

# Enhanced Multiple Sequence Alignment via Modified Genetic Algorithms and Localized Deep Reinforcement Learning

Aceto Andrea, Frino Gerardo, Frugieri Nicola

## Abstract

Multiple Sequence Alignment (MSA) is a fundamental challenge in bioinformatics, with applications in evolutionary analysis, motif discovery, and structural biology. However, due to its classification as an NP-complete problem, existing methods struggle to achieve both high accuracy and computational efficiency. Recent advances in Deep Reinforcement Learning have shown promise in improving alignment quality but suffer from significant training overhead and scalability issues. In this work, we propose a modified genetic algorithm framework, integrated with localized DRL, to enhance MSA efficiency and accuracy. Our approach introduces novel adaptive mutation strategies, an improved crossover mechanism, and a dynamic selection process to optimize alignment without excessive computational demands. Experimental results demonstrate that our enhanced GA-based framework outperforms state-of-the-art MSA tools, achieving higher Sum-of-Pairs scores while maintaining computational feasibility for large-scale sequence alignments.

## 1 Introduction

### 1.1 Background on the multiple sequence alignment problem

Multiple Sequence Alignment (MSA) is a fundamental problem in bioinformatics, essential for various tasks such as phylogenetic analysis, motif discovery, and protein structure prediction. The goal of MSA is to align homologous regions across multiple biological sequences, maximizing the conservation of functionally or evolutionarily significant motifs.

However, MSA is classified as an NP-complete problem, meaning that the computational resources required to find an optimal alignment grow exponentially with the number and length of sequences. As a result, heuristic and meta-heuristic approaches are often employed to approximate optimal alignments in a reasonable amount of time.

## 1.2 Existing Approaches and Their Limitations

Traditional MSA methods rely on progressive alignment (e.g., ClustalW, MAFFT, MUSCLE-5), hidden Markov models (e.g., Clustal Omega), or iterative refinement (e.g., MSA-Probs, PASTA). While these methods are efficient, they often suffer from local optima, where early alignment errors propagate throughout the process and degrade the final result.

Recent advancements in artificial intelligence and deep learning have introduced novel approaches for MSA. Reinforcement learning based methods, such as DPAMSA, leverage deep Q-learning and attention mechanisms to improve alignment accuracy. However, these methods still face significant scalability challenges, requiring extensive computational resources and long training times, particularly when dealing with large genomic datasets.

## 1.3 Motivation of This Work

The limitations of both traditional and AI-based methods highlight the need for an MSA approach that combines the efficiency of heuristic algorithms with the adaptability of machine learning techniques. This work presents GA-DPAMSA, an improved version of DPAMSA that integrates:

- Genetic Algorithms (GA) as an optimization strategy, acting as a computational orchestrator to guide the alignment process.
- Deep Reinforcement Learning (DRL) for localized sequence alignment, trained on smaller sub-alignments and applied adaptively to larger datasets.
- New training and inference mechanisms to improve scalability and computational efficiency.
- Benchmarking with state-of-the-art tools, including ClustalW, Clustal Omega, MSA-Probs, MAFFT, MUSCLE-5, UPP, and PASTA.

## 1.4 Contributions of This Work

This work extends the existing GA-DPAMSA framework by introducing several key improvements:

1. **Code optimization and refactoring**
  - Removal of redundancies and duplicated code.
  - Enhanced computational efficiency.
2. **Updated training and inference system**
  - New benchmark modes for training and inference.
  - Real-time visualization of training metrics for better monitoring.
3. **Expanded benchmarking and comparative analysis**

- Inclusion of additional MSA tools for evaluation.
- Comparative performance assessment between GA-DPAMSA and traditional methods.

#### 4. Improved genetic algorithm objectives

- Maximization of Column Score as an optimization criterion.
- Multi-objective optimization of Column Score (CS) and Sum of Pairs (SP) score.

