Modified Progressive Strategy for Multiple Proteins Sequence Alignment

GAMIL ABDEL-AZIM[1,2] MEMBER IEEE , MOHAMED BEN OTHMAN[1], SENIOR MEMBER IEEE. AND zaher ABO-ELENEEN[3]

Abstract— One of the important research topics of bioinformatics is the Multiple proteins sequence alignment. Since the exact methods for MSA have exponential time complexity, the heuristic approaches and the progressive alignment are the most commonly used in multiple sequences alignments. In this paper, we propose a modified progressive alignment strategy. Choosing and merging the most closely sequences is one of the important steps of the progressive alignment strategy. This depends on the similarity between the sequences. To measure that similarity we need to define a distance. In this paper, we construct a distance matrix. The elements of a row of this matrix correspond to the distance between a sequence to other sequences. A guide tree is built using the distance matrix. For each sequence we define a descriptor which is called also feature vector. The elements of the distance matrix are calculated based on the distance between the descriptors of the sequences. The descriptor reduces the dimension of the sequence then yields to a faster calculation of distance matrix and also to obtain preliminary distance matrix without pairwise alignment in the first step. The principle contribution in this paper is the modification of the first step of the basic progressive alignment strategy ie the computation of the distance matrix which yields to a new guide tree. Such guide tree is simple to implement and gives good result's performance. A comparison between the results got from the proposed strategy and from the ClastalW over the database BAliBASE 3.0 is analyzed and reported. The Results of our testing in all dataset show that the proposed strategy is as good as Clustalw in most cases.

***Keywords*——** Proteins sequences; Alignment; Progressive methods; Sequence descriptor; Computational molecular biology.

I. 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.

One of the most widely used heuristic search for multiple sequence alignments is known as progressive technique (also known as tree method). It combines pairwise alignments beginning with the most similar pair and progressing to the most distantly related, which finally builds up a MSA solution. The basic progressive alignment strategy is summarized in the following (see fig 1):

1. Compute D, a matrix of distances between all pairs of sequences
2. From D, construct a "guide tree" T
3. Construct MSA by pairwise alignment of partial alignments ("profiles") guided by T.

G. A. Azim: gazim3@gmail.com, abdaladiem@qu.edu.sa
M. B. Othman : mtothman@gmail.com, bn_athaman@qu.edu.sa
Z. A. Abo-Eleneen: zaher_aboeleneen@yahoo.com
College of Computer, Qassim University, Saudi Arabia.[1]
College of Computer & Informatics, Suez Canal University, Ismailia , Egypt. [2]
College of Computer & Informatics, Zagazig University, Egypt. [3]

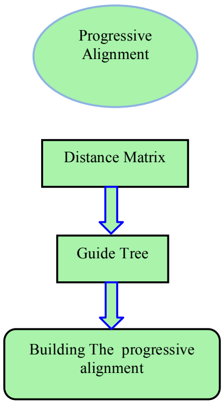

Fig. 1: Progressive Alignment Strategy

Progressive alignments solution cannot be globally optimal. Firstly, the main problem is that any error made at any stage in building the MSA, this error is propagated through to the final result.

Secondly, the performance is also particularly bad when all of the sequences in the set are rather distantly related. Progressive alignment methods are efficient enough to implement on a large scale for many (100s to 1000s) sequences. The most popular progressive alignment method has been implemented in the Clustal family [13], especially the weighted variant ClustalW [14]. Some early works on multiple sequence alignment can be found on [15-27].

The guide tree in the basic progressive strategy is determined by an efficient clustering method such as neighbor-joining, or un-weighted average distance (UPGMA).

In this paper we proposed a measurement of the similarity between the sequences, which play an important role in the building of the guide tree, then in the performance of the quality of the MSA solution. The measurement of the similarity between the sequences is based on their descriptors which will be described in section 4.

The similarity is defined by a new distance between the sequences. For each sequence we define a descriptor which is called also Sequence Feature Vector (SFV). The elements of the distance matrix are calculated based on the distance between the descriptors of the sequences. The descriptor reduces the dimension of the sequence then yields to a faster calculation of distance matrix and also to obtain preliminary distance matrix without pairwise alignment in the first step.

Our proposed algorithm consists of 3 phases similar to Clustalw. The only different part from Clustalw is how to build distance matrix (see fig 2). 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' and un-weighted average methods 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.

