that link the profile entry to the PDBSTR protein structure database entries and to the nonredundant amino acid sequence database entries.

SMART

To investigate individual profiles with a 3D perspective and to utilize the profile library for sequence analysis, we have developed a computer tool, SMART. Among its many capabilities, SMART can first be used to search a motif library such as StrProf and PROSITE for patterns or profiles that fit best to a query sequence. Then, SMART provides additional information of 3D structure and biological function taken from related proteins in the databases (Ogiwara et al., 1993). The SMART system works in a client-server mode, where the search engine as well as the databases and libraries are located in the server machine on the Internet. The client program may be obtained from the GenomeNet anonymous FTP server (ftp://ftp.genome.ad.jp/pub/hgc/smart/).

Figure 1 shows how the SMART system works. Here cytochrome c was used as a query sequence and searched against the StrProf library constructed with $C = 0.7$ and $U = 0.3$. The upper right portion of Figure 1 is the search result window, which contains a list of found motifs, the pattern representation of a selected motif, the location on the query sequence, and the lists of related sequence and structure database entries. The 3D structure selected from the structure list is displayed in the large window with the

Table 1. *Statistics of extracted motifs with two sets of parameters (C, U)*

	Total in PDBSTR	Extracted with (0.8, 0.8)	Extracted with (0.7, 0.3)
Number of original superfamilies	306	103 (34%)	182 (59%)
Number of groups	372	119 (32%)	220 (59%)
Number of sequences	1,600	483 (30%)	797 (50%)

