International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 10, Issue 2 (February 2021), PP. 10-20

Multiple Sequence Alignment of Proteins using an Extension of SARELI

Arturo Chavoya[*], Ricardo Ortega[**]

** Professor, Department of Information Systems - CUCEA, Guadalajara University, Guadalajara, Jalisco,Mexico,*

***Ph.D.graduate, Department of Information Systems - CUCEA, Guadalajara University, Guadalajara, Jalisco, Mexico*

Corresponding Author:achavoya@cucea.udg.mx

Abstract:*A new method for aligning multiple protein sequences as well as refining methods to realign them is presented in this document. The algorithm involved in the initialization of the progressive algorithm for the alignment of sequences is computed by a radius parameter that estimates the variation between sequences; afterwards, a guide tree is created using the neighbor joining algorithm. For scoring the alignment, we introduced as metric a threshold of the number of correctly aligned symbols, and marked each sequence that reached the threshold; this metric was used to realign sequences with different thresholds in order to adjust the alignments. Our proposed method SARELI, which stands for Sequence Alignment by Radial Evaluation of Local Interactions, was previously reported to only generate the guide trees, but in this article we extended the algorithm to generate the final alignment of proteins. The results from this method in the alignment of the sequences was compared with the results from Clustal W version 2, Clustal Omega, MAFFT, MUSCLE,and T-Coffeee on the BAliBASE, PREFAB, and SABmark protein sequence databases, using the column score and the sum of pairs scorings. After aligning the sequence datasets, our proposal obtained statistically superior scores on test cases where the number of sequences was between 5 and 25 with less than 30% of original identity between the sequences from all of the three databases considered.*

Keywords:*SARELI; multiple sequence alignment; guide tree metrics; MUSCLE; protein databases.*

Date of Submission: 28-02-2021 Date of acceptance: 16-03-2021

I. INTRODUCTION

In bioinformatics, the multiple sequence alignment (MSA) of related proteins is one of the most relevant problems, since its solution can help predict both protein structure and function, as well as enlighten researchers on the phylogenetic relationship of species. However, despite significant advances in the performance of alignment algorithms, finding consistently accurate alignments can prove difficult [1].

MSA algorithms start with a set of three or more possibly related biological sequences (proteins or nucleic acids) and proceed to obtain a set of sequences of the same length that matches as many homologous symbols (representing amino acids or nucleotides) as possible from the initial sequences. In order to obtain a better alignment, gap symbols can be introduced to displace the columns of the sequences.

SARELI, which stands for Sequence Alignment by Radial Evaluation of Local Interactions, is a software tool that has been used to produce guide trees that are employed in protein MSA algorithms [2]. Guide trees determine the order in which pair sequences are to be compared, usually starting with the most similar sequences and proceeding with the most dissimilar [3]. On the other hand, in addition to their own guide trees, MSA tools such as MUSCLE [4] and Clustal Omega [5] can use external guide trees, such as those generated by SARELI, as input to produce a final alignment [2].

In a previous report it was shown that when MUSCLE uses the guide trees from SARELI, it can produce statistically better sum of pairs and column scores of alignments on some protein benchmark databases, than when MUSCLE uses its original guide trees [2]. In the present article we explore the use of an extension of SARELI to perform the totality of steps required for the MSA of proteins, from the generations of the guide trees to the production of the final alignment file; for the remainder of this article, SARELI will refer to this extension of the software.

The rest of this paper is divided into the following sections. In Section II the protein databases, as well as the scoring methods used to measure the quality of the alignments, are described. In Section III we present our proposed metrics used to enhance the score quality of the alignments. The procedure followed by the extension of SARELI is described in Section IV, whereas the results on the alignment runs and the discussion on the benefits of using our proposed metrics are presented in Section V. Finally, conclusions and future work are considered in Section VI.

II. DATABASES AND SCORING METHODS

This section describes the protein benchmark databases used in the present study, as well as the definition of the scores applied to evaluate the alignments.

2.1 Databases

Three different benchmark databases were used in this work to corroborate the validity of our proposed method: BAliBASE 3 [6], PREFAB 4.0 [4], and SABmark [7]. The BAliBASE database was designed as an evaluation resource for addressing problems that arise when aligning complete sequences [8] and has been widely used for testing and comparison purposes [4], [9]–[11]. PREFAB is a database formulated from an automated protocol to select a set of sequences from published works [4]. Finally, SABmark uses a more systematic method to select the sequences based on the ASTRAL database [12].