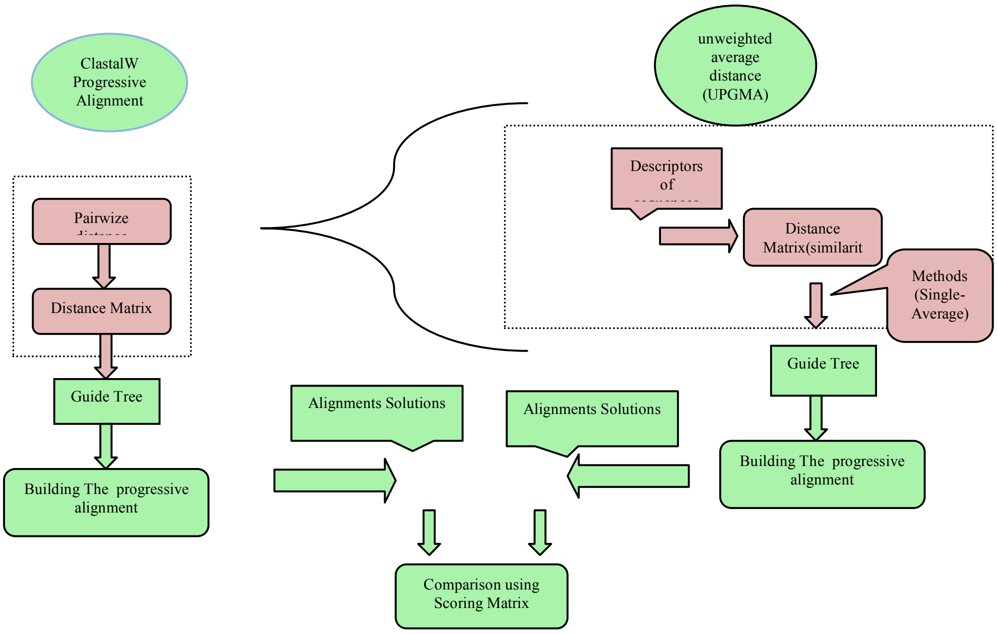
ClastalW Progressive Alignment
Pairwize
Distance Matrix
Guide Tree
Building The progressive alignment
unweighted average distance (UPGMA)
Descriptors of
Distance Matrix(similarit
Methods (Single-Average)
Guide Tree
Building The progressive alignment
Alignments Solutions
Alignments Solutions
Comparison using Scoring Matrix

Fig. 2: Proposed Progressive Strategy

II. PROTEINS SEQUENCES ALIGNMENTS

Let S = {S1, S2, . . ., Sn} be the input sequences and assume that n is at least 2. Let ∑ be the input alphabet that form the sequences; we assume that ∑ does not contain the character '–', which can be used to denote a gap in the alignment. A set S'= {S'1 , S'2 , . ., S'n } of sequences over the alphabet ∑' = ∑ U {–}, is called an alignment of S if the following two properties satisfied :

1. The strings in 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 Si. Two letters of distinct strings are called aligned under S if they are placed into the same column. See Figure (1) with three proteins sequences.

$$AP = \begin{bmatrix} \text{A R N - D C Q E G H I L} & \text{M F - W T W Y V} \\ \text{- R - N D C Q E G H I L} & \text{M F S - T W Y V} \\ \text{A R N - D C Q E G H I L} & \text{M F S - T W Y V} \end{bmatrix}$$

Fig. 3: Example of multiple alignments of three proteins sequences

III. EXISTENT 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.

IV. DISTANCE USING SIMILARITY DESCRIPTOR

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: PrS → R^n, f(s) = Dsq.

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:

Dsq = {N_i ; T_i ; D_i; 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, …, and N_V is the number of As, Rs, Ns,……….. and Vs in the sequence respectively. T_A, T_R, T_N , , and T_V are defined by:

$$T_i = \sum_{j=1}^{N_i} t_j \qquad (1)$$

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_j is the distance from the first amino acids to the same j^th amino acids. The parameter's D_A, D_R, D_N, …, and D_V are defined as follows:

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

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 summarizes 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:

$$L = \sqrt{\sum_j \sum_i (j_i - \hat{j}_i)^2}$$

$$i = \text{A, R, N, ...and V} \quad and \ j = N, T, D. \ (3)$$

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

V. .General Methodology of progressive Alignments

We briefly describe the General Methodology of Progressive Alignments (GEMPA) as following (see fig. 2):
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 pairswise 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 (Figs (4,5)).

