Received: 26 March 2025 | Revised: 22 January 2026 | Accepted: 30 January 2026 | Published online: 11 February 2026

RESEARCH ARTICLE

Medinformatics
2026, Vol. 00(00) 1–10
DOI: 10.47852/bonview62025784

BON VIEW PUBLISHING

A Hybrid Approach to Solve Multiple Sequence Alignment Problems Using Chaotic Metaheuristics

Gargi Nandi[1], Sweta Roy[1], Yeasmin Khatun[1], Nibedita Chakraborty[1], Shrestha Pal[1], Meheria Sultana Khatun[1] and Shouvik Chakraborty[1,*]

[1] *Department of Computer Science & Technology, Women's Polytechnic, India*

Abstract: Aligning multiple sequences of amino acids or nucleotides is considered a challenging task in biology, like fitting puzzle pieces together to find the best matches. Due to the huge computational overhead of checking all possible combinations, simple metaheuristic approaches, such as genetic algorithms (GAs) that are inspired by nature, are good because they can effectively optimize the gap positions to get better scores. This work addresses this problem by combining GAs with chaotic sequences to obtain a better diversity in the search space that leads to near-optimal alignment. Chaotic sequences are known for their unpredictable patterns but structured behavior, which helps explore different solutions effectively. Integrating chaos theory into metaheuristics helps in achieving effective and accurate alignments of multiple sequences by handling complexity, finding optimal alignment, and improving efficiency. Users can adjust hyperparameters such as mutation probability, crossover probability, etc., making the approach flexible. The experiments are carried out on various inputs that are obtained from the BAliBASE dataset to establish the effectiveness and the superiority of the proposed approach. The proposed approach is compared with the state-of-the-art approaches, and the obtained outcomes are promising enough and encouraging to apply the proposed approach to real-life problems.

Keywords: multiple sequence alignment, chaos, genetic algorithm, elitism, BAliBASE

1. Introduction

Multiple sequence alignment (MSA) is a fundamental task in bioinformatics with applications ranging from evolutionary analysis to structure–function prediction [1]. Despite its significance, MSA remains computationally challenging due to its NP-complete nature and the exponential growth of the search space with increasing sequence length and count [2]. Traditional alignment techniques, including progressive alignment and dynamic programming (DP), often fail to provide optimal alignments, especially in complex or large-scale datasets [3]. Although genetic algorithms (GAs) have been widely adopted to mitigate these limitations due to their global search capability, they are not immune to premature convergence and reduced population diversity, often leading to suboptimal solutions [4]. This manuscript addresses these challenges by proposing a hybrid approach that incorporates chaotic dynamics—specifically the Logistic map—into the core operations of GAs. By embedding chaos in population initialization, selection, crossover, and mutation, the model introduces deterministic randomness that enhances the exploration of the solution space while maintaining structured search behavior. The primary aim of this research is to improve the accuracy and robustness of MSA by overcoming the common pitfalls of conventional GAs. The methodology is evaluated using benchmark datasets from BAliBASE, and performance is assessed against established alignment techniques.

Key contributions of this work are summarized as follows:

- A simple metaheuristics-based MSA strategy is proposed.
- The traditional genetic algorithm (TGA) is modified and hybridized with the chaotic environment.
- Genetic operators are modified to handle the encoded gaps
- The chaotic Logistic map introduces an element of unpredictability into the GA's search process, enabling a more comprehensive exploration of the solution space.
- By using chaotic sequences for initializing populations and guiding mutation processes, the hybrid GA can generate more diverse and varied candidate solutions. This diversity is crucial for avoiding convergence on suboptimal alignments and improving the quality of the final alignment.
- The inherent randomness of the chaotic Logistic map helps mitigate the risk of premature convergence—a common issue in traditional GAs. By promoting exploration over exploitation, the hybrid approach increases the likelihood of discovering globally optimal or near-optimal alignments.

MSA has long been recognized as a fundamental problem in computational biology, serving as the basis for tasks such as phylogenetic analysis, protein structure prediction, and functional annotation. However, due to its NP-hard nature,

***Corresponding author:** Shouvik Chakraborty, Department of Computer Science & Technology, Women's Polytechnic, India. Email: shouvikchakraborty@ieee.org

obtaining optimal alignments for large and complex datasets remains computationally challenging. To overcome these limitations, researchers have increasingly turned to metaheuristic algorithms, which offer flexible and efficient strategies for exploring vast search spaces. Recent studies have applied diverse approaches—including GAs, particle swarm optimization, and bacterial foraging optimization—to enhance alignment accuracy and computational efficiency. This section also reviews these developments, highlighting key methodologies, innovations, and emerging trends in the use of metaheuristics for MSA. Issa et al. [5] provide an overview of recent advancements in MSA through the application of metaheuristic algorithms. Particular emphasis is placed on two emerging approaches. The first is the Fragmented Protein Sequence Alignment method, which employs a two-layer particle swarm optimization strategy. The second is an MSA framework that utilizes a multi-objective bacterial foraging optimization algorithm. Lajevardy et al. [6] introduced a GA combined with a specialized chromosome structure to address a mathematical model for MSA. The model establishes a foundation for deriving optimal solutions through various strategies, utilizing an X-mediated matrix composed of binary elements to represent the sequences. The approach is then implemented on a web-based platform using the GA framework, and the experimental results demonstrate the effectiveness of GA in solving the MSA problem. Chowdhury and Garai [7] introduce a novel GA–based alignment method called Bi-objective Sequence Alignment using Genetic Algorithm (BSAGA). The key innovation of this approach lies in its selection strategy: a portion of the population is chosen according to the sum-of-pairs (SP) score, while the remaining part is selected based on the Total Conserved Columns measure. To efficiently encode alignments, an integer-based chromosome representation is employed that specifies only the gap positions, thereby enhancing the search capability even for longer sequences. The proposed BSAGA method was evaluated using benchmark datasets such as BAliBASE and SABmark, with performance compared against existing alignment techniques. Experimental results, supported by the Wilcoxon signed-rank test, demonstrate that BSAGA achieves superior alignment accuracy relative to competing methods. Some comprehensive reviews on this topic can be found in [8–15].