In order to characterize the databases, we considered four main features from the point of view of symbols, rather than from the biological point of view. First, we counted the number of files per database, and as Table 1 shows, PREFAB was the most extensive, followed by SABmark, and with BAliBASE having the fewest files.

Table 1 Number of files per database

Database	**Number of files**
BAliBASE	386
PREFAB	1682
SABmark	425

The second distinctive characteristic considered in the databases was the number of sequences per file. To generate the distributions presented in Table 2, we sorted in ascending order the number of sequences per file for each database and made a numerical regression to obtain an equation representing the distribution, calculating the corresponding *R*-Squared values to validate the representation.

Table 2Number of sequences per database

Database	**Distribution**	***R*-Squared**
BAliBASE	$e^{1.174+0.009x}$	0.9941
PREFAB	First 400: $-7.43138+2.77544\sqrt{x}$ Rest:50	0.9672
SABmark	$e^{1.06803+0.0000121257\,x^2}$	0.9877

The third characteristic in the databases we analyzed was the average length of the sequences. In a similar manner to the process described for Table 2, we sorted in ascending order the average length per file for each database and made a numerical regression to find the distributions shown in Table 3.

Table 3Average length of sequences per file

Database	**Distribution**	***R*-Squared**
BAliBASE	$e^{3.67227+0.148005\sqrt{x}}$	0.9828
PREFAB	$e^{4.38954+0.00109666\,x}$	0.9801
SABmark	$e^{3.96002+0.00476144\,x}$	0.9616

The last characteristic considered is the p (proportion) distance [13], which can be used for comparing the degree of sequence divergence even with sequences of different length. The p-distance for each pair of sequences is estimated by dividing the number of amino acid differences (without considering insertions, deletions, or gaps) by the total number of amino acids compared; the p-distances were obtained using the MEGA software. For every file, an average of the p-distances between all pairs of sequences in the file was calculated. Applying a numerical regression to the sorted average distances, we obtained the distributions presented in Table 4.

Table 4Average length of sequences per file

Database	**Distribution**	***R*-Squared**
BAliBASE	$e^{1.87485+0.0913229\sqrt{x}}$	0.9824
PREFAB	$21.1275+0.000011366x^2$	0.9680
SABmark	$e^{1.76961+0.0620877\sqrt{x}}$	0.9548

2.2 Scoring

For the evaluation of multiple alignments, the sum of pairs (SP) is a common way to measure the quality of the alignments [14]–[19], and is calculated by adding all the possible pairs from each column of the alignment, without repetition, using a substitution matrix as a score for the aligned symbol and is calculated as

$$SP(A) = \sum_{j=1}^{M} \sum_{i_1=1}^{N-1} \sum_{i_2=i_1+1}^{N} Score(A_{i_1 j}, A_{i_2 j}), \qquad (1)$$

where *A* is the set of sequences to be scored, *M* is the length of the alignment, *N* is the number of sequences in the set, and *Score* is a function that returns the score of a substitution matrix, which in our case was the BLOSUM62 matrix [20].

Other common evaluation measure for MSA is the column score (CS), which add one unit per column where all the residues are aligned. This score is commonly used in combination with SP to evaluate the quality of the alignments. The column score (CS) is calculated as

$$CS(A) = \sum_{i=1}^{M} C_i, \qquad (2)$$

where *A* is the set of sequences, $C_i = 1$ if all the residues in the i-th column are aligned or 0 otherwise, and *M* is the length of the sequences in the alignment[15].

III. PROPOSED METRICS

Two metrics used for the alignment of sequences are proposed, one for the construction of the initial guide tree used by the neighbor joining algorithm (Radial Distance), and one for a score scheme that allows refinement methods to be applied to the already aligned sequences (Column Error Score). These metrics are detailed below, and examples are presented to illustrate their implementation.

3.1 Radial distance

The Radial Distance was previously reported as a metric that, when comparing two sequences, measures the distance between them, taking into consideration not only the symbol in the column to be aligned, but also the symbols surrounding the column [2]. The Radial Distance takes a radius parameter value that limits the number of symbols around each column of the pairwise alignment to be considered into the sum. As the distance from the referenced column is increased, the influence on the score is decreased with an asymptotic function. As previously reported, the Radial Distance (RD) between sequences *A* and *B* is defined as

$$RD(A,B) = \sum_{i=1}^{M} \sum_{j=i-R>0}^{i+R\leq M} \frac{Score\,(A_i,B_j)}{Abs\,(i-j)+1}, \qquad (3)$$

where *M* is the length of the initially aligned sequences using dynamic programming, and *R* is the radial parameter value that indicates how far the weights of the adjacent columns will affect the score [2]. The *Score* function used was the BLOSUM62 substitution matrix [20].

