An Algorithm to Compute Protein Homology Based On Hydrophobic Cluster Analysis

Chanin Chanma[1], Rath Pichyangkura[2], Chotirat Ann Ratanamahatana[1], Prabhas Chongstitvatana[1]

[1]Department of Computer Engineering
[2]Department of Biochemistry
Chulalongkorn University
Phayathai Rd., Pathumwan, Bangkok, Thailand, 10330
chanin.c@student.chula.ac.th, {prath, chotirat.r, prabhas}@chula.ac.th

***Abstract*—Current techniques in protein homology testing involve a 1-dimensional alignment of Nucleotide or Amino acid sequencing. Due to its various constraints and low sequence identity values, a 2-Dimensional Hydrophobic Cluster Alignment has increasingly been used to predict the structure and functionality of protein. This work proposed an algorithm based on a secondary-structure Hydrophobic Cluster Alignment to compute a similarity score of protein sequences automatically, which helps reduce interventions of a human expert for a manual alignment. Additional techniques are introduced to speed up the calculation, as well as to resolve some greedy-based alignment limitation in the previous work. The alignment results and the classification accuracies from the well-known HOMSTRAD database have demonstrated an improvement in both accuracy and the computation time.**

Keywords-bioinformatic; hydrophobic cluster analysis; protien homology; automatic alignment

I. INTRODUCTION

As biological data have grown tremendously in the past decade, they provide an avenue of researches in analyzing and extracting useful knowledge that give us better understanding of the rules of nature. Genome projects are parts of the largest resources of life science data, which mainly include nucleotide and amino acid sequences. However, effective retrieval of these data is still a great challenge. More specifically, we need a high-quality tool to determine protein homology via sequence alignment. Detection of protein homology has become a large research field in bioinformatics. Several crucial analyzed protein databases, such as UniProt [1], PDB [2], SCOP [3], and PFam [4] have been created. These databases contain useful knowledge, e.g., protein homology, structure, or functionalities. By detecting protein similarity, the newly discovered protein sequences can be used to predict their functionalities from the known information in the database.

Early methods, such as Maximum Matching, Basic Alignment Search Tool (BLAST) [5], and FASTA [6], measure protein homology from protein's primary structure information. These methods still have major limitations and drawbacks; they are unable to provide a proof of sequence homology if the sequence identity appears to be too low, a situation that typically occurs in proteins of the same functionality, but belong to organism from different species.

The similarity can generally be measured from an alignment of either nucleotide sequences or amino acid sequences. However, nucleotide sequence similarity is not suitable for protein function discovery; amino acid sequences are typically exploited instead since they contain much more information, such as hydrophobic and hydrophilic properties. Unfortunately, the current one-dimensional alignment tools mentioned earlier all have some limitation that yields poor alignment results. Therefore, higher level structures (Secondary (2-dimensional), Tertiary (3-dimensional), and Quaternary (4-dimensional)) have been increasingly put into consideration. Functionality of a protein is generally based on its 3-Dimensional structure. Some researchers have attempted to predict this protein structure by amino sequence folding based on each amino acid property, but this approach turns out to be unfeasible in practice that extremely high computational power is needed.

Instead, our proposed method is based on the idea of a 2-Dimensional structure, Hydrophobic Cluster Analysis (HCA). Hydrophobicity property is the key in protein folding. The reason is that as protein creating and folding occur in water, they try to preserve the structure by compacting and turning their hydrophobic part inside and resting the hydrophilic part outside for an easy access to water. Hydrophobic Cluster Analysis (HCA) approach has been developed from this belief. HCA approach visualizes amino acid sequence as a 2-D helical pattern. In a Hydrophobic Cluster Analysis, amino acid sequences are laid into a 2-D helical pattern by twisting the protein into a smoothed helix, where each twist will contain 3.6 amino acids [7]. Then, this cylinder is cut lengthwise and spread into a 2-Dimensional plane, and the hydrophobic amino acid will be highlighted and grouped together [8], as shown in Fig. 1. This representation is then used in protein alignment. However the actual hydrophobic cluster alignment requires a human expert to perform the alignment manually.