In this work, we aim to improve the performance of MSA, a well-known NP-complete problem characterized by an exponentially large search space and complex trade-offs between alignment accuracy and computational cost. Traditional methods, including DP and progressive alignment, often struggle with scalability and are prone to suboptimal solutions when dealing with long or numerous sequences. While GAs offer a robust heuristic framework for navigating large solution spaces, they are frequently limited by issues such as premature convergence and loss of population diversity. To address these limitations, we propose a hybrid model that integrates chaotic dynamics—specifically the Logistic map—into the genetic framework. Chaotic sequences are deterministic yet highly sensitive to initial conditions, enabling a pseudorandom and diverse exploration of the search space. This integration enhances the exploratory capacity of the GA while maintaining structured randomness, helping avoid local optima and improving alignment accuracy. Our approach modifies key GA components such as population initialization, selection, crossover, and mutation using chaos theory principles, thereby introducing adaptive randomness to balance exploration and exploitation. Unlike conventional GAs, our model leverages chaotic behavior to maintain population diversity and search efficiency throughout the optimization process. This work thus fills a significant gap by combining the strengths of chaotic systems and evolutionary computation to develop a more effective and scalable solution for MSA.

The remainder of the manuscript is structured as follows: Section 2 presents relevant background and theoretical underpinnings of the GA and chaos theory. Section 3 details the proposed hybrid methodology, including algorithmic design and implementation. Section 4 discusses experimental results and comparative analyses. Finally, Section 5 concludes the paper with insights into the significance of the findings and outlines future research directions. This improved framework contributes to the growing body of intelligent bioinformatics tools and offers a promising direction for enhancing large-scale sequence analysis.

2. Methodology

In this work, the chaos theory and a GA are combined to address the challenge of the MSA task. In this section, some relevant background information is discussed.

2.1. Genetic algorithm

A GA is a computational method inspired by natural selection and was proposed by John Holland [16]. It starts with a set of initial possible solutions and then combines the best ones to make a new set of solutions. GAs can be thought of as a game where different solutions are explored on a trial-and-error basis to find the best one. The process starts by making some guesses and then mixing and matching them to make better guesses until we find the right answer. They work by evolving solutions over time, selecting the best ones, and combining them to produce even better solutions. It does this by combining good solutions and making small random changes (such as mutations) to create new, possibly better solutions [17]. For the sake of completeness, some relevant terminologies are explained as follows:

Population: A set of candidate solutions to the optimization problem. Each individual in the population represents a possible solution and is typically encoded as a string (chromosome).

Fitness Function: A function that evaluates how good a candidate solution is relative to others. It assigns a fitness score to each individual based on how well it solves the problem or achieves the desired objective.

Selection: The process of choosing individuals from the current population to create the next generation. Selection methods, such as roulette wheel selection or tournament selection, favor individuals with higher fitness scores, thereby increasing the likelihood of their genes being passed on.

Crossover (Recombination): A genetic operator used to combine the genetic material of two parent individuals to produce offspring. Crossover mimics biological recombination and aims to create new individuals with potentially better solutions by mixing characteristics from both parents.

Mutation: A genetic operator that introduces random changes to an individual's genes. Mutation helps maintain genetic diversity within the population and can prevent the algorithm from prematurely converging on suboptimal solutions.

2.2. The chaotic logistic map

The Logistic map is a mathematical function that is often used in chaos theory and dynamical systems. It can be used to generate chaotic sequences that introduce randomness into alignment algorithms or methods. A Logistics map refers to a

mathematical model used to simulate and optimize the alignment of multiple sequences. Applying Logistic maps, researchers can enhance the accuracy of aligning biological sequences, such as DNA or protein sequences. This helps in exploring diverse solution spaces and improving the outcomes of computational analyses in biological research. The Logistic map has applications in various fields, including physics, biology, etc. [18, 19].

The Logistic map is a mathematical function that exhibits chaotic behavior, particularly when its controlling parameter r is in the range [3.57, 4]. The Logistic map is mathematically defined in Equation 1:

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n) \tag{1}$$

where X_n represents the value of the variable at the$n$th iteration and r is a parameter known as the growth rate or chaotic factor, typically ranging from 0 to 4.

The behavior of the Logistic map is highly sensitive to the parameter r. For values of r between 0 and 1, the system converges to a fixed point. As r increases past 1, the system exhibits periodic behavior [20, 21]. Beyond a certain threshold, the system transitions into chaotic behavior, characterized by sensitive dependence on initial conditions and a random sequence of values. In the chaotic regime, the Logistic map exhibits a chaotic attractor, a set of values toward which the system evolves over time. The Lyapunov exponent, a measure of the average rate at which nearby trajectories converge or diverge, is positive in the chaotic regime, indicating sensitive dependence on initial conditions.

2.3. Integration of the chaos theory with genetic algorithm

Chaotic systems are deterministic, meaning their future behavior is entirely determined by their initial conditions [22, 23]. Tiny differences in initial conditions can lead to vastly different outcomes over time. Chaotic sequences are sequences of numbers that are generated with the help of some chaotic maps, and these sequences are used strategically within algorithms or methods aimed at improving the accuracy and efficiency of aligning multiple biological sequences. Chaotic systems provide a mechanism to explore the search space more thoroughly by generating diverse and non-repetitive sequences. This helps in avoiding local optima and encourages a more global search approach. Chaotic sequences are used to inject randomness into the GA's operations. Instead of using purely random numbers, they decide how chromosomes are initially set up, where traits are exchanged during crossover, and which parts of chromosomes get mutated. This randomness helps the algorithm explore new possibilities more effectively, especially in complex problems where finding the best solution is tough. Integrating GAs with chaotic sequences helps computers to solve problems more efficiently by mixing natural selection with a touch of randomness. The unpredictability of chaotic systems can help prevent the algorithm from getting stuck in suboptimal regions of the search space, making the GA more robust against various types of optimization challenges.

3. Proposed Approach

In this section, the proposed hybrid MSA approach is discussed in detail. The proposed approach integrates the Logistic map into the GA to exploit the chaotic properties of the Logistic map to introduce randomness and variability into the algorithm's operations. A detailed elaboration of the proposed approach is presented in the following subsections.

