Proceedings of Engineering & Technology (PET)
pp. 267-270

2nd International Conference on Automation,Control, Engineering and Computer Science (ACECS-2015)
Proceedings of Engineering & Technology (PET)

Alt_Align: Progressive Multiple Alignment Algorithm Using New Clustering method and New Score Function

Ahmed Mokaddem
Higher School of Sciences and Technologies
(LaTICE),
Tunis, Tunisia
Moka.ahmed@yahoo.fr

Mourad Elloumi
Higher School of Sciences and Technologies
(LaTICE),
Tunis, Tunisia
Mourad.Elloumi@fsegt.rnu.tn

***Abstract*— In this paper, we present a new multiple sequence alignment algorithm called *Alt_Align*. Our algorithm uses the *distance based on motif* [1], new clustering method to construct a guide tree and new score function for profile-profile alignment. We assess our algorithm *Alt_Align* on different datasets extracted from different benchmarks of protein sequences. We obtain interesting results.**

***Keywords*— Multiple sequence alignment; algorithms; complexities; profiles; clustering; distances; guide tree.**

I. INTRODUCTION

The comparison of biological sequences makes a very important contribution in the analysis of biological macromolecules. In fact, it can reveal information about shared biologic functions and structures of macromolecules. This operation can be achieved via Multiple Sequence Alignment (MSA). Multiple Sequence Alignment consists in optimising the number of matches between the residues occurring in the same order for a set of N sequences where $N>2$. MSA is an NP-complete problem [2]. There are several approaches to solve this problem:

- Iterative approach: Algorithms adopting this approach construct an initial alignment. Then, they perform a set of modifications on the current alignment to construct a new one. These modifications are repeated until no improvement can be made on the score of alignment. Among iterative algorithm, we mention HMMER [3], SAGA [4] and QOMA [5].
- Divide and conquer approach: These algorithms operate in three steps. First, they choose a position in each sequence, which subdivides the sequence into two smaller ones. Then, they reiterate recursively this operation until we obtain small sequences that can aligned by an optimal algorithm such us MSA [6]. Finally, they concatenate the alignments of the small sequences to obtain the final alignment. Among algorithm adopting this approach, we mention DCA [7] .
- Progressive approach: Progressive approach is the most used and the most effective one, it operates in three steps:

1) Pairwise comparison: during this step, we compute distances, such as percent of similarity [8], Kimura distance [9], *k*-mer distances [10] and normalized scores [11] between each pairs of sequences. Thus, we can estimate the similarity between pairs of sequences in order to distinguish the sequences that are the first to be aligned. We store these distances in a matrix called *distance matrix*.

2) Sequences clustering: during this step, we use several algorithms, such us UPGMA [12] and Neighbor-Joining [13], in order to define the branching order of aligning sequences by constructing a guide tree.

3) Aligning alignment: during this step, we align sequences using profiles [14], [11] to construct a multiple sequence alignment following the guide tree.

In order to address the drawbacks of the progressive approach, several algorithms can apply a refinement stage. Among most efficient multiple sequence alignment algorithm adopting the progressive approach, we mention: CLUSTALW [8], PRRP [15], T-COFFEE [16], MUSCLE [10], MAFFT [17], PLASMA [18], PROBCONS [19] and GRAMALIGN [20].

II. NOTATION AND DEFINITIONS

Let $\mathcal{A}$ be a finite alphabet, a *sequence* is an element of $\mathcal{A}^*$, it is a concatenation of elements of $\mathcal{A}$. The length of a sequence w, denoted by $|w|$, is the number of the characters that constitute this sequence. A *substring* sw is a portion of w beginning at the position i and ending at the position j, $0<i\leq j\leq|w|$.

Let f a set of N sequences. *Aligning* a set of N sequences consists in optimizing the number of matches between the residues appearing in the same order in each sequence. When $|f|>2$, aligning the sequences is called *Multiple Sequence*

ISSN: 2356-5608

Alignment (MSA). A *profile* associated to a multiple alignment is a sequence constructed by selecting for each column of the multiple alignments the residue that has the maximum occurrences in this column.