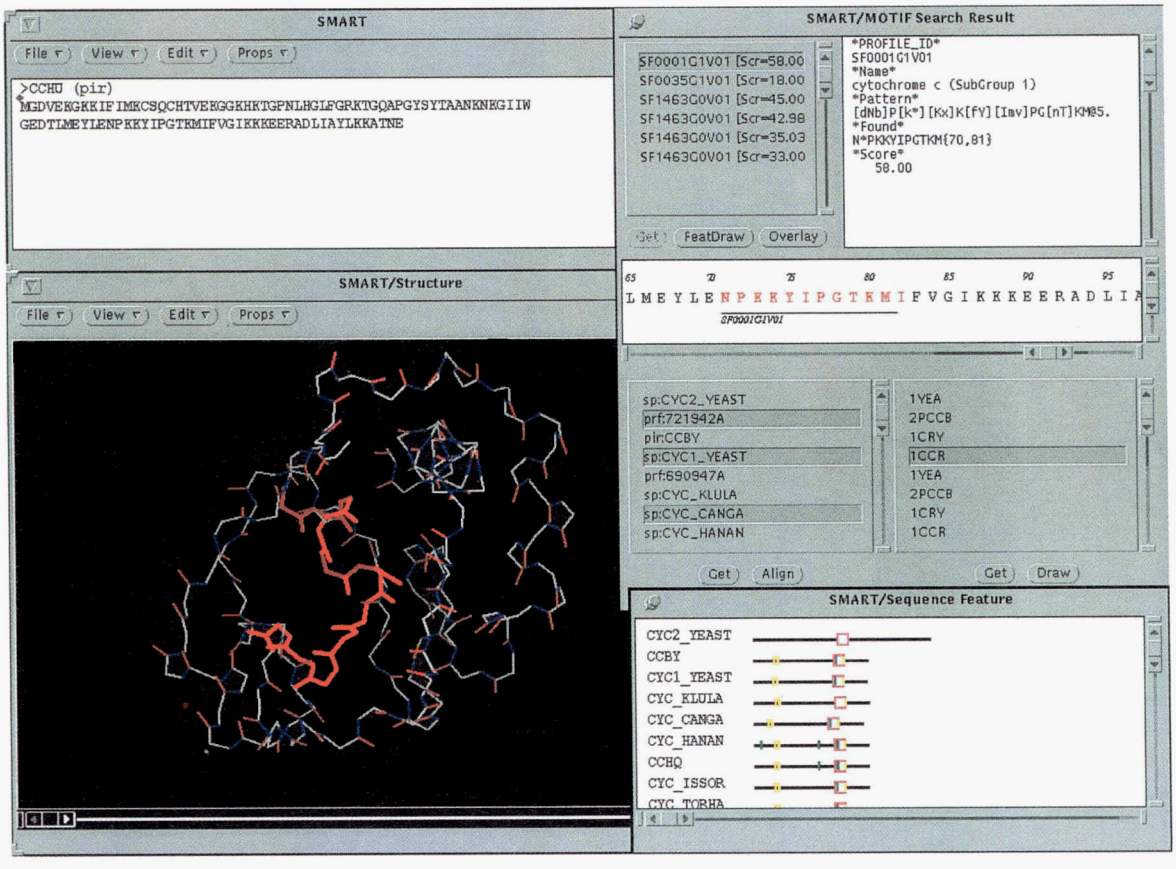

Fig. 1. Demonstration of the SMART system with cytochrome *c* as an example. Upper left window is the query sequence input window, where the query sequence (cytochrome *c*, in this example) in FASTA format should be typed in or read from a file. Upper right window is the motif search result window, where known sequences and 3D structures having the same sequence motif are shown. A selected structure may be analyzed on the 3D model window, where the user can rotate, move, or scale the 3D wire-frame model. The precision level (C^{α} only, main chain, or all atoms) and other modeling parameters can be changed. In addition, known functional sites in related sequences may be displayed (lower right), and a multiple sequence alignment may be performed for further analysis (not shown).

black background. It is possible to manipulate the graphics on this window. In the skeleton drawing of C^{α} atoms, the found motif shown as the red-colored segment is located near the molecular surface, which can easily be confirmed by rotating the graphics object. In the bottom right corner of Figure 1, functional features of related sequences are shown graphically in conjunction of the found motif shown by the red box. The features are taken from the FEATURES tables of the sequence databases with the yellow and green bars corresponding to the binding and modification sites, respectively.

In the case of cytochrome *c*, the motif region defined by StrProf included the metal binding site. The actual pattern in StrProf is:

[dNb] P [k*] [Kx] K [fY] [Imv] P G [nT] K M @5,

where the last methionine is the binding site. In the pattern representation, the letters in the brackets are multiple choices for an amino acid at a certain position, where the distinction of upper and lower cases is made depending on whether the frequency of a given amino acid is above or below the average of all amino acids. An asterisk represents the choice of a deletion. The number at the end preceded by a "@" corresponds to the conservation level of this block graded from 5 to 1, which corresponds to 100%, more than 80%, more than 50%, more than 10%, and less than 10% of conservation.

It is also possible to perform a multiple sequence alignment in SMART. Sequence entries selected from the sequence list of the search result window were aligned by a tree-based pairwise method. The result of this multiple alignment can be piled onto the query sequence with the marks of functional sites. These results will help users to annotate an unknown query sequence not only by indicating the location of motifs, but also by presenting structural or functional instances in related proteins.

Comparison of StrProf entries with PROSITE patterns

We compared the result of StrProf ($C = 0.7$, $U = 0.3$) with PROSITE (Release 13.0, November 1995). Table 2 shows how many entries were common or distinct in each library. The degree of coverage of superfamilies defined by PIR was roughly the same; 182 and 186 of 306 superfamilies were identified in StrProf and PROSITE, respectively. Among them there were 114 common superfamilies.

To compare the two libraries, because every entry in StrProf was derived from a certain PDB entry, we restricted the PROSITE entries having the same links to PDB. In PROSITE, there were 255

Table 2. *Comparison of StrProf and PROSITE*

	StrProf	PROSITE
Number of entries	220	1,167
Number of entries having links to PDB	220	255
Number of superfamilies having links to PDB (A)	182	186
Number of entries sharing the same PDB links	148	138
Number of superfamilies sharing the same PDB links (B)	114	114
Number of distinct superfamilies (A)-(B)	68	72

entries (corresponding to 186 superfamilies) that had cross-reference links to the PDB database. By comparing the names of these PDB entries, we found that 138 entries (114 superfamilies) of PROSITE could correspond to 148 entries (114 superfamilies) of the StrProf library. Thus, there were 72 superfamilies that were not present in StrProf, but were present in PROSITE. Conversely, there were 68 superfamilies that existed exclusively in StrProf.

Next, we examined examples of such StrProf-specific profiles in detail. The first example is the signature pattern for the influenza virus exo-alpha-sialidase family. Using influenza virus neuraminidase as a query, the search against PROSITE produced only common modification site patterns like glycosylation or phosphorylation and, in some cases, Kringle domain signature and EGF-like cysteine patterns. Actually, there was no description in PROSITE of exo-alpha-sialidase. In contrast, the search against StrProf resulted in:

Query: PDBSTR:1NCAN,

Pattern: [dkNQr] I L R T Q E S @5,

Found: N I L R T Q E S At: 142–149.

According to the cross-reference records in StrProf, one PDB entry (1NNA) had the binding sites information, which described that the region contained three substrate-binding residues (Ile 141, Arg 143, and Glu 146).