3.1. Initial population with the logistic map

The GA begins by creating an initial set of potential solutions (chromosomes), known as the initial population. The size of the initial population can be supplied externally. This number, denoted as P, is typically determined based on the complexity of the problem. In this work, the initial population is created with the help of the chaotic Logistic map. The initialization process begins with the chaotic Logistic map that requires a random initial value x0 within the range [0, 1] and Equation 1 to generate a sequence of values. This value acts as the starting point for the Logistic map iteration. The Logistic map iteratively generates a sequence of values <x1, x2, x3, ... xn>. Each iteration computes the next value based on the previous value and the chaotic factor r. Each value of this sequence falls in the range [0, 1]. The length of the chromosome CL is determined using Equation 2:

$$c_L = \sum_{i=1}^{nSeq} \left(len_{eq} - s^i_{len} \right) \tag{2}$$

where $nSeq$ denotes the sequence count, s^i_{len} denotes the length of the ith sequence, and len_{eq} denotes the maximum permissible length of each sequence after insertion of the gaps, and it is defined in Equation 3:

$$len_{eq} = len_{mx} \cdot f_{gap} \tag{3}$$

where len_{mx} denotes the length of the longest sequence and f_{gap} is the gap insertion factor, and this value is supplied externally.

So, for the ith sequence, $len_{eq} - s^i_{len}$ number of gaps can be inserted, and these gap positions must be sequence-wise unique. A chaotic sequence of length c_L is generated by iterating the chaotic logistic map. The members of this sequence will fall in the range [0, 1]. The gene values should belong to the range [1, len_{eq}]. A value of the chaotic sequence c^i is transformed into the gene values g^i using Equation 4.

$$g^i = \left\lceil 1 + c^i \cdot \left(len_{eq} - 1 \right) \right\rceil \tag{4}$$

Integrating chaotic sequences into this process introduces randomness, helping to explore different alignment possibilities more thoroughly and potentially finding better alignments, making it an important tool in bioinformatics for tasks such as understanding genetic relationships and predicting protein structures based on sequence similarities.

3.2. Designing the objective function

In sequence alignment, the fitness function evaluates how well two sequences match or mismatch, considering gaps and using a scoring matrix like PAM 250. PAM 250 stands for "Point Accepted Mutation 250." PAM 250 is a scoring matrix used in bioinformatics to assess the similarity between amino acid sequences during protein sequence alignment. It checks each position in the aligned sequences to see if the characters match (are the same), mismatch (are different), or if one sequence has a gap where the other has a character. Using the PAM 250 scoring matrix, which gives scores based on similarities between amino

acids, the fitness function adds these scores together across all positions to calculate a total fitness score. The proposed work uses the SP fitness score fit, and it is defined in Equation 5.

$$fit = \sum_{i=1}^{nSeq-1} \sum_{j=i+1}^{nSeq} scoringMat(i, j) \tag{5}$$

In this equation, scoringMat() represents the matrix from which the alignment scores can be calculated, and in this work, PAM 250 [24] is used. In this work, the value of the gap penalty is considered as −1. On some occasions, some chromosomes may produce some sequences where gaps are aligned with gaps, and those positions are ignored (by adding zero to the fitness function). The objective is to maximize the fitness to obtain near-optimal alignment.

3.3. Chaotic selection

The selection process is about picking the best individuals to make the next generation better. Normally, those who are stronger (fitter) are chosen based on their scores. Typically, individuals with higher fitness scores have a greater chance of being selected. When chaotic sequences are used in selection, it adds randomness. This randomness helps the algorithm explore different paths to find solutions. This is useful for complex problems where there are many ways to succeed. In this work, the binary tournament selection procedure is used, which selects two random individuals (chromosomes) from the population. Use their fitness values to determine the better individual. The chaotic sequence influences the probability of choosing the better individual, adding an element of unpredictability. Overall, chaotic sequences make selection more dynamic and adaptable. They can improve the algorithm's ability to find the best or nearly the best solutions for various challenges. In this work, the chaotic logistic map is used to generate two values of the chaotic sequence. In this stage, a different seed value is used to generate these two values compared to the seed values used in generating the initial population. These two obtained values belong to the range [0, 1]. Now these two values t are transformed to the range [1, P] using Equation 6.

$$t^i = \lceil 1 + c^i \cdot (P - 1) \rceil \tag{6}$$

3.4. Elitism

Elitism in GAs refers to a strategy where a certain number of the best individuals (chromosomes) from the current generation are directly transferred to the next generation without undergoing genetic operations such as crossover and mutation [25]. This ensures that the best solutions found so far are preserved across generations, helping to maintain or improve the overall quality of solutions over successive iterations. In this work, a single solution is preserved (i.e., the best solution of the previous generation) by replacing the worst solution of the present population. While elitism focuses on preserving high-quality solutions, it is often combined with mechanisms that promote exploration (such as mutation and crossover) to maintain diversity in the population. This balance helps the algorithm avoid premature convergence while still exploiting the best solutions. By retaining the best individuals, elitism prevents the potential loss of good solutions that might occur due to random variations introduced by genetic operators. This helps in maintaining a high quality of solutions throughout the evolutionary process.

3.5. Crossover

Crossover is a method where genetic material from two parent chromosomes is exchanged to create new offspring [26]. Crossover introduces new combinations of genes into the population, which helps in maintaining genetic diversity and exploring different areas of the solution space [27]. When chaotic sequences are used in this process, they introduce randomness to how crossover points are chosen. Crossover is like mixing traits from two parents to create new children. In this work, uniform crossover is used, where the crossover operation is performed with the help of a binary mask. The genes from the parent chromosomes are exchanged, where the value of the mask is 1. This crossover mask is generated with the help of the chaotic logistic map. The chaotic logistic map is used to create a pseudorandom bit sequence *prbs* (using Equation 7) of length CL that serves as the mask for the uniform crossover operation:

$$prbs_i = \begin{cases} 1 & if\ x_i > y_i \\ 0 & otherwise \end{cases} \tag{7}$$

where x_i and y_i are the outcomes of the chaotic logistic map with two different seed values and with the same value of the controlling parameter r, as illustrated in Equations 8 and 9, respectively.

$$x_{i+1} = r \cdot x_i \cdot (1 - x_i) \tag{8}$$

$$y_{i+1} = r \cdot y_i \cdot (1 - y_i) \tag{9}$$

The crossover process ensures we avoid repeating the same genetic sequence and allows for a wider exploration of potential solutions. By using chaotic sequences, GAs can explore more options and increase the chances of finding better solutions to complex problems. The crossover operation is illustrated in Figure 1.