A *substring* extracted from a *profile* and not forming by gaps is called *motif.*

Let be *w* and *w'* two sequences, a list of *substrings* that appear in the same order and without overlapping in two sequences *w* and *w'* is called *subsequence*. The length of a subsequence is the sum of the lengths of substrings that compose the subsequence.

Distance based on motifs between two sequences w_i and w_j is calculated using the following formula:

$$D_{motif}(w_i, w_j) = 1 - \frac{\sum_k^{nb} \sum_l occ(T_k, w_l) \times |T_k|}{|w_p| \times N} \quad (1)$$

Where $occ(T_k,w_l) = 1$ if the motif T_k appear in the sequence w_l, else $occ(T_k, w_l) = 0$, w_p is the profile obtained by aligning the sequence w_i and w_j , N is the number of sequences and nb is the number of motifs.

III. *Alt_Clust*: New Method For Guide Tree Construction

To construct a guide tree for a set of sequences, we used clustering algorithm; the most important and efficient algorithms, i.e., UPGMA [12] and Neighbor-joining [13], use the same distance to construct a guide tree that can represents a drawback. In fact, at each step in progressive processes, new nodes are created representing new multiple alignments. Although, the distance used to compare between sequences is not efficient to compare between different alignments or to compare between alignment and sequences.

Thus, we propose a new method of sequence clustering, called *Alt_Clust*, based on different distances in each step of guide tree construction. By our new method, we compute new distance between all nodes at each step, the new distances uses alignments and profiles constructed in the current step.

Our new method to guide tree construction is based on two distances. Thus, our method allows simultaneous construction of the guide tree and the multiple alignment. We use the distance based on motifs [1] to compute the distances between sequences and the length of longest common subsequence [21] to compute between profiles and sequences. Our algorithm *Alt_Clust* operates as follows:

1) During the first step, we construct leaves by assigning a node for every sequence.

2) During the second step, we compute the distance based on motifs [1] between every pair of sequences and we store these distances in the distance Matrix M.

3) During the third step, we select the two closest nodes from the distance Matrix M.

4) During the fourth stage, we align these two sequences to get the first node of the guide tree and we construct the corresponding profile.

5) In the fourth step, we compare all nodes in the current step by computing the length of longest common subsequence [22] between each nodes pairs; the comparison used the profile of each alignment. We obtain the new distance Matrix.

6) In the next step, we back to the third step and we reiterate all this process until we have two nodes.

During the last step, we align the two last nodes to construct the root of the guide tree.

IV. New Score Function For Profile-Profile Alignment

In this section, we present our new score function for *profile-profile* alignment, called MPSP. By using our new score function MPSP; we assign a value to each pair of column from two multiple alignments by using residues occurrences and the scores between residues. In fact, MPSP promotes columns having the maximum occurrences of the same character in the two columns. Thus, by our function, we assign higher scores to the columns that, by aligning them, we maximize the occurrence of this character. In addition, our score function penalizes columns having the higher number of gap in the two columns. To compute the score of two columns, we can use the substitution matrices such as PAM [22] or BLOSUM [23] for scoring two residues.

The MPSP score between two columns *x* and *y* is done by the following formula:

$$MPSPG(x,y) = \sum_i \sum_j (\max\{f_i(x,y), f_j(x,y)\} * (N - f_g(x,y)) * f_{x,i} * f_{y,j} * s(i,j)) \quad (2)$$

Where $f_i(x, y)$ the occurrences of the residue i in both columns x and y, $f_{g,x}$ the number of occurrences of gap character in column x, $f_{g,y}$ the number of occurrences of gap character in column y, $s(i, j)$ the score of the substitution matrix used between residues i and j and N the number of sequences.

V. *Alt_Align* Algorithm

In this section, we present our new algorithm *Alt_Align* that adopts our new clustering method called *Alt_Clust* to construct the guide tree and our new score function called MPSP to construct the *profile-profile* alignment. *Alt_Align* operates as follow:

1) First, we compute the distance base on motifs [1] between each pair of sequences from the initial set *f* and we store the computed distances in a distance matrix *M*.