#### 5. Integration of real datasets and preprocessing improvements

- Testing on both synthetic and real-world genomic data.
- Enhanced dataset generation and preprocessing scripts for variable sequence lengths.

#### 6. Automated reporting and visualization

- Generation of detailed reports, including performance metrics and graphical results.
- Improved documentation for usability and reproducibility.

### 1.5 Paper Organization

The paper is structured as follows:

- **Section 2** reviews the state-of-the-art in MSA, covering both traditional and AI-based approaches.
- **Section 3** describes the original GA-DPAMSA framework.
- **Section 4** presents the modifications and improvements introduced in this work.
- **Section 5** details the experimental setup and benchmarking results.
- **Section 6** discusses findings, limitations, and future research directions. These contributions position GA-DPAMSA as a scalable and efficient solution for complex MSA tasks, bridging the gap between heuristic alignment strategies and learning-based optimization techniques.

## 2 State of the art

### 2.1 Traditional approaches to multiple sequence alignment

Over the years, several methods have been developed to address the Multiple Sequence Alignment problem, each with its own strengths and limitations. These methods can be broadly categorized into progressive alignment, iterative refinement, and probabilistic models.

#### 2.1.1 Progressive Alignment Methods

Progressive alignment is one of the most widely used strategies for MSA due to its computational efficiency. It follows a hierarchical approach where sequences are aligned in a step-by-step manner based on a guide tree. Some of the most prominent tools in this category include:

- **ClustalW[1]**: Uses a hierarchical clustering approach and position-specific gap penalties to improve alignment sensitivity.
- **Clustal Omega[2]**: An improvement over ClustalW, leveraging Hidden Markov Models to achieve higher accuracy in large-scale alignments.
- **MAFFT[3]**: Employs fast Fourier transform to accelerate sequence comparison, making it one of the most efficient tools for large datasets.
- **MUSCLE-5[4]**: Uses a progressive approach with refinement phases to optimize alignments, balancing speed and accuracy.

Despite their efficiency, progressive methods suffer from the “once aligned, never realigned” problem: early alignment errors propagate and cannot be corrected later in the process.

#### 2.1.2 Iterative Refinement Methods

To mitigate the errors introduced by progressive alignment, iterative methods refine the alignment by repeatedly realigning subsets of sequences:

- **MSA-Probs[5]**: Uses a probabilistic consistency model to improve accuracy, iteratively refining the alignment.
- **PASTA[6]**: Extends progressive alignment by using iterative refinement and tree-based partitioning, allowing it to handle large-scale datasets.
- **UPP (Ultra-large Alignment using Phylogenetic Placement)[7]**: A recent approach that leverages phylogenetic information to improve the accuracy of large-scale alignments.

While iterative refinement improves alignment quality, these methods tend to be computationally expensive and may not scale efficiently to very large datasets.

### 2.1.3 Probabilistic and Hidden Markov Model-Based approaches

Some alignment methods incorporate probabilistic modeling to improve robustness:

- **Clustal Omega**: Uses profile HMMs to align sequences iteratively, making it particularly useful for very large datasets.
- **T-Coffee[8]**: Employs a consistency-based scoring approach, leveraging multiple alignment algorithms for better accuracy.

Although HMM-based methods enhance alignment reliability, their higher computational cost remains a significant drawback.

## 2.2 Machine learning and reinforcement learning for MSA

Given the computational complexity of MSA, recent research has explored machine learning and reinforcement learning based approaches to improve both efficiency and accuracy.

### 2.2.1 Reinforcement learning based methods

Reinforcement learning has been investigated as an alternative approach to MSA, where an agent learns to optimize sequence alignment through trial and error.

- **RLALIGN[9]**: One of the first RL-based MSA approaches, using the Asynchronous Advantage Actor-Critic (A3C) framework to optimize alignment quality.
- **Q-Learning-Based Alignment[10]**: Uses Q-learning and actor-critic models to iteratively refine alignments, balancing policy-based and value-based optimization.
- **EdgeAlign[11]**: Applies reinforcement learning in a sliding window approach, focusing on small regions of sequences to optimize local alignment.