Another interesting example was interferon γ (IFNγ). In this case also, only trivial modification sites were found in PROSITE, whereas StrProf contained the interferon gamma family profile:

Query: PDBSTR:1HIGA,

Pattern: [Q*] S Q I [IV] S F Y [Fl*] [K*] [fL*] [F*] @4 -

x(48,48) - [dQ] R K A [Iv] @4 -

x(0,75) - R [Kr] R S [QR] @5 -

x(2,2) - F [qR] G R R @3,

Found: Q S Q I V S F Y F K L F At: 46–57,

Q R K A I At: 106–110 &

R K R S Q At: 129–133.

These regions were spatially close in the 3D structure (data not shown). From the database information only, we could not conclude whether these regions were related to the activity of IFNγ, but the residue at position 45, immediate upstream of the first found block, was reported to be important in maintaining the protein structure by an experiment using IFNγ analogues (Hsu et al., 1986).

Generally, the regions covered by the StrProf profiles were much longer than those of PROSITE patterns. StrProf tended to contain many motif blocks spread along the sequence. This may be due to the simplicity of our screening method based on the conservation and uniqueness, the small diversity of sequence data in a group, or both. In contrast, because PROSITE entries are based on experimental observations well targeted to specific sites, the patterns tend to be shorter. This also suggests that, in some cases, PROSITE patterns are not sufficient to cover all functional residues, and an automatic procedure such as ours is necessary to identify additional patterns.

A typical example is a thyroid hormone-binding protein transthyretin (prealbumin) shown in Figure 2. In PROSITE, there were two separate entries for transthyretin signatures but, in StrProf, all blocks, including two PROSITE patterns, were merged into one profile entry. There were six blocks in this StrProf entry, and they covered more than half of the whole sequence. Judging from the description in SWISS-PROT, we could confirm that there were at least two thyroid hormone binding sites in this sequence. The first one (Lys 15) was included in both PROSITE and StrProf, but the second one (Glu 54) existed only in StrProf (Fig. 2). Although there was no further FEATURES description in SWISS-PROT, it was reported (Alves et al., 1993) that there were also other binding sites at Ser 117 and Thr 119, which were included in both PROSITE and StrProf.

There were many other cases where a StrProf entry corresponded to more than one PROSITE entry. In the case of the StrProf entry for cholinesterase family:

[nqS*]E[Dh]CL[tY][iL]N@4-x(0,137)-FG[En]S[As]G@2

-x(65,65)-SEDEL@1,

the second block corresponded to a PROSITE pattern, carboxylesterases type-B signature 2:

(F-G-G-x(4)-[LIVM]-x-[LIV]-x-G-x-S-[STA]-G),

and the first block to another PROSITE pattern, carboxylesterases type-B serine active site:

([ED]-D-C-L-[YT]-[LIV]-[DNS]-[LIV]-[LIVFYW]-x-[PQR]),

but in this case each PROSITE pattern was somewhat longer than the block in StrProf.

In the case of the StrProf entry for glutamate–ammonia ligase family, there were three PROSITE patterns corresponding to this group: glutamine synthetase signature 1, glutamine synthetase putative ATP-binding region signature, and glutamine synthetase class-I adenylation site. The StrProf entry contained six blocks that covered all the PROSITE patterns; the first block matched almost exactly to the first signature pattern of PROSITE. The second and the third PROSITE patterns, however, were only partially overlapping with the profile blocks. In the cases of heat shock protein 70 family and subtilisin family, three individual PROSITE patterns corresponded to each StrProf entry, but in both cases only two PROSITE patterns overlapped with StrProf blocks.

```
<PROSITE PATTERN>                    <PROSITE PATTERN>
Transthyretin signature 1            Transthyretin signature 2
S-K-C-P-L-M-V-K-V-L-D-A-V-R-G.       S-P-[FY]-S-Y-S-T-T-A-[LIVM]-V-[ST]-x-P.

<StrProf pattern representation>
transthyretin family
<Pattern>
[G*][DE*][S*][K*][C*]PLMVKVLDAVRG[rS]PA@5-x(19,19)-
A[NT]GKT@2-x(0,0)-[ST]WEPFASGK[gT]@4-x(2,6)-
[St*][G*][E*][L*]H[EG]LTT@5-x(13,13)-
DT[KS]SYWK[Agst]LG[Ilv]SPFHE@5-x(1,1)-
A[DE]VVF[sT]ANDSG@5-x(3,3)-
YTIAASLLSP[FY]SYSTTA[IV]@5.
 <Found>
GESKCPLMVKVLDAVRGSPA{6,25}-x(19)-
ASGKT{45,49}-x(2)-SGELHGLTT{52,60}-x(13)-
DTKSYWKALGISPFHE{74,89}-x(1)-
AEVVFTANDSG{91,101}-x(3)-
YTIAALLSPYSYSTTAV{105,121}
```