An example of the calculation of the RD for two sequences for a radius value of 2 is shown in Fig. 1, where the sixth step of the process is presented for position 6, with the corresponding RD values computed up to that point; the symbol *nc* represents an RD value not yet calculated. At the end of the process, the sum of all the individual values represents the radial distance between the two sequences.

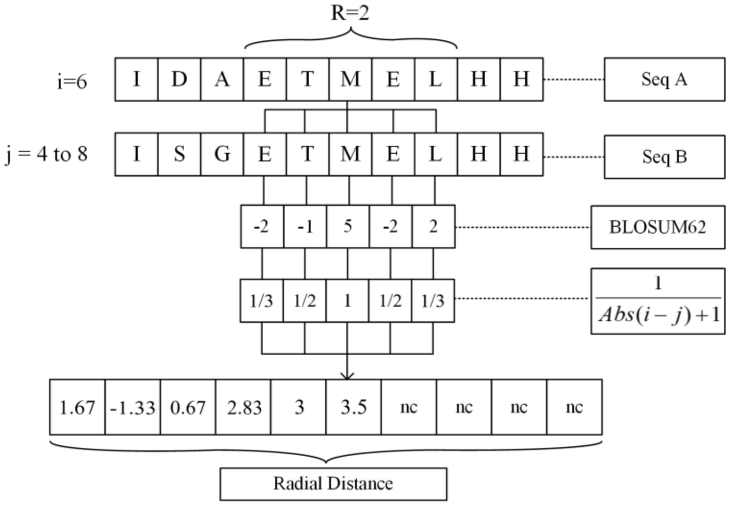


Fig. 1. Example of the partial calculation of the Radial Distance for sequences *A* and *B*.

3.2Column Error Score

In order to obtain a better column score, we devised a scoring method that, given a threshold (a percentage of the column that is already aligned), it identifies a potential column that might become better aligned. The method starts by flagging columns whose most repeated symbol is above the given threshold (as illustrated in Fig. 2 for a threshold of 75%). The *Flags* vector is calculated for each position by

$$Flags_j = \begin{cases} 1 & if 100\left(\frac{1}{N}\sum_{i=1}^{N} Q(A_{ij})\right) \geq Threshold \\ 0 & otherwise \end{cases}, \quad (4)$$

where *N* is the number of sequences, *A* is the set of sequences, *Threshold* is a percentage of already aligned symbols, and *Q* is the function

$$Q(A_{ij}) = \begin{cases} 1 & if A_{ij} = S_j \\ 0 & otherwise \end{cases}, \quad (5)$$

where S_j is the most repeated symbol in the *j*-th column of the alignment.

After the *Flags* vector is built, the number of symbols different from the most repeated symbol is counted in the flagged columns for each sequence; this number is the Column Error Score (CES) for the sequence. The CES for the *i*-th sequence is formulated as

$$CES(i) = \sum_{j=1}^{M}\left(1 - Q(A_{ij})\right) \cdot Flags_j, \quad (6)$$

where *M* is the alignment length, and *Q*, *A* and *Flags* are as defined above.

In our proposed alignment method, this score is used to realign the sequences with a higher CES. In the example of Fig. 2, which illustrates how to obtain this score, sequence A_3 would be selected to be realigned, as it has the highest CES; this realignment process can be iterated as needed.

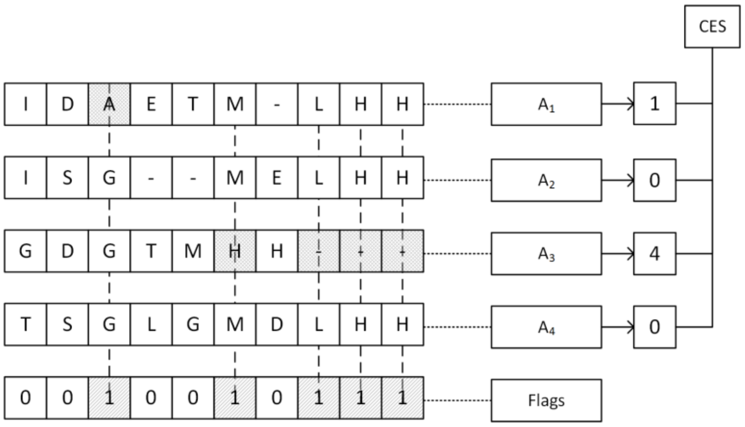


Fig. 2. Example of calculation for the Column Error Score

Figure 3 shows how Sequence A3 from Fig. 2 (the sequence with the highest CES) is realigned, yielding a better column score and sum of pairs score. The refining method based on the CES is used in our proposed method with a threshold parameter and a number or iterations per set of sequences.