While RL-based methods show promise, they face significant scalability limitations. Training deep RL models for MSA requires substantial computational resources, making them difficult to apply to large genomic datasets.

### 2.2.2 Deep Reinforcement Learning and DPAMSA

One of the most advanced RL-based methods for MSA is DPAMSA[12] (Deep Positionally Aware MSA), which combines Deep Reinforcement Learning (DRL) with positional encoding. DPAMSA approximates the Q-function in progressive column alignment, enhancing sequence feature extraction while preserving positional information.

### 2.2.3 Limitations of DPAMSA:

1. Computational inefficiency: training DRL models on large MSA datasets is computationally expensive and time-consuming.
2. Scalability issues: the method does not generalize well to longer sequences and requires frequent retraining when dealing with larger datasets.
3. Lack of flexibility: DPAMSA does not allow for direct benchmarking against other tools or dynamic adjustments to training parameters.

## 2.3 Genetic algorithms for MSA

Genetic Algorithms (GAs) have been explored as an alternative heuristic method for optimizing sequence alignment. GAs mimic the process of natural evolution, using selection, crossover, and mutation to improve alignment quality over successive generations. BSAGA (Bi-objective Sequence Alignment using Genetic Algorithm)[13] is a novel GA-based method for Multiple Sequence Alignment. Anyway, this algorithm suffers from premature convergence and suboptimal gap positioning.

While GAs excel in navigating complex search spaces, they often suffer from premature convergence, like BSAGA, and high computational costs in large-scale MSA tasks.

## 2.4 Hybrid approaches: combining GA and RL for MSA

To address the shortcomings of both Genetic Algorithms (GAs) and Reinforcement Learning, hybrid approaches have been proposed, leveraging the strengths of both techniques.

### 2.4.1 GA-DPAMSA: A hybrid GA and DRL approach for MSA

GA-DPAMSA, originally introduced in previous work, combines Genetic Algorithms with Deep Reinforcement Learning to overcome the scalability challenges of DPAMSA. The GA acts as an orchestrator, guiding the localized application of DRL agents to manageable sub-alignments, which are then integrated into a global alignment solution. The advantages of GA-DPAMSA are different and can potentially overcome the limitations of pure GA. In detail, GA-DPAMSA can have:

- **Scalability:** By localizing DRL to small sub-alignments, GA-DPAMSA avoids the high computational costs of training on full sequences.
- **Flexibility:** The GA framework dynamically selects the best alignment strategies, allowing real-time adaptation to different dataset sizes.

#### **2.4.2 Research Gaps and Motivation for Further Improvements**

While the original GA-DPAMSA approach demonstrates superior alignment performance compared to DPAMSA and traditional methods, several areas for improvement remain:

- Optimization of training and inference processes to enhance computational efficiency.
- Improved benchmarking tools to compare GA-DPAMSA against a wider range of state-of-the-art MSA methods.
- Integration of real-world genomic datasets for validation beyond synthetic benchmarks.

These limitations motivated the enhancements introduced in this work, which are detailed in the next sections.

### 3 GA-DPAMSA: original framework and methodology

As said before, GA-DPAMSA is a hybrid optimization framework that combines genetic algorithms and deep reinforcement learning to address the MSA problem. This approach leverages the exploratory power of genetic algorithms to guide the search space while using reinforcement learning-based local alignment refinements for enhanced accuracy.

#### 3.1 Original framework of GA-DPAMSA

GA-DPAMSA integrates two key computational paradigms:

##### 1. Genetic algorithm for global optimization

- Provides population-based search over the alignment space.
- Evolves multiple candidate alignments through selection, crossover, and mutation.
- Incorporates a fitness function (Sum of Pairs score) to drive the optimization.