3.6. Mutation

Mutation is like making small, random changes to genetic information within chromosomes. When we introduce chaotic sequences into this process, they add an element of unpredictability to where and how these changes occur [28]. Chaotic sequences generate random numbers that determine which parts of a chromosome get altered. This randomness helps keep the algorithm flexible and able to explore different possibilities for solving problems. By using chaotic sequences for mutation, GAs can try out

Parent #1 →	1	0	1	1	0	1	0	1
Parent #2 →	0	0	1	0	0	0	1	0
Chaotic Mask →	1	0	0	1	0	0	1	1
Offspring #1 →	0	0	1	0	0	1	1	0
Offspring #2 →	1	0	1	1	0	0	0	1

Figure 1. The uniform crossover operation with the chaotic mask

more variations, which increases the chance of finding better solutions to difficult problems [27]. This method keeps the algorithm from getting stuck and helps it keep improving over time. In this work, the bit-flip mutation is used. The chaotic response of the logistic map is used to compare with the mutation probability to execute the mutation at a certain place.

A step-by-step implementation of the proposed hybrid MSA approach is illustrated in Algorithm 1.

Algorithm 1: Chaotic–Metaheuristic-Based Hybrid Approach for the Multiple Sequence Alignment Problem

Input: A set of sequences to be aligned, scoring matrix, gap penalty
Output: Optimally aligned sequences and the corresponding alignment score

1. Initialize the guiding parameters, including the population size, crossover probability, mutation probability, seed value for the chaotic sequence, and the control parameters of the chaotic map.
2. Generate the initial population using the chaotic logistic map as described in Section 3.1, where each chromosome encodes gap positions within the sequences.
3. Repeat until the termination criterion is satisfied
    a. Evaluate the fitness of each chromosome using Equation (5).
    b. Select parent individuals using chaotic binary tournament selection, as detailed in Section 3.6.
    c. Apply uniform crossover using a crossover mask generated via the chaotic logistic map.
    d. Perform bit-flip mutation by comparing the mutation probability with the chaotic response value.
    e. Preserve elitism by replacing the worst solution of the $(i-1)$-th generation with the best solution of the i-th generation.
4. Output the final aligned sequences along with the corresponding alignment score.

4. Experimental Outcome

All simulations are performed in MATLAB R2022a environment in a computer that is equipped with an Intel i3 Processor, 8GB RAM, and 512 GB SSD. To establish the effectiveness and superiority of the proposed approach, the well-known BAliBASE dataset (https://www.lbgi.fr/balibase/) is used. The GA uses chaotic dynamics to improve the balance between exploring and exploiting solutions. Key parameters, such as population size, crossover probability, etc., are tuned manually, and these parameters are adjusted to ensure sufficient exploration of the solution space. Chromosome representation is chosen to effectively encode solutions using chaotic sequences or algorithms derived from chaotic maps such as the logistic map. The controlling parameters and their values are reported in Table 1. The proposed approach is compared with some other state-of-the-art approaches, for example, TGA [29], elitist GA [25], and progressive alignment [30].

The BAliBASE 3.0 dataset was used as the primary benchmark for evaluation due to its established reputation and structure-specific sequence groupings. We selected five representative protein families from reference sets RV11, RV12, RV30, and RV40 to cover a range of challenges including highly divergent sequences, variable lengths, and internal insertions. The selected sequences were RV12_BB12009, RV30_BB30020, RV40_BB40045, RV40_BB40010, and RV11_BB11035. Accession numbers and sequence content for each test case are provided in the supplementary materials. These cases were chosen to ensure the proposed algorithm is evaluated across structurally diverse and biologically realistic alignment scenarios.

Table 1. Controlling parameters and their values

Parameter	Value
Size of the initial population	100
No. of generations	200
Crossover probability	0.8
Mutation probability	0.3
Value of r (the controlling parameter of the chaotic logistic map)	3.98
Gap penalty	–1

The proposed approach is tested on the BAliBASE. This dataset is designed in such a way that it can meet the requirements of sequence exploration. The proposed approach uses the PAM 250 scoring matrix [24]. The fitness score is calculated to quantitatively analyze the performance of the sequence alignment approaches. A higher fitness score indicates a better alignment quality. For the sake of conciseness, the outcomes for only five protein families (randomly selected) are reported. To test the effectiveness and the practical applicability, the proposed hybrid approach is compared with the three standard approaches where the MSA problem is attempted to be solved with GA, elitist GA, and progressive alignment. The comparative outcome is reported in Table 2.

From the comparative analysis, it can be observed that the proposed approach performs well and outperforms some standard solutions to the MSA problem. Convergence curves obtained by applying the traditional GA, elitist GA, and the proposed hybrid approach are reported in Figure 2.

The results are analyzed generation-wise, comparing the performance with and without the use of chaotic sequences in the GA, as summarized in Table 3.

The alignment shown includes gaps inserted to align sequences, ensuring that each sequence is matched with others despite variations in length or content. Gaps are introduced strategically to optimize alignment scores, reflecting evolutionary relationships or functional similarities between sequences. This method allows for meaningful comparisons across sequences, revealing conserved regions and evolutionary changes crucial for understanding biological function or genetic relationships.

The alignment was achieved using a GA enhanced with chaotic sequences. This approach optimizes sequence alignment by iteratively refining solutions through selection, crossover, and mutation processes influenced by chaotic parameters.

Another alignment was created using traditional methods such as DP, which compares sequences step-by-step to find the best alignment based on predefined rules. These methods adjust for gaps and differences between sequences to maximize alignment scores, showing how sequences relate to each other over time. While effective, these methods don't use chaotic dynamics to speed up searches or explore different alignments quickly. The input protein sequence and the corresponding GA enhanced with chaotic sequences and DP-based representations are provided in Supplementary Table 1.