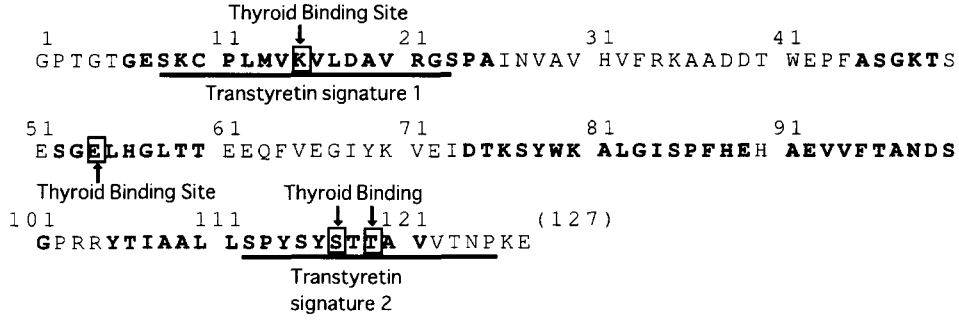

Fig. 2. Comparison of motifs between StrProf and PROSITE with respect to a transthyretin sequence. Bold letters in the StrProf motif indicate the residues actually found. In the lower part, the whole transthyretin sequence is shown, where the StrProf motif is shown in bold letters and the PROSITE motifs are shown underlined. Boxes correspond to binding sites as described in the SWISS-PROT database or in the literature.

Other examples that a StrProf entry corresponded to more than one PROSITE entries are: catalase, glutathione peroxidase, superoxide dismutase (Cu-Zn), nitrogenase iron protein, EPSP synthase, rhodanese, zinc carboxypeptidase, tryptophan synthase, xylose isomerase, hemocyanin, tropomyosin, homeobox, bacteriorhodopsin, and transferrin. However, StrProf blocks did not always correspond to PROSITE patterns.

Using StrProf to classify a query sequence

In order to obtain a rough estimate of the accuracy of classification when performing a profile search using StrProf, we checked the result of classification for every sequence in the initial PDBSTR entries. The sequence was assigned to the group whose profile matching score (see Materials and methods) was the highest, given a threshold score value for deciding whether an assignment should be made. We note, however, that this is not a rigorous cross-validation; the original training data set is used as a test data set. The result is shown in Table 3. We adopted the highest scored entry as the classification answer. If no answer could be obtained above the threshold cutoff, we counted such entries as "unclassified." To access the classification accuracy, we defined the sensitivity and the selectivity as described in Table 3.

In both cases of strict and loose conditions, sensitivity was 35–60%. That is, classification failed for 40–65% of entries because the best score was below the threshold. However, selectivity was higher (87–95%) in both cases. This means once the classification was made, it was relatively reliable. Table 3 also suggests that it is safer to use the StrProf library obtained with the strict condition of $C = 0.8$ and $U = 0.8$, but then the library covers a smaller fraction of the original PDBSTR groups.

For a comparison, we examined the sensitivity and the selectivity of using PROSITE in a similar manner. We also tested the BLOCKS search as well as the original pattern representation of PROSITE. The evaluation of PROSITE was slightly different. Because it was difficult to assign scores for selection from alternative choices in the pattern matching, we simply counted the total num-

Table 3. *Result of classification by StrProf*

	(C,U) = (0.8, 0.8)	(C,U) = (0.7, 0.3)
Number of PDB entries tested (A = B + C + D)	119	220
Number of truly classified entries (B)	67	78
Number of falsely classified entries (C)	3	11
Number of unclassified entries (D)	49	131
Sensitivity of classification (B/(B + D))	57.8%	37.3%
Selectivity of classification (B/(B + C))	95.7%	87.6%

ber of PDB entries that contained the pattern correctly or incorrectly. It was thus possible for a sequence to contain a correct pattern and an incorrect pattern at the same time; namely, a sequence could be assigned to multiple groups. The BLOCKS search was performed with the BLIMPS program (version 3.0.0, Wallace & Henikoff, 1992). As shown in Table 4, sensitivity was good in both cases. In the case of PROSITE, selectivity was comparatively low because we had overcounted the classification answers. However, this result also reflects the tendency of PROSITE patterns to be too small for the specific classification. This is consistent with our observation that StrProf profiles cover longer regions than PROSITE.

Discussion

The similarity search is a most popular method to interpret unknown protein or nucleic acid sequences. It is based on our empirical knowledge that, if sequences are similar, then functions are also similar. On the other hand, it is often the case that no similarity is detected in two sequences, yet they still retain common 3D structures and short conserved sequence segments corresponding to functional sites (Matsuo & Nishikawa, 1994). The motif libraries such as PROSITE focus on the functional sites determined by experiments, but the patterns involved tend to be too short to discriminate real functional sites from random matches. Thus, it is desirable to describe functional motifs with 3D structures as background information.

