CACiC 2023
CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN

Analysis of Bioinformatic algorithms for MSA

Adrián Díaz (student)[1] and Gabriela Minetti (director)[2] [0000-0003-1076-6766]

Universidad Nacional de Quilmes

Facultad de Ingeniería, Universidad Nacional de La Pampa, Argentina
diaz.adrian.g@gmail.com, minettig@ing.unlpam.edu.ar

Abstract. Aligning three or more biological sequences, such as DNA, RNA, or protein, is known as multiple sequence alignment (MSA). MSA is crucial in identifying important information about the sequences, including function, evolution, and structure. It serves as the first step in analyzing phylogenetic, protein, and genomic data. However, as sequence scale increases and the demand for alignment accuracy grows, MSA faces new challenges. Therefore, developing an efficient and precise tool for MSA and comparing its performance with existing ones has become a research hotspot in Bioinformatics. In this magister thesis, we propose a metaheuristic algorithm to solve MSA and a methodology to compare the performance of algorithms for aligning multiple sequences.

Keywords: Multiple sequence alignment, Bioinformatics, Simulated Annealing, Metaheuristics

1. Introduction

Bioinformatics is an interdisciplinary field that brings together experts in molecular biology, mathematicians, engineers, and physicists to align biological sequences, analyze genomic sequences, identify and predict molecular structures, determine the gene expression profile, etc. This discipline includes the computational techniques and applications that carry out these activities; being multiple sequence alignment (MSA) one of the most important bioniformatic problems [1].

Multiple sequence alignment is the process of aligning three or more biological sequences, typically DNA/RNA/protein. As the first step in phylogenetic, protein, and genomic analysis, MSA reveals the potential information about biological sequences, such as function, evolution, and structure. Obtaining an MSA is no trivial task because it involves complex calculations, and the results are not always biologically accurate. In addition, the required computational effort is highly dependent on the number of sequences to be aligned, becoming this kind of optimization in an NP-hard problem [1]. Although the execution speed and the result quality have improved over the years, producing alignments with insertions, deletions, and gaps in positions closer to biological reality is still a problem to be studied by bioinformatics researchers. As a consequence, the amount of software available to perform multiple alignment is large and diverse. Notice that, according to Google Scholar, in May 2023, the five most cited MSA software systems are: Clustal Omega [15], MUSCLE [6], KAlign [10], T-Coffe [5], and MAFFT [8].

Based on the above, our main goals are to design an algorithm that solves efficiently MSA and propose a well-defined methodology to compare different algorithms for MSA. At this research stage, we adapt an stochastic algorithm to align multiple sequences and introduce an algorithmic comparison methodology. A future step in this

magister thesis is to implement the suggested comparison method and to evaluate the performance of the proposed algorithm compared to these five algorithms on various biological sequences.

2. Multiple Sequence Alignment

Srinvias Aluru [1] defines an alignment of a set of sequences ($\bar{s_1}$, $\bar{s_2}$, ..., $\bar{s_N}$) as a correspondence among elements, ${}^{s'_{n_i}}s$, one taken from each sequence, allowing for the absence of correspondents but keeping the order within each sequence. Where s_n is a string that represent a DNA or protein sequence; consequently, the s_n alphabets can be $\bar{\Sigma} = \{A, C, G, T\}$ or $\Sigma = \{the\ 20\ amino\ acids\}$, respectively. When $N = 2$, an alignment is called a pairwise sequence alignment (PSA), but if $N \geq 3$ is named a multiple sequence alignment (MSA). An alignment can be represented by a rectangular matrix $A = \{a_{nl}\}(1 \leq n \leq N, 1 \leq l \leq L)$ over $\Sigma' = \Sigma \cup \{-\}$, where a dash '-' denotes a null implying the absence of the correspondent. One or more contiguous nulls in a row is called a gap or an indel (insertion-deletion). As a long gap may be produced by a single evolutionary event, gaps and nulls refer to related yet distinct entities.

