Fold Recognition Algorithms used in Protein Structures Prediction

Yuting Xiao
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
xiaox232@morris.umn.edu

ABSTRACT

Protein structure prediction is a critical topic in bioinformatics due to its importance for designing novel drugs and studying proteins' functionalities. In most cases, the procedure of protein structure prediction involves finding known structures and aligning unknown protein structure to single or known protein sequences, in a process called sequence alignment. The procedure of finding templates and aligning unknown protein sequence to templates simultaneously is called fold recognition, or protein threading. In this paper, we will examine the use, implementation, and consequences of using protein fold recognition algorithms to predict protein structure. In particular, we will explore two different algorithms that are both widely used: the first algorithm is called sequence profile-profile alignments (PPA), and the second algorithm uses profile-hidden Markov models (HMMs). Then, we will look into recent improvements upon these two algorithms.

Keywords

fold recognition, protein structure prediction, profile-profile alignment, profile-hidden Markov models, I-TASSER, HHpred

1. INTRODUCTION

Protein's functionalities are considered to be closely related to their structures. In general, protein structures have three levels: *primary structure*, *secondary structure* and *tertiary structure*. Using a protein's primary structure, sometimes assisted by secondary structure information, to build this protein's three dimensional complete structure is called *protein structure prediction*. Protein structure prediction has been one of the main topics in biological field like biomedical science and bioinformatics. Better understanding protein structure can lead us to finding the link between their structure and functionality, and therefore, help us understand the underlying mechanisms of biological functions in cells. In addition, this information is extremely helpful in discovering novel drugs. However, using traditional experimental methods to process protein sequences is expensive and time consuming. Considering the rapidly growing size of sequence data and increasing demand for analysis, we need to adopt computational methods. [17]

UMM CSci Senior Seminar Conference, April 2017 Morris, MN.

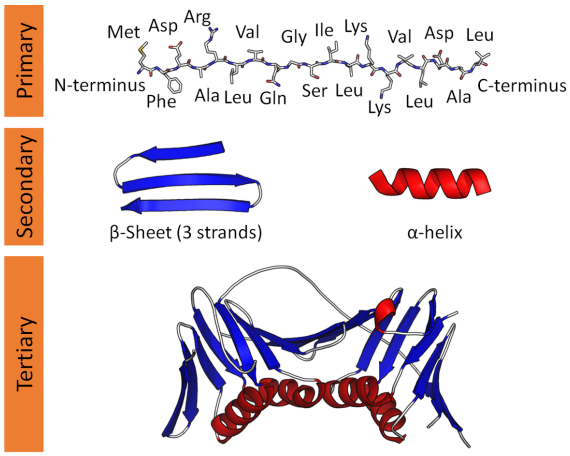

Figure 1: Protein Primary, Secondary and Tertiary Structures. Taken from [14]

In the next section, we will introduce protein structures in details. Then, we will introduce the strategy used for protein structure prediction as well as the *templated-based modeling* and *free modeling* methods based on this strategy. There are two main procedures in a templated-based modeling, where the first procedure, called *fold recognition*, will be the primary interest of this paper and the free modeling will not be discussed in details. The algorithms covered in this paper all use in the template-based modeling for fold recognition. The first algorithm, sequence profile-profile alignments (PPAs) is adopted in the I-TASSER server. The second algorithm uses Hidden Markov Models (HMMs) is used in the HHpred server [16].

2. BACKGROUND

Amino acids, also known as *residues*, are the basic structural units of proteins, and there are 20 different types of amino acids in total. As mentioned in the previous section, protein structures have three levels. Amino acids are bound together linearly in structures that are called polypeptide chain [10]. The linear sequence of amino acids is called the *protein primary structure*, which is shown in the first part of Figure 1. Certain local regions of a primary structure can be organised and folded into regular structures depending on the different amine and carboxyl hydrogen bond formation; those structures are referred to the *Protein secondary*

structures. There are two main types of secondary structures: the *α-helix* and the *β-strand*, which are shown in the second part of Figure 1. *Protein tertiary structures* are three-dimensional complete structures of amino acid chain with multiple secondary structures, as shown in the last part of Figure 1. [14] In order to store protein information on computers, we normally use a unique single letter symbol to represent each amino acid type. For example, "LEVK" represents a very short protein sequence, where L stands for leucine, E stands for glutamic acid, V stands for valine and K stands for lysine.

