

A New Similarity Measure among Protein Sequences

Kuen-Pin Wu, Hsin-Nan Lin, Ting-Yi Sung and Wen-Lian Hsu*

Institute of Information Science
Academia Sinica, Taipei 115, Taiwan

Abstract

Protein sequence analysis is an important tool to decode the logic of life. One of the most important similarity measures in this area is the edit distance between amino acids of two sequences. We believe this criterion should be reconsidered because protein features are probably associated more with small peptide fragments than with individual amino acids.

In this paper, we design small patterns that are associated with highly conserved regions among a set of protein sequences. These patterns are used analogous to the index terms in information retrieval. Therefore, we do not consider gaps within patterns. This new similarity measure has been applied to phylogenetic tree construction, protein clustering and protein secondary structure prediction and has produced promising results.

Keywords: protein sequences, pattern, highly conserved region, protein similarity, protein clustering, phylogenetic tree, protein secondary structure prediction.

1. Introduction

In protein sequence analysis, we often need to know the evolutionary, functional, or structural relationships between two sequences. We are mainly interested in how *similar* they are. One common similarity measure is the *edit distance* [9, 15]. Edit distance concerns with how many edit operations (insertion, deletion, or substitution) on individual amino acids are required to transform one sequence into another. However, we suspect that protein features are probably associated more with small peptide fragments than with individual amino acids. In this paper, we define a new similarity measure based on small fragments, and re-examine several important applications based on this new measure to illustrate its usefulness.

There are various approaches to discover motifs, including statistical approaches [1, 4, 8, 11, 12] and computational approaches [3, 16]. However, many motifs discovered by such approaches are not so *conserved* in the biological sense. We aim to find patterns associated with more *probable* conserved regions. We propose a coding

* Corresponding author. Postal address: Institute of Information Science, Academia Sinica, Taipei 115, Taiwan. Email: hsu@iis.sinica.edu.tw. Phone: +886-2-27883799 ext. 1804. Fax: +886-2-27824814.

scheme to find all short patterns for a set of protein sequences. To maintain computational efficiency, we restrict ourselves to find only patterns of length 4. These patterns are used analogous to the index terms in information retrieval. Therefore, we do not consider gaps within patterns.

To verify the usefulness of patterns, we have carried out several experiments based on the new similarity measure. A pattern-based protein secondary structure prediction algorithm has been implemented. We use DSSP protein dataset to test the algorithm, and have obtained good Q3 evaluation scores. We also use PIR protein sequences for experiments and have obtained a 98% consistency in protein clustering. Our approach does not require time-consuming training or any tuning to cluster proteins. These experiments can be carried out in a few seconds using an Intel Pentium 4 2.4GHz processor with 512MB main memory.

The rest of the paper is organized as follows. In Section 2, we discuss our approach to find short patterns. In addition, we compare pattern similarity and sequence similarity. In Section 3, a pattern-based protein secondary structure prediction approach is implemented and results are reported. Pattern-based approaches to phylogenetic analysis and protein clustering are discussed in Section 4. Finally, concluding remarks are given in Section 5.

2. Finding Short Patterns

Highly conserved regions among a set of protein sequences are useful and interesting targets in proteomic research. The term *highly conserved* regions implies that the corresponding subsequences are not exactly the same, but only *similar*. Therefore, exact matching becomes less desirable for identifying these regions. However, if we can apply exact matching on a set of sequences whose partial similarity has been pre-indexed, the problem would become much easier to solve. To achieve this, we transform protein sequences by grouping amino acids into different sets according to their similarity, and assign each set a unique *code*. Protein sequences are then transformed into *coding sequences*. Exact matching is performed on the coding sequences (rather than the original amino acids) to identify patterns.

In this section, we present an algorithm to find *short patterns* of protein sequences. This algorithm serves as a basic module for further applications including phylogenetic analysis, protein clustering, and protein secondary structure prediction.