##### 2. Deep reinforcement learning for local refinement

- Uses an agent-based approach to refine local sequence alignment decisions.
- Applies Deep Q-learning and attention mechanisms to improve sequence feature extraction.
- Adjusts gap positions dynamically to enhance alignment quality.

By combining these two methodologies, GA-DPAMSA balances computational efficiency and alignment accuracy, making it scalable to large genomic datasets.

#### 3.2 Genetic Algorithm Component

##### 3.2.1 Population Encoding and Initialization

GA-DPAMSA represents MSA solutions as chromosomes in a genetic algorithm. The population consists of multiple candidate alignments, each encoded using a gap-based representation. Given a board, we first build the board matrix (e.g., the  $4 \times 8$  matrix in Figure 1), and then each nucleotide is mapped to an integer as follows:

$$A:1, T:2, C:3, G:4, a:1, t:2, c:3, g:4, -:5$$

Note that gaps are encoded as 5.

The sequences are represented as integer matrices, forming the GA’s population of individuals, where each individual corresponds to a potential alignment. The initial population  $P$  is composed of several identical copies of the initial integer matrix.

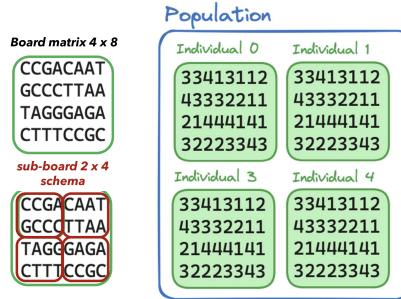


Figure 1: Board, sub-boards schema, and population of the GA orchestrator

### 3.3 Fitness function

The fitness value of each individual  $A_i \in P$  is evaluated in terms of SP score. This score represents a measure of the alignment quality, quantifying how well the sequences in an individual are aligned, and is crucial for guiding the evolutionary process, as individuals with higher SP scores represent more optimal alignments.

The GA uses the fitness to rank individuals, promoting those with better alignments to contribute more significantly to the creation of the next generation.

#### 3.3.1 Operators of the Genetic Algorithm

GA-DPAMSA can use two **mutation** methods:

1. **Random**: individuals are randomly chosen for mutation, introducing unbiased variation in order to prevent premature convergence; let  $M$  be the set of sub-boards on the schematized board matrix; for each distinct  $m \in M$ , a random individual  $A_i$  is chosen from the population, and the RL base agent is applied to  $A_i(m)$ , where  $A_i(m)$  is the sub-board of  $A_i$  corresponding to  $m$ ; this approach iterates through all  $m \in M$  until each sub-board has been processed.
2. **Worst fitted**: for each of the  $k$  individuals with the lowest fitness scores, the RL base agent operates on the sub-board  $m$  that contributes most to the low score; this strategy balances exploration and exploitation, with improved precision at a higher computational cost compared to the random strategy.

The **crossover** operator is fundamental in the GA, enabling the exchange of genetic material between two selected parent individuals  $A, B \in P$ . This operator creates a new individual  $C$  by combining portions of  $A$  and  $B$ , allowing the population to inherit favorable traits and potentially achieve higher fitness. The crossover is performed horizontally: let  $\frac{h}{2}$  be the horizontal split point, where  $n$  is the number of rows in the board matrix; the parents  $A$  and  $B$  are partitioned horizontally along the rows using  $h$  and the new individual  $C$  is created as follows:

$$C = A[1 : h, :]B[h + 1 : n, :]$$

where  $A[1 : h, :]$  (resp.  $B[h + 1 : n, :]$ ) is the upper half of  $A$  (resp. lower half of  $B$ ); this configuration preserves row continuity from each parent, introducing diversity in the alignment's horizontal structure (see Figure 2).