Example 1:

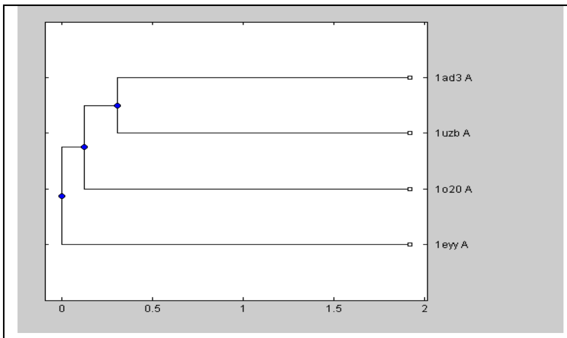	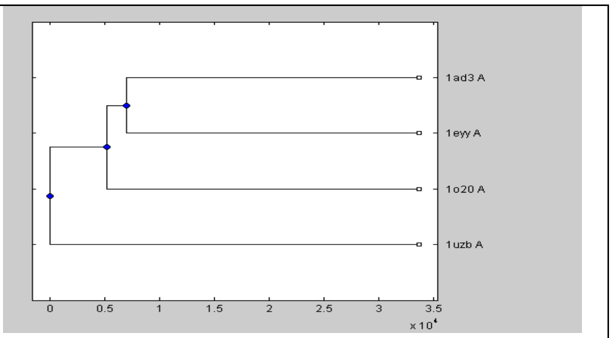
TreePW: with Pairwise distance. Scoring Value is 144.7000	TreePro with proposed distance. Scoring Value is 464.0000

Fig. 4: 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:

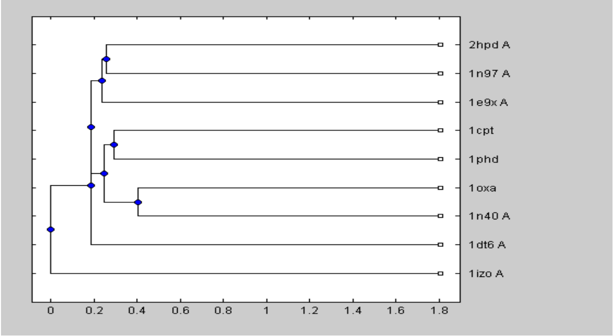	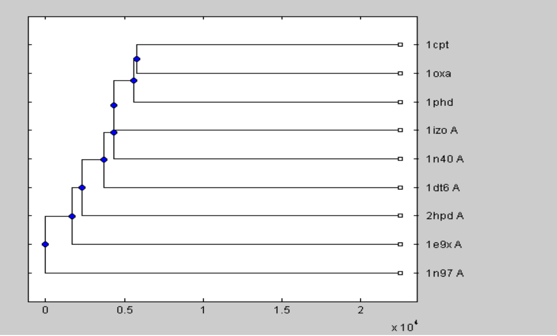
TreePW: with Pairwise distance.	**TreePro with proposed distance**

Fig. 5: TreePW and TreePro

The Scoring Value of the solution alignments using Gonnet matrix is = 2.2277e+003 for Pairwise distance matrix, and is 2.4690e+003 for the proposed distance matrix.

VI. . Results and Discussions

We used the protein database BAliBASE 3.0 for testing our strategy performance. 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 the following options:

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

To compare the solutions alignments given by our progressive strategy, which is implemented as following:

Pro=multialign(seqs,TreePro,'ScoringMatrix',{'pam150','pam200','pam250'});

where TreePW and TreePro are Phylogentics guide trees that are built using pairwise distance and proposed distance matrix respectively. PW and Pro 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 PW and Pro. Figs 6-8 give the comparison between PW and Pro (Solution Alignment Scoring Value) of different examples over the datasets RV11, RV12, and RV20 using single method and gonnet's substitution matrix.

Table I summarizes the set of figures attached in the appendix for the results of the ClastalW and our strategy using two different methods single and average to build the guide tree and different

substitution matrices Gonnet, Pam150, Pam200, and Pam250 over the data set RV11.

Table I: summary of figures for different methods and substitution matrices

Substitution Matrix	Method	
	Single	Average
Gonnet	Fig. 9	Fig. 13
Pam150	Fig. 10	Fig. 14
Pam200	Fig. 11	Fig. 15
Pam250	Fig. 12	Fig. 16

The obtained results show that for the single method over data set RV11 our strategy is as good as ClastalW in 82% of the examples. Over the data set RV12 and RV20 our strategy is similar than ClastalW.However, using the average method the performance of our strategy is better than ClastalW in some examples and similar over the rest.

