Multiple Proteins Sequence Alignment Based on Progressive Methods with New Guide Tree

Gamil Abdel-Azim¹ Member IEEE, Mohamed Ben Othman¹, Senior Member IEEE.

and Abo-Eleneen, Z. A²

College of Computer, Qassim University, Saudi Arabia.¹

gazim3@gmail.com, abdaladiem@qu.edu.sa mtothman@gmail.com bn_athaman@qu.edu.sa

College of Computer & Informatics, Zagazig University, Egypt. ²

Abstract

Multiple proteins sequence alignment is one of the important research topics of bioinformatics. In multiple sequence alignment, it is emphasized to find optimal alignment for a group of sequences. All sequences are constituted of residues i.e. nucleotides for DNA/RNA, or amino acids for proteins. The objective is to maximize the similarities between them by adding and shuffling gaps. To do this, we propose a guide tree based on new distance definition. This distance is based on a Sequence Feature Vector (SFV). The SFV is built using a similarities descriptor of the sequence. We are studying the progressive alignment methodology with the proposed guide tree which is constructed using the similarity of all possible pairs of sequences. The proposed guide tree is simple to implement and give good result's performance. The comparison between the proposed guide tree and the tree built using pairwise distance is analyzed and then obtained solution qualities are reported. The Results of our testing in all dataset show that the proposed guide tree is as good as Clustalw in most cases.

Keywords: Proteins Sequences alignment; Progressive methods; Computational molecular biology.

1. Introduction

Multiple sequence alignment (MSA) of DNA, RNA and proteins sequences is one of the most common and important tasks in Bioinformatics. It is one of the most important and challenging task in computational biology because the time complexity for solving MSA grows exponentially with the size of the considered problem see [1] for an overview on existing multiple alignment approaches. Finding the optimal alignment of given sequences is known as a nondeterministic polynomial-time (NP)-complete problem [2]. The solution of MSA using dynamic programming requires \( O((2m)n) \) time complexity (\( n \) is the number of sequences, and \( m \) is the average sequence length) and \( O(m n) \) memory complexity [3-5]. Therefore, carrying out
MSA by dynamic programming (DP) becomes practically intractable as the number of sequences increases. Multiple alignment methods can be divided into two main categories: methods aligning sequences over their entire length (global) and methods aligning regions of only high similarity (local). In this paper we focus in global alignment. The fact that the MSA problem is of high complexity has led to the development of different algorithms. In addition, the MSA of proteins sequences offers important tools in studying proteins. This is very useful 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.

The search for the best possible alignment for a set of sequences is not trivial. Finding a global optimal alignment of more than two sequences that include matches, mismatches, and gaps and that take into account the degree of variation in all sequences at the same time is especially difficult. The DP algorithm is used to obtain optimal alignment of a pair of sequences and can be extended to global alignment of three sequences, but for more than three sequences, only a small number of relatively short sequences may be treated. Thus, approximate methods are used for global sequence alignment. One class of these methods is progressive global alignment, which starts with an alignment of the most alike sequences and then builds an alignment by adding more sequences. Some early works on multiple sequence alignment can be found on [13-22].

The progressive alignment method utilized by the widely-used series of Clustalw. This program implements a progressive method for multiple sequence alignment. As a progressive algorithm, Clustalw adds sequences one by one to the existing alignment for build a new alignment. Although the progressive alignment methods are faster than (DP) algorithms, they also suffer from entrapment in local optima because they only optimize the alignment in a pairwise manner not taking the entire alignment into account. The order of the sequences to be added to the new alignment is indicated by a pre computed phylogenetic tree, which is called a guide tree. The guide tree is constructed using the similarity of all possible pairs of sequences. This similarity is built upon a certain distance between the sequences.

In this paper, we are studying the progressive alignment methodology with a new guide tree that is constructed using the similarity of all possible pairs of sequences.

This similarity is defined by a new distance between the sequences. This distance is based on a Sequence Feature Vector (SFV) which is built using a descriptor of the sequence.