The advantage of an automatic procedure to define sequence motifs is the following. First, it is free from cumbersome manual alignment and refinement and thus applicable to making use of a rapidly expanding body of sequencing data. Second, based on a systematic analysis, it will identify previously unreported sequence motifs. Third, even for known motifs, it will identify additional, surrounding patterns. By comparing StrProf and PROSITE, we found that roughly the same number of superfamilies were covered in the two motif libraries, and two thirds of them were common in both libraries. We could identify 68 new motifs that were not present in PROSITE, although we have not been able to check if all of them have functional significance. In the cases of common motifs, StrProf tended to cover longer regions of sequences than did PROSITE, which may reflect the conservation of 3D structure surrounding functional sites.

Simply judging from the comparison of the sensitivity and the selectivity, BLOCKS seemed to perform best. However, we note that BLOCKS depends on PROSITE. No motifs not found in PROSITE can be found in BLOCKS. In contrast, we note again that some novel motifs that did not exist in PROSITE, such as exo-alpha-sialidase and IFNγ, could be discovered.

Some problems may still remain in our automated procedure. Of special note, merging blocks on each sequence to construct a consensus profile may be complicated by the existence of repeat patterns and the lack of patterns in some sequences. As described in Materials and methods, because we used the dynamic programming procedure, which resembles a pairwise multiple alignment method, it depends on the order of merging. Some multiple alignment-free method (Stormo & Hartzell, 1989; Cardon & Stormo, 1992) may be effective in solving the problem.

Because StrProf itself lacks the description of functional meanings, it is important for the motif library to be linked to other existing databases. Thus, we have developed a computer tool, SMART, to search, analyze, and interpret motifs. SMART works on a client/server mechanism, and because no local database resource is required for the client part, the user needs only a workstation and the Internet connection. There are other computer programs that search a motif library, such as PROSITE, and report the names and locations of found motifs (Fuchs, 1991; Sibbald et al., 1991). In contrast, SMART provides not only the names and the locations of motifs found, but also additional information of related proteins sharing the same motifs. We believe such additional information is crucial in understanding biological significance and annotating a newly determined sequence. Another advantage of SMART is the database integration. There are many databases that provide cross-references or links to related entries in other databases. When a linked entry is requested, SMART follows the link and returns the content of the entry from another database. As the increasing popularity of WWW indicates, this style of integration is becoming increasingly familiar in the biological research community. The link-based approach taken in SMART and WWW, and the knowledge-based approach taken in StrProf and PROSITE, are a best solution at the moment to make full use of huge and complex data in molecular biology.

Table 4. *Result of classification by PROSITE and BLOCKS*

	PROSITE	BLOCKS
Total number of motif entries searched	1,167	3,201
Number of motif entries having 3D links	255	194
Number of PDB entries tested (A)	184	177
Number of times truly classified (B)	227	165
Number of times falsely classified (C)	716	6
Number of unclassified entries (D)	6	6
Sensitivity of classification (B/(B + D))	97.4%	96.5%
Selectivity of classification (B/(B + C))	24.1%	96.5%

Materials and methods

Database

We used the April 1994 version of PDB. The flow of data collection is illustrated in Figure 3. PDBSTR is a reorganized database derived from PDB and is released through the Internet (for details, http://www.genome.ad.jp/htbin/show_man?pdbstr). Basically, a multiple chain entry of PDB is separated into different entries in PDBSTR, where an entry is identified by the four-letter PDB code plus one-letter chain identifier. The release of PDBSTR we used was 68.0 (April, 1994), which contained 3,894 entries.

First, identical sequences were removed by the nrdb program developed by the U.S. National Center for Biotechnology Information, and we obtained 1,600 nonredundant protein sequences. We then searched the PIR protein sequence database for homologous sequences of these 1,600 sequences and classified them into superfamilies according to the PIR superfamily classification. The BLAST program was applied to search homologies. The PIR database used was release 41.0 (June, 1994) containing 3,166 superfamilies. As a result, we obtained 306 superfamilies containing 1,234 sequences. Some sequences were not classified because no homologous sequences were found. To refine the classification, we then examined the similarity score of every pair of sequences in a group and, if the score was less than 30%, the group was split into