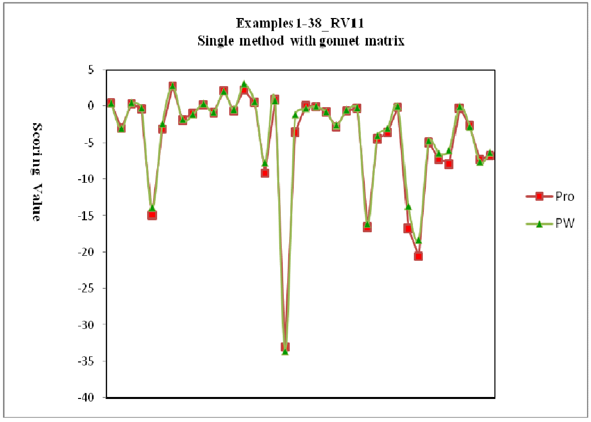

Fig. 6: Performance using Single method gonnet matrix (1-38 RV11)

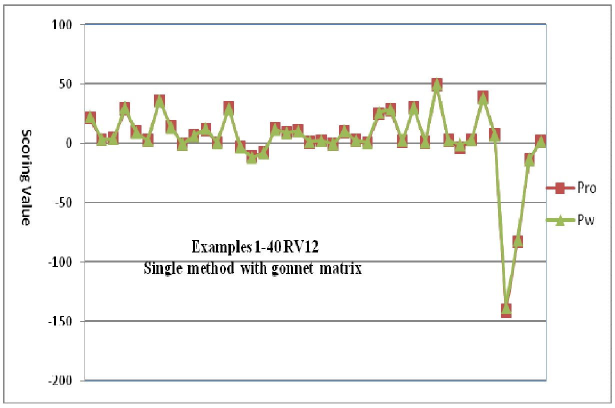

Fig. 7: Performance using Single method and gonnet matrix (1-40-RV12)

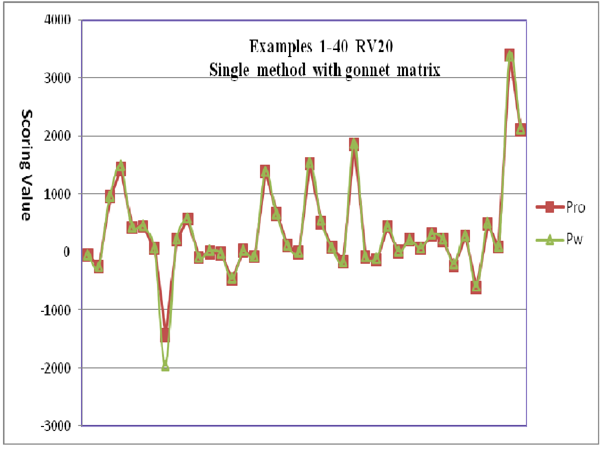

Fig. 8: Performance using Single method and gonnet matrix (1-40-RV20)

VII. . Conclusions.

We propose a modified progressive alignment strategy based on a modified distance matrix which is built using defined sequences' descriptors also called Sequence Feature Vector (SFV). Firstly to reduces the dimension of the sequences. Secondly to use it to measure the distance between the sequence and the others sequences. This can be summarized into two steps:

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

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 strategy and ClastalW is analyzed and the obtained solution qualities are reported. The results of our testing on all the dataset show that the proposed strategy obtains good quality solutions. The obtained solutions using the proposed strategy are as good as those obtained by ClastalW.