2) Then, we use the distance matrix *M* to build the guide tree. This is done by adopting our new clustering method *Alt_Clust*. We align sequences following the guide tree using the Needleman and Wunsch algorithm [24] for pairwise alignment. For aligning alignment we adapt the Needleman and Wunsch algorithm to construct the *profile-profile* alignment by using our new score function MPSP to score each pairs of columns instead of the substitution matrix.

3) Finally, we make a refinement step [10].

Time complexity of *Alt_Align* algorithm is $O(N^4 + N*L^2)$ in computing time.

VI. Experimental Study

We assess our program *Alt_Align* using a set of datasets extracted different benchmarks for protein sequences, i.e., BALIBASE [25], OXBENCH [26] and HOMSTRAD [27]. We compared the results obtained by our program with the most used multiple alignment programs, i.e., CLUSTALW2 [8], MUSCLE [10] and MAFFT [17] using the Column Score (CS) [25] and the Sum of Pairs Scores (SPS) [25]. The results of MUSCLE, MAFFT and CLUSTALW2 algorithms are respectively obtained using MUSCLE [10], the online web server of MAFFT [28] and the online web server of CLUSTALW2[8]. TABLE I and TABLE II represent respectively the SPS scores and the CS obtained for datasets extracted from BALIBASE.

TABLE I. Results obtained using the SPS on BALIBASE

DATASETS	CLUSTALW2	MUSCLE	MAFFT	*Alt_Align*
1ZIN (REF1)	0,960	**0,985**	0,960	**0,985**
1DOX(REF1)	0,914	0,924	0,920	**0,932**
1ABOA (REF2)	0,825	0,814	0,804	**0,839**
1CSY(REF2)	0,826	**0,835**	0,770	**0,850**
1IUKY (REF3)	0,486	**0,581**	0,554	0,507
1MFA(REF4)	0,399	0,348	**0,432**	0,426
1LKL(REF4)	0,735	0,767	**0,859**	0,795
2CBA(REF5)	0,746	0,835	**0,845**	0,819
S52(REF5)	0,894	0,893	0,881	**0,902**
KINASE1(REF5)	0,811	**0,841**	0,810	0,812
KINASE2(REF5)	0,806	0,851	**0,877**	0,816
1THM2(REF5)	0,843	0,875	0,891	**0,896**

TABLE II. Results obtained using the CS on BALIBASE

DATASETS	CLUSTALW2	MUSCLE	MAFFT	*Alt_Align*
1ZIN (REF1)	0,920	**0,970**	0,920	**0,970**
1DOX(REF1)	0,850	**0,860**	**0,860**	**0.860**
1ABOA (REF2)	**0,330**	0,320	**0,330**	**0,330**
1CSY(REF2)	0,110	**0,240**	0,110	0,000
1IUKY (REF3)	0,110	0,180	0,120	**0,190**
1MFA(REF4)	0,110	0,070	0,100	**0,130**
1LKL(REF4)	0,410	0,610	**0,710**	0,610
2CBA(REF5)	0,340	**0,660**	0,580	0,590
S52(REF5)	0,830	0,860	0,820	**0,880**
KINASE1(REF5)	0,630	**0,690**	0,620	0,630
KINASE2(REF5)	0,350	0,590	**0,630**	0,610
1THM2(REF5)	0,610	0,670	0,700	**0,740**

We benchmarked also our program *Alt_Align* on 50 datasets extracted from OXBENCH Benchmark. TABLE III shows the average of the scores (TC) and the Q-scores (Q) obtained. The scores TC and Q are respectively similar to CS and SPS. For these 50 datasets, our algorithm gives the best TC and Q scores. Thus, our new algorithm *Alt_Align* can improves results of the multiple sequence alignment algorithm.