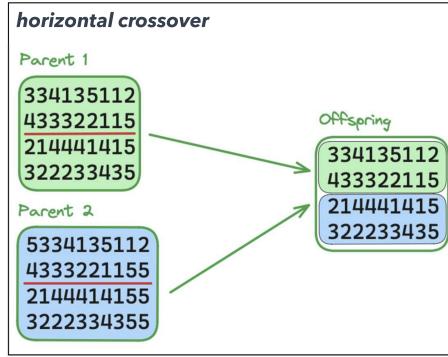


Figure 2: Crossover operator

The **selection** guides the GA toward optimal solutions by ranking individuals based on their fitness and selecting the top  $n$  for the next generation. This approach maintains high-quality alignments, balancing exploration and exploitation by preserving the best solutions while discarding lower-ranking individuals. The GA terminates after a fixed maximum number of epochs. The proposed approach is agnostic to the specific RL base model used, so replacing it only requires modifying the mutation phase.

### 3.4 Limitations of the original GA-DPAMSA Framework

Despite its innovative combination of GA and DRL, GA-DPAMSA exhibited several limitations that affected both its performance and usability. The original implementation suffered from code redundancy and inefficiencies, leading to unnecessary computational overhead. Several critical bugs impacted the alignment accuracy and genetic algorithm operations, including improper gap removal, where entire nucleotide columns were erroneously deleted, an overactive mutation function, which affected more individuals than intended, and a crossover issue, where offspring improperly referenced parent lists, causing unintended mutations in both generations.

Additionally, the training and inference pipeline lacked flexibility, preventing users from dynamically selecting benchmarking modes. In addition, real-time metric statistics were absent, making it difficult to assess model training effectiveness, while the benchmarking was limited to a few numbers of MSA tools in the literature.

In the end, the algorithm only used one metric (Sum of Pairs) to find the solution. While it's relatively efficient to compute, which has made it a popular choice in many alignment tools, it has some limitations:

- **Pairwise focus:** because it only considers pairs of sequences, it might miss higher-order correlations and evolutionary relationships among more than two sequences at a time.
- **Bias toward conserved regions:** the metric can favor alignments that score well in pairwise comparisons even if they miss functionally or structurally important features present in a multi-sequence context.
- **Local vs global quality:** It might not adequately capture the overall biological relevance of an alignment, such as the preservation of secondary or tertiary structures.

These limitations motivated the improvements introduced in this work, which aim to optimize GA-DPAMSA's efficiency, expand its benchmarking capabilities, and enhance genetic algorithm flexibility. The next section details these modifications and enhancements.

## 4 GA-DPAMSA enhancements

In this chapter, we detail the comprehensive enhancements implemented to address the limitations of the original GA-DPAMSA framework. These modifications encompass bug fixes, code optimization, integration of advanced benchmarking tools and the introduction of multi-objective optimization strategies, collectively aimed at elevating the framework's performance and adaptability.

### 4.1 Code refactoring and bug fixes

A comprehensive code refactoring and optimization were conducted to eliminate redundancies and replicated code, resulting in a more efficient and maintainable codebase.

#### 4.1.1 Code refactoring

Specifically, multiple repetitive lines of code were eliminated by introducing functions to replace them when needed. Moreover, Python loops were substituted with library functions wherever feasible, resulting in increased efficiency and speed. Furthermore, additional menus were introduced to select the operation mode of the genetic algorithm and benchmarks. Additionally, workflow information and result reporting were enhanced to provide a clearer understanding of the algorithm's processes and debugging. The training and inference system of DPAMSA was updated to allow users to select between training and inference modes, facilitating more streamlined benchmarking processes. Real-time visualization of training metrics was integrated, enabling immediate assessment of training quality and progress.

#### 4.1.2 Bug fixes

Several critical bugs within the genetic algorithm (GA) logic of GA-DPAMSA have been identified and rectified, particularly concerning the selection, mutation, and crossover operators.

