A Method of Multiple Protein Sequence Alignment Using a Hybrid Approach

Ismot Ara*

Abstract—Multiple protein sequence alignment is an extension of pairwise alignment to incorporate more than two sequences at a time. Multiple protein sequence alignment methods try to align all of the sequences in a given query set. Multiple protein sequence alignments are often used in identifying conserved sequence regions across a group of sequences hypothesized to be evolutionarily related. Many approaches are available to align multiple protein sequences. The most accurate and fastest method is MUSCLE. In this paper, a new method has been proposed for multiple protein sequence alignment, which combines progressive, iterative and probabilistic approaches. This method performs pairwise alignment, constructs guide tree and then uses progressive approach to do profile-profile alignment. The iterative approach is used for refining the aligned sequences to improve the results. It will be expected that this method will produce better results than existing methods.

Index Terms— Protein, Multiple Sequence, Progressive, Iterative, Alignment, Protein Sequence.

1 Introduction

Bioinformatics, is as conceptualizing biology in terms of macromolecules (in the sense of physical-chemistry) and then applying the technique "informatics" (resulting from disciplines such as applied mathematics, computer science, and statistics) to know and systematize the information associated with these molecules, on a large-scale. In short, bioinformatics influence the techniques taken from computer science to solve problems in molecular biology [1]. Protein Sequence alignment is a technique in bioinformatics for visualizing the relationships between residues in a collection of evolutionarily or structurally related proteins. Protein sequence alignment is the task of identifying evolutionarily related positions in a collection of amino acid sequences [2]

Sequence alignment aims at constructing an alignment of two or more sequences in such way that similarities of these sequences can be minimized. When there are two sequences occupied, then it is called pair wise alignment. For more than two sequences it is called multiple sequence alignment [3]. Domains may be repeated or shuffled for many biologically important protein families [4].

Multiple alignments of protein sequences are vital points in studying proteins. The basic information they provide is identification of conserved sequence regions. This is very helpful in designing experiments to test and modify the function of specific proteins, in predicting the function and structure of proteins, and in identifying new members of protein families.

There are two types of alignment, global and local. Global alignments, which attempt to align every residue in every sequence, are most important when the sequences in the query set are similar and of roughly the same size. (This does not mean global alignments cannot end in gaps.) Local alignments are more useful for dissimilar sequences that are suspected to contain regions of similar sequence motifs within their larger sequence.

Amino acid sequences in a protein are defined by the sequence of a gene, which is encoded in the genetic code. In general, a protein sequence is a string of amino acids, each represented by a single letter. There are 20 different amino acids. Typically proteins are about 300 amino acids long sequences as shown in bellow.

… I M U L A N K V K V K M U T …

The distinctive sequence of amino acid residues the unique characteristic of each protein in its polypeptide chain. The amino acid sequence is the relation between the genetic information in DNA and the three-dimensional structure that performs a protein's biological task. Protein sequence comparison among similar proteins yields insights into the evolutionary relationships of proteins and protein function [6]. The comparison of sequences between normal and mutant proteins provides invaluable information toward identification of discovery residues consisted of protein function as well as detection and treatment of diseases. The knowledge of amino acid sequences is essential to the protein's three-dimensional structure, mechanism of actions, and design [6]. The importance of protein sequence is to compare two or more sequences and to determine the similarities in their functions.

Multiple sequence alignment mainly refers to searching for similarity in three or more sequences [7]. In general Multiple Protein Sequence Alignment is a sequence alignment of three or more protein sequence. These sequences can be for DNA or RNA [4]. A vital role in molecular sequence analysis is played by sequence alignment. It can help to build a phylogenetic tree of related DNA sequences or to predict the function or structure of unknown protein sequences by aligning with other known sequences. Protein is one of three domains of biological

- *Ismot Ara-Computer Science and Engineering Discipline, Khulna University, Khulna-9208, Bangladesh. Dept. of Computer Science and Engineering, Faculty of Science and Technology, Atish Dipankar University of Science and Technology, Dhaka-1213, Bangladesh.*
- *Corresponding author: Ismot Ara, E-mail: preetycse07@yahoo.com*

sequences. Other two is DNA and RNA. Only when all involved sequences are defined in the same domain then the sequence alignment makes sense. The target of sequence alignment to construct an alignment of two or more sequences so as to maximize similarities of these sequences [3].