Given that the number of feasible alignments grows exponentially concerning the number of sequences and their length, the MSA objective is to find an alignment that maximizes its quality. The alignment quality is measured by an alignment score that generally takes the form of Equation 1, considering an MSA of N sequences (A), where $S_N(\mathbf{a}_l)$ indicates the similarity score assigned to the column vector $\mathbf{a}_l$[13,3,12], $G_N(*_l)$ refers the gap penalty given to column l, and the subscript N denotes the number of sequences involved.

$$H_N(\mathcal{A}) = \sum_{1 \leq l \leq L} \{S_N(\mathbf{a}_l) - G_N(*_l)\} \tag{1}$$

3. Simulated Annealing for MSA

Simulated Annealing (SA) [9] is an stochastic optimization algorithm that models the physical process of heating material. SA evolves by a sequence of changes between states generated by transition probabilities, which are calculated involving the current temperature. Therefore, SA can be modeled mathematically by Markov chains and avoids getting stuck prematurely at a local optimum by applying the Boltzmann probability. We select SA to solve MSA because this algorithm solved efficiently several NP-hard optimization problem in different real-world domains [7], including Bioinformatics [14]. To adapt SA to the MSA domain and obtain MSASA algorithm (MSA solved by SA), we represent an alignment as a state (or solution, S_i), define specific-domain move operators to generate a new state (S_j also called neighbor solution), and specify a scoring system to measure the state energy.

Alignment representation. To represent an alignment as a solution, we use a rectangular matrix $\bar{A}$ defined in Section 2, where the total number of sequences is N and the length of the longest sequence is L, including gaps. In Figure 1) five sequences, s_n, are aligned in a matrix A with $N = 5$ and $L = 9$.

	a_{n1}	a_{n2}	a_{n3}	a_{n4}	a_{n5}	a_{n6}	a_{n7}	a_{n8}	a_{n9}
$\overline{s_1}$	M	V	L	Y	C	D	-	-	-
$\overline{s_2}$	*V*	*C*	*D*	*E*	*F*	*Y*	*V*	*D*	*Y*
$\overline{s_3}$	*Y*	*F*	*A*	*A*	*D*	-	-	-	-
$\overline{s_4}$	*A*	*C*	*D*	*Y*	*C*	*V*	-	-	-
$\overline{s_5}$	*A*	*C*	*D*	*F*	-	-	-	-	-

Fig.1. Alignment of five sequences, A = $\{a_{nl}\}(1 \leq n \leq 5, 1 \leq l \leq 9)$.

InDelGap, a specific-domain move operator. InDelGap operator is based on the insertion operator for permutation representations and adapted to the MSA domain. InDelGap generates a new state from a previous one by inserting or deleting a gap in a sequence, s_n. At beginning, the operator uniformly selects either an insertion or deletion operation. Next, a sequence s_n is picked at random, which must contain at least one gap if the operation is a deletion. Then, a position is randomly chosen within s_n, and a gap is inserted or deleted at that location. Finally, the InDelGap operator ensures that all rows in A reach the new maximum length, L, by filling any shorter rows with gaps.

Scoring sytems. Measuring the state energy means evaluating the alignment quality through an MSA score, H_N(A). In this work, we compute the score using Equation 2, which sums the score calculated for each sequence pair. In this sense, we employ three different heuristics to compute the score for a sequence pair: maximum conservation (MC), substitution matrix (SM), and variability (Vb). MC penalizes the pairs of residuals that are not equal. In this way, MC tries to minimize the difference in residues per column and to propend to high conservation in each column. SM uses the *BLOSUM62* matrix to obtain the score between two residues. Consequently, biological context is incorporated into the alignment evaluation. Vb is Similar to MC but allows more variability per column since no penalty is applied when differences in pairs of residuals appear.

$$H_N(\mathcal{A}) = \sum_{1 \leq i \leq N} \sum_{1 \leq j \leq N} \{score(s_i, s_j)\} \quad (2)$$

4. Comparison methodology

To compare various algorithms that perform MSA in an automated and reproducible way, we propose a method that includes a set of tasks for executing each algorithm, utilizing test cases with varying characteristics, and applying tools to assess alignment quality, as is described in the following paragraphs.

