Neutrosophic Sets and Systems, Vol. 61, 2023
University of New Mexico

Harnessing Tree Soft Set and Soft Computing Techniques' Capabilities in Bioinformatics: Analysis, Improvements, and Applications

Mona Gharib[1], Fatima Rajab[1] and Mona Mohamed[2]
[1]Department of Mathematics, Faculty of Science, Zagazig University, 44519 Zagazig, Egypt.
E-mail: monagharib@zu.edu.eg; monagharib@yahoo.com
[2]Higher Technological Institute, 10th of Ramadan City 44629, Egypt.
E-mail: mona.fouad@hti.edu.eg.

Abstract

A crucial aspect of bioinformatics is sequence comparison, which entails matching recently discovered biological sequences with previously identified sequences kept in databases. To find similarities between two or more nucleotide or amino acid sequences, sequence alignment organizes the sequences. Understanding the functional, structural, and evolutionary links between the sequences is made easier by looking at these areas of commonality. This study highlighted types of alignment. Also, proposed an effective methodology for deciding which algorithm can be utilized and satisfying the objective. Hence, Multi-Criteria Decision-Making (MCDM) techniques have been harnessed with Neutrosophic theory as a supporter in uncertain situations. Herein Single Value Neutrosophic Sets (SVNSs) as a type of uncertainty theory-Neutrosophic. This process requires a set of criteria leveraged in judgment. Also, Tree Soft Sets (TrSS) are applied for the first time to model the required criteria to facilitate the decision process. The hybrid techniques are applied to support stakeholders in making optimal decisions for optimal alignment algorithms among various algorithms such as pairwise and sequence algorithms. The results of the implementation of this decision technique indicated that multiple sequence alignment is the best compared with pairwise algorithms. Thus, we implemented multiple sequences in our study and employed logic programming to perform sequence matching. To ensure optimal alignment, the approach is tested on different sets of 16S rRNA gene of Actinobacteria (Streptomyces) sequences taken from NCBI. Then, the results are compared with MEGA.

Keywords: Bioinformatics; Multiple sequence alignment; Logic Programming; Multi-Criteria Decision-Making (MCDM); Single Value Neutrosophic Sets (SVNSs); Tree Soft Sets (TrSS)

1. Introduction

All living organism cells are composed of genetic codes that are passed from one generation to another. This is the reason for some living organisms are biologically similar and some are distinct. The genetic code can be represented as a sequence of alphabets, such as four base pairs of DNA and RNA, or twenty amino acids of protein [1]. These sequences are called biological sequences and over time a lot of changes called mutations occur in these sequences.

The field of Bioinformatics aims to align many biological sequences to derive their evolutionary relationships through comparative sequence analysis.

Bioinformatics applies computations to biological sequences to analyze and manipulate them. Sequence alignment (SA) is the most basic and essential module of computational bioinformatics and has varied applications in sequence assembly, sequence annotation, structural and functional prediction, and evolutionary or phylogeny relationship analysis.

SA is a field of research that focuses on the development of tools for comparing and finding similar sequences of (RNA, DNA, or amino acids) base pairs with the help of computers. The degree of similarity is used to measure gene and protein homology, classify genes and proteins, predict biological function, secondary and tertiary protein structure, detect point mutations, construct evolutionary trees, etc.

This study works into two phases. The first one is analyzing and examining existing alignment algorithms for deciding and utilizing optimal ones. In this phase, we are volunteering a combination of effective techniques to achieve the phase's objective. These techniques are utilized for preferencing and prioritizing SA alternatives based on a set of criteria. Hence, MCDM techniques are one of the utilized techniques in our study. Due to the ability of MCDM to treat this circumstance. TrSS model is volunteered for modeling the determined criteria and clarifying the relationship between these criteria. MCDM has been boosted by SVNSs in opacity circumstances Second phase: the results of phase one received by phase two to apply as optimal alternative for alignment. According to the results of the first phase, multiple sequences are the optimal alternative which applies for alignment.

Accordingly, we developed an algorithm that applied a logic program to align multiple biological sequences. SWI-Prolog (*http://www.swi-prolog.org*) is used to implement our proposed algorithm. Furthermore, we apply our implemented algorithm on eight different sets of 16S rRNA gene of Actinobacteria (Streptomyces) sequences: Seq1, Seq2, Seq3, Seq4, Seq5, Seq6, Seq7, and Seq8, were collected from GenBank at National Center for Biotechnology Information (NCBI). Also, we will use MEGA (Molecular Evolutionary Genetic Analysis Software for microcomputer), available at (*http://www. megasoftware.net*) to align the selected eight sequences. Each sequence set will be aligned using both methods fifty times and the execution times for all the fifty runs will be averaged.

Based on the average execution time, we compare the two methods to see which method reduces the execution time, speeds the performance, and decreases the memory location used to make the sequence alignment.

The objective of this study is summarized into several points:

1. Conducting surveys for prior studies and perspectives related to our scope.
2. Next, the results of the previous step entailed determining the effective and popular algorithms for alignment and we treated them as alternatives (Alts).

3. Leveraging decision techniques such as MCDM, SVNSs, and TrSS model to analyze the alternatives based on determined aspects and recommend the optimal.
4. We employ the recommended alternative to implement in our study.
5. We are observing the results of implementing the recommended algorithm and discussing it in the results and discussion section.

The outline of this study is as follows: Section 2 reviews the literature related to sequence alignment. The methodology used for sequence alignment of two methods is discussed in Section 3. Experimental results and their discussions are presented in Section 4. Finally, Section 5 discusses the obtained results. Finally, our conclusion of the study is represented in Section 6.

2. Prior Perspectives: Theoretical background related to our scope.

In this section, we conducted surveys for prior studies that embraced our notion. Firstly, we exhibited the principles for the concept alignment by showcasing its types and branches. Secondly, we collected the previous perspectives and studies from other scholars.

2.1 Comprehensive Visions for Sequence Alignment

A biological sequence is a sequence of characters from an alphabet. For DNA sequence, the character alphabet is {A, C, G, T}, for RNA sequence, the alphabet is {A, C, G, U}, and for RNA sequence is composed of A, C, G, U. For protein sequence, character set is {A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V}. Sequence alignment is the process of identifying one-to-one correspondence among subunits of sequences to measure the similarities among them. These similar regions provide functional, structural, and evolutionary information about the sequences under study. Aligned sequences are generally represented as rows within a matrix. Gaps (‘-‘) are inserted between the characters so that identical or similar characters are aligned in successive columns. Gaps represent the insertion of a character in or a deletion of a character from a biological sequence. Sequence alignment of two biological sequences is called pair-wise sequence alignment, and in case more than two biological sequences are involved, it is called multiple sequence alignment [2]. The sequence alignment is divided into:

2.1.1 Global Alignment

Closely related sequences which are of the same length are very much appropriate for global alignment. Here, the alignment is carried out from the beginning till the end of the sequence to find out the best possible alignment as in Figure 1

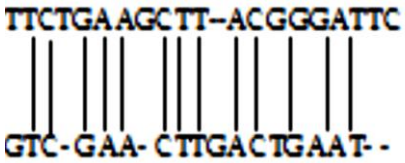

Fig. 1: Global Alignment of two biological sequences

2.1.2 Local Alignment

Sequences that are suspected to have similar or even dissimilar sequences can be compared with the local alignment method. It finds local regions with a high level of similarity as in Figure 2.

Pairwise Sequence Alignment is used to identify regions of similarity that may indicate functional, structural, and/or evolutionary relationships between two biological sequences (protein or nucleic acid). This type of alignment is based on numbers. Multiple sequence alignment (MSA) is the alignment of three or more biological sequences of similar length and therefore it is included in the alignment based on numbers. From the output of MSA applications, homology can be inferred and the evolutionary relationship between the sequences can be studied.