Table 2. Comparative analysis of the proposed approach

Sequence ID	Sequence count	Approaches			
		Traditional Genetic Algorithm	Elitist Genetic Algorithm	Progressive	Proposed Approach
RV12_BB12009	5	–1780	–1653	–2907	–1620
RV30_BB30020	10	–2350	–1975	–3812	–753
RV40_BB40045	9	–5230	–4735	–8706	–4338
RV40_BB40010	12	–5047	–4826	–4144	–4521
RV11_BB11035	5	–1022	–886	–1570	–873

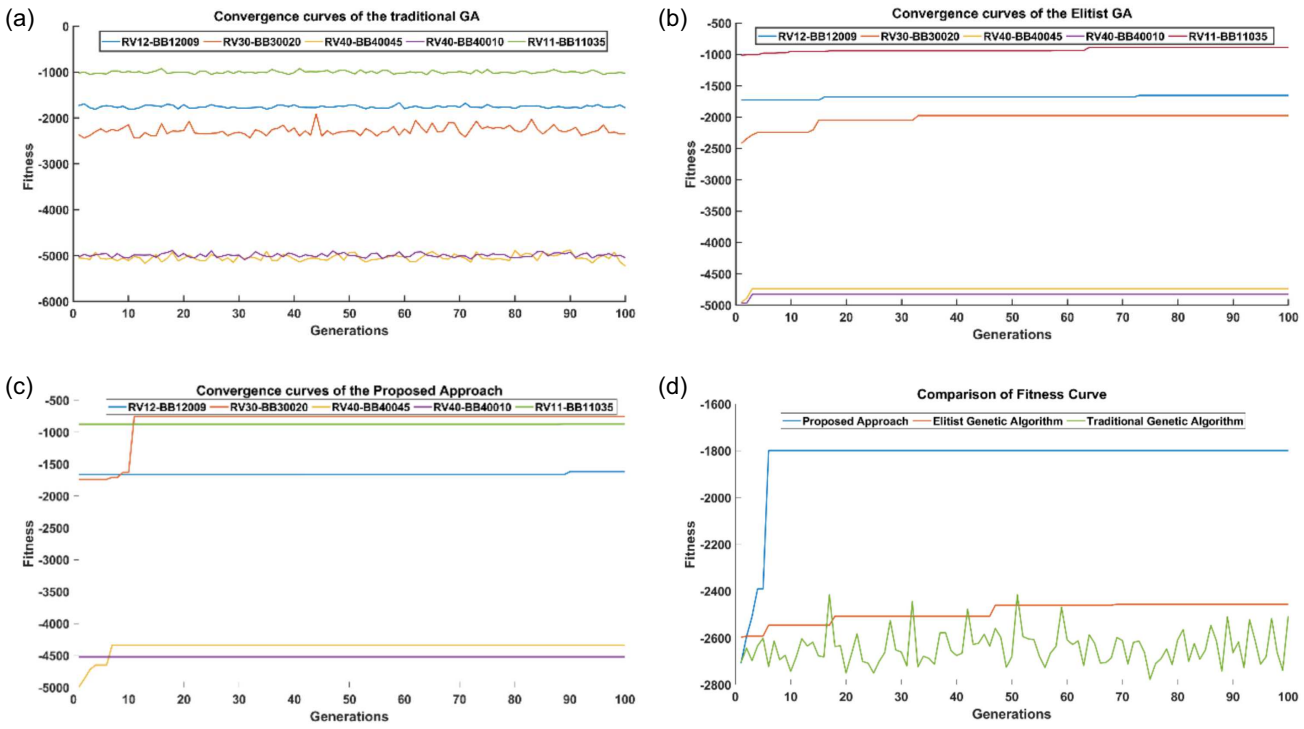

Figure 2. Graphical illustration of convergence. Convergence curves were obtained using the (a) traditional genetic algorithm, (b) elitist genetic algorithm, (c) proposed approach, and (d) a comparison of the convergence curves of three approaches (applied on RV12_BB12003)

Table 3. Generation-wise analysis of three different approaches

Method	Generations	Sequences				
		RV12_BB12009	RV30_BB30020	RV40_BB40045	RV40_BB40010	RV11_BB11035
Traditional GA	1	–1807	–2430	–5088	–4889	–1011
	10	–1800	–2319	–4987	–5087	–981
	25	–1792	–2219	–5009	–5028	–916
	30	–1786	–2071	–4817	–4989	–994
	40	–1752	–2354	–4909	–5042	–976
	50	–1748	–2459	–4897	–5040	–912
	60	–1772	–2303	–5125	–5013	–970
	70	–1776	–2300	–5057	–4982	–971
	80	–1771	–2282	–5204	–5012	–931
	100	–1780	–2350	–5230	–5047	–1022

(*Continued*)

Table 3. (*Continued*)

Method	Generations	Sequences				
		RV12_BB12009	RV30_BB30020	RV40_BB40045	RV40_BB40010	RV11_BB11035
Elitist GA	1	–1713	–2275	–5030	–4934	–1029
	10	–1740	–2203	–4931	–4937	–897
	25	–1710	–2066	–4867	–4927	–911
	30	–1664	–2162	–4763	–4856	–961
	40	–1667	–1855	–4745	–4910	–896
	50	–1679	–2031	–4814	–4843	–902
	60	–1670	–1949	–4833	–4849	–890
	70	–1663	–1990	–4797	–4859	–905
	80	–1678	–2097	–4804	–4882	–913
	100	–1653	–1975	–4735	–4826	–886
Proposed Approach	1	–1685	–1579	–4807	–4524	–881
	10	–1677	–1467	–4498	–4579	–879
	25	–1639	–1359	–4523	–4503	–850
	30	–1609	–1395	–4374	–4474	–899
	40	–1668	–1242	–4354	–4540	–889
	50	–1651	–1012	–4265	–4548	–851
	60	–1644	–870	–4366	–4482	–874
	70	–1660	–971	–4525	–4544	–849
	80	–1655	–1025	–4288	–4522	–869
	100	–1620	–753	–4338	–4521	–873

Table 4. Comparative evaluation using SP and TC scores (mean ± std. dev.)