Multiple sequence alignment which is an extension of pairwise alignment used to incorporate more than two sequences at a time. Multiple alignment methods try to align all of the sequences in a given query set. And also used in identifying conserved sequence regions across a group of sequences hypothesized to be evolutionarily related [4].

A formal definition can be defined as follows [7]:

Let us assume that, a set of sequences, s1… sk, where each character is taken from an alphabet Σ, and does not contain the particular gap character '-'. A multiple alignment of protein sequences is called rectangular array, consisting of characters taken from another alphabet Σ', which is Σ plus the gap character "-", that satisfies the following three conditions:

1. There are absolutely **k** rows.

2. Avoiding the gap character; i, row number, is exactly the sequence s_i

3. Each column contains at least one character unlike from "-".

As for Example, There are three protein sequences as shown in bellow-

X= TISCTGSSSNIGAGNHVKWYQQLPG

Y= VTISCTGTSSNIGSITVNWYQQLPG

Z= LRLSCSSSGFIFSSYAMYWVRQAPG

The final alignments of these three sequences are as follows,

X= TISCTGSSSNIGAG NHVKWYQQLPG

Y= VTISCTGTSSNIGS—ITVNWYQQLPG

Z= LRLSCSSSGFIFSS—YAMYWVRQAPG

In this paper the objective is to study different field in Bioinformatics(Protein and protein sequence) and sequence alignment. The target is to solve the following problems.

- The choice of the sequences
- The choice of an objective function
- The optimization of that function

2 Related Work

2.1 MUSCLE

MUSCLE (Multiple Sequence Comparison by Log-Expectation), which is proposed by R.C Edgar, which uses two distance measures for a pair of sequences: a k-mer (A k-mer is a contiguous subsequence of length k) distance (for an unaligned pair) and the Kimura (the fractional uniqueness calculated from a global alignment of the two sequences) distance (for an aligned pair). The first stage to progressive alignment build is the draft progressive. At first the similarity of each pair of sequences is computed, either using *k*-mer counting or structuring a global alignment of the pair and determining the fractional identity. The calculation of a triangular distance matrix is taken from the pair-wise similarities. A tree is constructed from the distance matrix using UPGMA or neighbor-joining, and a root is identified. Then the progressive alignment is built by following the tree of branching order, constructing a multiple alignment of all input sequences at the root. The second stage attempts to develop the tree and constructs a new progressive alignment according to this tree. This stage may be iterated. Match of each pair of sequences is found out using fractional uniqueness computed from their mutual alignment in the current multiple alignments. Then, a tree is constructed by computing a Kimura distance matrix and applying a clustering method to this matrix and the previous and new trees are compared, identifying the set of internal nodes for which the branching order has changed. If this stage has executed more than once, and the number of changed nodes has not decreased, the process of developing the tree is measured to have converged and iteration terminates. A new progressive alignment is built. The existing alignment is retained of each sub tree for which the branching order is unchanged; new alignments are formed for the possibly empty set of altered nodes. When the alignment at the root is completed, the algorithm may terminate. The third stage performs iterative refinement using a variant of tree-dependent restricted partitioning. This tree is divided into two sub trees. Taking an edge of the tree to create two groups. The sequences in the subtree are used to build a multiple alignment and then a profile. By realigning the two profiles a new multiple alignments is build. If this new alignment improves the score, it kept. Otherwise it is discarded.

The MUSCLE algorithm uses k-mer to determine distance matrix. It is an iterative process [4]. The iterative process is so much time consuming and very much unreliable [1].

2.2 Unweighted Pair Group Method with Arithmetic mean (UPGMA)

Unweighted: All pair wise distances contribute similarly.
Pair-group : Groups are united in pairs (dichotomies only)
Arithmetic mean : Pairwise distances to each group (cable) are mean distances to all members of that group.