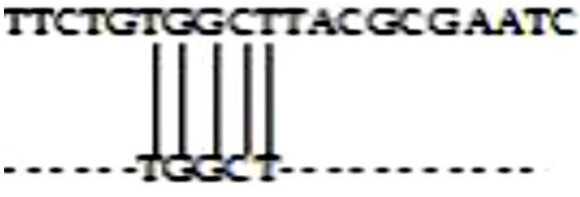

Fig. 2: Local Alignment of two biological sequences

2.2 Comprehensive Related Studies

Biological sequences databases are growing exponentially resulting in extensive demands on the implementation of new fast and efficient sequence alignment algorithms. Most of the work in the sequence alignment field has been primarily intended to provide new fast and efficient alignment methods.

The Needleman-Wunsch algorithm [3] employs a global alignment on two query sequences and is used widely in bioinformatics to align protein or nucleotide sequences. It uses a dynamic programming method to ensure the alignment is optimum by exploring all possible alignments and choosing the best.

While, the Smith–Waterman algorithm is a well-known algorithm for performing local sequence alignment that is for determining similar regions between two nucleotide or protein sequences

[4],[5]. Instead of looking at the total sequence, the Smith–Waterman algorithm compares segments of all possible lengths and optimizes the similarity measure.

In all the algorithms that had been proposed, the main objective of the researchers had been to apply different techniques to provide efficient alignment algorithms in terms of time and memory requirements.

Logic programming has been applied to develop logical databases to retrieve information about metabolic pathways, to identify and model genome structure [6] and to model protein interaction networks [7], [8]

3. Methodology: TrSS for Modelling Sequence Alignment Algorithms and Selection Procedures

Herein, we are leveraging the soft set notion represented in TrSS which was introduced by Smarandache. [9]. In TrSS we are clarifying and modeling various algorithms of sequence alignment (SA) into nodes at some levels. The purpose of modelling the determined algorithms into TrSS for make optimal decisions for selecting optimal and appropriate algorithms in our study. Hence, we are taking advantage of MCDM techniques and utilizing these techniques in the constructed tree to bolster us in making optimal decisions as clarified in the following steps:

Step 1: Construct a Tree and determine its nodes.

- ✓ At level 1: this level includes main aspects of sequence alignment {Matching Efficiency Node 1(N_1), Producing Phylogenetic Trees =Node2 (N_2), Prediction Efficiently= Node3 (N_3)}.
- ✓ At level 2: this level is divided into various branches based on previous branches of N_1, N_2 , and N_3 . Thereby, {Identify unknown sequence =$N_{1.1}$, Accuracy=$N_{1.2}$ } are considering sub-node of N_1. Also, {Finding out the relationship between the matched sequences = $N_{2.1}$, Easy of analyzing= $N_{2.2}$}are considering sub-node of N_2 . Finally, {Predicting protein efficiently= $N_{3.1}$, Predicting gene locations efficiently= $N_{3.2}$ } are considering sub-node of N_3.

Step 2: Determining Influential Aspects.

- ✓ in this step, the crucial factor in decision-making is determining the influential factors which impact the decision process. In this study, the decision process is conducted on three main aspects and six sub-aspects.
- ✓ The role of MCDM techniques is starting to work. Herein, we are employing entropy as a technique of MCDM to analyze determined sequence alignment's aspects. For boosting entropy, we are merging Neutrosophic theory for bolstering entropy in ambiguous situations. This theory is proposed by Smarandache [10]. Due to the ability of neutrosophic to apply in indeterminacy situations as mentioned

in [11] through measuring possible degrees of membership as truth, false, also indeterminancy a. Hence, we are implementing SVNSs as in [12] as a type of Neutrosophic theory. The aspects' weights are derived from entropy analysis and these weights have been obtained through the following several steps.

Step 2.1: We had an encounter with three specialists in this field to prioritize the determined alternative algorithms through determined aspects in Figure 3.

Step 2.2: Resulted from the encounter with three Neutrosophic decision matrices for three specialists. These matrices formed as in Eq. (1)

$$X^{n} = \begin{pmatrix} Asp_{11}^{n} & Asp_{12}^{n} & \cdots & Asp_{1n}^{n} \\ \vdots & \ddots & & \vdots \\ Asp_{m1}^{n} & Asp_{m2}^{n} & \cdots & Asp_{mn}^{n} \end{pmatrix} \quad (1)$$

Where:

X^{n} indicated to prioritize each specialist − based decision matrix.

Step 2.3: Eq. (2) is employed for transforming neutrosophic matrices into crisp matrices.

$$\mathbf{s}(Q_{ij}) = \frac{(2 + Tr - Fl - In)}{3} \quad (2)$$

Where:

Tr, Fl, In refer to truth, false, and indeterminacy respectively.

Step 2.4: Crisp matrices are amalgamated based on Eq.(3) into a single matrix so-called an aggregated decision matrix.

$$\partial_{ij} = \frac{(\sum_{j=1}^{N} Q_{ij})}{S} \quad (3)$$

Where:

Q_{ij} refers to the value of the criterion in the matrix, and S refers to the number of specialists.

Step 2.5: Eq. (4) is normalizing an aggregated matrix.

$$Nor_{ij} = \frac{\partial_{ij}}{\sum_{j=1}^{n} \partial_{ij}} \quad (4)$$

Where:

$\sum_{j=1}^{n} \partial_{ij}$ represents the sum of each aspect in an aggregated matrix per column.

Step 2.6: Entropy of the normalized matrix is computed through Eq. (5).

$$\mathrm{En}_{j=-h\sum_{i=1}^{n}\mathrm{Nor}_{ij}}\ \ln \mathrm{Nor}_{ij} \quad (5)$$

Where:

$$h = \frac{1}{\ln(\mathrm{Alts})} \quad (6)$$

Step 3: Reaching the optimal Decision for sequence alignment algorithm.

- ✓ This is the final step in the decision-making process, selecting the optimal algorithm between two SA algorithms. Alternative 1(Alt_1) = Pairwise alignment; Alternative 2(Alt_2) = Multiple SA algorithms.
- ✓ COPRAS is employed in this study as a technique of MCDM with hybridization of SVNs for ranking and prioritizing two Alts based on aspects and sub-aspects of SA. This process facilitates decision-making for optimal Alt. The hybridization process is implemented as follows:

  Step 3.1: Leveraging normalized matrix produced from previous step two and aspects' weights generated from entropy based on SVNSs to produce a weighted decision matrix through following Eq. (7).

$$\mathcal{B}_{ij} = w_i * \mathrm{Nor}_{ij} \quad (7)$$

Step 3.2: Eqs (8) and (9) are employed for computing the Sum of the weighted decision matrix.

$$S_{+i} = \sum_{j=1}^{n} \mathcal{B}_{+ij}, \text{ for beneficial criteria} \quad (8)$$

$$S_{-i} = \sum_{j=1}^{n} \mathcal{B}_{-ij}, \text{ for nonbeneficial criteria} \quad (9)$$

Step 3.3: the relative importance of alternatives is calculated based on Eq. (10).

$$Q_i = s_{+i} + \frac{s_{-\min}\ \sum_{i=1}^{m} s_{-i}}{s_{-i}\ \sum_{i=1}^{m}(s_{-m}/s_{-i})} \quad (10)$$

where I = 1, 2,…,m, and $s_{-m} = s_{-i}$ all aspects and sub-aspects are beneficial.

Step 3.4: quantity utility U_i for each Alt is based on Eq. (11) to rank Alts.

$$U_i = \left[\frac{Q_i}{Q_{\max}}\right] \times 100\% \quad (11)$$

4. Comprehensive Analysis

Herein, this section is divided into two sub-sections, each one responsible for exhibiting results and Consequent to each other. The first sub-section involving the results of the application of the methodology has been exhibited. The second sub-section is prepared based on the results of the first sub-section.