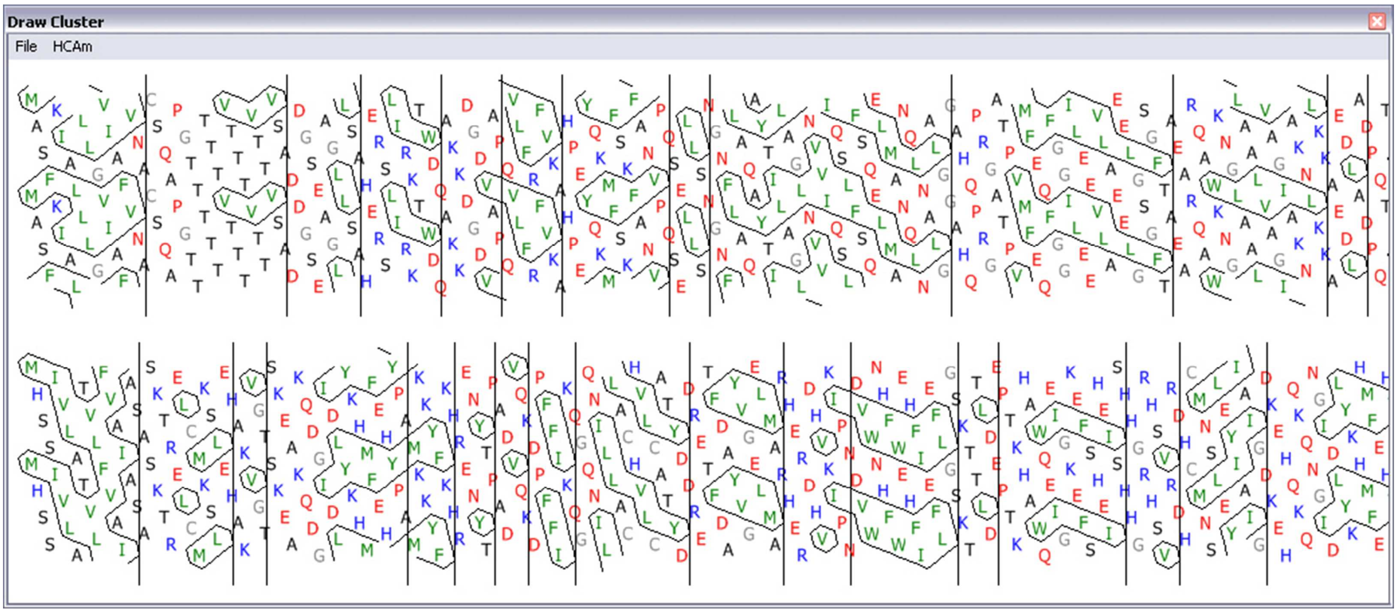


Figure 1. Two-dimensioanal representation of the amino acid sequence using HCA method.

An earlier work of Automatic 2-D Hydrophobic Cluster Alignment [9] introduced the new representation of amino acid sequences and used dynamic programming approach to measure their similarity. The homologous proteins with the same functionalities will have a high sequence identity score and others with different functionalities will have a low sequence score.

In more detail, the algorithm in [9] is based on a dynamic programming approach. Every cluster block from the test sequence will be compared with all cluster blocks from another sequence to discover the best alignment and score. First, a matrix as large as the number of cluster blocks in each sequence is created. Then each cell in the matrix is filled accordingly, starting from the first pair to the last pair. To determine a cumulative value in each current cell, the maximum score of the three neighboring cells (Top, Left, and Diagonal) is added to the current cell's alignment score. This cell's alignment score reflects the best alignment score by shifting residues one-by-one from left to right and updating the remaining residues in the matrix that will be used later. After the score of the last pair is calculated, the best score of the alignment is obtained. The actual alignment can be constructed by tracing the path back to the first alignment pair. Even though this approach achieves good accuracy improvement over the one-dimensional alignment approaches, its alignment is still not optimal and its computational complexity can be much improved.