UPGMA is simple agglomerative or hierarchical clustering method used in bioinformatics. UPGMA assumes a constant rate of evolution. UPGMA was initially designed for utilize in protein electrophoresis studies, however, is currently most often used to produce guide trees for more sophisticated rebuilding algorithms.

The structure present in a pair wise distance matrix is examined by the algorithm that construct a rooted tree. At each step, the nearest two clusters are combined into a higher-level cluster. The distance between any two clusters A and B is taken to be the standard of all distances between

pairs of objects "x" in A and "y" in B, that is, the mean distance among elements of each cluster:

UPGMA characteristics :

1. UPGMA is the simplest method for constructing trees.
2. Generates rooted trees (re-rooting is not allowed)
3. Generates ultrametric trees

UPGMA characteristics have been shown in figure 1.

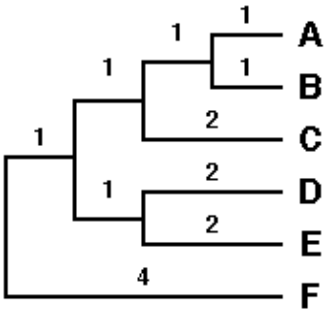

Figure 1: Example of UPGMA

The UPGMA algorithm:

- UPGMA starts with a matrix of pair wise distances D[1..n, 1..m].
- In the following text each sample as for example taxon, operational taxonomic unit (OTU) is denoted as a 'cluster'.
- starts by assigning all clusters (samples) to a star-like tree
- Find that pair (cluster i and j) with the smallest distance value in the distance matrix: D[i,j].
- Define a new cluster comprising cluster i and j: Cluster i is connected by a branch to the common predecessor node. The same condition applies for cluster j.
  Therefore, the distance D[i,j] is split onto the two branches. So, each of the two seperate branches obtains a distance of D[i,j]/2.
- If i and j were the last 2 clusters, the tree is finished. If not the algorithm finds a new cluster called u.
- Go back to step 1 with one less cluster. Clusters i and j are removed, and cluster u is added to the tree.

2.3 MAFFT

MAFFT (Multiple Protein Sequence Alignment Method Based on Fast Fourier Transform) is
proposed by K. Katoh *et al*, one of the fastest methods among the available update multiple alignment tools and used in projects [11]**.**

In this method an initial distance matrix is constructed from the pairwise scores, instead of shared 6- tuples and a *guide* tree is build with the UPGMA (Unweighted Pair Group Method with Arithmetic Mean) with modified linkage. Each pairwise alignment is splited into gap-free segments and number *n* is assigned to each segment. The information of these segments is stored in a set of arrays, which represents the alignment score of the *nth* gap-free segment between sequences. The frequency value, which represents how frequently a fixed place of sequence is involved in gap free segments, is calculated. The procedure of progressive and iterative refinement methods is used to generate the alignment of subsets referred to as 'group'. The group alignment is improved by the iterative refinement method. Though MAFFT is a progressive alignment approach, it depends on initial pairwise alignment [11, 12]. And it is so much unreliable and the method cannot be globally optimized.

3. Method

To study gene evolution across a wide range of organisms, biologists need accurate platform for multiple sequence alignment of protein group. Obtaining accurate alignments, however, is a difficult computational problem because of not only the high implementation cost but also the lack of proper objective functions for measuring alignment quality. In this paper, *probabilistic algorithm* has been introduced, a novel scoring function for multiple sequence comparisons. It has been presented Pair Hidden Marcov model, a practical tool for progressive protein multiple sequence alignment based on probabilistic algorithm, and has been evaluated its performance on several standard alignment.

To implement this method the following steps are performed.

Given m sequences, $S = \{s(1), \ldots, s(m)\}$:

Step 1: Hidden Marcov model for Calculating of posterior-probability matrices

For every pair of sequences x, $y \in S$ and all $i \in \{1,\ldots, |x|\}$, $j \in \{1,\ldots,|y|\}$, find out the matrix Pxy, where $Pxy(i, j) = \mathbf{P}(xi - yj \in a^* \mid x, y)$ is the probability that xi and yj are combined in a^*.