4.1 Analysis of Implementing Proposed Methodology

Herein, we discuss the results of implementing entropy-COPRAS under SVNSs based on TrSS. The resulting Alt as optimal SA is applied in this study.

4.1.1 Encounter with specialists: three specialists contributed to rating and prioritizing two Alts based on aspects and sub-aspects of SA which were modelled in the TrSS model.

4.1.2 Analyzing and obtaining weights for aspects and sub-aspects: this step involves two dimensions. First dimension, we obtain the main aspects' weights. The second dimension is obtaining sub-aspects' weights.

- **First dimension: Extracting the main aspects' weights Procedures**.

1. Three constructed neutrosophic decision matrices based on the SVNS scale which applied in [13] are transformed into crisp matrices based on Eq.(2).
2. These crisp matrices are amalgamated into the aggregated matrix by Eq. (3) as in Table 1.
3. Table 2 represents a normalized matrix based on Eq. (4).
4. Entropy for normalized matrix is calculated by Eq. (5) as in Table 3.
5. Final Aspects' weights are exhibited in Figure 3 through Eq. (6). This Figure indicates that main Aspect 1 outperforms main Aspect 2 and main Aspect 2.
6.

Table 1. An aggregated matrix of Aspects at level 1 for N1-N2

	$\mathbf{ASP_1}$	$\mathbf{ASP_2}$	$\mathbf{ASP_3}$
$\mathbf{Alt_1}$	0.226666667	0.594444444	0.777777778
$\mathbf{Alt_2}$	0.494444444	0.36	0.66

Table 2. Normalized matrix of Aspects at level 1 for N1-N2

	$\mathbf{ASP_1}$	$\mathbf{ASP_2}$	$\mathbf{ASP_3}$
$\mathbf{Alt_1}$	0.314329738	0.622817229	0.540958269
$\mathbf{Alt_2}$	0.685670262	0.377182771	0.459041731

Table 3. Entropy of Normalized matrix of Aspects at level 1 for N1-N2

	$\mathbf{ASP_1}$	$\mathbf{ASP_2}$	$\mathbf{ASP_3}$
$\mathbf{Alt_1}$	-0.363777805	-0.29490531	-0.3323719
$\mathbf{Alt_2}$	-0.258743457	-0.36776278	-0.3574164
$\sum_{i=1}^{m} \mathbf{X_{ij}}$	-0.622521262	-0.662668097	-0.689788258
$-h\sum_{i=1}^{m} \mathbf{X_{ij}} \ln X_{ij}$	0.568592766	0.540771009	0.521976737

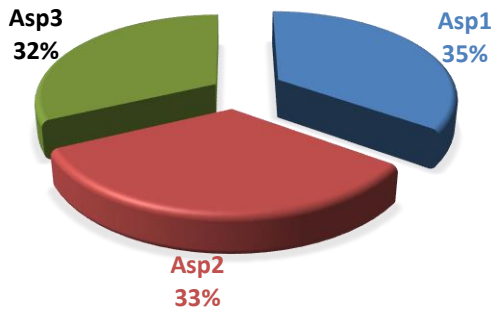

Fig 3. Weights of Main Aspects in Level 1 for N1-N2

- **Second dimension: Extracting Sub- aspects' weights Procedures**.

1. Three Neutrosophic decision matrices are constructed for {Sub-Asp $_{1.1}$, Sub-Asp $_{1.2}$ }; { Sub-Asp $_{2.1}$, Sub-Asp $_{2.2}$ }; { Sub-Asp $_{3,1}$, Sub-Asp $_{3.2}$ } and transformed into crisp matrices based on Eq.(2).
2. Eq.(3) is exploited for aggregating each pair of sub_Aspects into an aggregated matrix belonging to the main node (Aspect) at level 1.
3. Figure 4 indicates that sub_Aspect 1.1 outperforms sub_Aspect 1.2.
4. Figure 5 indicates that sub_Aspect 2.1 outperforms sub_Aspect 2.2.
5. Figure 6 indicates that sub_Aspect 3.1 outperforms sub_Aspect 3.2.

4.1.3 Ranking and prioritizing SA algorithms

1. Eq. (7) plays a critical role in the normalized matrix to generate a weighted decision matrix as in Table 4.
2. Eq. (8) is applied to obtain a sum weighted where all Aspects are beneficial
3. through Eq. (11), Quantity utility U_i for each alternative is calculated to rank the alternatives and results illustrated in Figure 7. Alt $_2$ (Multiple Alignment algorithm) is an optimal algorithm.

Table 4. Weighted decision matrix

	ASP_1	**ASP_2**	**ASP_3**
Alt_1	0.109557517	0.206456898	0.173089327
Alt $_2$	0.238985759	0.125031841	0.146878658

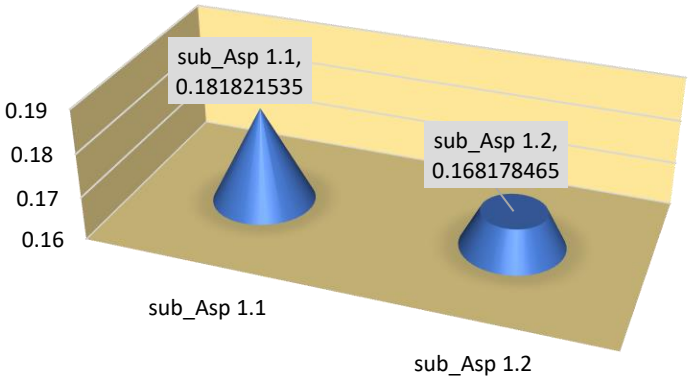

Fig.4. Final Weights of Sub Aspects 1.1 to 1.2 in Level 2

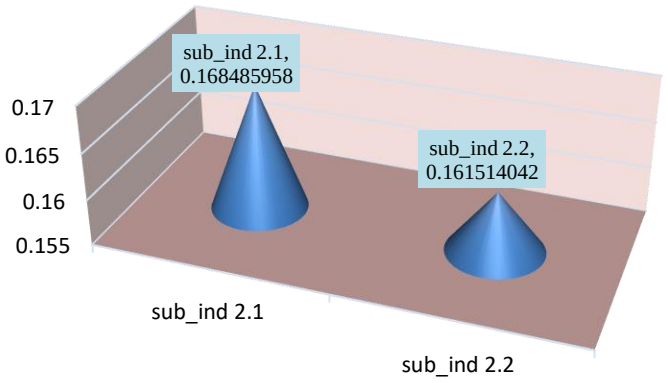

Fig.5. Final Weights of Sub Aspects 2.1 to 2.2 in Level 2

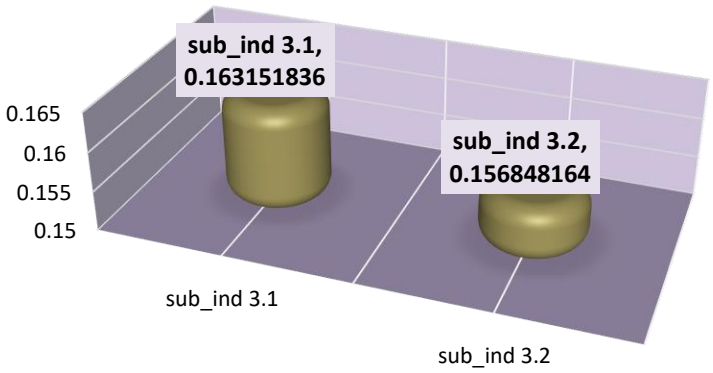

Fig.6. Final Weights of Sub Aspects 3.1 to 3.2 in Level 2

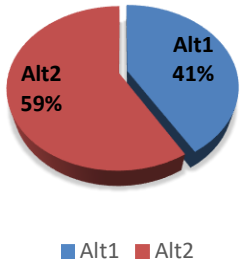

Fig. 7. Ranking two sequence algorithms

4.2 Analysis of Implementing Multiple Sequence Algorithm

Based on the results of the implementation of MCDM techniques under SVNS based on TrSS, the multiple sequence algorithm outperforms another algorithm. Hence, we used multiple sequences for aligning to

determine the similarity between the sequences, and then based on their degree of similarity the sequences were aligned.