TABLE III. Results obtained using the TC and Q scores on OXBENCH

PROGRAM	TC	Q
MUSCLE	0,669	0,786
MAFFT	0,652	0,780
CLUSTALW2	0,667	0,785
Alt_Align	**0,680**	**0,800**

We benchmarked also our algorithm *Alt_Align* on different datasets extracted from HOMSTRAD Benchmark and we compute the Q scores and the TC scores. TABLE IV and TABLE V represent respectively the Q scores and the TC scores obtained.

TABLE IV. Results obtained using the Q on HOMSTRAD

DATASETS	MAFFT	CLUSTALW2	MUSCLE	*Alt_Align*
beta/gamma crystallins	0,495	0,874	0,652	**0,889**
Msb	0,830	0,838	0,862	**0,909**
HLH	0,824	0,785	**0,931**	**0,937**
Nucleotide kinase	0,849	0,849	**0,869**	0,860
Rubis	**0,954**	0,938	0,947	0,922
Peroxidase	0,866	0,885	**0,908**	0,864
Abc_tran	0,611	0,422	**0,616**	0,577
Intb	0,514	0,497	0,523	**0,540**
Bv	0,094	0,115	**0,176**	0,136
LUXS	0,814	0,785	0,881	**0,889**
PHC	0,957	0,841	0,936	**0,960**

TABLE V. Results obtained using the TC on HOMSTRAD

DATASETS	MAFFT	CLUSTALW2	MUSCLE	*Alt_Align*
beta/gamma crystallins	0,016	0,657	0,022	**0,680**
Msb	0,628	**0,697**	0,634	**0,697**
HLH	0,753	0,684	**0,869**	**0,869**
Nucleotide kinase	0,541	0,533	**0,618**	0,610
Rubis	**0,855**	0,792	0,843	0,770
Peroxidase	0,670	0,713	**0,770**	0,676
Abc_tran	**0,381**	0,151	0,335	0,330
Intb	0,308	0,340	0,321	**0,371**
Bv	0,000	0,000	0,000	**0,009**
LUXS	0,753	0,684	0,797	**0,829**
PHC	**0,910**	0,870	0,850	0,872

We note that our algorithm gives, for different datasets extracted from different benchmarks, the best SPS and CS scores. In fact, our new algorithm adopts new clustering method and new score function that allow aligning columns having the maximum occurrence of the same residue, as a consequence, we can align the maximum number of similar residues.

VII. Conclusion And Perspectrives

In this paper, we presented new solutions to improve progressive alignment approach. First, we define new clustering algorithm called *Alt_Clust* in order to construct the guide tree. We used the distance based on motifs [1] and the edit distance [21]. Then, we define new score function for *profile-profile* alignment called MPSP, our new score function attributes to each column of pairs of profiles a score using the occurrences of the residues in the two profiles to be aligned.

We define new multiple progressive alignment algorithm called *Alt_Align*. Our algorithm adopts our new clustering method *Alt_Clust* and the MPSP score function.

We assessed our new algorithm *Alt_Align* on different datasets extracted from different benchmarks, i.e., BALIBASE, OXBENCH and HOMSTRAD of protein sequences, and we compared with other typical programs, i.e., MUSCLE, MAFFT and CLUSTALW using the Sum of Pairs Score (SPS) and the Column score(CS). We proved that we obtain and interesting results for many datasets.

In future work, we can in future work applying more efficient distances in our clustering algorithm *Alt_Clust* and we can improve the aligning alignment step. We would like also to develop new reffinment algorithm to improve the multiple sequence alignment score.

References

[1] A. Mokaddem, M. Elloumi, New Distances For Improving Progressive Alignment Algorithm, in Proc. The 2nd International Conference on Advances in Computing and Information Technology, ACITY'12 (Chennai, India), Lecture Notes in Computer Science (LNCS), Springer-Verlag, Berlin, Heidelberg, Germany (Publish.) : (July 2012).

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

[3] S. R Eddy, Multiple alignment using hidden markov models, in Proc. International Conference on Intelligent Systems for Molecular Biology : (1995), p114-120.

[4] C. Notredame, D. Higgins, SAGA: sequence alignment by genetic algorithm. Nucl. Acids. Res., 24:1515-1524, 1996.