The basic strategy for predicting unknown protein structure is by copying similar known protein structures. When predicting the tertiary structure of an unknown protein, we can decide on the modeling method based on whether its primary structure or secondary structure is similar to some known protein structures. The freely accessible Protein Data Bank (PDB) contains solved three-dimensional structural data of large biological molecules. Information of known proteins is stored in the form of coordinate files, which contains list of atoms in each protein and their 3D location in space [13]. Protein structure prediction can be categorized into two types: template-based modeling and free modeling [18]. Template-based modeling methods are used when we can find known structures called templates for predicting the unknown protein's structure in PDB. Free modeling methods are used when no similar structures are found; complicated biochemistry principles are needed for calculating predicted structure for unknown proteins.

Template-based modeling has four steps. The first step is finding known structures (templates) in the database that are related to the unknown sequence (target). The second step is aligning the target sequence to the template structure. These two steps together are called *fold recognition*, or *protein threading*, for they are processing at the same time. At the end of the fold recognition process, some regions of the target are aligned with templates, and the remaining regions are unaligned. The third step is building structural frameworks for aligned regions by copying the 3D structures in the database according to appropriately aligned regions of templates and target, where each atom follows spatial constraints. The final step is constructing the unaligned regions using knowledge-based techniques. [18] This paper will focus only on the fold recognition process, and the last two steps of template-based modeling will not be discussed in detail. In the following section, we will examine and discuss the use of the PPA and HMM algorithms in protein fold recognition process and their improvements.

3. PROTEIN THREADING ALGORITHMS

In this section, we will explore two different algorithms used in protein sequence alignment of protein threading process. These two algorithms are both widely used approaches in the protein fold recognition procedure [16]. Both algorithms aim to minimize the "distance" of each amino acid and heuristically find the best alignments between target structure and structural templates. In particular, we will look into the fold recognition PPA program used in the I-TASSER Suite, and the fold recognition using profile-HMMs used in the HHpred server.

3.1 Profile-profile Alignments (PPAs)

In this section, we will discuss the idea of pairwise sequence alignment first, then introduce the concept of a profile. Finally, we will explore the process of the PPA program used in I-TASSER and one of its improvements made in the newly released I-TASSER Suite.

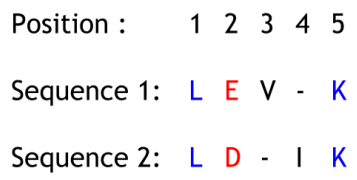

Figure 2: Pairwise Alignment

3.1.1 Pairwise Sequence Alignment

Pairwise Sequence Alignment is the most basic method for comparing two protein sequences and finding the similarity between them. There are two approaches for sequence alignment in general, *global alignments* and *local alignments*. Global alignments aim to find a global optimization or similarity throughout the entire length of all sequences. Local alignments, on the contrary, aim to find only regions of similarity within sequences. The *Needleman-Wunsch algorithm* is generally used for global alignments, and the *Smith-Waterman algorithm* is generally used for local alignments. Both algorithms are based on dynamic programming, which means that both algorithms will find the best choice for each step in order to find the best global or local alignments. We will not discuss the details of these algorithms in this paper, however; detail information about Needleman-Wunsch algorithm can be found in [4], and detail information about Smith-Waterman algorithm is in [7].