APPENDIX

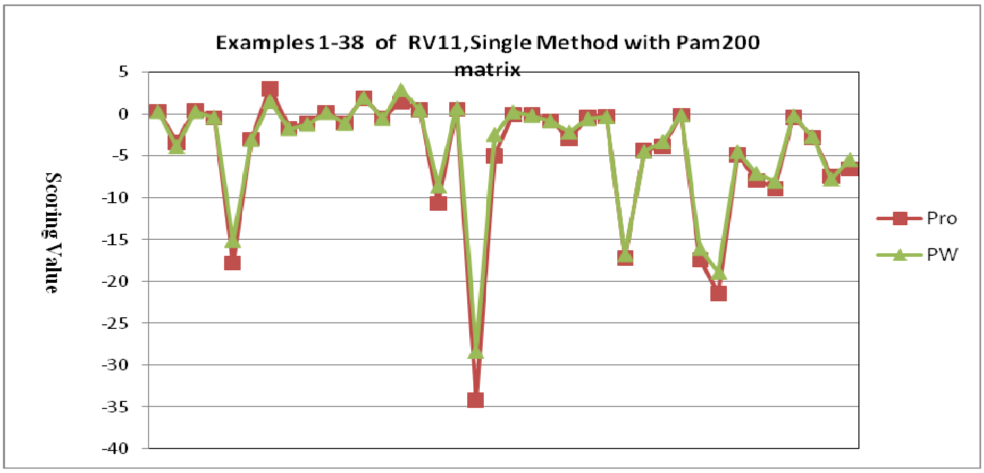

Fig. 9: Performance using Single method and gonnet matrix

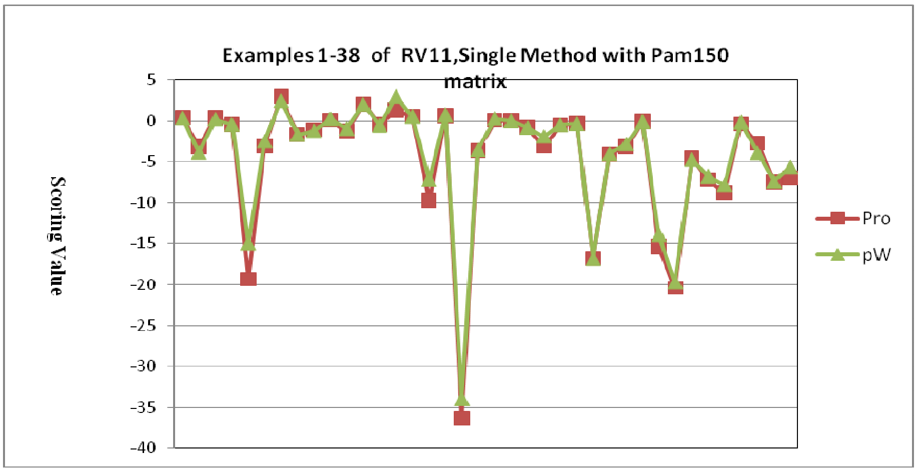

Fig. 10: Performance using Single method with Pam150

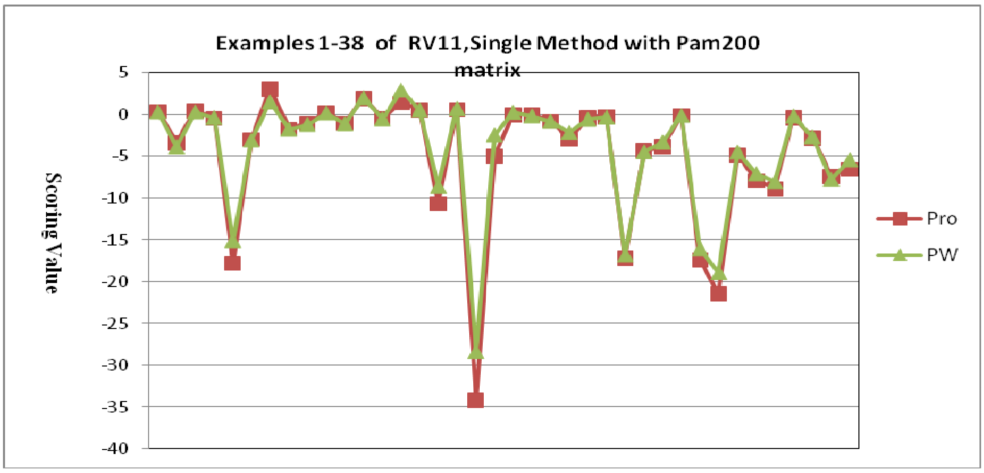

Fig. 11: Performance using Single method with Pam200

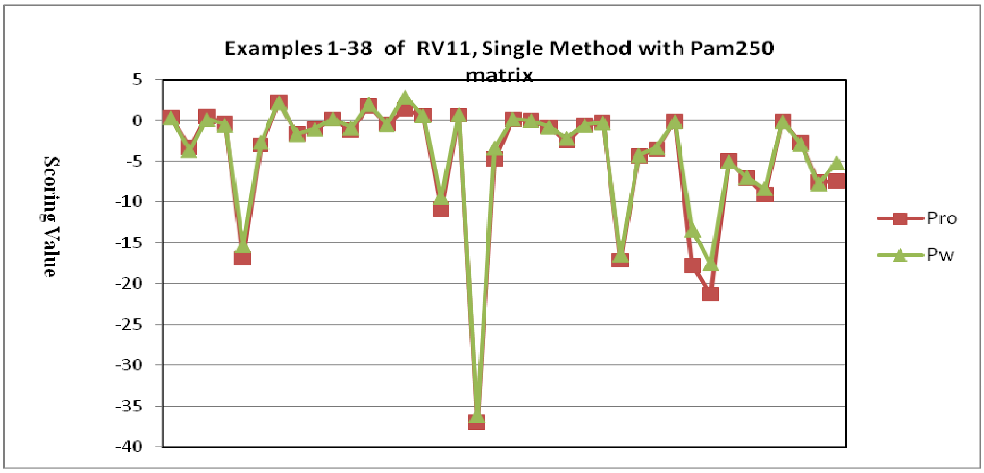

Fig. 12: Performance using Single method with Pam250

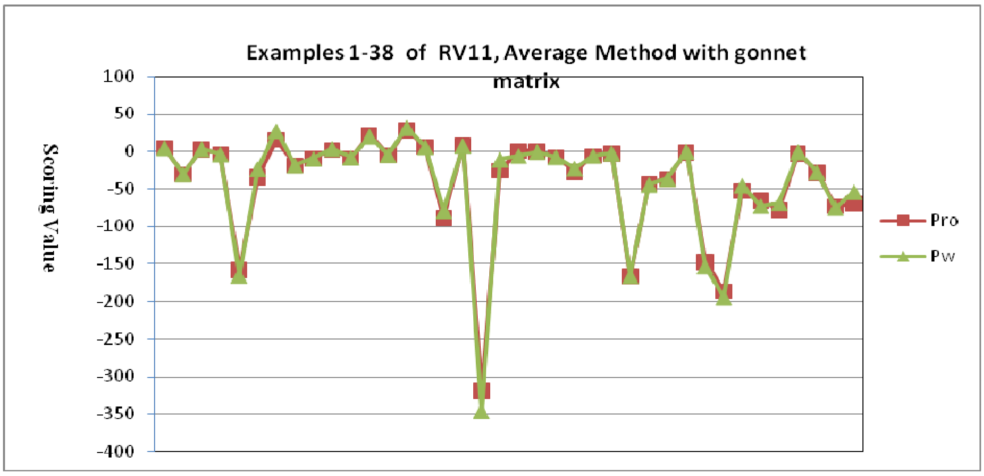

Fig. 13: Performance using Average method with gonnet matrix

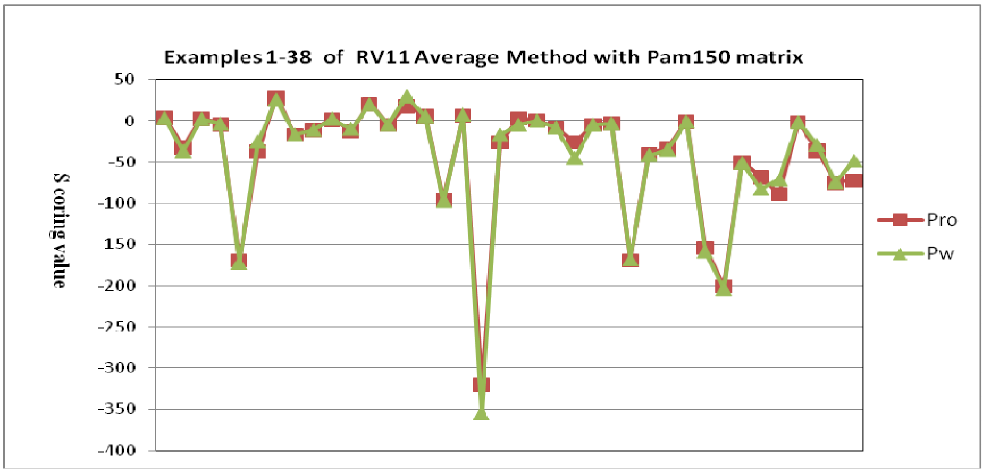

Figure 14: Performance using Average method with Pam150

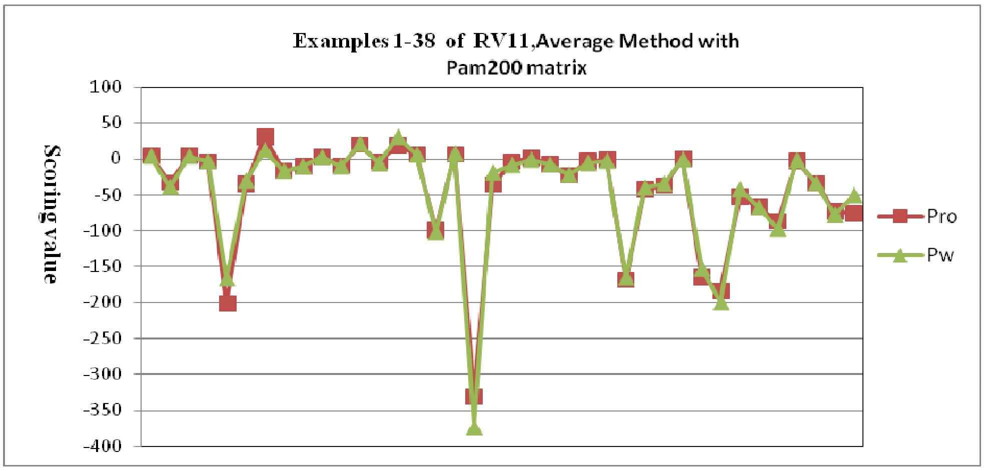

Fig. 15: Performance using Average method with Pam200

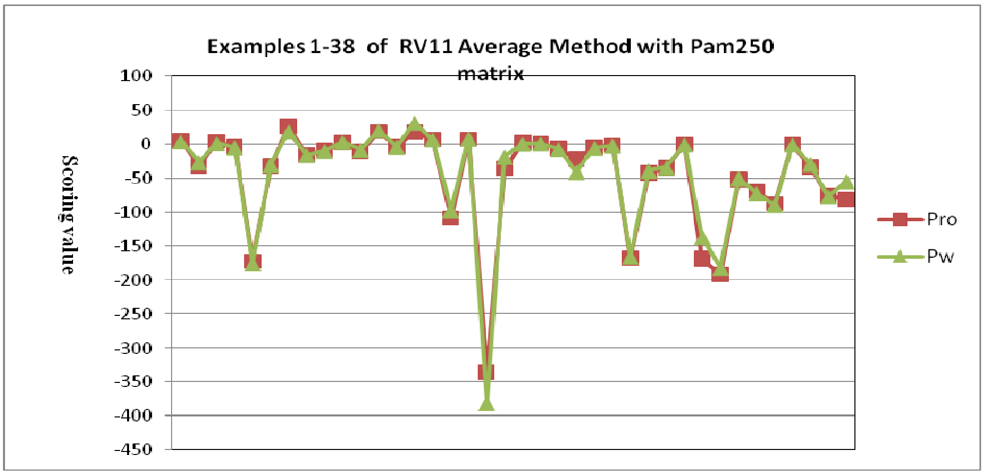

Fig. 16: Performance using Average method with Pam250

REFERENCES

[1] Notredame,C. (2002) Recent progress in multiple sequence alignment: a survey. Pharmacogenomics, 3, 131–144.

[2] Wang, L., and T. Jiang. 1994. On the complexity of multiple sequence alignment. J. Comput. Biol. 1:337–348.