- **Selection phase:** previously, the selection mechanism failed to handle cases where two individuals possessed equal fitness scores, leading to inconsistent selection outcomes. This issue has been resolved by implementing a tie-breaking strategy that ensures deterministic and fair selection among individuals with identical fitness levels.
- **Mutation phase:** the mutation operator was erroneously applied to a greater number of individuals than specified, disrupting the intended genetic diversity and convergence rates. This over-application has been corrected, aligning the mutation process with the predefined parameters to maintain appropriate evolutionary pressure.

- **Crossover phase:** in the crossover operation, offspring were inadvertently referencing the genomes of their parent individuals. This reference linkage caused unintended alterations in both parent and offspring genomes during subsequent genetic operations, leading to chaotic and unpredictable evolutionary behavior. The issue has been addressed by ensuring that offspring genomes are independently instantiated, thereby preserving the integrity of both parental and progeny genetic information.
- **Remove unnecessary gaps:** The function was designed to eliminate trailing columns in a MSA alignment that contained only gap characters. However, due to a bug, it occasionally removed nucleotide data, thereby altering the original sequences. This issue arose from improper handling of columns, leading to unintended deletion of columns containing actual nucleotide information.

These corrections collectively enhance the robustness and reliability of the GA-DPAMSA framework, ensuring that genetic operations perform as intended without unintended side effects.

## 4.2 Added benchmark tools

To ensure robust performance evaluation, benchmarking tools such as Clustal Omega, ClustalW, MSA-Probs, MAFFT, MUSCLE-5, UPP, and PASTA were incorporated, allowing for comprehensive comparisons. A user-friendly menu was added to facilitate benchmarking comparisons between these tools, DPAMSA, and GA-DPAMSA. Following the benchmark, a comprehensive automated report is generated, detailing the results of all the tools for comparison against both DPAMSA and GA-DPAMSA. Furthermore, the benchmark now produces various plots to illustrate the differences among the tools, facilitating data analysis.

## 4.3 Optimization objectives and improvements

### 4.3.1 Additions to the objectives of the GA

As said in previous sections, the original GA was designed to optimize the Sum of Pairs score. This solution obtained discrete results, but not quite sufficient. We contributed to the work with two additional modes:

1. **Column Score optimization:** the GA works like the original one, but the fitness score is based on the Column Score instead of the Sum of Pairs.
2. **Multi-objective optimization:** the GA is improved with a multi-objective fitness function that calculates the Pareto front between the Sum of Pairs and the Column Score metrics.

#### 4.3.2 The Hall of Fame

Another addition to the project was the introduction of the *Hall of Fame*, a specialized data structure designed to preserve the top-performing individuals (alignments) encountered throughout the evolutionary process of the genetic algorithm. This mechanism ensures that the best solutions are retained across generations, providing a repository of elite alignments that can be analyzed or reintroduced into the population as needed.

The key features of the *Hall of Fame* are:

- **Preservation of elite individuals:** the Hall of Fame maintains a record of the highest-scoring alignments based on predefined fitness criteria, safeguarding them from potential loss due to genetic operations like crossover and mutation.
- **Dynamic Updating Mechanism:** As the genetic algorithm progresses, the Hall of Fame is continuously updated. New individuals that surpass the fitness of the current members are added, ensuring that only the most optimal alignments are stored.
- **Facilitating convergence analysis:** by analyzing the contents of the Hall of Fame over successive generations, researchers can gain insights into the convergence behavior of the algorithm and the diversity of high-quality solutions.
- **Enhancing solution quality:** The elite alignments stored in the Hall of Fame can be reintroduced into the population, promoting the propagation of superior genetic material and potentially accelerating the convergence towards optimal solutions.

#### **4.3.3 Modification to the population generation**

The initial population is a diverse set of candidate multiple sequence alignments that serves as the starting point for the genetic algorithm’s optimization process. With our modifications, the generation function now generates an initial set of candidate alignments using exact copies and modified copies, that have random gap insertions. The population is built by creating a fraction of exact copies, based on a CLONE\\_RATE variable and from the input sequences with some modified versions, created by inserting random gaps into sequences, using a GAP\\_RATE variable.