2.1. Pattern Generation Algorithm

Many substitution matrices are available on the web for representing similarity among amino acids, where each entry m_{ij} of a substitution matrix M represents the “normalized probability” (score) that amino acid i can mutate into amino acid j . Usually a threshold of scores is determined by users to define similar amino acids according to

user preference on recall or precision. One could choose a lower threshold to obtain better recall, and a higher threshold for better precision.

Among many substitution matrices, PAM [2] and BLOSUM [5] families are most widely used. We use BLOSUM62 as our substitution matrix in this paper based on the comparison mentioned in [5]. Two amino acids are considered *similar* if they have a *positive* score in BLOSUM62. Note that BLOSUM62 is a symmetric matrix, i.e., a being similar to b implies that b is similar to a . We shall consider peptides consisting of four amino acids. The reason to consider peptides of size 4 (rather than 3 or 5) is to balance sensitivity and specificity.

Note that similarity of amino acids is not a transitive relation. For example, based on the score in BLOSUM62, T is similar to S and S is similar to N, but T is not similar to N. Hence, it is not possible to *partition* the 20 amino acids into *similarity classes*. This fact inevitably complicates our computation of similar peptides. It further creates a problem when we consider a set of peptides to be similar.

To clarify these points, let us consider the following similarity graph G on peptides of size 4. The vertex set of G is the set of all peptides of size 4. Two such peptides are considered similar if the two amino acids in each corresponding position (i.e. 1, 2, 3 and 4) are similar. The edge set of G is the set of similar peptide pairs. In graph G , a set of peptides are pair-wise similar iff they form a clique (a complete subgraph) in G . However, this entails that, at every position of this set of peptides, the set of amino acids must be mutually similar. Such a condition is rather restrictive since the clique size is not likely to be very large. Furthermore, finding cliques in a graph is also a time-consuming process. Therefore, such a criterion does not seem to be robust enough to discover similarity for protein sequences.

In this paper we shall bypass the above similarity graph and consider a new similarity measure, one that is easy to implement with relatively high sensitivity and specificity (as illustrated by the applications in Sections 3 and 4). We shall use *codes* to represent similarity among amino acids and use coding sequences to represent similarity among peptides. To generate codes, construct a set $A = \{b \mid a \text{ is similar to } b\}$ for each amino acid a . Note that a itself must be in A . Twenty such sets are generated (see Figure 1). We specifically write a as the first element of the set A to distinguish it from the other sets. Not all elements in such a set are similar; for example, in the set $\{S, A, T, N\}$, S is similar to A, T and N, but A is not similar to T and T is not similar to N. However, those dissimilar pairs in such a set tend to have a relatively high scores (even though they are not positive) in BLOSUM62. Furthermore, if we add up the total scores of each element with respect to the other elements in each set, they all tend to be reasonably high.

{C}, {S, A, T, N}, {T, S}, {P}, {A, S},
 {G}, {N, S, D, H}, {D, N, E}, {E, D, Q, K}, {Q, E, R, K},
 {H, N, Y}, {R, Q, K}, {K, E, Q, R}, {M, I, L, V}, {I, M, L, V},
 {L, M, I, V}, {V, M, I, L}, {F, Y, W}, {Y, H, F, W}, {W, F, Y}.

Figure 1. 20 sets of similar amino acids.

For ease of exposition and implementation, we remove redundant sets and keep the remaining 15 sets. Each set is assigned a code. The codes and their corresponding sets are listed in Table 1. In the following exposition, each code represents a set of *code similar* amino acids. We shall use the terms “code” and “code set” interchangeably. A set of amino acids are code similar (or *c-similar*) if they are contained in the same code.

Table 1. Codes and their corresponding code sets

Code	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Amino acids	C	S	T	P	A	G	N	D	E	Q	R	H	M	Y	F
		A	S		S		S	N	D	E	Q	N	I	H	Y
		T					D	E	Q	R	K	Y	L	F	W
		N					H		K	K		V	W		

Define a *pattern* to be a sequence of codes of length 4. Each amino acid a can be included in more than one code. For example, amino acid S is included in codes 1, 2, 4, and 6. So its corresponding set of codes is $\{1, 2, 4, 6\}$. A peptide of length 4 is said to be an *instance* of a pattern P if the amino acid at each position is included in the code at that position of P . A collection of peptides of length 4 are said to be *c-similar* if there exists a pattern P such that each peptide is an instance of P .