When comparing two protein sequences, there are many possible ways to align them. There are three possible alignment results at each specific position. A *match* is defined as both sequences have the same residue at a specific position. A *mismatch* is defined as both sequences have residues at a specific position, but they are two different types of residues. Also, *gaps*, denoted as "-", are inserted in protein sequences, so that similar residues in following positions can be aligned together. For example, in Figure 2, we have matches at positions 1 and 5, a mismatch at position 2 and gaps at positions 3 and 4. In order to measure the quality of an alignment, we need to assign a quantitative value for each possible alignment. Therefore, a scoring system is needed, where higher score indicating more matches in a possible alignment, and the best alignment is the one with maximum match and highest score. Consider, for example, a simple scoring system which given plus 2 for each match position, plus 0 for each mismatch position, and minus 1 for each gap position. Then the alignment shown in Figure 2 would have a score of 2.

3.1.2 Profile

With the idea of pairwise sequence alignment introduced above, the problem we often encounter in real world is multiple sequence alignment (MSA), a sequence alignment with more than three sequences. Applying pairwise sequence alignment repeatedly for MSA is extremely inefficient and expensive for large biodata sets. A new approach is adopted by researchers. When comparing multiple protein sequences,

the target sequence is aligned optimally to a family of similar sequences. This comparison uses position-specific scoring matrix (PSSMs) and gap penalties, which are based on the frequency of amino acid at each position. As we introduced before, there are 20 types of amino acids in total. Therefore, for a sequence with length n, the corresponding PSSM would be a 20 by n table, where each entry value P_{ki} represents the likelihood of observing any amino acid k at position i. This table is called a *profile*. Each likelihood value is calculated using *log-odds* score. Define $q_{i,j}$ as the probability that amino acids i and j correspond to each other in alignments of related sequences, and p_i, p_j are the probability with which residue i and j occurs, respectively [1]. Then log-odds score is defined as:

$$s_{i,j} = log\frac{q_{i,j}}{p_i p_j}$$

Portions of a profile example are shown in Table 1. A profile is a better representation of conserved features of a protein than the sequence itself. The PPAs algorithm we discuss in this paper uses *Position-Specific Iterative Basic Local Alignment Search Tool* (PSI-BLAST) with multiple iterations to generate all protein profiles needed in a alignment process.

3.1.3 PPA Program

First, the PPA program searches the target in database using PSI-BLAST, which is an easy and cheap way to find templates with similar primary structures that we can use to align with the target. A multiple sequence alignment will be returned as a search result. This MSA result then will be used to generate target profiles for aligned regions, using target sequence as the master sequence for each profile. Then the PPA program will align the target profile with template profiles that are pre-generated by PSI-BLAST, and each template profile represents a specific set of protein families [6]. Therefore, PPA program reduces MSA comparison to a comparison that is similar to a pairwise alignment. The log-odds score in PPA program is calculated in a similar way as we discussed above. Define $Q(\vec{x})$ as the probability of observing the data under the assumption of relatedness for alignment column $\vec{x}$, and $P(\vec{x})$ under the assumption of non-relatedness. Then the log-odds score for this column is defined as given in [1]:

$$S(\vec{x}) = log\frac{Q(\vec{x})}{P(\vec{x})}$$

$F_q(i,k)$ represents the frequency of the k-th amino acid at the i-th position of the target multiple sequence alignment, using PSI-BLAST. $L_t(j,k)$ denotes the log-odd profile of a template for the k-th amino acid at the j-th position which was pre-calculated for each template by the PSI-BLAST search. Using $F_q(i,k)$ and $L_t(j,k)$, we can then calculate the residue frequencies likelihood between the target profile and the template profile at corresponding positions, which would be the first term in PPA program's scoring function. As we introduced before, there are mainly two types of secondary structures. Let $s_q(i)$ be the secondary structure at the i-th position of the target sequence, and let $s_t(j)$ represent the secondary structure at the j-th position of a template, then a piecewise function, σ, can be defined: return 1 if the target and template have the same type of secondary structure, and return 0 otherwise. This σ function is used to measure the fitness of secondary structures between target and templates. Parameters of c_1 (=0.65), shift (=−0.96) were used in PPA program, where shift is introduced to avoid the alignment of unrelated residues in the local regions [15]. With these terms defined, the scoring function used in PPAs is defined as:

$$S_{PPAS}(i,j) = \sum_{k=1}^{20} F_q(i,k)L_t(j,k) + c_1\sigma[s_q(i), s_t(j)] + \text{shift}$$

.