In this study, we applied the developed algorithm given in [14]. The algorithm described an application for the logic programming paradigm for large-scale comparison of complete microbial genomes. We used SWI-Prolog language to implement our proposed algorithm. Where Prolog is a general-purpose logic programming language associated with artificial intelligence and computational linguistics.

4.2.1 Implementing the Algorithm

We have divided the implementation of the algorithm into three stages, the First stage, extracting genome information from GenBank, the Second stage, identifying homologous genes using BLAST [15], and the Third stage, alignment of homologous gene pairs using the Smith-Waterman software. The Smith-Waterman algorithm[16],[17] is a matrix-based dynamic programming technique to align two sequences. Smith–Waterman algorithm is a local sequence alignment; that is, for determining similar regions between two strings or nucleotide or protein sequences. Instead of looking at all the sequences, the Smith–Waterman algorithm only compares segments of all possible lengths and then optimizes the similarity measure.

4.2.2 Obtained Sequences from Genbank

Eight different sets of 16S rRNA gene of Actinobacteria (Streptomyces) sequences: Seq1, Seq2, Seq3, Seq4, Seq5, Seq6, Seq7, and Seq8, were collected from Genbank at NCBI (see Appendix). Identification of bacteria by using the molecular method (16S rDNA sequence) is more accurate than the traditional biochemical methods. The use of 16S rRNA gene sequences to study bacterial phylogeny and taxonomy has been by far the most common housekeeping genetic marker used for some reasons. These reasons include:

(i) It is present in almost all bacteria, often existing as a multigene family or operons.

(ii) The function of the 16S rRNA gene over time has not changed, suggesting that random sequence changes are a more accurate measure of time (evolution); and

(iii) The 16S rRNA gene (1,500 bp) is large enough for informatics purposes [18].

Details of the obtained sequence sets are listed in Table 5.

Table 5: Identification of Streptomyces.

Sequences	Streptomyces isolates	GenBank number	Base pair (bp)
Seq1	S. albidofuscus	Later name is S. pyridomyceticus BankIt1507621 JQ625331	900
Seq2	S. ambofaciens	BankIt1507642 JQ625332	703
Seq3	S. canarius	BankIt1507650 JQ625337	849
Seq4	S. chibaensis	Later name is S. corchorusii BankIt1507649 JQ625336	851
Seq5	S. coelicolor	BankIt1507648 JQ625335	944
Seq6	S. corchorusii	BankIt1507647 JQ625334	834
Seq7	S. nigrifaciens	Later name is S. flavovirens BankIt1507149 JQ625330	716
Seq8	S. parvullus	BankIt1507645 JQ625333	787

In this study, we will use MEGA to align the selected sequences. MEGA software is an integrated suite of tools for statistics-based comparative analysis of molecular sequence data based on evolutionary principles [19], [20]. MEGA is being used by biologists in a large number of laboratories for reconstructing the evolutionary histories of species and inferring the extent and nature of selective forces shaping the evolution of genes and species. Additionally, MEGA is used in many classrooms as a tool for teaching the methods used in evolutionary bioinformatics.

4.2.3 Results of Multiple Sequence

We have been extracting an algorithm that employs logic programming to measure the similarity of sequences. To guarantee the optimal alignment of the sequences we are using prolog language.

The algorithm is tested on various sets of real genome sequences taken from NCBI, and the processing time for the alignment process on these data sets has been computed.

To evaluate the performance of this approach, eight sets (Seq1, Seq2, Seq3, Seq4, Seq5, Seq6, Seq7, and Seq8) of 16S rRNA gene of Actinobacteria (Streptomyces) sequences have been used.

Data sets are used to find out the effect of varying the number of sequences being aligned on the processing time. The alignment of eight sequences by using MEGA is shown in Fig. 8.

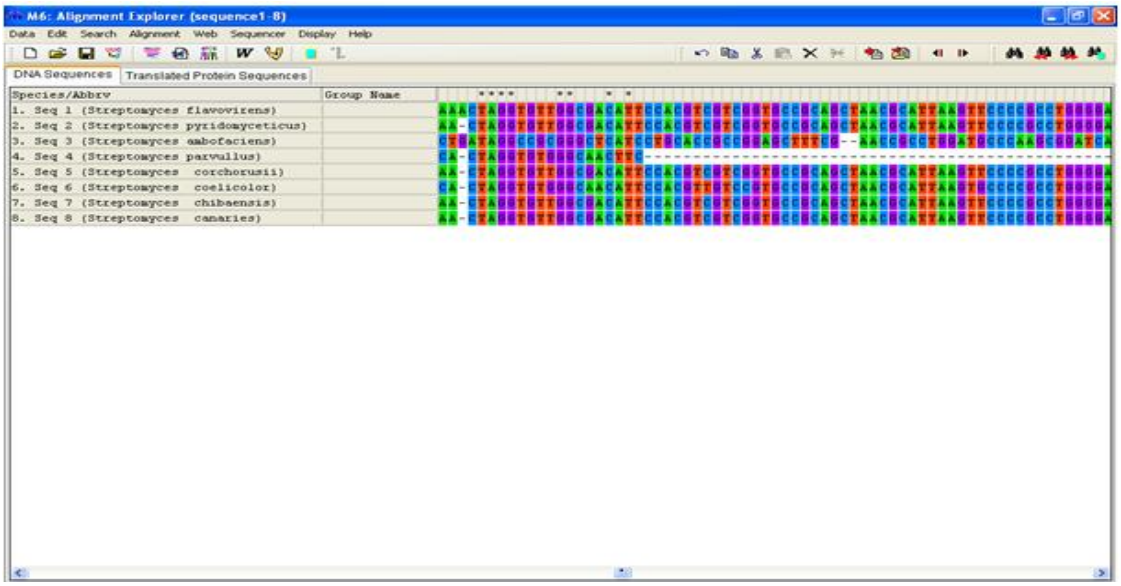

Fig. 8: Alignment of Eight Sequences.

To compare the amount of time needed to process the two methods of alignment being discussed, the processing time has been calculated. Each sequence set has been aligned using both methods fifty times and the execution times for all the fifty runs have been averaged. This average execution time has been used for the comparison. The average processing time for eight sets (Seq1, Seq2, Seq3, Seq4, Seq5, Seq6, Seq7 and Seq8) of 16S rRNA gene of Actinobacteria (Streptomyces) sequences are tabulated in Table 6 and Table 7 respectively.

Table 6: Average processing time (in seconds) for sequences Seq1, Seq2, Seq3, and Seq4.

Number of Sequences	Seq1 (Length:900 bp)		Seq2 (Length: 703 bp)		Seq3 (Length: 849 bp)		Seq4 (Length: 851 bp)	
	Logic Programming Method	MEGA Method	Logic Programming Method	MEGA Method	Logic Programming Method	MEGA Method	Logic Programming Method	MEGA Method
10	20.25	15.64	3.98	4.30	15.12	13.95	16.86	14.98
20	35.36	29.40	6.95	7.59	28.26	27.12	29.52	28.10
50	50.21	45.43	26.62	27.98	43.20	41.56	45.34	43.87
70	66.52	58.23	37.80	39.13	60.53	59.96	62.20	59.93
100	109.32	96.30	69.76	71.05	89.82	87.16	91.25	89.75
120	123.31	116.54	85.34	86.98	115.34	112.19	117.52	111.63
150	226.62	207.14	130.65	132.12	207.20	199.92	209.89	197.23

Table 7: The average processing time (in seconds) for sequences Seq5, Seq6, Seq7, and Seq8.