For each protein sequence, we can generate a coding sequence consisting of its corresponding sets of codes. For example, given a protein sequence V-L-S-T-D-N, its corresponding coding sequence is $\{12\}$ - $\{12\}$ - $\{1, 2, 4, 6\}$ - $\{1, 2\}$ - $\{6, 7, 8\}$ - $\{1, 6, 7\}$. We then use a sliding window of length 4 and scan through this coding sequence. For each subsequence of length 4 in the window, we generate all possible patterns. For example, consider the coding subsequence $\{12\}$ - $\{12\}$ - $\{1, 2, 4, 6\}$ - $\{1, 2\}$ created from V-L-S-T. We can generate the following 8 patterns: 12-12-1-1, 12-12-1-2, 12-12-2-1, 12-12-2-2, 12-12-4-1, 12-12-4-2, 12-12-6-1, and 12-12-6-2. For a given pattern, there can be peptides from different protein sequences realizing it (when we consider the similarity among a collection of protein sequences). We can store these peptides as a record of this pattern. We use Protein $i(j)$ to denote the subsequence of protein sequence i starting from position j . We show an example of the record of pattern 12-12-1-1 in Figure 2.

(Pattern: 12-12-1-1)
Protein 0(23): V-L-S-T
Protein 3(25): V-L-A-T
Protein 4(22): V-I-S-T
Protein 6(23): V-L-S-T
Protein 8(139): M-L-A-A

Figure 2. Record of pattern 12-12-1-1.

2.2. Pattern-Based Similarity Measure

Once all short patterns have been found, we define $Pattern(p)$ to be the set of all patterns contained in the protein sequence p . Two similarity scores, S_1 and S_2 , are defined as follows.

Case 1. If the lengths of different protein sequences can be ignored, define $S_1(p_1, p_2)$ as

$$S_1(p_1, p_2) = c \times |Match(p_1, p_2)| / (|Pattern(p_1)| + |Pattern(p_2)|),$$

where $Match(p_1, p_2)$ is the set of patterns shared by sequences p_1 and p_2 , constant c is a normalizing factor.

Case 2. If two protein sequences are required to have similar length, define $S_2(p_1, p_2)$ as

$$S_2(p_1, p_2) = c \times |Match(p_1, p_2)| / (|Pattern(p_1)| + |Pattern(p_2)| - |Length(p_1) - Length(p_2)|).$$

Our coding mechanism reduces the number of candidate fragments to be checked. For example, given a peptide sequence V-L-S-T-D-N, the corresponding similar pattern is $\{V, M, I, L\}$ - $\{L, M, I, V\}$ - $\{S, A, T, N\}$ - $\{T, S\}$ - $\{D, N, E\}$ - $\{N, S, D, H\}$, which generates $4 \times 4 \times 4 \times 2 \times 3 \times 4 = 1536$ possible sequences to be checked. However, in using the coding sequence $\{12\}$ - $\{12\}$ - $\{1, 2, 4, 6\}$ - $\{1, 2\}$ - $\{6, 7, 8\}$ - $\{1, 6, 7\}$, the number of sequences to be checked is reduced to $1 \times 1 \times 4 \times 2 \times 3 \times 3 = 72$. As the length of sequences increases, the coding sequence approach becomes more favorable.

3. Using Patterns for Protein Secondary Structure Prediction

Let Q be a set of proteins whose structural information is known, which shall be used as our training set. To perform protein secondary structure prediction, we first find all patterns in Q . Each pattern is recorded by storing its corresponding proteins, as well as their structural information.

We assume that the secondary structure of a peptide is independent of its length; we use S_1 to calculate similarity score. To predict the secondary structure of a protein p , we first find all patterns of p , and then determine the set $P = \{q \mid q \in Q, S_1(p, q) > t\}$, where t is a cut-off threshold that is used to capture “similar” proteins. Define $W(q) = S_1(p, q)$ as the weight of q with respect to p ; intuitively, similar protein sequences have similar structures.