The scoring function defined above can be used to determine the best possible match between the target MSA and a template MSA at each corresponding column. Based on this function, the Needleman-Wunsch dynamic programming algorithm is then applied to find the best possible global alignment between the target profile and the template profile.

Table 1: Portion of A Profile Example

Position	1	2	3	4	5
L	−10	−12	−27	0	60
E	−33	1	−36	11	−25
V	−18	−16	11	−2	−29
...	...	...	...	...	...

3.1.4 Improvement

In the newly developed independent package based on the I-TASSER server, which is called I-TASSER Suite, variations of the PPA program are introduced to replace some of the original set of protein recognition programs used in the I-TASSER server [5, 15].

One of the variations of PPA program is called the *Env-PPAS* protein threading program. The Env-PPAS scoring function is developed based on the PPAs program discussed above, with a new environment potential term added. Moreover, the new parameter added in this protein threading program contains information about 3D structural environment fitness, such as *torsion angle*, *solvent accessibility* and secondary structure similarities between the target sequences and template sequences. Within a residue molecule, three atoms can define a plane in the 3D space, and the angle between two planes are defined as a torsion angle [12]. Solvent accessibility refers to the area that one residue molecule can be accessed by outside water [9]. These information are quantified and represented in the new term in order to increase the sensitivity of the algorithm to random alignments. Let $AA_q(i)$ denote the i-th residue of the target sequence in the structural environment, and let $E(j, AA_q(i))$ denote the fitness score between this i-th residue of the target sequence structure and j-th amino acid of the template structure in the structural environment. Parameter c_2 (=0.45) is chosen based on a set of training data [15], then the new Env-PPAS scoring function is defined as:

$$S_{Env-PPAS}(i,j) = S_{PPAS}(i,j) + c_2E(j, AA_q(i))$$

Moreover, based on the new scoring function defined above, the Smith-Waterman local dynamic programming algorithm is then used, instead of the Needleman-Wunsch dynamic programming algorithm, to identify the maximum-match pathway.

3.2 Profile-Hidden Markov Models

Table 2: Transition Probabilities between States

	State A	State B
State A	0.9	0.2
State B	0.1	0.8

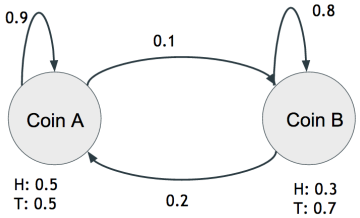

Figure 3: HMM for Coin Toss Example

In this section, we will discuss the general concept of hidden Markov model, and how it was adopted in protein structure prediction in the form of profile-hidden Markov model. Then, we will explore the use of profile-hidden Markov models in the fold recognition process in HHpred server and an improvement proposed by researchers for the scoring function.

3.2.1 General Hidden Markov Models (HMMs)

Hidden Markov Models (HMMs) are used widely in pattern recognition problems, like speech recognition. Moreover, they have been extensively used in bioinformatics since the primary databases in bioinformatics are in the form of string sequences. Many protein structure prediction servers are built on HMMs, including the HHpred server. Generally, a hidden Markov model contains two layers: (1) a visible layer that has symbols, which represent observed events; (2) an invisible layer that has states, which represent invisible internal factors underlying the observation. Each state is distinct from others states. Within the hidden states, a *hidden Markov chain* can be formed based on the *transition probability* between two states [17]. A transition probability $t_{i,j}$ is the probability of switching from current state i to state j. When proceeding a Markov chain, the future state of each step depends only on the current state. Moreover, a hidden Markov model can also has a start state and an end state. In general, we can consider a hidden Markov model as a probabilistic finite state machine.