Step 2: Hidden Marcov model for probabilistic consistency transformation

Estimate the match quality scores $\mathbf{P}(xi - yj \in a^* \mid x, y)$ by applying the *probabilistic consistency transformation*, which incorporates match of x and y to other sequences from S into the x–y pairwise comparison:

$$\mathbf{P1}(xi - yj \in a^*[x,y) \leftarrow 1/[S] \sum_{z \in S} \sum_{zk} \mathbf{P}(\text{xi-zk} \in a^*[x,z)\ \mathbf{P}(zk - yj \in a^*[z,y).$$

In matrix form, the transformation may be written as

$$\mathbf{P1}_{xy} \leftarrow 1/[S] \sum_{z \in S} Pxz, Pzy.$$

Since most values in the Pxz and Pzy matrices will be near zero. This step may be repeated as many times as desired.

Step 3: Computation of guide tree using clustering with the Unweighted Pair Group Method with Arithmetic mean (UPGMA).

Construct a guide tree for S through hierarchical clustering. A measure of similarity between two sequences x and y use $E(x, y)$. Define the similarity of two clusters by a weighted average.

Step 4: Progressive alignment using Sum-of-Pairs scoring function

Align sequence groups hierarchically according to the order specified in the guide tree. Alignments are scored by using a sum-of-pairs scoring function in which aligned residues are assigned the transformed match quality scores $\mathbf{P}1(x_i - y_j \in a^* \mid$

x, *y*) and gap penalties are set to zero.

Step 5: Iterative refinement
Randomly partition alignment into two groups of sequences are realigned. This step may be repeated as many times as desired. In addition to the steps shown, it has been experimented also with the generation of automatic column reliability annotations for the alignment based on the posterior matrix formulation.

3 Description

The pair HMM works by
(1) computing posterior-probability matrices,
(2) applying the probabilistic consistency transformation,
(3) computing guide tree, and
(4) performing progressive alignment.
As a default, iterative refinement is performed as a post-processing step. In this section each of these steps has been considered in detail.

Step 1: Posterior probability matrices
Let *x* and *y* be two proteins represented as character strings in which x_i is the *i*th amino acid of *x*. An alignment *a* corresponds uniquely to a sequence of state emission pairs, $(s_1, o_1), \ldots (s_n, o_n)$. The probability of *a* is given by

$$\mathbf{P}(a|x,y) = \pi(s1) \left(\prod_{i=1}^{n-1} \alpha(s_i \rightarrow s_{i+1})\right) \left(\prod_{i=1}^{n} \beta(o_i \mid s_i)\right)$$

where $\pi(s)$ is the *initial probability* of starting in state s, $\alpha(s_i \rightarrow s_{i+1})$ is the *transition probability* from s_i to s_{i+1}, and $\beta(o_i \mid s_i)$ is the *emission probability* for either a single letter or aligner residue pair o_i in the state s_i.

In the derivation which follows, let *a** be the (unknown) alignment from *A* that most nearly represents the biological alignment of *x* and *y*. Ideally, we wish to determine *a** based on the sequence information in *x* and *y* alone. To do this the distribution $\mathbf{P}(A \mid x, y)$ is used to represent our beliefs regarding *a**, i.e., we assume that $\mathbf{P}(a \mid x, y)$ is the possibility that an alignment *a* is equal to *a**.

Let the notation $x_i - y_j \in a$ denote the event that two positions x_i and y_j are matched in an alignment *a*. Formally, the posterior probability of $x_i - y_j \in a^*$ is

$$\mathbf{P}(x_i - y_j \in a^* | x,y) = \sum_{a \in A} \mathbf{P}(a|x,y)\mathbf{1}\{x_i - y_j \in a\}$$

where the common indicator notation **1**{*condition*} is used to describe a function that sets to 1 whenever *condition* is true and 0 otherwise. Then, the posterior probability matrix *Pxy* for the alignment of *x* and *y* is a table of $\mathbf{P}(x_i - y_j \in a^* \mid x, y)$ values for $1 \le i \le |x|$, $1 \le j \le |y|$. The pair HMM algorithm begins by calculating these posterior probability. This computation step takes time $O(m^2L^2)$, where *m* denote the number of sequences and *L* is denoted for the length of each sequence.