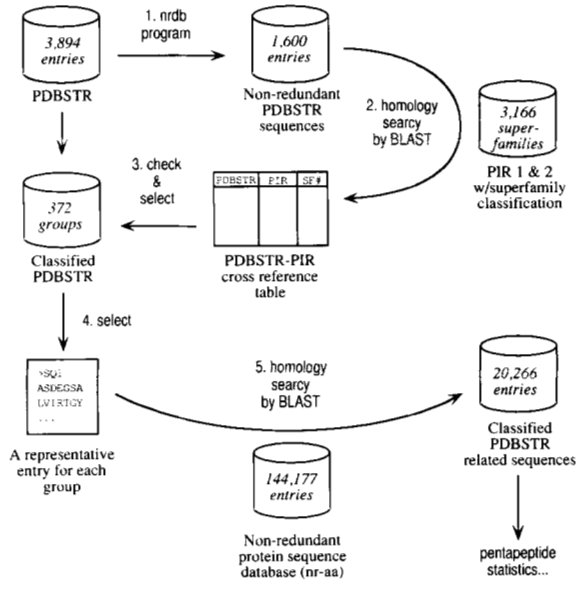

Fig. 3. Flow of data collection for constructing StrProf. The initial data set was taken from the PDBSTR 3D structure database containing 3,894 sequences. Removing redundant sequences with the nrdb program, 1,600 nonredundant sequences became available. Comparing with the PIR database classified into superfamilies, they fell into 372 groups. To increase the size of related sequences, the nonredundant protein sequence database was searched for a representative of each group. The final data set contained 20,266 sequences in total.

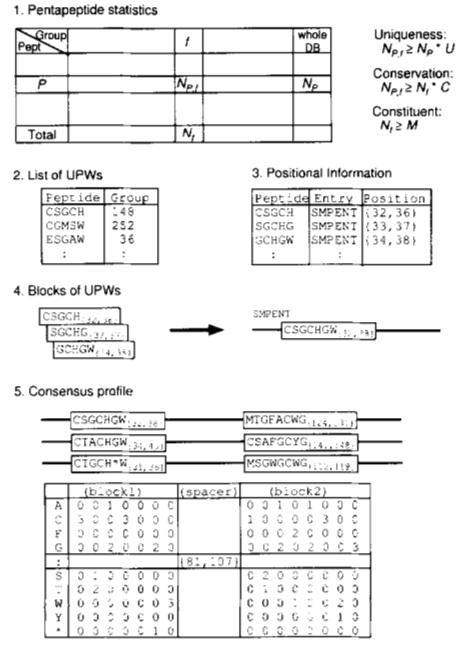

Fig. 4. Schematic drawing of the procedure to construct StrProf. First, the statistics of pentapeptides were taken for every protein group derived from the data set described in Figure 3. Next, using the criteria of uniqueness and conservation, a list of unique peptide words (UPWs) was created. Then, locations of UPWs in the original sequences were examined, and continuous blocks of overlapping UPWs were made. Finally, using a multiple alignment method, chains of blocks were merged into a consensus, which is expressed by a sequence of profile matrices separated with spacers.

two or more. After the refinement, we obtained 372 groups of similar sequences.

In order to investigate sequence variations and to define sequence motifs, we gathered for each group as many similar sequences as possible by searching the nonredundant protein sequence database (the daily updated version on October 3, 1994, containing 144,177 sequences derived from SWISS-PROT, PIR, PRF, and GenPept databases). By applying the BLAST program to a representative sequence of each group, we gathered 20,266 sequences in total. These sequences, including the initial PDBSTR sequences, were used to count frequencies of all possible pentapeptide patterns.

Procedure to construct a profile library

The procedure to construct a profile library, which is illustrated in Figure 4, is similar to our previous procedure to construct a pattern library (Ogiwara et al., 1992). First we count the frequency of all possible pentapeptide patterns, i.e., $20^5 = 3{,}200{,}000$ pentapeptides from AAAAA to YYYYY in alphabetical order, that appear in each group and in the whole database. Using this statistics we select "unique peptide words" (UPWs) that satisfied the following conditions:

$$N_{P,f} \geq C \times N_f,$$

$$N_{P,f} \geq U \times N_P,$$

where $N_{P,f}$ is the number of sequences having a UPW P in a group f, N_f is the total number of sequences in the group f, C is the parameter defining "conservation" ($0 \leq C \leq 1$), N_P is the total number of sequences having the UPW P in the whole database, and U is the parameter defining "uniqueness" ($0 \leq U \leq 1$). In addition, we require that the group f must have at least M entries for the statistics to be meaningful. The actual parameters we have chosen are described in the result section.