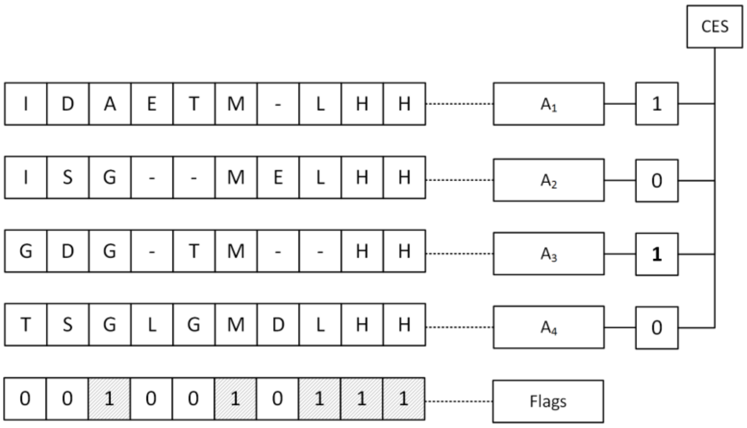


Fig.3.Example of a refined alignment.

IV. SARELI EXTENSION

The process followed by SARELI for aligning the sequences in each file for the three databases considered (BAliBASE, PREFAB, and SABmark) is depicted in Fig. 4. In our proposed method, the construction of the initial distance matrix is achieved using the Radial Distance metric and the refinement process is improved by use of our Column Error Score metric, whereas the rest of the process corresponds to a common progressive alignment method.

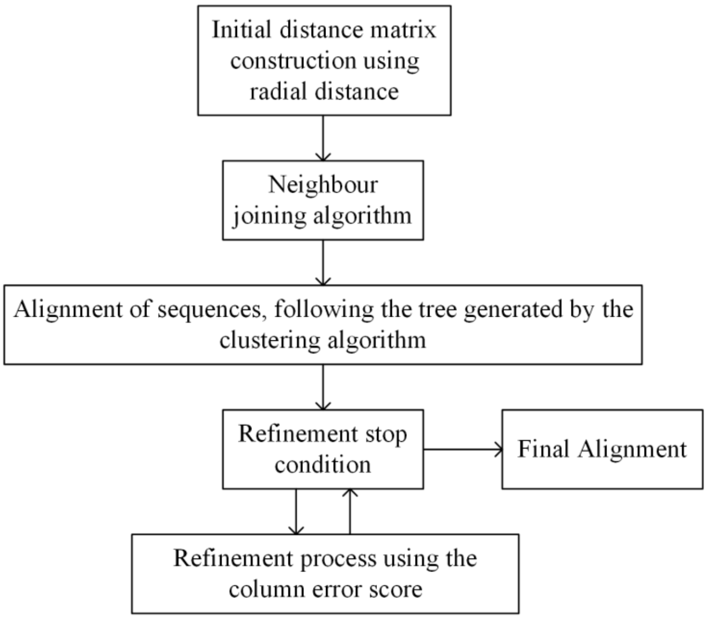
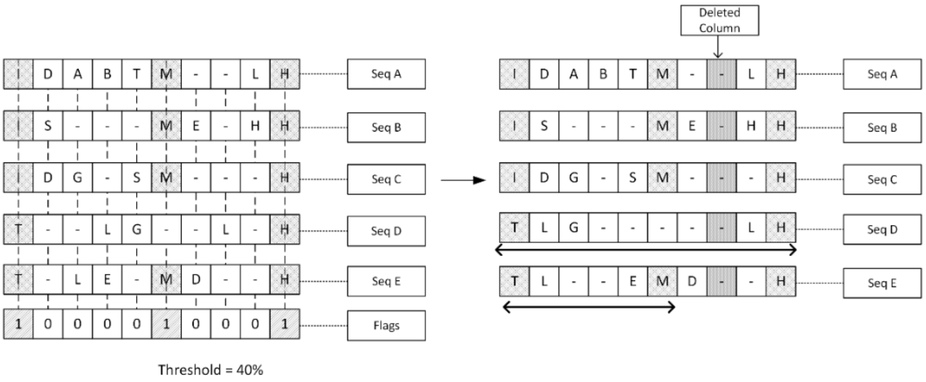

Initial distance matrix construction using radial distance
Neighbour joining algorithm
Alignment of sequences, following the tree generated by the clustering algorithm
Refinement stop condition
Final Alignment
Refinement process using the column error score

Fig. 4. Alignment process used in SARELI.