Step 2: Probabilistic consistency transformation
Here the *probabilistic consistency*, a method for obtaining more accurate substitution scores when a third homologous sequence *z* is available.

For a sequence *z*, let $z_{(k,k+1)}$ denote the interletter regions (or gaps) between amino acids *k* and *k* + 1 of *z* for $0 \le k \le |z|$ (here $z_{(0,1)}$ and $z_{(|z|,|z|+1)}$ denote the gaps at the beginning and ends of *z*). Generalizing the notation for posterior probabilities of matches, an alternative estimate for the quality of an $x_i - y_j$ match is given by marginalized probability,

$$\mathbf{P}(x_i - y_j \in a^* | x,y,z) = \sum_{z_k} (x_i - y_j - z_k \in a^* | x,y,z) + \sum_{z_{k+1}} (x_i - y_j - z_{(k,k+1)} \in a^* | x,y,z)$$

where *a** refers to three sequence alignment of *x*, *y*, and *z*. It has been referred to the concept of re-estimating pairwise alignment match quality scores based on three-sequence information as *probabilistic consistency*.
As stated, computing $\mathbf{P}(x_i - y_j \in a^* \mid x, y, z)$ values for each $x_i - y_j$ pair requires $O(L3)$ time ; to avoid this, the computation is simplified as follows. First, it is heuristically ignored the second summation over gaps in *z* to get

$$\sum_{z_k} (x_i - y_j - z_k \in a^* | x,y,z)$$

Second, the inner condition is changed to an equivalent expression,

$$\sum_{z_k} \mathbf{P}((x_i - z_k \in a^*) \wedge (z_k - y_j \in a^*) | x,y,z)$$

Finally, heuristic independence is made for assumptions to get

$$\mathbf{P}(x_i - z_k \in a^* | x,z)\, \mathbf{P}(z_k - y_j \in a^* | z,y)$$

With the procedure described above, it can be aligned two sequences given information from a third sequence. To align two sequences *x* and *y* given a set of sequences, *S*, it would be ideally liked to estimate $\mathbf{P}(x_i - y_j \in a^* \mid S)$. In practice, the following heuristic decomposition is used.

$$\frac{1}{[s]} \sum_{z \in S} \sum_{z_k} \mathbf{P}(x_i - z_k \in a^* | x,z) \mathbf{P}(z_k - y_j \in a^* | z,y)$$

where $\mathbf{P}(x_i - x_j \mid x)$ is set to 1 if $i = j$ and 0 otherwise.
In the derivations above, it is clear that several unjustified assumptions were needed to obtain an proficiently computable formation for probabilistic consistency. In the first step, the simplification of not considering positions that are gap in a sequence *z* is difficult. In the fourth step, the independence assumptions required for the transformation do not hold for sets of linked sequences. Furthermore, the decomposition of $\mathbf{P}(x_i - y_j \in a^* \mid S)$ into an average over the different intermediate sequences in *S* is also not well grounded. Nevertheless, these methods work well in practice.

Ignoring gapped positions in the first simplification hurts only when x_i is aligned to y_j through a gap in *z*; for reliably alignable regions in which all sequences are present, this has little effect. Averaging $\mathbf{P}(x_i - y_j \in a^* \mid z)$ values in the final step can be interpreted as a linear regression-like method for predicting $\mathbf{P}(x_i - y_j \in a^* \mid S)$ where all inputs are given identical weight.