Algorithm Selection. For selecting the algorithms for comparison in an unbiased way, we propose choosing the algorithms more cited in the literature. For this work, we use the five most cited MSA software systems, according to Google Scholar in May 2023: Clustal Omega, MUSCLE, KAlign, T-Coffe, and MAFFT.

Sequence Database Selection. To determine a representative database of test sequences, we also recommend using a benchmark widely accepted by the Bioinformatic community to evaluate and compare different MSA software. As a

consequence, we select BAli BASE (Benchmark Alignment dataBASE) [2] that contains high-quality, manually constructed multiple sequence alignments together with detailed annotations, whose alignments are all based on threedimensional structural superpositions and include linear motifs. Moreover, this database is organized into ten reference groups of sequences with different degrees of similarity and complexity.

Scenario Configuration. Configuring the test scenarios from the chosen sequence database is imperative to select the appropriate test cases. We develop the test scenarios based on three dimensions that are crucial in determining the accuracy and effectiveness of the research: *i)* sequence length, short (< 100 residues), medium (≥ 100 and < 400 residues), and long (≥ 400);*ii)* the number of sequences to align, low (< 100 sequences) and high (≥ 100 sequences); and *iii)* the identity percentage of the reference alignment, which is set in 30% as a general rule by the Bioinformatic community. Therefore, 23 test cases with diverse characteristics and complexity levels are used for this comparison.

Metric Selection. Selecting the metrics to compare the results obtained by each MSA software is essential to analyze their performance from different points of view. Because of this, we use metrics provided by different origins: *i) Sumof-pairs* (SP) and *Column Score* (CS) from BAli Score [2]; *ii)* percentages of identity and coverage from Mumsa [11]; and *iii)* the transitive consistency score (TCS) [5] from T-Coffee.

Execution Environment Configuration. To ensure the same execution environment in an algorithmic comparison, we configure and use virtual machines with the same initial hardware and software conditions. Furthermore, since we are working with stochastic algorithms or components (such as MSASA, Clustal Omega, and MAFFT), we need 30 independent runs for each test case and algorithmic configuration. The experimentation is carried out by using the NextFlow programming language, which provides a framework for creating workflows [4], as shown the Fig. 2, and allows to collect the results in a structured and automatic way, facilitating their subsequent analysis.

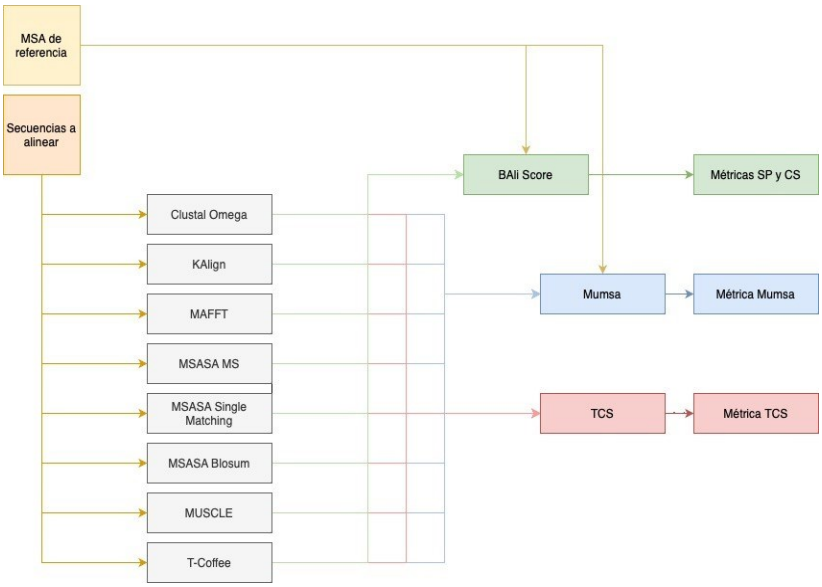

Fig.2. Experimentation Workflow.