Now for each pattern x in p , we predict the secondary structure of each amino acid x_j in x , where x_j denotes the j -th amino acid of x . For a type of structure, say helix, we find all proteins $q_i \in P$ that contain x and annotate x_j as helix. We then compute the score $H(x_j)$ that x_j is a helix as:

$$H(x_j) = \sum_i W(q_i).$$

The score of each type of structure can be obtained similarly, and the type with the highest score is predicted to be the structure of x_j . If no protein in P contains x , we assign x_j to be a helix because of its relative high frequency of occurrences.

We have randomly downloaded 10,000 proteins, as well as their structural information from the DSSP database (<http://www.cmbi.kun.nl/gv/dssp/>) to evaluate our approach. The DSSP database is a database containing secondary structure assignments for all protein entries in the Protein Data Bank (<http://www.rcsb.org/pdb/>). We setup several experiments with different data sizes and cut-off thresholds. The prediction precision is measured by $Q3$ as follows:

$$Q3 = \frac{\text{number of residues correctly predicted}}{\text{number of all residues}} \times 100,$$

where $Q3$ considers three conformational states: helix, strand, and coil. The mean precision is defined as follows:

$$\text{Mean precision} = \frac{\sum Q3}{n},$$

where n is the number of proteins to be predicted.

In each experiment, data sets are randomly selected from the 10,000 proteins. Each dataset is partitioned into 10 equal-size groups; one group is used as a testing set and the other 9 groups are used as a training set. Each group will in turn be used as the testing set. Experimental results are listed in Tables 2, 3, and 4 with cut-off thresholds $t = 0.25, 0.35, \text{ and } 0.5$, respectively. The field *recall rate* in these tables is the probability that a protein in the testing set can find at least one protein in P with similarity greater than t .

Table 2. Experimental results with cut-off threshold 0.25.

Training data size	Testing data size	Mean precision	Recall rate
450	50	86.32	0.868
900	100	85.81	0.899
1350	150	85.68	0.906
1800	200	86.16	0.917
9000	1000	89.33	0.964

Table 3. Experimental results with cut-off threshold 0.35.

Training data size	Testing data size	Mean precision	Recall rate
450	50	92.84	0.734
900	100	91.84	0.762
1350	150	91.80	0.783
1800	200	91.22	0.800
9000	1000	92.36	0.893

Table 4. Experimental results with cut-off threshold 0.5.

Training data size	Testing data size	Mean precision	Recall rate
450	50	95.62	0.682
900	100	94.72	0.703
1350	150	94.42	0.708
1800	200	94.16	0.734
9000	1000	94.31	0.842

In comparison to previous works [6, 7, 13, 14], we achieve better mean precision (see Table 5). Currently we cannot find test data of results reported in other papers nor a benchmark for comparison. We believe there are two reasons attributing to this. First, two sequences having low sequence similarity may have high pattern similarity, which better reflects the similarity of secondary structure. Second, previous work may employ low sequence similarity whereas our datasets are randomly selected from the DSSP database and are not restricted to low sequence similarity, say 10~30%. Proteins with low similarity in general have lower accuracy of structure prediction.

Table 5. Accuracy comparison results.

Methods	PSIPRED	PHD	SVM	PROF	Pattern-based
Accuracy	76.5~78.3	72~75	73.5	76	85.7~95.6

4. Using Patterns for Phylogenetic Tree Construction and Protein Clustering

Phylogenetic tree construction and protein clustering are conventionally carried out based on sequence similarity. We shall use patterns to build a phylogenetic tree and to cluster proteins. Among superfamilies with more than 20 proteins from PIR (<http://pir.georgetown.edu/cgi-bin/nbrflist?super>), we randomly select 30 of them. Within each superfamily, we randomly select 8 to 16 proteins. Partition these 30 superfamilies into six datasets, each containing five superfamilies. All protein sequences are well-mixed and no prior knowledge about the superfamilies is used to build the tree. These six datasets are used to perform protein clustering experiments. We illustrate phylogenetic tree construction using an arbitrary dataset chosen from these six datasets. Phylogenetic trees are constructed using UPGMA algorithm [10]. Figure 3 shows the phylogenetic tree of the fourth experiment (listed in Table 6). Note that the five superfamilies of the experiment are arranged into five subtrees as marked in the figure. We assume that homologous proteins have similar lengths, so we apply S_2 similarity measure to build the trees.