Step 3: Computation of guide tree
Most of the progressive multiple sequence alignment pro-

grams use evolutionary lengths estimated from pairwise alignments or *k*-mer statistics to build an approximate evolutionary tree via neighbor joining or UPGMA.
Given a set *S* of sequences to be aligned, denote the desired correctness for aligning any of two sequences *x* and *y* as *E*(*x*, *y*). Initially, each sequence is placed in the owner of cluster. The two clusters *x* and *y* with the maximum desired accurateness are combined to form a new cluster *xy*; then the expected accuracy of aligning *xy* is defined with any other cluster *z* as *E*(*x*, *y*)(*E*(*x*, *z*) + *E*(*y*, *z*))/2. This process is iterated until only one cluster remains.

Like UPGMA, the guide-tree computation procedure used here relies on customized arithmetic averaging term to estimate the "distance" of newly created clusters to other clusters. However, the significant distinction is that the calculation here has the goal of finding clusters that can be reliably aligned.

In multiple sequence alignment, one possible simplification is to estimate the expected approximation of accurate pairwise matches in each one column of alignment. Here a set *C* of the aligned residues in a particular column, this expected proportion of approximate pairwise matches ψ(*C*) is given by

$$\Psi(C) = \left(\binom{|c|}{}\right)^{-1} \sum_{x_i, y_j \in c} \mathbf{P}(xi - yj \in a^* | S)$$

which is expressed in Step 1.

Step 4: Progressive alignment
The final progressive alignment step in pair HMM is a routine. Since the alignments within each group are fixed, it may be ignored matches between sequences in each group. Thus, for each progressive alignment step, a profile–profile Needleman–Wunsch alignment procedure is run in which the score for matching a column consisting of *n*1 non-gap letters to one with *n*2 non-gap letters is accounted by summing *n*1*n*2 values from the corresponding pairwise posterior matrices. No gap penalties are used in this final step, thus greatly simplifying the task of profile–profile alignment.

Step 5: Iterative refinement
In this approach, the sequences of the existing multiple alignment are randomly splited into two groups of possibly not the same size by randomly assigning each sequence to one of the two groups to be realigned. Then, the similar dynamic programming procedure used for progressive alignment is employed to realign the two estimated alignments. This fine-tuning procedure can be iterated either for a fixed number of iterations or convergence; for the simplicity, only the former of these options is implemented in pair HMM. Because gap penalties are not used during each realignment step.

There are different approaches to perform multiple protein sequence alignment. The most efficient approach is progressive and iterative. To perform progressive approach most cases used dynamic programming. But in this paper probabilistic algorithms is used. The reasons to select probabilistic approach to proposed method are given bellow:

First, there are different categories of optimal algorithms. But progressive and iterative are best of them.

Second, the most accurate progressive alignment in MSA performs both global and local alignment. Only dynamic programming and HMMs (Probabilistic approaches) perform both global and local alignment.

Third, the complexity of dynamic programming is much more than HMMs (probabilities approaches).
For example, the complexity of two sequences in dynamic programming for length L is O(L2) and space complexity O(L). But for log odd scoring of HMMs is

$$s_{ij} = \log_2 (q_{ij} / e_{ij})$$

Where s is the alignment score, q_{ij} is the real frequency, and e_{ij} is the expected random frequency.

Forth, the complexities of dynamic programming are being increased when numbers of sequences are being increased.

Fifth, in dynamic programming same complexity is applied for gap.

Sixth, Dynamic programming is not feasible for every case but heuristic alignment does.

4. Experimental result and comparison

To implement this method for multiple protein sequence alignment, Microsoft visual C++, 2012 has been used. several protein sequences are used, which are found in European Molecular Biology Laboratory (EMBL) database. The result of MUSCLE has been taken using the same protein sequences.

Table 2: Q score

Nummber of sequences	*Average length*	*Mafft Algorithm*	*Muscle Algorithm*	*Proposed Method*
		Q score	*Q score*	*Q score*
4	282	.99	.996	.996
4	363	.989	.991	.994
3	414	.984	.99	.992
7	2036	.917	.978	.997

In the table 1 the total Q score (Q score means quality score which is the number of correctly aligned residue pairs divided by the number of residue pairs in the reference alignment) has been shown for the MUSCLE, Mafft and proposed method.