We used the sum of pairs and column score for measuring the quality of the alignments employing the BLOSUM62 [20] substitution matrix on the BAliBASE, PREFAB, and SABmark benchmark databases. Each sequence set from these databases was aligned using our proposed method and the scores compared against those from ClustalWversion 2[21][10], ClustalOmega[22], MUSCLE[4], MAFFT [23], and T-Coffee [24]. We heuristically determined the radius value that maximized the scoring methods for each sequence file, using a value from the range of 3 to 10.

In order to refine the alignments in our proposed method, the CES was configured with a threshold of 80% and the sequence with the highest error score was selected for realignment. This process iterated until the number of possible aligned columns decreased (indicating that the columns began to misalign), or when the same score was maintained for a third of the number of sequences (indicating a potential lack of improvement in the scores). The value for the latter condition and the threshold values were determined heuristically after a number of trials.

As a second refining method, we first calculated the CES for each sequence in the set using a threshold of 40% to use the flagged columns as anchors to potentially rearrange the symbols between them. When gaps were present in the sequence, the symbols lying at each side of a flagged column were displaced towards the closest column that acted as anchor. This process is illustrated in Fig. 5, where the anchor columns are presented with a shadowed background and arrows are placed below sequence fragments that were realigned after this refinement process was applied. At the end of each iteration, a verification step is made to delete columns that only contain gaps.



I	D	A	B	T	M	-	-	L	H	Seq A
I	S	-	-	-	M	E	-	H	H	Seq B
I	D	G	-	S	M	-	-	-	H	Seq C
T	-	-	L	G	-	-	L	-	H	Seq D
T	-	L	E	-	M	D	-	-	H	Seq E
1	0	0	0	0	1	0	0	0	1	Flags

Threshold = 40%

Deleted Column

I	D	A	B	T	M	-	-	L	H	Seq A
I	S	-	-	-	M	E	-	H	H	Seq B
I	D	G	-	S	M	-	-	-	H	Seq C
T	L	G	-	-	-	-	-	L	H	Seq D
T	L	-	-	E	M	D	-	-	H	Seq E

Fig. 5.Example of the second refinement method.

All the tests were performed on an Apple Mac Pro computer with two Quad-core 2.80-GHz Intel Xeon processors, 6 GB of RAM, and a Quadro 4000 for Mac NVIDIA card. The operating system used was Windows 10, whereas the library containing the alignment algorithm for SARELI was coded in C# using the Visual Studio 2013 IDE and compiler, and version 6.5 of the CUDA library. The source code for the library used by our method is freely available at [25], or can be installed into the solution directly from the NuGet repositories, running "Install-Package SARELI_DLL" in the command line of the Package Manager Console. To use the CUDA functions, the file "kernel.cu" that is distributed with the library needs to be compiled into a .PTX file specific for the graphic card used; a batch file is packed as well to compile this file after the Visual Studio and CUDA software development kit are correctly installed. This software has only been tested on the Microsoft Windows 10 OSand is released under the MIT License. Finally, the statistical analysis was performed using STATGRAPHICS Centurion XVI.

V. RESULTS AND DISCUSSION

After running kurtosis, skewness, chi-squared, and Shapiro-Wilk statistical tests on the BAliBASE, PREFAB, and SABmark databases, the scores obtained for the sum of pairs (SP) score and column score (CS) for each set of samples did not show a normal distribution; therefore, we used a non-parametric statistical test to compare the medians of the scores. Since the experiment consisted of more than two sets of related samples, we used the Friedman test to determine if there was at least one score statistically different—if the *p*-value indicated that there was a difference between the medians, a Wilcoxon test should be applied to compare the scores by pair of set of samples. Table 5 shows the results after applying the Friedman test, whereas Tables 6, 8 and 10 show the medians by database. According to the *p*-values from Table 5, there was at least one database statistically different; thus, the Wilcoxon test was applied and the results per database are presented in Tables 7, 9 and 11.