[3] Carrillo, H., and D. Lipman. (1988). The multiple sequence alignment problem in biology. SIAM J. Appl. Math. 48:1073–1082.

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

[5] Smith, T. F., and M. S. Waterman. (1981). Identification of common molecular sub sequences. J. Mol. Biol. 147:195–197.

[6] Saitou N. and Nei, M., (1987). "The neighbor-joining method: a new method for reconstructing phylogenetic trees," Mol Biol Evol, vol. 4, no. 4, pp. 406–425,

[7] Choudry, R. (1999). Application of Evolutionary Algorithms for Multiple Sequence Alignment. Stanford University.

[8] Lipman, D. J., Altschul S. F., Kececioglun J. D., (1989). A tool for Multiple Sequence Alignment. Vol. 86.pp 4412-4415, Biochemistry. Proc. Natl. Acad. Sci. USA

[9] Thompson, J. D., Koehl, P., Ripp, R., and Poch, O. (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. Proteins, 61, 127–136.

[10] Thompson J.D., Plewniak F, Poch O. (1999). BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics, 15, 87-8.

[11] Thompson J.D., Plewniak F, Poch O. (1999). A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Res. 27, 2682-90.

[12] Bahr A., Thompson J.D. (2001). Thierry JC, Poch O. BAliBASE (Benchmark Alignment dataBASE): enhancements

for repeats, transmembrane sequences and circular permutations. Nucleic Acids Res. 29, 323-6.

[13] Higgins DG, Sharp PM (1988). "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer". Gene 73 (1): 237–244.

[14] Thompson JD, Higgins DG, Gibson TJ (1994). "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice". Nucleic Acids Res 22 (22): 4673–4680.

[15] Ben Othman, M., Azim, G. A. and Hamdi-Cherif, A. (2008). Pair wise Sequence Alignment Revisited – Genetic Algorithms and Cosine Functions, NAUN conference, Attica, Greece, June 1-3.

[16] Ben Othman, M., Hamdi-Cherif, A. and Azim, G. A. (2008). Genetic algorithm and scalar product for pairwise sequence alignment International Journal of Computer, NAUN North Atlantic University Union, 2, 134-147.