Table 2: TC score

Nummber of sequences	Average length	Mafft Algorithm	Muscle Algorithm	Proposed Method
		TC score	TC score	TC score
4	282	.976	.992	.993
4	363	.978	.989	.990
3	414	.975	.985	.993
7	2036	.783	.926	.992

In table 2 the TC score (Total Column Score) is the number of correctly aligned columns divided by the number of columns in the reference alignment.

From table 1 and table 2 it can be seen that this method gives the better result than others. It can also be seen that the proposed method also improves the result with increment of the number of sequences.

5. Conclusion

Though the problem of multiple protein sequence alignment is hardly new, the computation of high accuracy multiple sequence alignments is still now an open problem. In this method it has been discussed the way of solving the problem very efficiently. The proposed method is a combine method of progressive, iterative and probabilistic approaches. Progressive approach is used for pair wise alignment for the input sequences using distance matrix and guide tree. The iterative approach is used for refinement of the aligned sequences to get the better results. This technique enables high throughput applications to get average accurateness both in score and match column comparable to the most accurate tools previously available. The combination of probabilistic and dynamic algorithms show better result than before. In future, it will be tried to experiment for DNA sequence also.

6 References

[1] M.F. Omar, R.A. Salam, R. Abdullah and N.A. Rashid, "Multiple Sequence Alignment Using Optimization Algorithms", *International Journal of Computational Intelligence*, vol.1, no. 2, pp.81-89, 2004.

[2] B.D. Chuong and K. Katoh, "Protein Multiple Sequence Alignment", *Metods in Molecular Biology*, vol. 484, pp. 379-413.

[3] H.D. Nguyen, I. Yoshihara, K. Yamamori and M. Yasunaga, "Aligning Multiple Protein Sequences by Parallel Hybrid Genetic Algorithm", *Genome Informatics,* vol.13, pp. 123-132, 2002.

[4] T.M. Phuong, B.D. Chuong, C. Robert, Edgar and S. Batzoglou, "Multiple alignment of proteinsequences with repeats and rearrangements", *Nucleic Acids Research*, vol. 34, no. 20, pp. 1-11, 2006.

[5] S.R. Eddy, "Multiple alignment using hidden Markov models", ISMB, vol. 95, pp.114-120, 1995.

[6] C.S Tsai, "An introduction to computational Biochemistry", *John Wiley & Sons, Inc*., 605 Third Avenue, New York, NY, pp. 10158-0012 (212) 850-6011,2002.

[7] R. Liu, "Strategies for improving multiple alignment of Retrotransposon sequences", *The University of Georgia*, pp.1-49, August. 2001.

[8] M. Shatsky, R. Nussinov and H. Wolfson, "A Method for Simultaneous Alignment of Multiple Protein Structures, *Proteins: Structure, Function, and Genetics*, vol. 51, no. 1, pp. 143-156, 2004.

[9] A. Andreeva and A.G. Murzin, "Structural classification of proteins and structural genomics: new insights into protein folding and evolution, *Acta Crystallogr Sect F Struct Biol Cryst Commun*, vol. 66, no. 10, pp. 1190-1197, Oct. 2010.

[10] G.J. Barton and M.J.E. Sternberg, "Protein Sequence Alignment and Database Scanning", *IRL Press at Oxford University Press*, pp. 1-35, 1996.

[11] K. Katoh, K. Kuma, H. Toh and T. Miyata, "MAFFT: Improvement in accuracy of multiple sequence alignment". *Nucleic Acids Research*, A.G. Murzin, "Structural classification of proteins and structural genomics: new insights into protein folding and evolution**,** vol. 33, no. 3, pp. 511-518, 2005.

[12] K. Katoh, K. Kuma, H. Toh and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on Fast Fourier Transform", *Nucleic Acid Research*, A.G. Murzin, "Structural classification of proteins and structural genomics: new insights into protein folding and evolution**,** vol. 30, no. 14, pp. 3059-3066, 2002.