Number of Sequences	Seq5 (Length:944 bp)		Seq6(Length: 834 bp)		Seq7 (Length: 716 bp)		Seq8 (Length: 787 bp)	
	Logic Programming Method	MEGA Method	Logic Programming Method	MEGA Method	Logic Programming Method	MEGA Method	Logic Programming Method	MEGA Method
10	24.36	19.15	12.38	10.23	4.65	5.75	8.5	7.15
20	39.65	32.76	26.50	23.45	7.38	8.93	19.23	17.33
50	56.37	51.78	41.82	36.89	28.65	29.95	35.82	33.56
70	71.44	66.16	57.15	51.14	40.10	41.87	49.5	47.12
100	125.82	119.24	87.28	80.55	71.28	72.89	79.83	77.15
120	130.12	124.92	112.23	108.89	87.51	88.98	95.89	93.25
150	236.22	229.19	205.81	198.46	132.72	134.12	143.29	141.65

In the following, we give the line graph for the average processing time over fifty runs of both the methods on the eight sequences of (Seq1, Seq2, Seq3, Seq4, Seq5, Seq6, Seq7 and Seq8) of 16S rRNA gene of Actinobacteria (Streptomyces) in Fig. 9, Fig. 10, Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15 and Fig. 16 respectively.

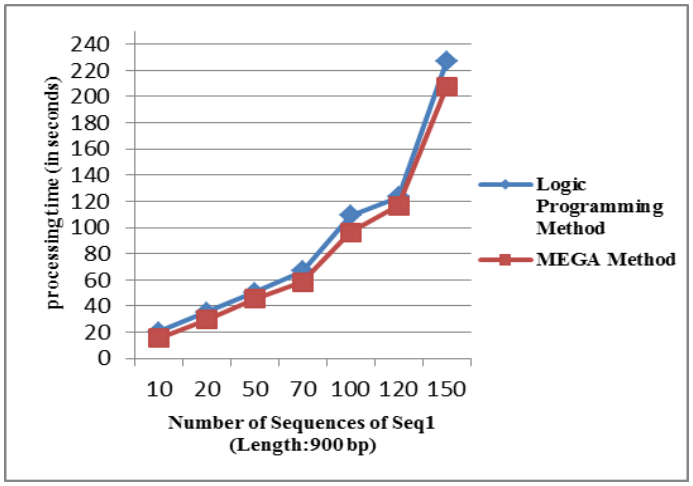

Fig. 9: Average processing time's line graph for Logic Programming and MEGA methods on Seq1.

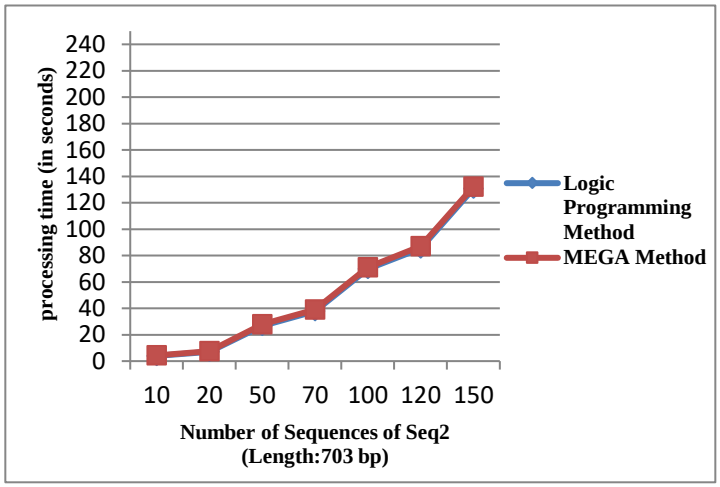

Fig. 10: Average processing time's line graph for Logic Programming and MEGA methods on Seq2.

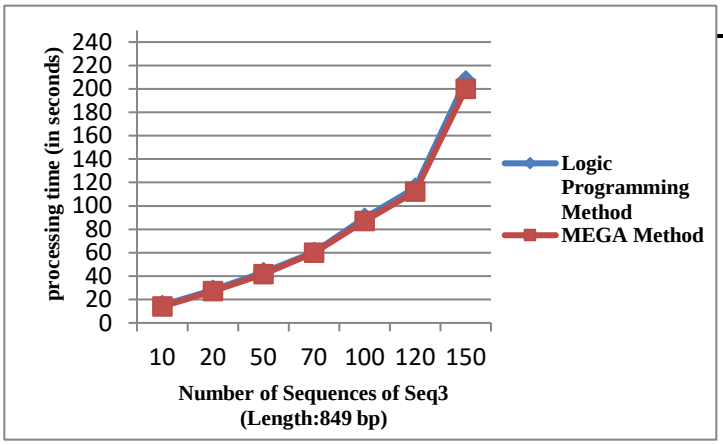

Fig. 11: Average processing time's line graph for Logic Programming and MEGA methods on Seq3

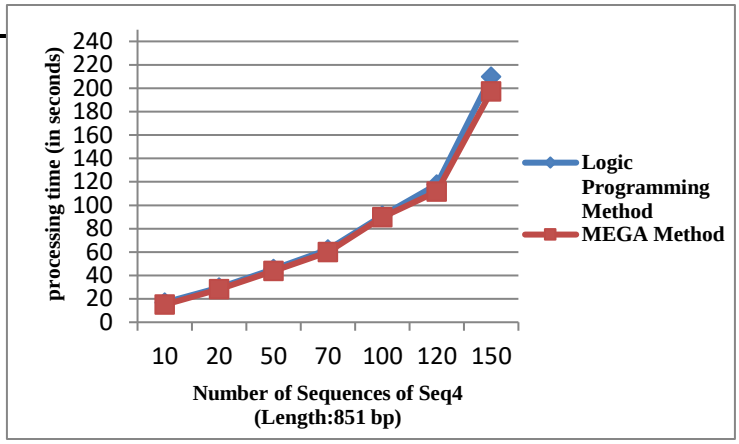

Fig. 12: Average processing time's line graph for Logic Programming and MEGA methods on Seq4

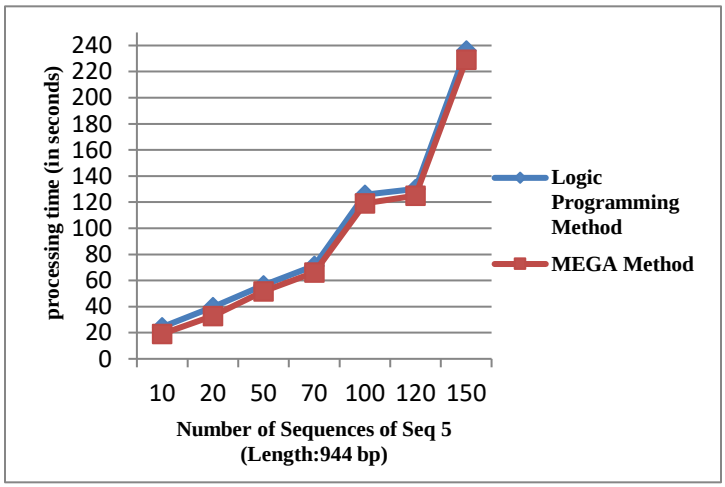

Fig. 13: Average processing time's line graph for Logic Programming and MEGA methods on Seq 5.