Dataset ID	Method	SP score (↑)	TC score (↑)	Stat. significance vs proposed
RV12_BB12009	Traditional GA	0.512 ± 0.021	0.314 ± 0.017	$p < 0.01$
	Elitist GA	0.534 ± 0.018	0.337 ± 0.015	$p < 0.01$
	MAFFT	0.581 ± 0.015	0.361 ± 0.013	$p < 0.05$
	Clustal Omega	0.563 ± 0.014	0.351 ± 0.012	$p < 0.05$
	Proposed Method	0.608 ± 0.012	0.387 ± 0.011	–
RV30_BB30020	Traditional GA	0.488 ± 0.029	0.306 ± 0.019	$p < 0.01$
	Elitist GA	0.502 ± 0.024	0.317 ± 0.017	$p < 0.01$
	MAFFT	0.579 ± 0.016	0.358 ± 0.013	$p < 0.01$
	Clustal Omega	0.572 ± 0.018	0.353 ± 0.014	$p < 0.01$
	Proposed Method	0.621 ± 0.011	0.394 ± 0.010	–
RV40_BB40045	Traditional GA	0.421 ± 0.031	0.281 ± 0.023	$p < 0.01$
	Elitist GA	0.447 ± 0.027	0.298 ± 0.021	$p < 0.01$
	MAFFT	0.504 ± 0.019	0.329 ± 0.015	$p < 0.05$
	Clustal Omega	0.493 ± 0.020	0.323 ± 0.016	$p < 0.05$
	Proposed Method	0.547 ± 0.013	0.356 ± 0.012	–

In addition to the raw fitness values, we evaluated alignment quality using standard MSA metrics: the SP score and the total column (TC) score. The SP score computes the sum of match/mismatch scores across all pairwise sequence combinations, while the TC score measures the proportion of columns that are fully conserved across all sequences. To assess robustness and consistency, each experiment was run independently 30 times per dataset using controlled seed values. The mean and standard deviation of SP and TC scores were calculated, and t-tests ($\alpha = 0.05$) were conducted to compare our method's results with MAFFT, Clustal Omega, and traditional GA approaches. Statistical significance is marked with asterisks in the result tables. This ensures our reported improvements are both consistent and statistically sound. It is reported in Table 4. Figure 3 (A) illustrates a comparison of SP scores, and Figure 3 (B) compares TC scores. SP score distributions across 30 runs for the RV30 dataset can be visualized in Figure 3 (C).

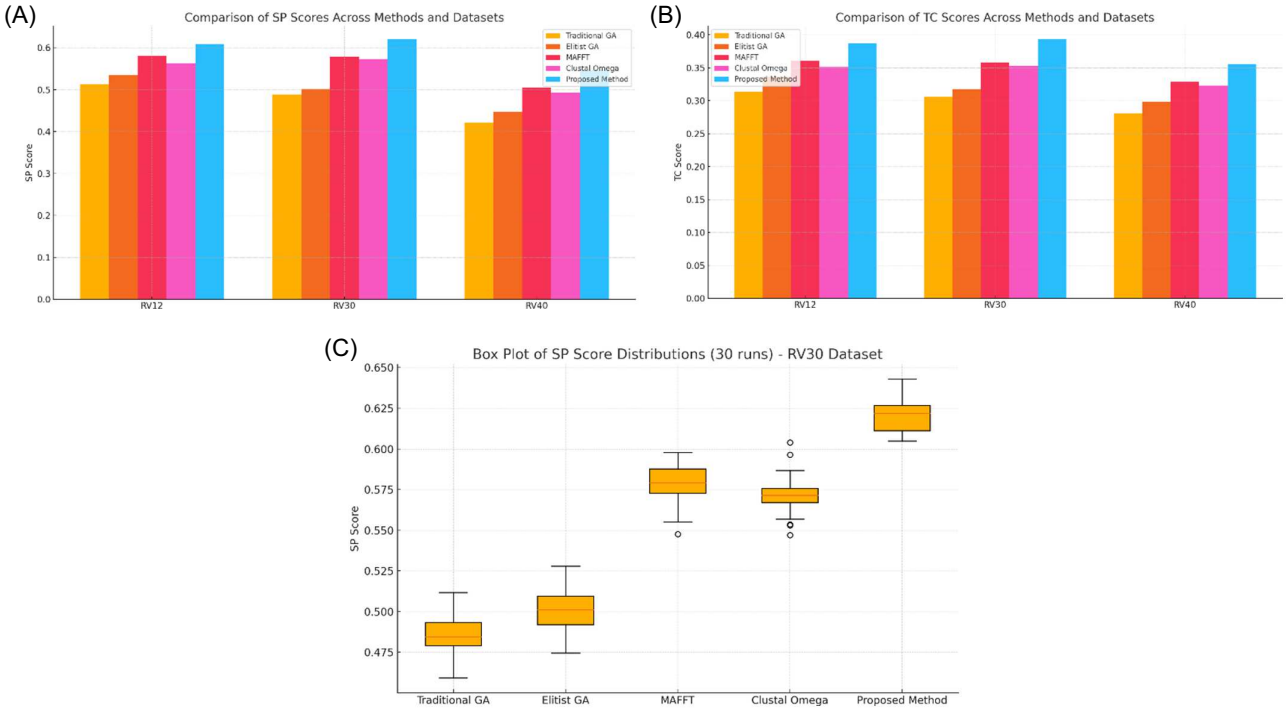

Figure 3. (A) Bar chart comparing SP scores across the three selected datasets (RV12, RV30, RV40) for all methods. (B) Bar chart comparing TC scores across the three selected datasets (RV12, RV30, RV40) for all methods. (C) SP score distributions across 30 runs for the RV30 dataset

5. Conclusion

Addressing the multiple sequence alignment problem using the proposed chaotic GA-based hybrid solution represents a significant advancement in computational biology and bioinformatics. By integrating chaotic systems with GAs, this approach leverages the inherent unpredictability and exploration capabilities of chaotic dynamics to enhance the search efficiency and solution quality for complex alignment problems. The hybrid model effectively navigates the vast and rugged solution landscape, overcoming the limitations of traditional alignment methods and yielding more accurate and biologically relevant alignments. The results demonstrate that this innovative approach not only improves alignment performance but also offers a robust framework for tackling other intricate optimization problems in computational biology. From the experimental outcomes, it can be observed that the proposed approach can effectively outperform some state-of-the-art approaches, making it suitable for real-life applications. Future research can build on this foundation by exploring further refinements and applications, potentially integrating additional heuristic techniques and real-world biological data to advance our understanding and capabilities in sequence alignment.