[5] X. Zhang, T. Kahveci, QOMA: quasi-optimal multiple alignment of protein sequences, Bioinformatics, Vol. 23, N°2: (2007), p162–168.

[6] D. J. Lipman, S. F. Altschul, , J. D. Kececioglu, A tool for multiple sequence alignment, Proc. Natl. Acad. Sci. USA, N°86: (1989), p4412-4415.

[7] J. Stoye, Multiple sequence alignment with the divide-and conquer method, Gene, N°211: (1998), pGC45–GC56.

[8] J. D. Thompson, D. G. Higgins, T. J. Gibson, CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice, Nucleid Acids Research, Vol. 22, N°22: (1994), p4673-4680.

[9] M. Kimura: The neutral theory of molecular evolution. Cambridge University Press 1983.

[10] R. C. Edgar, MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics, (5), 113, Vol. (2004).

[11] T. J. Wheeler and J. D. Kececioglu, Multiple alignment by aligning alignments, Bioinformatics, Vol. 23, N°13: (2007), p559–568.

[12] P. Sneath, R. Sokal, Numerical taxonomy, Freeman (Publish.), San Francisco: (1973), p230–234.

[13] N. Saitou, M. Nei, The neighbor-joining method: A new method for reconstructing phylogenetic trees, Mol. Biol. Evol. N°4: (1987), p406–425.

[14] O. Gotoh, Further improvement in methods of group-to-group sequence alignment with generalized profile operations, CABIOS, N°10,: (1994), 379–387.

[15] O. Gotoh, Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments, J. Mol. Biol. Vol. 264, N°4: (1996), p823-838.

[16] C. Notredame, D. Higgins J. Heringa, T-COFFEE: A novel method for multiple sequence alignments. Journal of Molecular Biology, 302:p205–217, 2000.

[17] K. Katoh, M. Standley MAFFT version 5: Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability Nucleic Acids Res. Vol. 33, N°2: (2005), p511–518

[18] V. Derrien, J. M. Richer, J. K. Hao, PLasMA: un nouvel algorithme progressif pour l'alignement multiple des séquences, in Proc. Premières Journées Francophones de Programmation par Contraintes (JFPC'05) : (Juin 2005), p39-48.

[19] C. B. Do, M. S. Mahabhashyam, M. Brudno, S. Batzoglou, PROBCONS: Probabilistic consistency-based multiple sequence alignment, Genome Res. N°15: (2005), p330–340.

[20] D. J. Russell, H. H. Out, K. Sayood, Grammar-based distance in progressive multiple sequence alignment, BMC Bioinformatics, Vol. 9: (2008).

[21] D. S.: A linear space algorithm for computing maximal common subsequences. Commun.ACM 18 , 6 (1975), 341–343

[22] M. O. Dayhoff, R. M. Schwartz, B. C. Orcutt, A model of evolutionary change in proteins, in Atlas of Protein Sequence and Structure, chapter 22, National Biomedical Research Foundation, Washington, DC: (1978), p345–358.

[23] S. Henikoff, J. G. Henikoff, Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci. U.S.A., Vol. 89 : (Nov. 1992), p10915-10919.

[24] S. B Needleman, C. D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, Vol. 48, N°1: / p443-453.

[25] J. D. Thompson, F. Plewniak, O. Poch, A comprehensive comparison of multiple sequence alignment programs, Nucleic Acids Res., Vol. 27, N°13: (1999), p2682–2690.

[26] G. P. Raghava, S. M. Searle, P. C. Audley, J. D. Barber, G. J. Barton, OXBENCH: a benchmark for evaluation of protein multiple sequence alignment accuracy, BMC Bioinformatics, Vol. 4: (2003).

[27] L. A. Stebbings, K. Mizuguchi, HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database, Nucleic Acids Research, Vol. 32, Database issue: (2004), pD203-D207

[28] K. Katoh, K. Kuma, H. Toh, T. Miyata, MAFFT version 7: Improvement in accuracy of multiple sequence alignment, Molecular Biology and Evolution Vol. 30, p772-780. (2013).