Consider an example of tossing coins, with two coins available. One coin A is a fair coin with 0.5 probability to generate either a head (H) or a tail (T). The other coin B is a biased coin with a 0.3 probability to generate a H, and a 0.7 probability to generate a T. You are given that one person tossed coins 6 times without knowing which coin was used at each toss. In this case, a hidden Markov model can be used as a probabilistic model that best explains a sequence of observations for a coin tossing result like $O = \{H, H, T, T, H, T\}$. In order to know this, we have to know which coin was used for each toss. Therefor, two hidden states, A state and state B, are needed to represent the fair coin A and biased coin B in this model. Each state contains a set of probabilities of generating, or emitting, different events H and T; they are called the *emission probabilities* for each state. Given the transition probabilities

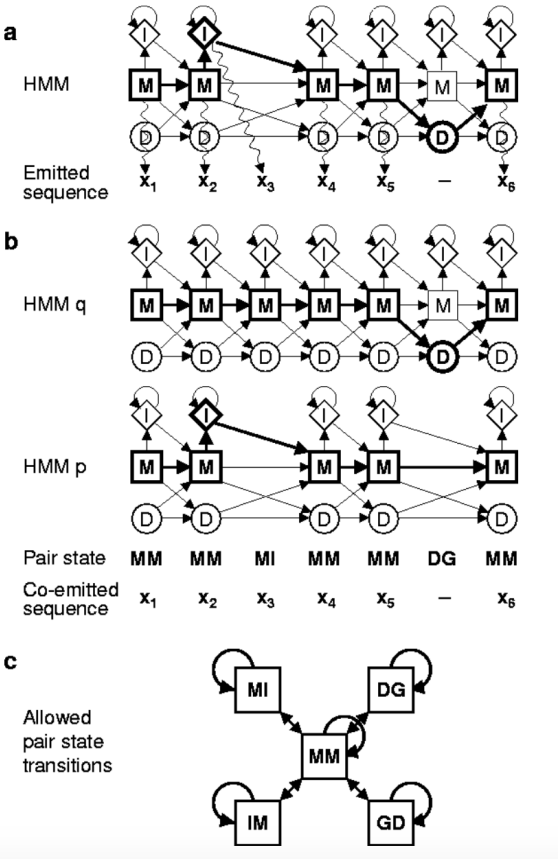

Figure 4: profile-HMMs and Pairwise alignment. Taken from [8]

between state A and B in Table 2, a corresponding hidden Markov model is shown in Figure 3. If a state sequence like $S = \{A, A, B, B, A, B\}$ is given, we then are able to calculate the probability for observations O. For example, with the state sequence S, the probability of producing a sequence O can be calculated as $P_{O,S} = 0.5*0.9*0.5*0.1*0.7*0.8*0.7* 0.2 * 0.5 * 0.1 * 0.7$. There are many different possible state sequences for a set of observations O; we do not know which one is actually used. The best we can do is then choose the model with maximum likelihood.

3.2.2 Profile-HMMs

The idea of HMMs has been adopted in computational biological science for aligning protein sequences, which is called *profile-HMMs*. Profile-HMMs are a variation of the general HMMs that were discussed above. They are structured specifically for modeling sequence profiles. Moreover, they are similar to protein sequence profiles in a way. Besides the residue frequencies at each position, profile-HMMs also contain information about the position-specific probabilities for insertions and deletions [8]. Profile-HMMs have linear left-to-right structures that contain three types of hidden states: match states (M_k), insert states (I_k) and delete states (D_k), which represent position-specific symbol frequencies, symbol insertions, and symbol deletions at the k-th state, respectively [17].

In profile-HMMs, only match states and insert states can

emit amino acids and they can align with each other, whereas delete states can not emit amino acids and they can only align with other delete states or gaps G. Therefore, in alignments between profile-HMMs, there are 5 possible pair states: MM, MI, IM, DG, GD (shown in Figure 4c), where pair states II and DD are excluded.