Acknowledgments

The authors are grateful to the anonymous reviewers and editors who have contributed to the enrichment of this manuscript.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors. The data were collected under the ethical approvals reported in the original study/dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

This article does not involve the creation or analysis of new data; therefore, the matter of data sharing does not apply to this study. The well-known BAliBASE dataset (https://www.lbgi.fr/balibase/) is used.

Author Contribution Statement

Gargi Nandi: Conceptualization, Methodology, Software, Investigation, Data curation, Writing – original draft, Visualization. **Sweta Roy:** Conceptualization, Methodology, Investigation, Data curation, Writing – original draft, Visualization. **Yeasmin Khatun:** Conceptualization, Methodology, Investigation, Data curation, Writing – original draft, Visualization. **Nibedita Chakraborty:** Validation, Formal analysis, Resources. **Shrestha Pal:** Validation, Formal analysis, Resources. **Meheria Sultana Khatun:** Validation, Formal analysis, Resources. **Shouvik Chakraborty:** Conceptualization, Methodology, Software, Investigation, Data curation, Writing – original draft, Writing – review & editing, Supervision, Project administration.

References

[1] Zhou, L., Feng, T., Xu, S., Gao, F., Lam, T. T., Wang, Q., . . . , & Yu, G. (2022). ggmsa: A visual exploration tool for multiple sequence alignment and associated data. *Briefings in Bioinformatics*, *23*(4), bbac222. https://doi.org/10.1093/bib/bbac222

[2] Jiang, Y., Shang, T., & Liu, J. (2025). Privacy-preserving multiple sequence alignment scheme for long gene sequence. In *Proceedings on Privacy Enhancing Technologies*, 236–249. https://doi.org/10.56553/popets-2025-0014

[3] Maiolo, M., Gatti, L., Frei, D., Leidi, T., Gil, M., & Anisimova, M. (2021). ProPIP: A tool for progressive multiple sequence alignment with Poisson Indel Process. *BMC Bioinformatics*, *22*(1), 518. https://doi.org/10.1186/s12859-021-04442-8

[4] Chen, J., Chao, J., Liu, H., Yang, F., Zou, Q., & Tang, F. (2023). WMSA 2: A multiple DNA/RNA sequence alignment tool implemented with accurate progressive mode and a fast win-win mode combining the center star and progressive strategies. *Briefings in Bioinformatics*, *24*(4), bbad190. https://doi.org/10.1093/bib/bbad190

[5] Issa, M., & Hassanien, A. E. (2020). Multiple sequence alignment optimization using meta-heuristic techniques. In *Data Analytics in Medicine: Concepts, Methodologies, Tools, and Applications* (pp. 565–579). IGI Global Scientific Publishing.

[6] Lajevardy, S. A., & Kargari, M. (2022). Developing new genetic algorithm based on integer programming for multiple sequence alignment. *Soft Computing-A Fusion of Foundations, Methodologies & Applications*, *26*(8), 3863–3870. https://doi.org/10.1007/s00500-022-06790-w

[7] Chowdhury, B., & Garai, G. (2020). A bi-objective function optimization approach for multiple sequence alignment using genetic algorithm. *Soft Computing*, *24*(20), 15871–15888. https://doi.org/10.1007/s00500-020-04917-5

[8] Amorim, A. R., Zafalon, G. F. D., de Godoi Contessoto, A., Valêncio, C. R., & Sato, L. M. (2021). Metaheuristics for multiple sequence alignment: A systematic review. *Computational Biology and Chemistry*, *94*, 107563. https://doi.org/10.1016/j.compbiolchem.2021.107563

[9] Paruchuri, T., Kancharla, G. R., Dara, S., Yadav, R. K., Jadav, S. S., Dhamercherla, S., & Vidyarthi, A. (2022). Nature inspired algorithms for solving multiple sequence alignment problem: A review. *Archives of Computational Methods in Engineering*, *29*(7), 5237–5258. https://doi.org/10.1007/s11831-022-09769-w

[10] Calvet, L., Benito, S., Juan, A. A., & Prados, F. (2023). On the role of metaheuristic optimization in bioinformatics. *International Transactions in Operational Research*, *30*(6), 2909–2944. https://doi.org/10.1111/itor.13164

[11] Zhang, Y., Zhang, Q., Zhou, J., & Zou, Q. (2022). A survey on the algorithm and development of multiple sequence alignment. *Briefings in Bioinformatics*, *23*(3), bbac069. https://doi.org/10.1093/bib/bbac069

[12] Ibrahim, M. K., Yusof, U. K., Eisa, T. A. E., & Nasser, M. (2024). Bioinspired algorithms for multiple sequence alignment: A systematic review and roadmap. *Applied Sciences*, *14*(6), 2433. https://doi.org/10.3390/app14062433

[13] Chao, J., Tang, F., & Xu, L. (2022). Developments in algorithms for sequence alignment: A review. *Biomolecules*, *12*(4), 546. https://doi.org/10.3390/biom12040546

[14] Tomar, V., Bansal, M., & Singh, P. (2024). Metaheuristic algorithms for optimization: A brief review. *Engineering Proceedings*,*59*(1),238.https://doi.org/10.3390/engproc2023059238

[15] Akay, B., Karaboga, D., & Akay, R. (2022). A comprehensive survey on optimizing deep learning models by metaheuristics. *Artificial Intelligence Review*, *55*(2), 829–894. https://doi.org/10.1007/s10462-021-09992-0

[16] Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73. https://doi.org/10.1038/scientificamerican0792-66

[17] Zafalon, G. F. D., Gomes, V. Z., Amorim, A. R., & Valêncio, C. R. (2021). A hybrid approach using progressive and genetic algorithms for improvements in multiple sequence alignments. In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) 2*, 384–391. https://doi.org/10.5220/0010495303840391

[18] Roy, M., Chakraborty, S., & Mali, K. (2023). An evolutionary image encryption system with chaos theory and DNA encoding. *Multimedia Tools and Applications*, *82*(22), 33607–33635. https://doi.org/10.1007/s11042-023-14948-3