Table 5Friedman test *p*-values for the databases

Method	**Score**	
	SP	**CS**
BAliBASE	0.0000	0.0000
PREFAB	0.0000	0.0000
SABmark	0.0000	0.0000

Table 6BAliBASE medians

Method	**Score**	
	SP	**CS**
SARELI	1613	5
ClustalWversion 2	399	4
ClustalOmega	229	6
MAFFT	446.5	6.5
MUSCLE	692	5.5
T-Coffee	-611	6

Table 7Wilcoxon test *p*-values for SARELI with BAliBASE

Method	**Score**	
	SP	**CS**
ClustalW version 2	0.0199	0.0000
ClustalOmega	0.0000	0.0004
MAFFT	0.0000	0.2837
MUSCLE	0.6315	0.0002
T-Coffee	0.0043	0.6206

According to the sum of pairs criterion from Tables 6 and 7, when testing the methods using BAliBASE, SARELI resulted statistically better than ClustalOmega, MAFFT, and T-Coffee at 99% of confidence, and it resulted better than ClustalW version 2 at 95% of confidence, whereas SARELI and MUSCLE were statistically equal at 99% of confidence. As for the column score, SARELI was better than ClustalW version 2, and equal to MAFFT and T-Coffee at 99% of confidence, whereas our method was not better than ClustalOmega or MUSCLE at 99% of confidence.

Table 8PREFAB medians

Method	**Score**	
	SP	**CS**
SARELI	66639.5	1
ClustalW version 2	117279	2
Clustal Omega	112439	2
MAFFT	108912.5	3
MUSCLE	123234	2
T-Coffee	79480.5	3

Table 9Wilcoxon test *p*-values for SARELI with PREFAB

Method	**Score**	
	SP	**CS**
ClustalW version 2	0.0000	0.4358
Clustal Omega	0.0000	0.0000
MAFFT	0.0000	0.0000
MUSCLE	0.0000	0.0000
T-Coffee	0.4395	0.4682

As for the PREFAB database, Tables 8 and 9 show that regarding the column score, SARELI was statistically equal to ClustalW version 2 at 99% of confidence, whereas ClustalOmega, MAFFT and MUSCLE were statistically better than our method at 99% of confidence. As for the sum of pairs, SARELI resulted statistically equal to T-Coffee, whereas all of the other methods had a statistically significant difference in their favor at 99% of confidence.

Table 10SABmark medians

Method	**Score**	
	SP	**CS**
SARELI	473	6
ClustalW version 2	47	6
Clustal Omega	74	3
MAFFT	160	3
MUSCLE	56	4
T-Coffee	51	4

Table 11Wilcoxon test *p*-values for SARELI with SABmark

Method	**Score**	
	SP	**CS**
ClustalW version 2	0.0000	0.0000
Clustal Omega	0.0000	0.0000
MAFFT	0.0000	0.0000
MUSCLE	0.0000	0.0000
T-Coffee	0.0017	0.0022

When comparing the methods using the SABmark database (Tables 10 and 11), SARELI resulted statistically better than all of the other algorithms for the column score and the sum of pairs score with 99% of confidence—as both SARELI and ClustalW version 2 had a column score value of 6, it was necessary to use a

Box-and-Whisker plot with median notch to determine that the former rendered better results than the latter for this particular score. A summary of the comparisons of SARELI against the other methods for all databases is presented in Table 12.

Table 12Summary of the comparisons of SARELI against the other methods

Method	Score	BAliBASE	PREFAB	SABmark
ClustalW version 2	SP	+	-	++
	CS	++	=	++
Clustal Omega	SP	++	-	++
	CS	-	-	++
MAFFT	SP	++	-	++
	CS	=	-	++
MUSCLE	SP	=	-	++
	CS	-	-	++
T-Coffee	SP	++	=	++
	CS	=	=	++

++: Better al 99% confidence; +: Better at 95% confidence; -: Worse; =: No statistically significant difference

The positive results obtained with SARELI when tested with the SABmark database gave us an insight as to the strength and weakness of our method. A further analysis of this database indicated that it has a distinctive pattern in the sequences with respect to the other two, i.e. the number of sequences is less than or equal to 25, and the initial average distance is less than or equal to 30% between pairs of sequences. We decided to take these parameter values to filter the databases for further testing—since the average length for all the databases was similar, we did not include this criterion for the filtered datasets. After applying the filter, the number of files obtained was 209 for BAliBASE, 36 for PREFAB, and the 425 files from SABmark. None of these datasets showed a normal distribution; thus, we used the Friedman test to verify if there was a statistically difference between the medians of the samples, and since the *p*-values indicated that at least one of the samples was statistically different, we used the Wilcoxon test per pairs to determine which of the medians was different. The medians for the filtered dataset are presented in Table 13, whereas the *p*-values from the Wilcoxon test are presented in Table 14. SARELI showed statistically better results at 99% confidence on the column and sum of pairs scores when compared against all the other MSA methods; we therefore recommend using our method when the sequence sets to be aligned meet the criteria used for the filtered datasets.