Next, we merge overlapping UPWs to form blocks of conserved, unique segments. To connect adjacent UPWs, we look up the original sequences and obtain the positions. This process is applied to all the sequences in a group, and a consensus form of block patterns is derived by a kind of multiple alignment based on a dynamic programming (DP) algorithm. In our previous study (Ogiwara et al., 1992), the consensus was given by a pattern representation, but, in this work, the consensus is expressed by a profile matrix. In the profile representation, each element of the matrix reflects the relative frequency of a certain amino acid at a certain position. The value is first normalized with the total number of residues at the position and then divided by the fraction of occurrences of the amino acid in the database. The size of our profile matrix is the block length times 24, i.e., 20 amino acids, B (Asn or Asp), Z (Gln or Glu), X (any amino acid), and a gap. Actually, however, we do not make use of the values for B, Z, and X in the search process described below.

We apply the DP algorithm twice to make consensus profiles. The first DP is to compare a new profile block on the newly merged sequence with the consensus matrix being constructed, and the second is to compare a chain of profile blocks with the sequence of consensus matrices. In the first DP, PAM120 (Dayhoff et al., 1978; Schwartz & Dayhoff, 1978) was used as the scoring matrix. In the second DP, the score was defined as follows:

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} + B_{i,j} \\ S_{i-1,j} - L_b \times G, \\ S_{i,j-1} - L_c \times G \end{cases}$$

where $S_{i,j}$ is the score of the best alignment at the position of the ith consensus block and the jth block of a newly merged entry. $B_{i,j}$ is the alignment score of the ith consensus block and the jth new block, which has been calculated in the first DP step. L_c and L_b are the lengths of the consensus block and the new block, respectively, and G (>0) is a penalty for a block deletion.

Thus, each entry of our profile library contains one or more blocks of profile matrices separated by the given numbers of spacer residues.

Search algorithm

A new search method was developed to make use of our profile library. The method is based on a dual dynamic programming (DP) algorithm, which is similar to the algorithm for the profile construction. In the first DP routine, the best alignment between a block of one profile entry and a region of a query sequence is made using the BLOSUM90 scoring matrix (Henikoff & Henikoff, 1992). The second DP routine makes alignments between the chain of aligned blocks and the whole query sequence. The similarity score is calculated from the score of the first DP, the conservation levels of blocks, and the divergence from the range of allowed spacer lengths.

In the first DP, the optimal alignment score (B_L) for a block of length L can be calculated from the following formula:

$$B_l = B_{l-1} + \max \begin{cases} \dfrac{\sum_{a \in A} (P(a,l) \times M(a,s_l))}{\sum_{a \in A} P(a,l)} \\ \qquad \times \dfrac{P(s_l,l)}{\sum_{a \in A} P(a,l)} & \text{if aligned} \\ G & \text{if a gap is inserted} \end{cases}$$

where $P(a,l)$ is the normalized value of the element of a profile matrix for amino acid a at position l ($1 \le l \le L$), s_l is the amino acid of the query sequence at position l in the block, $M(a,s_i)$ is the element of BLOSUM90 scoring matrix for amino acids a and s_i, and the set A stands for 20 amino acids.

The penalty for a gap is defined as follows:

$$G = \begin{cases} \text{minimum matching score (4)} & \text{if a gap is allowed at the position} \\ \text{gap to non-gap score (6)} & \text{otherwise.} \end{cases}$$

In the second DP routine, the global alignment score S_N between a chain of profile blocks consisting of N blocks and the query sequence is determined as follows:

$$S_n = S_{n-1} + B_n \times W_n - V_n$$

where W_n is the weight for the nth block ($1 \le n \le N$), and V_n stands for the penalty for violating the allowed spacer length between ($n - 1$)th and nth blocks. W is assigned to some value between 0 to 1, according to the level of conservation of the block described in StrProf. V equals 0 if the length of the spacer falls in the allowed range; otherwise, it is a penalty value proportional to the degree of error. Note that the second DP procedure makes it necessary to look up all the previously calculated nodes to determine the best score, because the score depends on the distance between the current position and the nearest aligned block. Thus, it requires much more time than the normal DP algorithm.

Computer tool

A computer workbench, named SMART, that assists sequence interpretation by motif search, was developed for Sun workstations using XView window system. Currently, SMART runs on a Sun SPARC system with SunOS 4.1.x. SMART was constructed based on a client/server model. The client part runs on a user's machine and provides the user interface, and the server part runs on the server machine in the Human Genome Center of the University of Tokyo (smart.genome.ad.jp) and provides the database retrieval and other calculations. The compiled version of the client part can be obtained by the anonymous ftp from the following address: ftp://ftp.genome.ad.jp/pub/hgc/smart/.