[19] Roy, M., Chakraborty, S., & Mali, K. (2023). An optimized image encryption framework with chaos theory and EMO approach. *Multimedia Tools and Applications*, *82*(20), 30309–30343. https://doi.org/10.1007/s11042-023-14438-6

[20] Patro, K. A. K., & Acharya, B. (2019). A simple, secure, and time-efficient bit-plane operated bit-level image. In J. Chattopadhyay, R. Singh, & V. Bhattacherjee. (Eds), *Innovations in Soft Computing and Information Technology*. Singapore: Springer. https://doi.org/10.1007/978-981-13-3185-5_23

[21] Singh, N., & Sinha, A. (2009). Gyrator transform-based optical image encryption, using chaos. *Optics and Lasers in Engineering*, *47*(5), 539–546. https://doi.org/10.1016/j.optlaseng.2008.10.013

[22] Liu, H., Abraham, A., & Clerc, M. (2007). Chaotic dynamic characteristics in swarm intelligence. *Applied Soft Computing*, *7*(3), 1019–1026. https://doi.org/10.1016/j.asoc.2006.10.006

[23] Chaabane, L. (2021). An effective cooperative aligner to resolve multiple-sequence alignment problem. *International Journal of Cloud Computing*, *10*(5-6), 507–521. https://doi.org/10.1504/IJCC.2021.120390

[24] Wheeler, D. (2003). Selecting the right protein-scoring matrix. *Current Protocols in Bioinformatics*, *1*, 3–5. https://doi.org/10.1002/0471250953.bi0305s00

[25] Chakraborty, S., Seal, A., & Roy, M. (2015). An elitist model for obtaining alignment of multiple sequences using genetic algorithm. In *2nd national conference NCETAS*, *4*(9), 61–67.

[26] García-Valdez, M., Mancilla, A., Castillo, O., & Merelo-Guervós, J. J. (2023). Distributed and asynchronous population-based optimization applied to the optimal design of fuzzy controllers. *Symmetry*, *15*(2), 467. https://doi.org/10.3390/sym15020467

[27] Hemanth, D. J., & Anitha, J. J. A. S. C. (2019). Modified genetic algorithm approaches for classification of abnormal magnetic resonance brain tumour images. *Applied Soft Computing*, *75*, 21–28. https://doi.org/10.1016/j.asoc.2018.10.054

[28] Shayeghi, H., Jalili, A., & Shayanfar, H. A. (2007). Robust modified GA based multi-stage fuzzy LFC. *Energy Conversion and Management*, *48*(5), 1656–1670. https://doi.org/10.1016/j.enconman.2006.11.010

[29] Chowdhury, B., & Garai, G. (2017). A review on multiple sequence alignment from the perspective of genetic algorithm. *Genomics*, *109*(5-6), 419–431. https://doi.org/10.1016/j.ygeno.2017.06.007

[30] Mount, D. W. (2009). Using progressive methods for global multiple sequence alignment. *Cold Spring Harbor Protocols*, *2009*(7), pdb.top43. https://doi.org/10.1101/pdb.top43

How to Cite: Nandi, G., Roy, S., Khatun, Y., Chakraborty, N., Pal, S., Khatun, M. S., & Chakraborty, S. (2026). A Hybrid Approach to Solve Multiple Sequence Alignment Problems Using Chaotic Metaheuristics. *Medinformatics*. https://doi.org/10.47852/bonview62025784

Supplementary Information

This table presents the original protein sequence along with the corresponding representations generated by the genetic algorithm (GA) and dynamic programming (DP) approaches. The GA and DP outputs are algorithmic encodings of a single input sequence and should not be interpreted as biological sequence alignments. Gap symbols ('-') arise from chromosome formatting (GA) and state transitions (DP).

Supplementary Table 1. Reference protein sequence and algorithm-generated representations

Entry	Description	Representation
Input	Original protein sequence	MAAKKKVVTQIKLQCPAGKATPAPPVGPALGPHGVSAPQFVQQFN-DRTKSMEPGLVVPVVTVYSDKSFSFVLKTPPAAVLIRKACGIEKGS TNSVKQKVARLSLAQLTEIAQVKLPDMSALTLDAAKRIIAGTARSM GVEVERSLQIKLQLPAGKATPAPPVGPALGQHGVNIMEFCKRFNAE TADKAGMILPVVITVYEDKSFTFIIKTPPASFLLKKAAGIEKGSSEPKR KIVGKVTRKQIEEIAKTKMPDLNANSLEAAMKIIEGTAKSMGIEVV
GA Output	Genetic algorithm–based chromosome representation	MAAKK-KV-VTQ-IKLQ–CPAGKATP-APPV—-GPA-LGPHGVSAPQFVQQFNDRTKS-MEPGLVVPVVTVY-SDKSF-SF-VLKT-PPAAV-LIR–KACGIEK—GS–TNSVKQKVARLSLA-QL-TEIA-QVKLPDMS-A-LTLD-AA–KRI-IAGTARSMGVEV-ERS—L—— QIKL-QLPAGKATP-APPVG-PALGQ-H-GVN–I-MEFCK-R-FNA–ETA–DK–AG-MI-LP-V-VI-TVYE-DKSFT-FII-KTP-PAS-FLL-KKAAGI-EK–G-S-S-EP—KRKIVG-KV-TR-KQ-IE-E-IAKTK-MP-DLNA-N-SLEAAMKI–IEGTA-KSM-G-I—EVV-
DP Output	Dynamic programming–based optimal path representation	MAAK-KKVVTQIK-L—QC-PAGKA-TPA-PPV-GP-ALGP-HGV—-S-APQFVQQ-F-N—DRTKSMEPGLVVP-VVT-VYSDKSF-SFVLKTPP-A-AV–LIRKA-C-GIEK-GS-TNSVKQ-KVARL-SLAQ-LT-EIAQVKL-PDMSAL-T-LDAAKRIIAGTARSM–G–VE-VERSL QI-KL-QL-PAGK-A-TPAP-PVG-PALGQHGVNIMEF–CKR—FNAETA-D-K-A-GMI-LP–V-VI-T-V-Y-ED-K-SFT-FII-KTPP-A-S–FLLKKAAGIEKGSSE-PKRK-IVGKV–T—RK–QIEE-IAK-T–KMPDLNA–NSLEAAMKI-I-EGTAKS-MGI-EVV-