Our proposed algorithm consists of 3 phases similar to Clustalw. The only different part from Clustalw is how to build distance matrix. The 3 phases are: a) building the Distance Matrix b) calculating the guide tree from the distance matrix using a neighbor joining algorithm [6], and c) processing the progressive alignment. The guide tree defines the order in which the sequences are aligned in the next stage. There are several methods for building trees, including distance matrix methods and parsimony methods. In this paper, we are using 'neighbor-joining' method as distance matrix approach. The sequences are progressively aligned following the guide tree.
The rest of the paper is organized as follows: In the next section, the description of multiple protein sequence alignment is presented. Section 3 will briefly review the existent optimization algorithms and section 4 shows a proposed distance base on a similarities descriptor of the sequences. Our algorithm called GEneral Methodology of Progressive Alignments (GEMPA) is decrypted in Section 5 with illustration by examples. The data set and results are discussed in section 6. Finally, concluding remarks and further research to be developed are presented.

2. Proteins Sequences Alignments

Let $\mathbf{S} = \{S_1, S_2, \ldots, S_n\}$ be the input sequences and assume that $n$ is at least 2. Let $\sum$ be the input alphabet that form the sequences; we assume that $\sum$ does not contain the character ‘–’, which can be used to denote a gap in the alignment. A set $\mathbf{S}' = \{S'_1, S'_2, \ldots, S'_n\}$ of sequences over the alphabet $\sum' = \sum \cup \{-\}$, is called an alignment of $\mathbf{S}$ if the following two properties satisfied:

1. The strings in $\mathbf{S}'$ have the same length.
2. Ignoring gaps, sequence $S'_i$ is identical with sequences $S_i$.

An alignment can be interpreted as an array with $n$ rows and $m$ columns, one row for each $S_i$. Two letters of distinct strings are called aligned under $\mathbf{S}$ if they are placed into the same column. See Figure (1) with three proteins sequences.

![Figure 1: Example of multiple alignments of three proteins sequences](image)

3. Exisitent Optimization Algorithms

There exist three categories of the optimization algorithms for multiple alignment [7]; exact, progressive and iterative. Numerous MSA programs have been applied using many techniques and algorithms. Most commonly used techniques are progressive and iterative techniques. The exact method [1,8] suffers from inexact sequence alignment. Most progressive alignment methods heavily rely on dynamic programming to perform multiple alignments starting with the most related sequences and then progressively adding fewer related sequences to the initial alignment. The existence of several progressive programs has broadened up the aligning techniques. This approach has the advantages of speed and simplicity [7]. They have the advantage of being fast and simple as well as reasonably sensitive. The main drawback is the ‘local minimum’ problem that stems from the greedy nature of the algorithm. Also the major problem with progressive alignment method is the errors in the initial alignments are the most closely related
sequences propagated to the multiple alignments [7]. Algorithms that construct multiple sequence alignment require a cost function as a criterion for constructing an optimal alignment. We are using Gonnet Matrix as a cost function [10].

In this paper, we interested on the progressive technique improvement by proposing a new guide tree based on new distance definition.

4. Proposed Distance

In this section, we define a descriptor for each sequence which is used to build the SVF. Over the SVF the distance between the sequences is then defined. The descriptor is defined as follows:

$$f : \text{Pr} \times S \to R^n, \ f(s) = D_{sq}.$$ 

Where PrS is the set of proteins sequences. The proteins Alphabets = { A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V } are twenty letters. Firstly, we describe the Feature Vector for each Sequence in the proteins and used the Euclidean distance to find the distance between the two features vectors of the two sequences $S_n, S_m$. Each SVF of a protein sequence has a length of sixty shown below:

$$D_{sq} = \{ N_i ; T_j ; D_i ; \text{with } i = A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V \}$$

Where: $N_A, N_R, N_N, \ldots$, and $N_V$ is the number of As, Rs, Ns,……….. and Vs in the sequence respectively. $T_A, T_R, T_N, \ldots$, and $T_V$ are defined by:

$$T_i = \sum_{j=1}^{N_i} t_{ij}$$

For an amino acids $i$ (i = A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V); $t_{ij}$ is the distance from the first amino acids to the same $j^{th}$ amino acids. The parameter’s $D_A, D_R, D_N, \ldots$, and $D_V$ are defined as follows:

$$D_i = \sum_{j=1}^{N_i} \frac{(t_{ij} - \mu_i)^2}{N_i} \text{ where } \mu_i = \frac{T_i}{N_i}$$

Where $D_i$ is the scatter of amino acids $i$ within the protein sequence. Between two proteins sequences the fact that the first two parameters are close does not mean that they have a high similarity, unless they have a close distribution. For this reason this new feature parameter $D_i$ is added. However, a combined feature vector that contains these three sets of feature parameters could be used to characterize the similarity between proteins sequences. The SVF can be used as a numerical measure of similarity in different proteins sequences.