Table 13Filtered dataset medians

Method	Score	
	SP	CS
SARELI	415	6
ClustalW version 2	-32	3
Clustal Omega	-42	4
MAFFT	74	5
MUSCLE	160	4
T-Coffee	-133	4

Table 14Wilcoxon test *p*-values for SARELI with the filtered dataset

Method	Score	
	SP	CS
ClustalW version 2	0.0000	0.0000
Clustal Omega	0.0000	0.0000
MAFFT	0.0000	0.0000
MUSCLE	0.0000	0.0000
T-Coffee	0.0026	0.0014

VI. CONCLUDING REMARKS

In this article we present a new method that enhances the multiple sequence alignment process for proteins, with a novel metric for constructing the initial distance matrix employed by the neighbor joining algorithm. This matrix is used to obtain a guide tree that seeks to maximize the sum of pairs and column scores by establishing the order in which the sequence pairs are to be aligned. The proposed metric was named Radial Distance, as it considers the effect adjacent symbols within a given radius can have on every symbol in a pair of aligning sequences. We additionally propose a metric termed Column Error Score used by two refining methods that further enhance the alignments: one method helps select the sequence in the set that needs to be realigned, and a second method identifies sequence segments that can be realigned. Our proposed MSA method SARELI, which stands for Sequence Alignment by Radial Evaluation of Local Interactions, was previously reported to generate guide trees, but in this article we present an extension of the algorithm to continue the steps to yield the final multiple sequence alignments.

We compared SARELI against the well-known MSA programs Clustal W version 2, Clustal Omega, MAFFT, MUSCLE, and T-Coffee using the BAliBASE 3, PREFAB 4.0, and SABmark protein sequence databases. We compared the resulting alignments using the column score and the sum of pairs scoring. Using BAliBASE and the sum of pairs score, SARELI was statistically better than Clustal W version 2, Clustal Omega, MAFFT, and T-Coffee, whereas the alignments from SARELI and MUSCLE were statistically equivalent; as for the column score, SARELI was better than Clustal W version 2 and equal to MAFFT and T-Coffee, but not better than Clustal Omega or MUSCLE. With PREFAB, SARELI was statistically equal to Clustal W version 2 and T-Coffee using the column score, whereas for the sum of pairs, T-Coffee was statistically equal to SARELI, but not better in the rest of cases. With the SABmark database, SARELI was statistically better than all of the other MSA methods, both in column score and sum of pairs. When limiting the number of sequences per set to 25 and 30% of initial similitude in BAliBASE and PREFAB (as in SABmark), SARELI was statistically better than all the other methods for the three databases, both in column score and sum of pairs. We therefore recommend using SARELI when these conditions apply.

As future work, we would like to use our two proposed metrics on already known algorithms from other packages to assess the behavior of the scoring on those implementations. We would also like to use additional benchmark databases with SARELI to perform further tests, and we would like to exploit the advantages of parallel architectures of GPUs and cluster computing to further improve execution performance. As for parameter determination of the radius value, we used a heuristic method to find the best radius for each sequence set file, but it would be of great help if this parameter could be automatically calculated using characteristics from the sequence set to be aligned, such as the initial distances, and the length and number of sequences. Finally, we would like to enhance the determination of the gap penalty to allow more sequences to be aligned without compromising the accuracy of the final alignment.

REFERENCES

[1] K. Katoh and H. Toh, “Recent developments in the MAFFT multiple sequence alignment program.,” *Brief. Bioinform.*, vol. 9, no. 4, pp. 286–298, Jul. 2008, doi: 10.1093/bib/bbn013.

[2] R. Ortega, A. Chavoya, C. López-Martín, and L. Delaye, “SARELI : Sequence Alignment by Radial Evaluation of Local Interactions,” *Curr. Bioinform.*, vol. 13, no. 3, pp. 290–298, 2018, doi: 10.2174/1574893613666180130143055.

[3] D.-F. Feng and R. F. Doolittle, “Progressive sequence alignment as a prerequisitet to correct phylogenetic trees,” *J. Mol. Evol.*, vol. 25, no. 4, pp. 351–360, 1987, doi: 10.1007/BF02603120.

[4] R. C. Edgar, “MUSCLE: multiple sequence alignment with high accuracy and high throughput,” *Nucleic Acids Res.*, vol. 32, no. 5, pp. 1792–1797, Mar. 2004.

[5] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, and W. Li, “Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega,” *Mol Syst Biol*, vol. 7, 2011, doi: 10.1038/msb.2011.75.