Therefore, this work proposes an extension of the previous technique in [9]. We revise the representation of the amino acid sequence and improve the dynamic programming algorithm. The goal is to offer a better accuracy alignment score and improve the efficiency of the computation.

II. Proposed Method

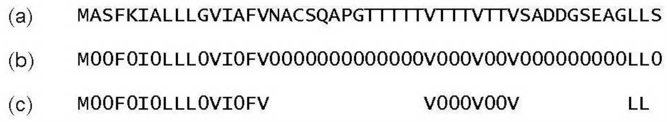
A. New representation

We improve the representation proposed in [9], whose amino acids are simply transformed into binary symbols; "1" represents hydrophobic amino acid and "0" represent hydrophilic amino acid. As a result, the information regarding different hydrophobic amino acids is lost. Instead, our representation replaces only hydrophilic amino acid into "0" since it has no role during the alignment, and keeps the orginal representation of all 7 types of the hydrophobic amino acids (Valine (V), Isoleucine (I), Leucine (L), Phenylalanine (F), Tryptophan (W), Methionine (M), and Tyrosine (Y)). Preserving this hydrophobic information enables our approach to achieve a more accurate score since we can assign different scoring to different types of hydrophobic amino acids. Our substitution matrix is extended from BLOSUM62 [10]. Specifically, we use only the scores of hydrophobic amino acid pairs and the hydrophilic amino acid is substituted with the average score of the hydrophobic amino acids, as shown in Fig. 2.

	M	I	L	V	F	Y	W	Other
M	**5**	1	2	1	0	-1	-1	-2
I		**4**	2	3	0	-1	-3	-3
L			**4**	1	0	-1	-2	-2
V				**4**	-1	-1	-3	-2
F					**6**	3	1	-2
Y						**7**	2	-2
W							**11**	-3

Figure 2. The substitution matrix for hydrophobic cluster alignment.

To extract an amino acid sequence into an individual hydrophobic cluster, we follow the previously proposed method whose streak of five “0”s or more in the sequence, as well as the Proline amino acid symbol (“P”), indicate the end of the cluster. Now our amino acid sequences are transformed into cluster blocks and are ready to be used in our alignment algorithm. Fig. 3 shows our new representation.



```
(a)    MASFKIALLLGVIAFVNACSQAPGTTTTTVTTTVTTVSADDGSEAGLLS
(b)    MOOFOIOLLLOVIOFV0000000000000V000V00V000000000LLO
(c)    MOOFOIOLLLOVIOFV               VOOOVOOV        LL
```

Figure 3. Our proposed representation.
(a) Original sequence. (b) Hydrophilic replaced. (c) Break to cluster block.

In order to recognize, identify, and understand features of cluster blocks easily, we use the visualization tool in [9]. An example visualization of the 2-D clusters in our representation is shown in Fig. 4.

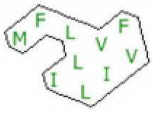
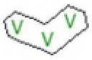
Example	2-D Cluster	Representation
1		M00F0I0LLL0VI0FV
2		V000V00V

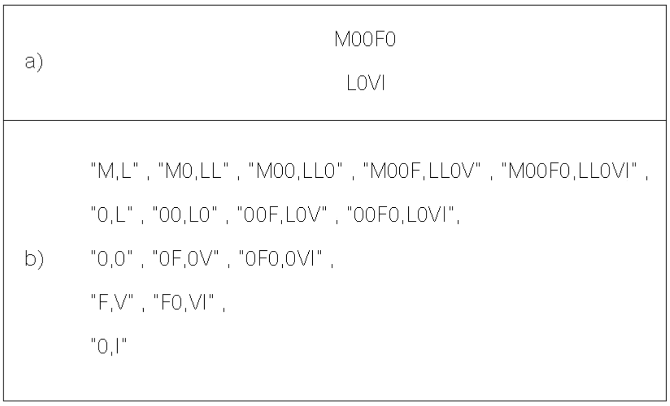
Figure 4. A Visualization corresponded to our representation