In order to measure the similarity and difference between proteins sequences, for each sequence we find SVF that represents it. Firstly to reduces the dimension of the sequences. Secondly to use it to measure the distance between the sequence and the others sequences.
This is summarized in two steps:

- Define the descriptor for each sequence (Sequence Feature Vector).
- Build the guide tree based on the distance defined by the descriptor.

Two proteins sequences are considered similar when the distance between their two feature vectors is small. The distance of two feature vectors is defined as follows:

\[ L = \sqrt{\sum_{j=1}^{N} \sum_{i=1}^{R} (j_i - \hat{j}_i)^2} \quad i = A, R, N, ... \text{ and } j = N, T, D. \]

Note that this distance can be used for DNA sequences by changing only the set of alphabet.

5. General Methodology of Progressive Alignments

We briefly describe the General Methodology of Progressive Alignments (GEMPA) as following:

1. Read the set of proteins sequences
2. Construct the distance between all sequences. (Distance Matrix)
3. Build the phylogenetic tree using distance matrix methods
4. Apply the progressive alignment methods with phylogenetic tree.
5. Output the resulting sequence alignment.

Now we will illustrate the GEMPA using two examples, 4 and 9 proteins sequences with minimum length of 390, 385 and maximum length of 456, 457 respectively. First, we calculate the distance matrix, second we build the phylogenetic tree. The guide trees are built using the proposed distance (section 4) for the first and the pairwise distance for the second. We implemented the two guide trees using Matlab functions as following:

TreePW = seqlinkage (DistancePW,'single',seqs), where seqlinkage is a matlab function, that implements Neighbor-joining algorithm. And,

DistancePW = seqpdist (seqs, 'ScoringMatrix',pam250), where seqs are the proteins sequences.

TreePro = seqlinkage (PDM,'single',seqs), where PDM is the proposed distance matrix. See Figures (2,3).

Example 1:
TreePW: with Pairwise distance. Scoring Value is 144.7000

TreePro with proposed distance. Scoring Value is 464.0000

Figure 2: TreePW and TreePro

The Scoring Value of the solution alignments using Gonnet matrix is 144.7000 for Distance PW, and is 464.0000 for the PDM.

Example 2:

TreePW: with Pairwise distance.

TreePro with proposed distance

The Scoring Value of the solution alignments using Gonnet matrix is \(2.2277 \times 10^3\) for Pairwise distance matrix, and is \(2.4690 \times 10^3\) for the proposed distance matrix.

6. Results and discussions

In this section, we discuss the results using protein database BALiBASE 3.0. The information concerning the data set taken from the database is summarized as following:

Reference 1: Equi-distant sequences with 2 different levels of conservation.

Reference 2: Families aligned with a highly divergent "orphan" sequence.

RV11: Reference 1, very divergent sequences (20 identities).
RV12: Reference 1, medium divergent sequences (20-40 identity).

RV20: Reference 2. See[9-12].

Also we are comparing the results between the two distances used in progressive algorithm;

The progressive algorithm appears to have the best performance in various research papers. It was implemented by multialign in Matlab function with option as follows:

multialign (S, 'terminalGapAdjust', true). See (Matlab version 7). To compare the solutions alignments given by the two options of the progressive algorithm, which are implemented as following:

SolPW = multialign (seqs, TreePW,'ScoringMatrix', {'pam150','pam200','pam250'});

SolPro = multialign (seqs, TreePro,'ScoringMatrix', {'pam150','pam200','pam250'}); where TreePW and TreePro are Phylogenetics guide trees that are built using pairwise distance and proposed distance matrix respectively. SolPW and SolPro are alignments solutions obtained using Phylogentic TreePW, and Phylogentic TreePro respectively. Note that the Gonnet scoring matrix is used to measure the two alignments solutions SolPW and SolPro. Figs 3-5 give the comparison between TreePW and TreePro (Solution Alignment Scoring Value) of different examples over the datasets.

Figure 3 : SolPW and SolPro performance Examples 1-38 (RV11)
7. Conclusions.

We propose a guide tree based on a new distance definition. This distance is based on a Sequence Feature Vector (SFV). The SFV is built using a similarity descriptor of the sequence. It is simple to implement, and gives good results performance. The comparison between the proposed distance and pairwise distance is analyzed and the obtained solution qualities are reported. The Results of our testing on all the dataset show that the proposed distance obtains good quality solutions. The solutions obtained using TreePro are as good as those obtained by TreePW.

8. References