[6] J. D. Thompson, F. Plewniak, and O. Poch, “BAliBASE: A benchmark alignment database for the evaluation of multiple alignment programs,” *Bioinformatics*, vol. 15, no. 1, pp. 87–88, 1999, doi: 10.1093/bioinformatics/15.1.87.

[7] I. Van Walle, I. Lasters, and L. Wyns, “SABmark—a benchmark for sequence alignment that covers the entire known fold space,” *Bioinformatics*, vol. 21, no. 7, pp. 1267–1268, Apr. 2005.

[8] A. Bahr, J. D. Thompson, J.-C. Thierry, and O. Poch, “BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations,” *Nucleic Acids Res.* , vol. 29, no. 1, pp. 323–326, Jan. 2001.

[9] K. Karplus and B. Hu, “Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set,” *Bioinformatics*, vol. 17, no. 8, pp. 713–720, 2001.

[10] M. A. Larkin *et al.*, “Clustal W and Clustal X version 2.0,” *Bioinformatics*, vol. 23, no. 21, pp. 2947–2948, 2007.

[11] T. Lassmann and E. L. . Sonnhammer, “Quality assessment of multiple alignment programs,” *FEBS Lett.*, vol. 529, no. 1, pp. 126–130, Oct. 2002, doi: 10.1016/S0014-5793(02)03189-7.

[12] R. C. Edgar, “Quality measures for protein alignment benchmarks,” *Nucleic Acids Res.* , Jan. 2010, doi: 10.1093/nar/gkp1196.

[13] R. H. Thomas, “Molecular Evolution and Phylogenetics,” *Heredity (Edinb).*, vol. 86, no. 3, p. 385, 2001, doi: 10.1046/j.1365-2540.2001.0923a.x.

[14] Q. Zhan, Y. Ye, T.-W. Lam, S.-M. Yiu, Y. Wang, and H.-F. Ting, “Improving multiple sequence alignment by using better guide trees,” *BMC Bioinformatics*, vol. 16, no. Suppl 5, pp. S4–S4, Mar. 2015, doi: 10.1186/1471-2105-16-S5-S4.

[15] J. D. Thompson, F. Plewniak, and O. Poch, “A comprehensive comparison of multiple sequence alignment programs,” *Nucleic Acids Res*, vol. 27, 1999, doi: 10.1093/nar/27.13.2682.

[16] I. Van Walle, I. Lasters, and L. Wyns, “Align-m-a new algorithm for multiple alignment of highly divergent sequences.,” *Bioinformatics*, vol. 20, no. 9, pp. 1428–1435, 2004, doi: 10.1093/bioinformatics/bth116.

[17] J. Stoye, V. Moulton, and A. W. M. Dress, “DCA: An efficient implementation of the divide-and-conquer approach to

simultaneous multiple sequence alignment," *Bioinformatics*, vol. 13, no. 6, pp. 625–626, Dec. 1997.

[18] D. J. Lipman, S. F. Altschul, and J. D. Kececioglu, "A tool for multiple sequence alignment," *Proc. Natl. Acad. Sci.*, vol. 86, no. 12, pp. 4412–4415, 1989.

[19] C. Lee, C. Grasso, and M. F. Sharlow, "Multiple sequence alignment using partial order graphs," *Bioinformatics*, vol. 18, no. 3, pp. 452–464, 2002.

[20] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 89, no. 22, pp. 10915–9, Nov. 1992.

[21] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Res*, vol. 22, 1994, doi: 10.1093/nar/22.22.4673.

[22] F. Sievers *et al.*, "Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega.," *Mol. Syst. Biol.*, vol. 7, no. 1, p. 539, Jan. 2011, doi: 10.1038/msb.2011.75.

[23] K. Katoh and D. M. Standley, "MAFFT multiple sequence alignment software version 7: Improvements in performance and usability," *Mol. Biol. Evol.*, vol. 30, no. 4, pp. 772–780, Apr. 2013, doi: 10.1093/molbev/mst010.

[24] C. Notredame, D. G. Higgins, and J. Heringa, "T-coffee: a novel method for fast and accurate multiple sequence alignment," *J. Mol. Biol.*, vol. 302, no. 1, pp. 205–217, 2000, doi: 10.1006/jmbi.2000.4042.

[25] R. Ortega, "SARELI Source code," 2016. https://github.com/icariantk/SARELI (accessed Nov. 18, 2020).

Arturo Chavoya. "Multiple Sequence Alignment of Proteins using an Extension of SARELI." *International Refereed Journal of Engineering and Science (IRJES)*, vol. 10, no. 02, 2021, pp 10-20.