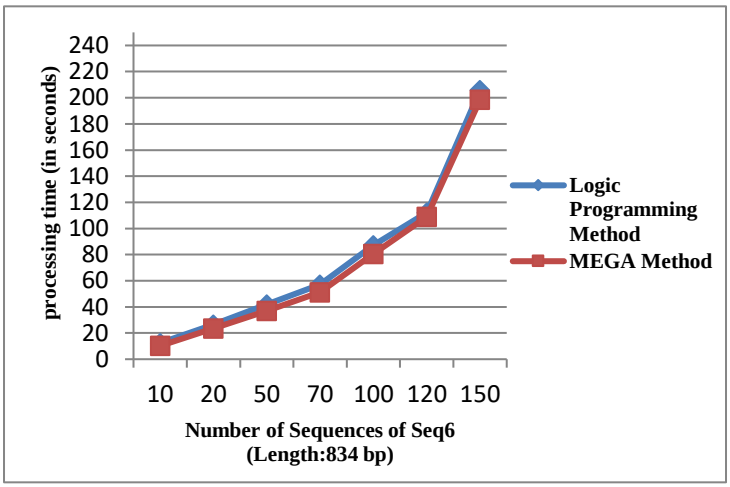

Fig. 14: Average processing time's line graph for Logic Programming and MEGA methods on Seq6

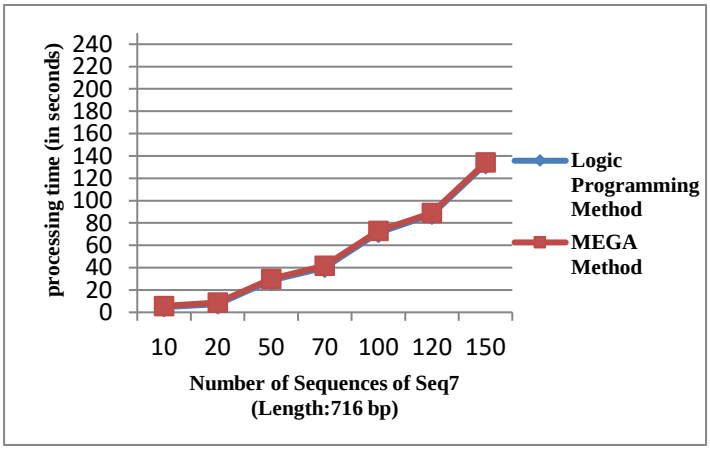

Fig. 15: Average processing time's line graph for Logic Programming and MEGA methods on Seq7.

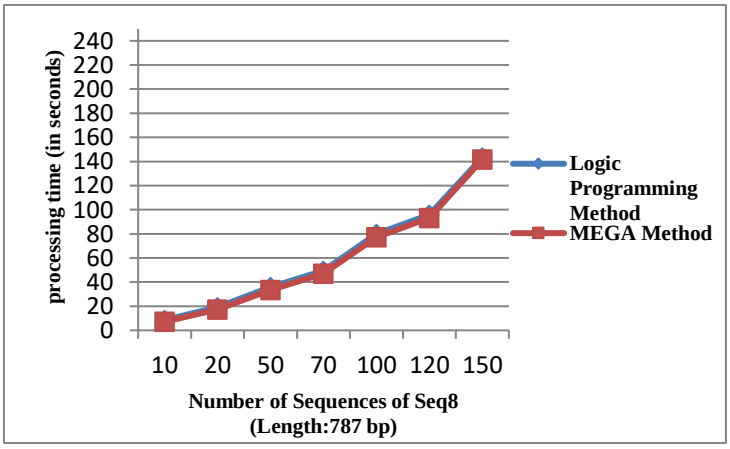

Fig. 16: Average processing time's line graph for Logic Programming and MEGA methods on Seq8.

5. Discussion

In this study, we leveraged the ability of MCDM techniques (i.e. entropy -COPRAS) under the authority of SVNSs for supporting MCDM in indeterminacy situations the objective of implementing these techniques is to recommend the optimal algorithm that we can utilize in our problem. The recommendation occurs based on a prioritizing process for a set of criteria/aspects and sub-aspects. Hence, we are modeling the decision-making process by using TrSS to express relationships between main aspects and sub-aspects. The results from the implementation of these techniques in the decision process indicated that multiple sequence algorithms in contrast to pairwise algorithms. Thus, we are implementing multiple sequences in our study. The experiments of applying multiple sequences for data sets in Table 2 and Table 3 show the effect of variation in the number of sequences on the processing time of the two alignment methods. From the processing times of sequences Seq2 and Seq7 in Table 5 and Table 6, we obtain that the processing time of the Logic Programming method takes less time as compared to the MEGA method for the sequences of length in the range 703-716 bp.

From the processing times of sequences Seq1, Seq3, Seq4, Seq5, Seq6, and Seq8 in Table 2 and Table 3, we obtain that the processing time of the Logic Programming method is higher than the MEGA method of sequences of length 787 – 944 bp.

From the obtained experimental results, we conclude that if the number and length of involved sequences are large, the Logic Programming method is very inefficient. Furthermore, we have that the Logic Programming method outperforms the MEGA method if the length of involved sequences is in the range 703-716 bp.

6. Conclusion

One of the most important steps in comparing biological sequences is thought to be sequence alignment. To find similarities between two or more nucleotide or amino acid sequences, sequence alignment organizes the sequences. Understanding the functional, structural, and evolutionary links between the sequences is made easier by looking at these areas of commonality.

Hence, utilizing suitable SA is critical. Herein, we discussed the methodology for selecting an optimal algorithm to perform the task of alignment. We utilized TrSS for the first time for modeling the aspects which contributed to the selection process. Also, MCDM worked with SVNSs to serve our objective. These techniques recommended multiple alignments for the alignment process.

References

[1] Mabrouk, M. S., Hamdy, M., Mamdouh, M., Aboelfotoh, M., & Kadah, Y. M. (2006). BIOINFTool: bioinformatics and sequence data analysis in molecular biology using Matlab. CAIRO INTERNATIONAL BIOMEDICAL ENGINEERING CONFERENCE.

[2] Mount, D. W. (2004). Bioinformatics: sequence and genome analysis.
[3] Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, *48*(3), 443–453.
[4] de Carvalho Junior, S. A. (2003). Sequence alignment algorithms. *Master's thesis. department of computer science, school of physical sciences, engineering. niig college, londol*.
[5] Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, *147*(1), 195–197.
[6] Videla, S., Guziolowski, C., Eduati, F., Thiele, S., Grabe, N., Saez-Rodriguez, J., & Siegel, A. (2012). Revisiting the training of logic models of protein signaling networks with asp. *Computational methods in systems biology: 10th international conference, cmsb 2012, london, uk, october 3-5, 2012. proceedings* (pp. 342–361). Springer.
[7] Fayruzov, T., Janssen, J., Vermeir, D., Cornelis, C., & De Cock, M. (2011). Modelling gene and protein regulatory networks with answer set programming. *International journal of data mining and bioinformatics*, *5*(2), 209–229.
[8] Gharib, M., & Rajab, F. (2013). Modeling Protein Interaction Networks with Monotonic Answer Set Programming. *International journal of computer science & communication networks*, *3*(4), 218.
[9] Smarandache, F. (2023). New Types of Soft Sets: HyperSoft Set, IndetermSoft Set, IndetermHyperSoft Set, and TreeSoft Set. *International journal of neutrosophic science*, *20*(4), 58–64. DOI:10.54216/IJNS.200404
[10] Smarandache, F. (2014). *Neutrosophic Theory and Its Applications, Vol. I: Collected Papers*. Infinite Study.
[11] Mohamed, M. (2023). Toward Smart Logistics: Hybrization of Intelligence Techniques of Machine Learning and Multi-Criteria Decision-Making in Logistics 5.0. *Multicriteria algorithms with applications*, *1*(1), 42–57.
[12] Sallam, K. M., & Mohamed, A. W. (2023). Single Valued Neutrosophic Sets for Assessment Quality of Suppliers under Uncertainty Environment. *Multicriteria algorithms with applications*, *1*(1), 1–10.
[13] Mohamed, M., Sallam, K. M., & Mohamed, A. W. (2023). Transition Supply Chain 4.0 to Supply Chain 5.0: Innovations of Industry 5.0 Technologies Toward Smart Supply Chain Partners. *Neutrosophic systems with applications*, *10*, 1–11.
[14] Bansal, A. K., & Bork, P. (1999). Applying logic programming to derive novel functional information of genomes. *International symposium on practical aspects of declarative languages* (pp. 275–289). Springer.
[15] Altschul, S. F., Gish, W. M., & Miller, W. (n.d.). W. Myers, EW and Lipman. DJ 1990. *Basic local alignment search tool. j. mol. biol*, *215*, 403–410.
[16] Setubal, J. C., Meidanis, J., & Setubal-Meidanis, . (1997). *Introduction to computational molecular biology*. PWS Pub. Boston.
[17] Waterman, M. S. (2018). *Introduction to computational biology: maps, sequences and genomes*. Chapman and Hall/CRC.
[18] Patel, J. B. (2001). 16S rRNA gene sequencing for bacterial pathogen identification in the clinical laboratory. *Molecular diagnosis*, *6*, 313–321.
[19] Kumar, S., Nei, M., Dudley, J., & Tamura, K. (2008). MEGA: a biologist-centric software for evolutionary analysis of DNA and protein sequences. *Briefings in bioinformatics*, *9*(4), 299–306.
[20] Tamura, K., Peterson, D., Peterson, N., Stecher, G., Nei, M., & Kumar, S. (2011). MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. *Molecular biology and evolution*, *28*(10), 2731–2739.