B. *Cluster Alignment Algorithm*

In this work, our cluster alignment algorithm also uses a dynamic programming approach and is based on the 2-dimensional string matching technique similar to that in [9]. However, we include additional techniques as follows to increase accuracy and to reduce computation cost.

1) *Score Dictionary Lookup:*

The computation time for an alignment of amino sequences or cluster blocks is spent mostly in score calculations of the sequence pairs. Generally, the scores will be calculated between the new query sequence and each of the sequences in a very large database. Calculating the score of the same sequence repeatedly clearly wastes a lot of computational resources. So, we introduce the score dictionary lookup technique to retrieve the already existed intermediate scores. After we calculate the score of each sequence pair, we add all possible sequences and their corresponding scores into a dictionary. Next time a pair of the sequences with the same substring in the dictionary, we simply lookup without the need of the score’s recalculation. Fig. 5 shows an example of some possible subsequence added to the dictionary.



a) M00F0
L0VI

b) "M,L" , "M0,LL" , "M00,LL0" , "M00F,LL0V" , "M00F0,LL0VI" ,
"0,L" , "00,L0" , "00F,L0V" , "00F0,L0VI",
"0,0" , "0F,0V" , "0F0,0VI" ,
"F,V" , "F0,VI" ,
"0,I"

Figure 5. Examples of some possible subsequences created and added to a dictionary
a) Query Sequence. b) Possible subsequences add to the dictionary.

2) *Local Window Search:*

The approach in [9] scores all pairs of cluster blocks in each sequence and find the best score and its path from the entire table, which evidently requires extremely high computation complexity. Moreover their approch uses a greedy choice in the search, where each score depends on the remaining residue from the previous cluster used. This leads to confusion when each protien sequence have similar cluster blocks but their positions in the sequences are very different. The subsequent cluster will get a mismatch score caused by wrong remaining residue.

Therefore, our work introduces a Local window search to reduce bias from a greedy choice and to create more efficient remaining residue. Before we perform a dynamic programming step, we find a pair of clusters that obtains the maximum score in the window. Then we do a dynamic programing from the starting point to this maximum point in the window. After the dynamic programing step is finished, we move the starting point to the last maximum point. Fig. 6 illustrates an example of our local window search space. As a result, this local window search greatly improves the quality of the scores and reduces the computation time, as will be demonstrated in the experiment section.

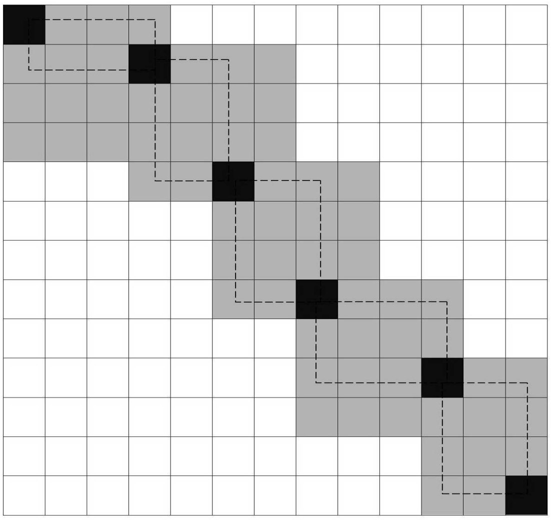


Figure 6. Local search space in the table. The gray areas denote local windows. The black areas denote the maximum score node in each local window. The starting point of a new local window is moved to the previous maximum score node. The dashed rectangles are the areas where we perform dynamic programing.