We denote the observed residue symbol sequence as $X = x_1, x_2, ..., x_L$, and the underlying state sequence as $Y = y_1, y_2, ..., y_L$, where y_n is the corresponding underlying state of the n-th residue x_n. The transition probability from state i to state j is denoted as $t(i, j)$. An example for an alignment between a sequence and profile-HMM is shown in Figure 4a, where the bold arrows represent a path through the HMM.

3.2.3 Log-sum-of-odds Score and Column Score

In the previous PPA section, we discussed the scoring function, a quantitative measurement for sequences' similarity, which is used for finding the best alignment between profiles. In HMMs, a different scoring function called *Log-sum-of-odds* is used. An example of pairwise alignment between HMM p and HMM q is shown in Figure 4b. In general, the log-sum-of-odds score measures the probability that a sequence is coemitted by both HMMs rather than by a random null model [8]. Thus, for a profile-HMM alignment with length n, the log-sum-of-odds score is defined in HHpred server as:

$$S_{LSO} = \log \sum_{x_1,...,x_m} \frac{P(x_1, ..., x_n|\text{co-emission on path})}{P(x_1, ..., x_L|\text{Null})} \quad (1)$$

In Equation 1, the numerator represents the probability that x_1, x_2, ..., x_L is coemitted by both HMMs along the alignment path. It is the product of the amino acid coemission probabilities for each match state pair on the path and the transition probabilities between match state pairs. The denominator represents the probability of a sequence generated from null model. $P(x_1, ..., x_L|\text{Null}) = \prod_{L=1}^{L} f(x_l)$, where $f(x_l)$ are the fixed amino acid background frequencies that are calculated based on target profile. For insert states, the probability of emitting an amino acid a is defined as the same as the fixed amino acid background frequency $f(a)$. We denote $q_{i(k)}(a)$ and $p_{j(k)}(a)$ as the probabilities that HMMs q and p emit amino acid a in match state i or j in k-th column, and use $t_i(X, X')$ and $t_j(Y, Y')$ as the transition probabilities from state X or Y in column i or j to a state X' or Y', where $X, X', Y, Y' \in \{M, I, D\}$. Using notations discussed above, a term called *Column score*, which represents the frequency likelihood between HMMs q and p at corresponding position, then can be defined as :

$$S_{aa}(q_i, p_j) = \log \sum_{a=1}^{20} \frac{q_i(a)p_j(a)}{f(a)} \quad (2)$$

Moreover, with the notations defined above, we can rewrite Equation 1 as:

$$aS_{LSO(i,j)} = \sum_{k:X_kY_k=MM} S_{aa}(q_{i(k)}, p_{j(k)}) + \log \mathcal{P}_{tr} \quad (3)$$

In Equation 3, $\mathcal{P}_{tr}$ represents the product of all transition probabilities from the path through p and q. $S_{aa}(q_{i(k)}, p_{j(k)})$ compares the amino acid distributions from the two HMMs up to position i and j. A positive column score means that two distributions are similar to each other, and a negative column score means otherwise.

3.2.4 Pairwise profile-HMMs Alignment

For a profile-HMM pairwise alignment, as shown in Figure 4b, the *Viterbi algorithm* is used for finding the path with the maximum log-sum-of-odds score through the two HMMs. Again, we will not discuss the process of Viterbi algorithm in this paper; detailed information about Viterbi algorithm is provided in [3]. Five dynamical programming matrices S_{XY} are defined for the 5 pair state, MM, MI, IM, DG, GD (shown in Figure 4c), in order to recursively calculate the score of the best partial alignment that ends in column i of HMM q and column j of HMM p in pair state XY. The total log-sum-of-odds score is then defined as the maximum over the whole matrix S_{MM} [8]. The 5 dynamical programming matrices with the base case at $(0, 0)$ are defined as below, where $S_{IM}(i, j)$ and $S_{GD}(i, j)$ are similar to $S_{MI}(i, j)$ and $S_{DG}(i, j)$.