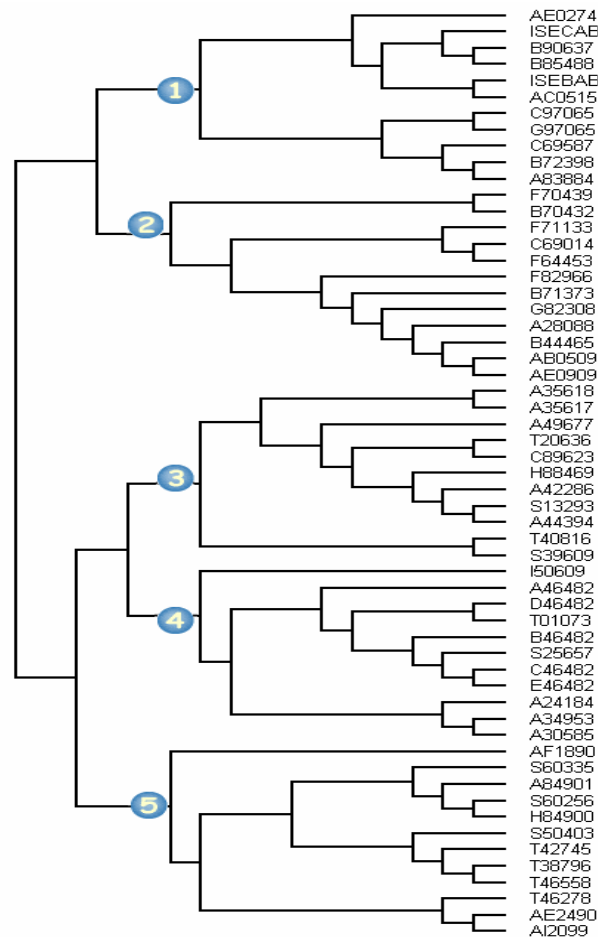


Figure 3. A phylogenetic tree of protein superfamilies.

Protein clusters can be naturally obtained from phylogenetic trees. In four out of

six experiments, our clusters match those of PIR perfectly. In the other two experiments, 98% of the proteins fall in the same PIR clusters. Note that our experiment does not require training, and all experiments can be done in a few seconds. Experimental results are listed in Table 6. The term *miss* means that a protein is arranged into a wrong family. *Accuracy* reflects the overall fraction of the clusters that matches PIR. Almost all 30 superfamilies (348 proteins) are successfully clustered into different subtrees. Only two proteins are arranged into different families in PIR, i.e., two misses.

Table 6. Experimental results of the 6 experiments.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Experiment 6
Superfamily 1	@6 (12)	A9 (12)	D652 (8)	K7 (11)	T229 (14)	X6 (9)
Superfamily 2	@39 (11)	D575 (9)	H1 (11)	K49 (12)	U21 (16)	Y86 (14)
Superfamily 3	@50 (11)	C2725 (10)	H15 (9)	L1 (11)	V6 (14)	Y37 (15)
Superfamily 4	@79 (8)	C3325 (12)	I71 (12)	T8 (11)	V114 (11)	Y39 (15)
Superfamily 5	@44 (13)	A52 (8)	I148 (8)	T36 (12)	V132 (13)	Y23 (16)
Number of proteins	55	51	48	57	68	69
Number of misses	0	1 (S60415)	0	0	1 (F36819)	0
Accuracy	100%	98%	100%	100%	98%	100%
Execution time	18.1 sec	15.1 sec	10.4 sec	12.0 sec	24.9 sec	22.4 sec

It is worth mentioning that our experiments show that proteins with the same secondary structure tend to be clustered together under our algorithm. For example, we have built a phylogenetic tree for the *3',5'-cyclic-GMP phosphodiesterase alpha chain* superfamily. The results are shown in Figure 4. The tree contains two subtrees. The protein names in the first subtree all contain the term “Beta Chain,” whereas those in the second subtree all contain the term “Alpha Chain.”