SMART may be regarded as a kind of integrated database system because it can treat sequence and structure databases uniformly as if they were stored in a single database. The integration strategy of SMART stands on a loosely coupled integration, where every database is left untouched and only protocols to refer other databases are established, instead of converting all databases into another single database. Reference to other databases is achieved by following the links stored in the profile library.

Acknowledgments

This work was supported by a grant-in-aid for scientific research on priority areas, "Genome Informatics," from the Ministry of Education, Science, Sports and Culture of Japan. The computation time was provided by the Human Genome Center, the Institute of Medical Science, the University of Tokyo, and the Supercomputer Laboratory, the Institute for Chemical Research, Kyoto University.

References

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. *J Mol Biol 215*:403–410.

Alves IL, Divino CM, Schussler GC, Altland K, Almeida MR, Palha JA, Coelho T, Costa PP, Saraiva MJ. 1993. Thyroxine binding in a TTR Met 119 kindred. *J Clin Endocrinol Metab 77*:484–488.

Bairoch A. 1992. PROSITE: A dictionary of sites and patterns in proteins. *Nucleic Acids Res 20*:2013–2018.

Bucher P, Bairoch A. 1994. A generalized profile syntax for biomolecular sequence motifs and its function in automatic sequence interpretation. In: Altman R, Brutag D, Karp P, Lathrop R, Searls D, eds. *Proceedings, second international conference on intelligent systems for molecular biology*. Stanford, California: Stanford University. pp 53–61.

Cardon LR, Stormo GD. 1992. Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *J Mol Biol 223*:159–170.

Dayhoff MO, Schwartz RM, Orcutt BC. 1978. A model of evolutionary change in proteins. In: Dayhoff MO, ed. *Atlas of protein sequence and structure 5*. Washington DC: National Biomedical Research Foundation. pp 345–352.

Fuchs R. 1991. MacPattern: Protein pattern searching on the Apple Macintosh. *Comput Appl Biosci 7*:105–106.

Galas DJ, Eggert M, Waterman MS. 1985. Rigorous pattern-recognition methods for DNA sequences. Analysis of promoter sequences from *Escherichia coli*. *J Mol Biol 186*:117–128.

Gribskov M, McLachlan AD, Eisenberg D. 1987. Profile analysis: Detection of distantly related proteins. *Proc Natl Acad Sci USA 84*:4355–4358.

Griffin AM, Griffin HG, eds. 1994. *Computer analysis of sequence data, part I*. New Jersey: Humana Press.

Henikoff S, Henikoff JG. 1992. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci USA 89*:10915–10919.

Hodgman TC. 1989. The elucidation of protein function by sequence motif analysis. *Comput Appl Biosci 5*:1–13.

Hsu YR, Ferguson B, Narachi M, Richards RM, Stabinsky Y, Alton NK, Stebbing N, Arakawa T. 1986. Structure and activity of recombinant human interferon-gamma analogs. *J Interferon Res 6*:663–670.

Matsuo Y, Nishikawa K. 1994. Protein structural similarities predicted by a sequence–structure compatibility method. *Protein Sci 3*:2055–2063.

Ogiwara A, Uchiyama I, Kanehisa M. 1993. Sequence motif analysis and retrieval tool. In: Takagi T, Imai H, Miyano S, Mitaku S, Kanehisa M, eds. *Proceedings, genome informatics workshop IV*. Tokyo: Universal Academy Press, Inc. pp 402–410.

Ogiwara A, Uchiyama I, Seto Y, Kanehisa M. 1992. Construction of a dictionary of sequence motifs that characterize groups of related proteins. *Protein Eng 5*:479–488.

Pearson WR, Lipman DJ. 1988. Improved tools for biological sequence comparison. *Proc Natl Acad Sci USA 85*:2444–2448.

Schneider TD, Stormo GD, Gold L. 1986. Information content of binding sites on nucleotide sequences. *J Mol Biol 188*:415–431.

Schwartz RM, Dayhoff MO. 1978. Matrices for detecting distant relationship. In: Dayhoff MO, ed. *Atlas of protein sequence and structure 5*. Washington DC: National Biomedical Research Foundation. pp 353–358.

Sibbald PR, Sommerfeldt H, Argos P. 1991. Automated protein sequence pattern handling and PROSITE searching. *Comput Appl Biosci 7*:535–536.

Stormo GD, Hartzell GW. 1989. Identifying protein-binding sites from unaligned DNA fragments. *Proc Natl Acad Sci USA 86*:1183–1187.

Taylor WR. 1988. Pattern matching methods in protein sequence comparison and structure prediction. *Protein Eng 2*:77–86.

Wallace JC, Henikoff S. 1992. PATMAT: A searching and extraction program for sequence, pattern and block queries and databases. *Comput Appl Biosci 8*:249–254.