Appendix

The Eight sets sequences Seq1, Seq2, Seq3, Seq4, Seq5, Seq6, Seq7 and Seq8, are given as follows:

Seq 1

GTTGGTGGGGTGATGGCCTACCAAGGCGACGACGGGTAGCCGGCCTGAGAGGGCGACCGGCCACACTGG
GACTGAACACGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCGAAAGCCTGA
TGCAGCGACGCCGCGTGAGGGATGACGGCCTTCGGGTTGTAAACCTCTTTCAGCAGGGAAGAAGCGAAA
GGGACGGTACCTGCAGAAGAAGCCTTCTTGAATAACTACGTGCCAGCAGCCGCGGTAATACGTAGGGCG
CAAGCGTTGTCCGGAATTATTGGGCGTAAAGAGCTCGTAGGCGGCTTGTCACGTCGGATGTGAAAGCCC
GGGGCTTAACCCCGGGTCTGCATTCGATACGGGCTAGCTAGAGTGTGGTAGGGGAGATCGGAATTCCTG
GTGTAGCGGTGAAATGCGCAGATATCAGGAGGAACACCGGTGGCGAAGGCGGATCTCTGGGCCATTACT
GACGCTGAGGAGCGAAAGCGTGGGGAGCGAACAGGATTAGATACCCTGGTAGTCCACGCCGTAAACGTT
GGGAAACTAGGTGTTGGCGACATTCCACGTCGTCGGTGCCGCAGCTAACGCATTAAGTTCCCCGCCTGG
GGAGTACGGCCGCAAGGCTAAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCAGCGGAGCATGTGGC
TTAATTCGACGCAACGCGAAGAACCT

Seq 2

CGCATGGGGGTTGGTGTAAAGCTCCGGCGGTGCAGGATGAGCCCGCGGCCTATCAGCTTGTTGGTGGGG
TAATGGCCTACCAAGGCGACGACGGGTAGCCGGCCTGAGAGGGCGACCGGCCACACTGGGACTGAGACA
CGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCGAAAGCCTGATGCAGCGAC
GCCGCGTGAGGGATGACGGCCTTCGGGTTGTAAACCTCTTTCAGCAGGGAAGAAGCGCAAGTGACGGTA
CCTGCAGAAGAAGCACCGGCTAACTACGTGCCAGCAGCCGCGGTAATACGTAGGGTGCGAGCGTTGTCC
GGAATTATTGGGCGTAAAGAGCTCGTAGGCGGCCTGTCGCGTCGGATGTGAAAGCCCGGGGCTTAACCC
CGGGTCTGCATTCGATACGGGCAGGCTAGAGTGTGGTAGGGGAGATCGGAATTCCTGGTGTAGCGGTGA
AATGCGCAGATATCAGGAGGAACACCGGTGGCGAAGGCGGATCTCTGGGCCATTACTGACGCTGAGGAG
CGAAAGCGTGGGGAGCGAACAGGATTAGATACCCTGGTAGTCCACGCCGTAAACGTTGGGAACTAGGTG
TTGGCGACATTCCACGTCGTCGGTGCCGCAGCTAACGCATTAAGTTCCCCGCCTGGGGAGTACGGCCGC
AAGGCTAAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCAGCGGAGCATGTGGCTTAATTCGACAGA
CCAACGCGAAGAACCTTACCAAGGCTTGACATATACCGGAAACGGCTAGAGATAGTCGCCCCCTTGTGG
TCGGTATACAGGTGGTGCATGGTTGTCGTCAGCTCGTGTCGTGAGATGTTGGGTTAAGTCCCGCAACGA
GCG

Seq 3

GCTCCTCAGCGTCAGTATCGGCCCAGAGATCCGCCTTCGCCACCGGTGTTCCTCCTGATATCTGCGCAT
TTCACCGCTACACCAGGAATTCCGATCTCCCCTACCGAACTCTAGCCTGCCCGTATCGACTGCAGACCC
GGGGTTAAGCCCCGGGCTTTCACAACCGACGCGACAAGCCGCCTACGAGCTCTTTACGCCCAATAATTC
CGGACAACGCTCGCGCCCTACGTATTACCGCGGCTGCTGGCACGTAGTTAGCCGGCGCTTCTTCTGCAG
GTACCGTCACTTGCGCTTCTTCCCTGCTGAAAGAGGTTTACAACCCGAAGGCCGTCATCCCTCACGCGG
CGTCGCTGCATCAGGCTTGCGCCCATTGTGCAATATTCCCCACTGCTGCCTCCCGTAGGAGTCTGGGCC
GTGTCTCAGTCCCAGTGTGGCCGGTCGCCCTCTCAGGCCGGCTACCCGTCGTCGCCTTGGTGAGCCGTT
ACCTCACCAACAAGCTGATAGGCCGCGGGCTCATCCTGCACCGCCGGAGCTTTCGAACCGCCTGGATGC
CCAAGCGGATCAGTATCCGGTATTAGACCCCGTTTCCAGGGCTTGTCCCAGAGTGCAGGGCAGATTGCC
CACGTGTTACTCACCCGTTCGCCACTAATCCCCACCGAAGTGGTTCATCGTTCGACTTGCATGTGTTAA
GCACGCCGCCAGC

Seq 4

ACGAACGCTGGCGGCGTGCTTAACACATGCAAGTCGAACGATGAACCACTTCGGTGGGGATTAGTGGCG
AACGGGTGAGTAACACGTGGGCAATCTGCCCTGCACTCTGGGACAAGCCCTGGAAACGGGGTCTAATAC
CGGATACTGACCTTCACGGGCATCTGTGAAGGTCGAAAGCTCCGGCGGTGCAGGATGAGCCCGCGGCCT
ATCAGCTTGTTGGTGAGGTAATGGCTCACCAAGGCGACGACGGGTAGCCGGCCTGAGAGGGCGACCGGC
CACACTGGGACTGAGACACGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCG
AAAGCCTGATGCAGCGACGCCGCGTGAGGGATGACGGCCTTCGGGTTGTAAACCTCTTTCAGCAGGGAA
GAAGCGAAAGTGACGGTACCTGCAGAAGAAGCGCCGGCTAACTACGTGCCAGCACCGCGGTAATACGTA
GGGCGCAAGCGTTGTCCGGAATTATTGGGCGTAAAGAGCTCGTAGGCGGCTTGTCACGTCGGTTGTGAA
AGCCCGGGGCTTAACCCCGGGTCTGCAGTCGATACGGGCAGGCTAGAGTTCGGTAGGGGAGATCGGAAT
TCCTGGTGTAGCGGTGAAATGCGCAGATATCAGGAGGAACACCGGTGGCGAAGGCGGATCTCTGGGCCG
ATACTGACGCTGAGGAGCGAAAGCGTGGGGAGCGAACAGGATTAGATACCCTGGTAGTCCACGCCGTAA
ACGGTGGGCACTAGGTGTGGGCAACTTC