[17] Azim, G. A. and Ben Othman, M. (2010). Hybrid iterated local search algorithm for solving multiple sequences alignment problem", Far East Journal of Experimental and Theoretical Intelligence 5, 1-17.

[18] Wang, Y. and Li, K.-B. (2004). An adaptive and iterative algorithm for refning multiple sequence alignment, Comput. Biol. Chem. 28, 141–148.

[19] Cai, J. J., Smith, D. K., Xia X. and Yuen, K. Y. (2006). MBEToolbox 2: an enhanced MATLAB toolbox for molecular biology and evolution, Evolutionary Bioinformatics 2, 189-192.

[20] Gondro C. and Kinghorn, B. P. (2007). A simple genetic algorithm for multiple sequence alignment, Genet. Mol. Res. 6, 964-982.

[21] Kumar, S. and Filipski, A. (2007). Multiple sequence alignment: pursuit of homologous DNA positions, Genome Res. 17, 127-135.

[22] Loots G. G. and Ovcharenko, I. (2007). Mulan: multiple-sequence alignment to predict functional elements in genomic sequences, Methods Mol. Biol. 395, 237-254.

[23]Roshan, U. and Livesay, D. R. (2006). Probalign: multiple sequence alignment using partition function posterior probabilities, Bioinformatics 22, 2715-2721.

[24]Wang, C. and Lefkowitz, E. J. (2005). Genomic multiple sequence alignments: refinement using a genetic algorithm, BMC Bioinformatics 6, 200.

[25] Yue, F., Shi J. and Tang, J. (2009). Simultaneous phylogeny reconstruction and multiple sequence alignment, BMC Bioinformatics 10(Suppl. 1) , S11.

[26] Hamdi-Cherif, A. 2010, "Integrating Machine Learning in Intelligent Bioinformatics", WSEAS Trans. on Computers, 9(4):406-417 ISSN: 1109-2750, Wisconsin, USA, http://www.worldses.org/journals/computers/computers-2010.htm

[27] Abdel Azim, G. , Ben Othman, M. and Abo-Eleneen, Z. A. (2010). " Multiple Proteins Sequence Alignment Based on Progressive Methods with New Guide Tree" International Conference on Bioscience and Bioinformatics (ICBB '10)-Vouliagmeni, Athens, Greece, December 29-31, 2010

Dr. Gamil Abdel Azim received his BSc in Mathematics from Cairo University and a DEPS (Diplôme des Etudes Pratiques Supérieures) from Poitiers University , France. He received MSc. and Ph.D. degrees in Computer Science from Paris Dauphine University, France, in 1988 and 1992, respectively. He worked as Associate Professor in the Department of Computer Sciences, College of Computer and Informatics, Canal Suez University, Egypt. He is currently an Associate Professor at Computer Science Department, Computer College, Qassim University, Saudi Arabia. His Current research interests include Neural Networks, Combinatorial Optimization, Pattern Recognition, Evolutionary Computation (Genetic algorithms and Genetic Programming), and Bioinformatics. He supervised about 30 BSc student projects. Dr. Gamil is Member of IEEE.

Mohamed Tahar Ben Othman received his Ph.D. in Computer Science from The National Institute Polytechnic of Grenoble INPG France in 1993, His Master degree in Computer Science from ENSIMAG "École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble" in 1989. He received a degree of Senior Engineer Diploma in Computer Science from Faculty of Science of Tunis. This author became a Member (M) of IEEE in 1997, and a Senior Member (SM) in 2007. He worked as post-doc researcher in LGI (Laboratoire de Genie Logiciel) in Grenoble, France between 1993 and 1995, Dean of the Faculty of Science and Engineering between 1995 and 1997 at the University of Science and Technology in Sanaa, Yemen, as Senor Software Engineer in Nortel Networks, Canada, between 1998 and 2001 and Assistant Professor in Computer College at Qassim University in Saudi Arabia from 2002 until present. His research interest areas are wireless networks, Adhoc Networks, communication protocols, and bioinformatics.

Abo-Eleneen, Z. A. received the B.Sc., M.Sc. and Ph.D. from Zagazig University, Zagazig, Egypt in 1988, 1994, and 2001, respectively. From 1999 to 2001 he was a Visiting Scholar, University of The Ohio State University, USA. He joined the Faculty of Computers and Informatics at Zagazig University, where he held the position of Associate Professor. His research interests include order statistics, estimation theory, information theory, mathematical modeling, and bioinformatics.