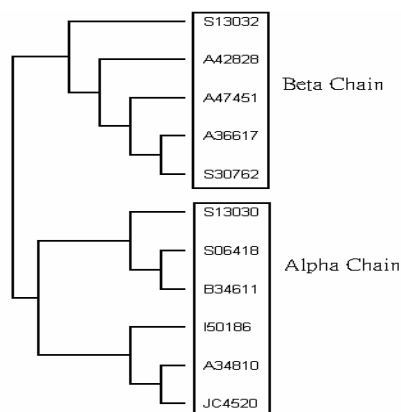


Figure 4. The *3',5'-cyclic-GMP phosphodiesterase alpha chain* protein family.

5. Concluding Remarks

In this paper, we define a new similarity measure based on short patterns. These patterns are used analogous to the index terms in information retrieval; therefore, we do not consider gaps within patterns. In our study, phylogenetic tree construction and protein clustering using existing algorithms on the basis of patterns works quite well. Protein secondary structure prediction using patterns seems to outperform other existing methods. In the future, we shall apply our approach to comparison-based biological sequence research, such as multiple sequence alignment.

References

1. Chang, B.C.H. and Halgamuge, K. (2002) Protein motif extraction with neuro-fuzzy optimization. *Bioinformatics*, **18**, 1084-1090.
2. Dayhoff, M.O., Schwartz, R.M. and Orcutt, B.C. (1978) A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structures*, **15(suppl. 3)**, 345-358.
3. Gao, Y., Mathee, K., Narasimhan, G. and Wang, X. (1999) Motif detection in protein sequences, *IEEE String Processing and Information Retrieval Symposium*, 63-72.
4. Gonnet, P. and Lisacek, F. (2002) Probabilistic alignment of motifs with sequences, *Bioinformatics*, **18**, 1091-1101.
5. Henikoff, S. and Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks, *Proceedings of the National Academy of Sciences of the USA*, 10915-10919.
6. Hua, S. and Sun, Z. (2001) A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure: Support Vector Machine Approach, *J. Mol. Biol.*, **308**, 397-407.
7. Jones, D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices, *J. Mol. Biol.*, **292**, 195-202.
8. Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F and Wootton, J.C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment, *Science*, **262**, 208-214.
9. Levenstein, V.I. (1966) Binary codes capable of correcting insertions and reversals, *Sov. Phys. Dokl.*, **10**:707-710.
10. Michener, C.D. and Sokal, R.R. (1957) A quantitative approach to a problem in classification. *Evolution*, **11**:130-162.
11. Neuwald, A.F., Liu, J.S. and Lawrence, C.E. (1995) Gibbs motif sampling: detection of bacterial outer membrane protein repeats, *Protein Science*, **4**, 1618-1632.
12. Nicodème, P., Doerks, T. and Vingron, M. (2002) Proteome analysis based on mo-

- tif statistics, *Bioinformatics*, **18(suppl. 2)**, S161-S171.
13. Rost, B. (1996) PHD: predicting one-dimensional protein structure by profile-based neural networks, *Methods Enzymol*, **266**, 525-39.
 14. Rost, B. (2001) Review: protein secondary structure prediction continues to rise, *J. Struct. Biol.*, **134**, 204–218.
 15. Sankoff, D. and Kruskal, J. (1983) *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley, Reading, MA.
 16. Smith, H.O., Annau, T.M. and Chandrasegaran, S. (1990) Finding sequence motifs in groups of functionally related proteins, *Proc. Natl. Acad. Sci. USA*, **87**, 826-830.