5. Discussion and future works

In this research phase, we can study some preliminary results of the whole experimentation, as shown the Fig. 3. The CS metric values show that the five most cited MSA algorithms outperform our proposal, indicating that MSASA must be improved by tuning the move operator. Another future research work involves identifying the most appropriate MSA algorithm according to the characteristics of sequences to align.

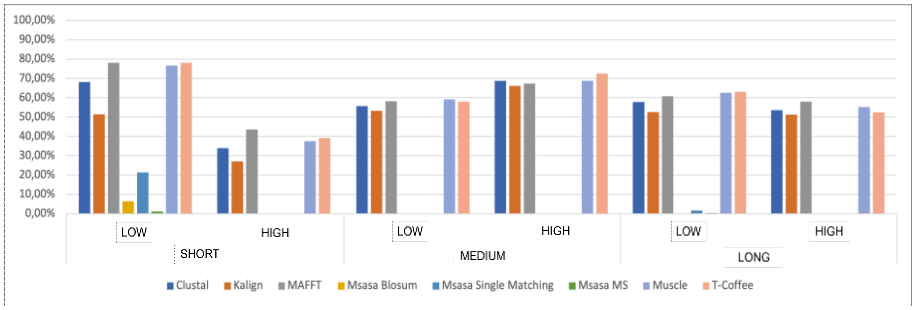

Fig.3. CS metrics values for the results obtained by each MSA algorithm grouped by number of sequences and sequence lengths.

References

1. Aluru, S.: Handbook of Computational Molecular Biology (Chapman & All/Crc Computer and Information Science Series). Chapman Hall/CRC (2005)
2. Bahr, A., Thompson, J., Thierry, J., Poch", O.: Balibase (benchmark alignment database): enhancements for repeats, transmembrane sequences and circular permutations. Nucleic Acids Res. 29, "323–326" (2001)
3. Carrillo, H., Lipman, D.J.: The multiple sequence alignment problem in biology. SIAM Journal on Applied Mathematics 48(5), 1073–1082 (1988)
4. Di Tommaso, P., Chatzou, M., Floden, E.W., Barja, P., Palumbo, E., Notredame, C.: Nextflow enables reproducible computational workflows. Nature Biotechnology 35, 316–319 (04 2017)
5. Di Tommaso, P., Moretti, S., Xenarios, I., Orobitg, M., Montanyola, A., Chang, J.M., Taly, J.F., Notredame, C.: T-coffee: a web server for the multiple sequence alignment of protein and rna sequences using structural information and homology extension. Nucleic acids research 39(Web Server issue), W13–W17 (2011)
6. Edgar, R.C.: Muscle: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research 32(5), 1792–1797 (2004)
7. Hernández, J., Salto, C., Minetti, G., Carnero, M., Sánchez, M.: Hybrid simulated annealing for optimal cost instrumentation in chemical plants. Chem. Eng. Trans. 74, 709–714 (2019)
8. Katoh, K., Standley, D.M.: MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Molecular Biology and Evolution 30(4), 772–780 (2013)
9. Kirkpatrick, S., Jr, C.G., Vecchi, M.: Optimization by simulated annealing. Science (220), 671–680 (1983)
10. Lassmann, T.: Kalign 3: multiple sequence alignment of large data sets. Bioinformatics 36(6), 1928–1929 (2019), advance online publication
11. Lassmann, T., Sonnhammer, E.: Kalign, kalignvu and mumsa: Web servers for multiple sequence alignment. Nucleic acids research 34, W596–9 (08 2006)
12. Pevzner, P.: Computational molecular biology: An algorithmic approach. The MIT Press (2000)
13. Sankoff, D., Cedergren, R.J.: Simultaneous comparison of three or more sequences related by a tree. In: Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Addison-Wesley, Reading, MA (1983)
14. Sarıyer, O.S., Güven, C.: Sequence alignment using simulated annealing. Physica A: Statistical Mechanics and its Applications 389(15), 3007–3012 (2010)
15. Sievers, F., Higgins, D.G.: Clustal omega, accurate alignment of very large numbers of sequences. Methods in Molecular Biology (Clifton, N.J.) 1079, 105–116 (2014)