III. Experiments and Results

We test our approach on HOMSTRAD [11] database, which contains more than 300,000 *annotated* protein sequences. This database classifies their protein sequences into 1032 families based on their structure alignments. We separate this database into training and test set. For each of the sequence in the test set, we search the training set to find the most similar sequence based on our alignment scores. If the best sequence we discover is in the same family as the test sequence, we denote it as a correct answer. In our experiments, test sets include one random sequence from each of the protein family, resulting in a total of 1032 sequences. The training set sizes are varied from 3448 sequences to the entire HOMSTRAD database of 336827 sequences. Note that both training and test data are distinct, where no sequences in the test set overlaps with those in the training set. We compare our classification accuracy with the previous method [9]; results are shown in Table 1.

TABLE I. Experimnetal result

Training size	**Testing size**	**Method**	**Accuracy**
3448	1032	**Our aprroach** Previous approach[9]	**93.50%** 91.47%
44813	1032	**Our aprroach** Previous approach[9]	**95.05%** 94.38%
86678	1032	**Our aprroach** Previous approach[9]	**95.93%** 95.15%
187991	1032	**Our aprroach** Previous approach[9]	**97.18%** 96.32%
336827	1032	**Our aprroach** Previous approach[9]	**98.64%** 97.09%

From the results, our proposed method reports higher classification accuracy based on protein's secondary structure and its homology. Our running time is also significantly improved. However, we decide not to report the raw running time here since it would be unfair to the previous approach as they are implemented under different platforms. However, a simple analysis of our algorithm theoretically confirms that our newly proposed approach will reduce the time complexity by a large margin, especially in massive databases.

IV. Conclusion

In this work, we propose an improved automatic 2-Dimensional Hydrophobic Cluster Alignment algorithm that yields higher classification accuracy with reduced time complexity. As a result, our proposed technique could facilitate and unveil a new opportunity of research in protein's homology, its functionality, and numerous Bioinformatics applications.

References

[1] R. Leinonen, F. G. Diez, D. Binns, W. Fleischmann, R. Lopez, and R. Apweiler, "UniProt archive," *Bioinformatics*, vol. 20, pp. 3236-3237, 2004.

[2] H. M. Berman, et al., "The Protein Data Bank," *Nucl. Acids Res.*, vol. 28, pp. 235-242, 2000.

[3] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *J Mol Biol*, vol. 247, pp. 536-540, 1995.

[4] A. Bateman, et al., "The Pfam protein families database," *Nucl. Acids Res.*, vol. 32, pp. D138-141, 2004.

[5] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J Mol Biol*, vol. 215, pp. 403-410, 1990.

[6] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," *Proc Natl Acad Sci U S A*, vol. 85, pp. 2444-2448, 1988.

[7] Christine Gaboriaud, et al., "Hydrophobic cluster analysis: an efficient new way to compare and analyse amino acid sequences," *FEBS Lett*, vol. 224, no. 1, pp. 149-55, 1987.

[8] L. Lemesle-Varloot, et al., "Hydrophobic-cluster analysis of plant protein sequences. A domain homology between storage and lipid-transfer proteins," *Biochem J*, vol. 255, pp. 901-905, 1990.

[9] P. Kannasut, R. Pichyangkura and C. A. Ratanamahatana, "Automatic 2-D Hydrophobic Cluster Alignment," *Int. J. Biomedical Engineering and Technology.*, inpress.

[10] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks." *Proc Natl Acad Sci U S A*, vol. 89, no. 22, pp. 10 915-10 919, November 1992.

[11] K. Mizuguchi, C. M. Deane, T. L. Blundell, and J. P. Overington, "Homstrad: a database of protein structure alignments for homologous families." *Protein Sci*, vol. 7, no. 11, pp. 2469-2471, November 1998.