Seq 5

AGTGGCGGACGGGTGAGGAATACATCGGAATCTACCTTGTCGTGGGGGATAACGTCTGGAAACGGGGTC
TAATACCGGATACCACTCTCGCAGGCATCTGTGAGGGTTGAAAGCTCCGGCGGTGAAGGATGAGCCCGC
GGCCTATCAGCTTGTTGGTGAGGTAATGGCTCACCAAGGCGACGACGGGTAGCCGGCCTGAGAGGGCGA

CCGGCCACACTGGGACTGAGACACGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAAT
GGGCGAAAGCCTGATGCACGACGCCGCGTGAGGGATGACGGCCTTCGGGTTGTAAACCTCTTTCAGCAG
GGAAGAAGCGAAAGTGACGGTACCTGCAGAAGAAGCGCCGGCTAACTACGTGCCAGCAGCCGCGGTAAT
ACGTAGGGCGCAAGCGTTGTCCGGAATTATTGGGCGTAAAGAGCTCGTAGGCGGCTTGTCACGTCGGGT
GTGAAAGCCCGGGGCTTAACCCCGGGTCTGCATTCGATACGGGCTAGCTAGAGTGTGGTAGGGGAGATC
GGAATTCCTGGTGTAGCGGTGAAATGCGCAGATATCAGGAGGAACACCGGTGGCGAAGGCGGATCTCTG
GGCCATTACTGACGCTGAGGAGCGAAAGCGTGGGGAGCGAACAGGATTAGATACCCTGGTAGTCCACGC
CGTAAACGGTGGGAACTAGGTGTTGGCGACATTCCACGTCGTCGGTGCCGCAGCTAACGCATTAAGTTC
CCCGCCTGGGGAGTACGGCCGCAAGGCTAAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCAGCGGA
GCATGT

Seq 6

TATTGGGCGTAAAGAGCTCGTAGGCGGCTTGTCAGCGTCGGTTGTGAAAGCCCGGGGCTTAACCCCGGG
TCTGCAGTCGATACGGGCAGGCTAGAGTTCGGTAGGGGAGATCGGAATTCCTGGTGTAGCGGTGAAATG
CGCAGATATCAGGAGGAACACCGGTGGCGAAGGCGGATCTCTGGGCCGATACTGACGCTGAGGAGCGAA
AGCGTGGGGAGCGAACAGGATTAGATACCCTGGTAGTCCACGCCGTAAACGGTGGGCACTAGGTGTGGG
CAACATTCCACGTTGTCCGTGCCGCAGCTAACGCATTAAGTGCCCCGCCTGGGGAGTACGGCCGCAAGG
CTAAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCGGCGGAGCATGTGGCTTAATTCGACGCAACGC
GAAGAACCTTACCAAGGCTTGACATACACCGGAAACGTCTGGAGACAGGCGCCCCCTTGTGGTCGGTGT
ACAGGTGGTGCATGGCTGTCGTCAGCTCGTGTCGTGAGATGTTGGGTTAAGTCCCGCAACGAGCGCAAC
CCTTGTCCCGTGTTGCCAGCAAGCCCTTCGGGGTGTTGGGGACTCACGGGAGACCGCCGGGGTCAACTC
GGAGGAAGGTGGGGACGACGTCAAGTCATCATGCCCCTTATGTCTTGGGCTGCACACGTGCTACAATGG
CCGGTACAATGAGCTGCGATACCGCGAGGTGGAGCGAATCTCAAAAAGCCGGTCTCAGTTCGGATTGGG
GTCTGCAACTCGACCCCATGAAGTCGGAGTCGCTAGTAATCGCAGATCAGCATTGCTGCGGTGAATACG
TTCCCGGGCCTTGTACACACCGCCCGTCACGTCACGAAAGTCGGTAACACCCGAAGCCGGTGGCCCAAC
CCCTTGTGGGAGGGAGCTGTCGAAGTGGGACTGGCGATGGACGAGTC

Seq 7:

GGATGAGCCCGCGGCCTATCAGCTTGTTGGTGAGGTAACGGCTCACCAAGGCGACGACGGGTAGCCGGC
CTGAGAGGGCGACCGGCCACACTGGGACTGAGACACGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGA
ATATTGCACAATGGGCGNAAGCCTGATGCAGCGACGCCGCGTGAGGGATGACGGCCTTCGGGTTGTAAA
CCTCTTTCAGCAGGGAAGAAGCGAAAGTGACGGTACCTGCAGAAGAAGCGCCGGCTAACTACGTGCCAG
CAGCCGCGGTAATACGTAGGGCGCAAGCGTTGTCCGGAATTATTGGGCGTAAAGAGCTCGTAGGCGGCT
TGTCACGTCGGGTGTGAAAGCCCGGGGCTTAACCCCGGGTCTGCATTCGATACGGGCTAGCTAGAGTGT
GGTAGGGGAGATCGGAATTCCTGGTGTAGCGGTGAAATGCGCAGATATCAGGAGGAACACCGGTGGCGA
AGGCGGATCTCTGGGCCATTACTGACGCTGAGGAGCGAAAGCGTGGGGAGCGAACAGGATTAGATACCC
TGGTAGTCCACGCCGTAAACGGTGGGAACTAGGTGTTGGCGACATTCCACGTCGTCGGTGCCGCAGCTA
ACGCATTAAGTTCCCCGCCTGGGGAGTACGGCCGCAAGGCTAAAACTCAAAGGAATTGACGGGGGCCCG
CACAAGCAGCGGAGCATGTGGCTTAATTCGACGCAACGCGAAGAACCTTACCAAGGCTTGACATACACC
GGAAAACCCTGGAGACAGGGTCCCCCTTGTGGTCGGTGTACAGGTGGTGCATGGCTGTCGTCAGCTCGT
GTCGTGAGATGTTGGGTTAAGTC

Seq 8:

AGCTTGTTGGTGAGGTAACGGCTCACCAAGGCGACGACGGGTAGCCGGCCTGAGAGGGCGACCGGCCAC
ACTGGGACTGAGACACGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCGAAA
GCCTGATGCAGCGACGCCGCGTGAGGGATGACGGCCTTCGGGTTGTAAACCTCTTTCAGCAGGGAAGAA
GCGAAAGTGACGGTACCTGCAGAAGAAGCGCCGGCTAACTACGTGCCAGCAGCCGCGGTAATACGTAGG
GCGCAAGCGTTGTCCGGAATTATTGGGCGTAAAGAGCTCGTAGGCGGCTTTCACGTCGGGTGTGAAAGC
CCGGGGCTTAACCCCGGGTCTGCATTCGATACGGGCTAGCTAGAGTGTGGTAGGGGAGATCGGAATTCC
TGGTGTAGCGGTGAAATGCGCAGATATCAGGAGGAACACCGGTGGCGAAGGCGGATCTCTGGGCCATTA
CTGACGCTGAGGAGCGAAAGCGTGGGGAGCGAACAGGATTAGATACCCTGGTAGTCCACGCCGTAAACG
GTGGGAACTAGGTGTTGGCGACATTCCACGTCGTCGGTGCCGCAGCTAACGCATTAAGTTCCCCGCCTG
GGGAGTACGGCCGCAAGGCTAAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCAGCGGAGCATGTGG
CTTAATTCGACGCAACGCGAAGAACCTTACCAAGGCTTGACATACACCGGAAAACCCTGGAGACAGGGT
CCCCCTTGTGGTCGGTGTACAGGTGGTGCATGGCTGTCGTCAGCTCGTGTCGTGAGATGTTGGGTTAAG
TCCCGCAACGAGCGCAACCCT

Received: May 06, 2023. Accepted: Dec. 09, 2023