$$S_{MM}(i, j) = S_{aa}(q_i, p_j) +$$

$$\max \begin{cases} S_{MM}(i-1, j-1) + \log[q_{i-1}(M, M)p_{j-1}(M, M)] \\ S_{MI}(i-1, j-1) + \log[q_{i-1}(M, M)p_{j-1}(I, M)] \\ S_{IM}(i-1, j-1) + \log[q_{i-1}(I, M)p_{j-1}(M, M)] \\ S_{DG}(i-1, j-1) + \log[q_{i-1}(D, M)p_{j-1}(M, M)] \\ S_{GD}(i-1, j-1) + \log[q_{i-1}(M, M)p_{j-1}(D, M)] \end{cases}$$

$$S_{MI}(i, j) = \max \begin{cases} S_{MM}(i-1, j) + \log[q_{i-1}(M, M)p_j(M, I)] \\ S_{MI}(i-1, j) + \log[q_{i-1}(M, M)p_j(I, I)] \end{cases}$$

$$S_{DG}(i, j) = \max \begin{cases} S_{MM}(i-1, j) + \log[q_{i-1}(M, D)] \\ S_{GD}(i-1, j) + \log[q_{i-1}(D, D)] \end{cases}$$

3.2.5 Improvement

One improvement made on the profile-HMMs method is to add protein structural information like protein solvent accessibility and torsion angles information to the scoring function [2].

A large data set called CASP9 is used for testing this new method. Results show that adding solvent accessibility and torsion angles information improve the accuracy of HMM-based pairwise profile-profile alignments. Additional evolutionary residue coupling information did not show significant improvement in the given experimental setting, however, it still may be a potential source of information for improvement [2].

We define the $S_{MMorig}(q_i, p_j)$ as the previous dynamic programming matrix for S_{MM}, $S_{ss}(q_i, p_j)$ as the secondary structure score between column i in profile-HMM q and column j in template profile-HMM p, $S_{sa}(q_i, p_j)$ as the solvent accessibility score between q and p, and $S_{tors}(q_i, p_j)$ as the torsion angle score between q_i and p_j, with corresponding weight w_{ss}, w_{sa}, and w_{tors}. The new dynamic programming matrix for S_{MM} is then defined as below, where the other 4 matrices remain the same.

$$S_{MM}(i, j) = S_{MMorig}(q_i, p_j) + \\ w_{ss}S_{ss}(q_i, p_j) + w_{sa}S_{sa}(q_i, p_j) + w_{tors}S_{ss}(q_i, p_j)$$

4. RESULT

In the previous section, we closely looked at the protein fold recognition algorithms used in the I-TASSER server, and one of the improvements made in its I-TASSER Suite package. Both the I-TASSER server and the I-TASSER Suite use meta-threading programs, which means that they run different protein threading programs in fold recognition process and use the best common result returned by those programs as the final best alignment. We discussed one of the threading program, the Env-PPA program used in the I-TASSER Suite, where many other variations of PPA program are also used. The set of variations of the PPA program is important for increasing the coverage of template detections in protein fold recoginition process [15]. *Critical Assessment of protein Structure Prediction (CASP)*, is a worldwide experiment for protein structure prediction taking place every two years since 1994. This experiment aims at testing various protein structure prediction methods, based on their performances on identifying protein three-dimensional structure from its amino acid sequence [11]. The I-TASSER Suite was tested in the recent CASP experiment, including CASP10. The protein structure prediction accuracy generated by I-TASSER Suite was 20 percent higher than that of the second-best method in the experiment for $4,271$ targets [15].

We also looked at how profile-HMMs were used in the protein fold recognition process, and one improvement proposed in paper [2]. In this research, the improved method along with the original profile-HMM based profile-profile alignment, which is used in HHpred server, were tested on the alignments between 106 targets of CASP9. As a result, both the number of correctly aligned pairs of residue in the predicted alignment and the number of correctly aligned columns are a little higher when using the improved scoring method [2].

5. CONCLUSION

In this paper, we explored the use of two different algorithms, PPA and profile-HMMs, for finding the best alignment between targets and templates in the fold recognition procedure in protein structure prediction. Both approaches are popular tools in biological sequence analysis [16], and they both have their own advantages and disadvantages in different situations.

We will not discuss the direct comparison between these two programs' performances. However, we have statistical evidence for the performances of the two servers that are using PPA and profile-HMMs for protein structure prediction in the CASP experiments. The I-TASSER server, which uses PPA with other methods for fold recognition, has been in the top 3 places for the last few CASP experiments, and the HHpred server, which uses profile-HMMs based fold recognition, has also been in the top 10 list.

In this paper, we discussed one variation of PPA program, the Env-PPA program used in the new independent I-TASSER Suite package, and one improvement for the profile-HMM scoring function proposed by researchers. Moreover, we discussed the result for both improvements. There are many other improvements that have been made for both algorithms, in order to find better performances in different situations. Nevertheless, there is still no single method that outperforms all others on every target [18], which makes the protein threading process challenging.

6. ACKNOWLEDGMENTS

I would like to thank Elena Machkasova, Peter Dolan, Nic McPhee and Kirbie Dramdahl for their time and valuable to the drafts of this paper.

7. REFERENCES

[1] S. F. Altschul, J. C. Wootton, E. Zaslavsky, and Y.-K. Yu. The construction and use of log-odds substitution scores for multiple sequence alignment. *PLoS Computational Biology*, 2010.

[2] X. Deng and J. Cheng. Enhancing HMM-based protein profile-profile alignment with structural features and evolutionary coupling information. *BMC Bioinformatics*, 2014.

[3] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[4] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[5] A. Roy, A. Kucukural, and Y. Zhang. I-TASSER: A unified platform for automated protein structure and function prediction. *Nature protocols*, 5(4):725–738, 2010.

[6] L. Rychlewski, L. Jaroszewski, W. Li, and A. Godzik. Comparison of sequence profiles. strategies for structural predictions using sequence information. *Protein Science*, 9(2):232–241, 2000.

[7] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[8] J. Söding. Protein homology detection by HMM–HMM comparison. *Bioinformatics*, 21(7):951–960, 2005.

[9] Wikipedia. Accessible surface area — Wikipedia, The Free Encyclopedia, 2017. [Online; accessed 24-April-2017].

[10] Wikipedia. Amino acid — Wikipedia, The Free Encyclopedia, 2017. [Online; accessed 30-April-2017].

[11] Wikipedia. CASP — Wikipedia, The Free Encyclopedia, 2017. [Online; accessed 9-April-2017].

[12] Wikipedia. Dihedral angle — Wikipedia, The Free Encyclopedia, 2017. [Online; accessed 24-April-2017].

[13] Wikipedia. Protein data bank — Wikipedia, The Free Encyclopedia, 2017. [Online; accessed 1-May-2017].

[14] Wikipedia. Protein structure — Wikipedia, The Free Encyclopedia, 2017. [Online; accessed 21-March-2017].

[15] J. Yang, R. Yan, A. Roy, D. Xu, J. Poisson, and Y. Zhang. The I-TASSER suite: Protein structure and function prediction. *Nature methods*, 12(1):7–8, 2015.

[16] J. Yang and Y. Zhang. Protein structure and function prediction using I-TASSER. *Current protocols in bioinformatics*, pages 5–8, 2015.

[17] B.-J. Yoon. Hidden markov models and their applications in biological sequence analysis. *Current genomics*, 10(6):402–415, 2009.

[18] Y. Zhang. Progress and challenges in protein structure prediction. *Current opinion in structural biology*, 18(3):342–348, 2008.