#### **4.3.4 Early stopping**

To enhance DPAMSA’s efficiency and prevent unnecessary iterations, an early stopping mechanism has been integrated into the training process. This technique monitors the model’s performance and halts training when improvements plateau, thereby avoiding unnecessary iterations and potential overfitting. Specifically, if the model’s reward metric does not exhibit significant enhancement over a predefined number of iterations, the training process is terminated. This approach ensures computational resources are utilized effectively and the model maintains optimal generalization capabilities.

## 5 Experimental Results

### 5.1 Experimental Setup

We conducted extensive experiments using both synthetic and real datasets of varying sizes and complexity. We compared our modified GA-DPAMSA framework against standard tools, as said above. The GA-DPAMSA results shown below are obtained using the ultimate GA-DPAMSA version, with Multi-objective mode, Hall of Fame and bug fixes. The DPAMSA model utilized in these experiments was trained on the 3x30 dataset. The aim is to assess whether the GA can enhance model performance on extended sequences without the need for training on such sequences. Every standard tool has been utilized with its default settings.

### 5.2 Performance Metrics

We evaluated the following metrics:

- **Sum-of-Pairs;**
- **Column score.**

### 5.3 Results and Analysis

The GA-DPAMSA framework was evaluated using four distinct datasets: Synthetic Dataset 3x30, Synthetic Dataset 6x30, Synthetic Dataset 6x60, and Real Dataset 4x101<sup>1</sup>, obtained from a dataset on <https://www.encodeproject.org/>. Each dataset was designed to assess the algorithm’s performance under varying conditions, including different sequence lengths and complexities. The synthetic ones have been generated using the Python script in the *Dataset* folder.

---

<sup>1</sup><https://www.encodeproject.org/files/ENCFF222FVM/>

### 5.3.1 Synthetic Dataset 3x30

In evaluations using smaller synthetic datasets, most tools yielded comparable results. Notably, GA-DPAMSA ranked third, following MSAProbs and ClustalW, while DPAMSA and Clustal Omega performed less favorably.

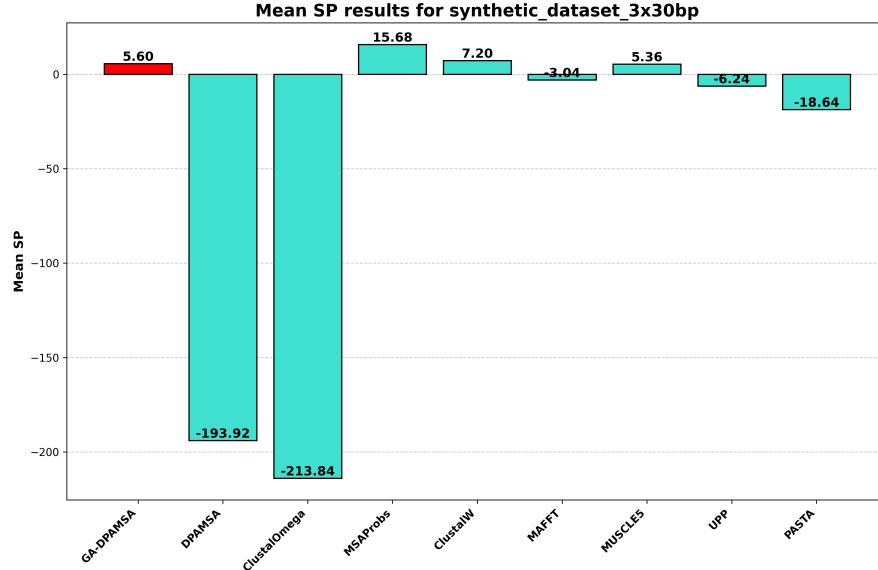


Figure 3: Barplot of the mean SP score results in 3x30 dataset

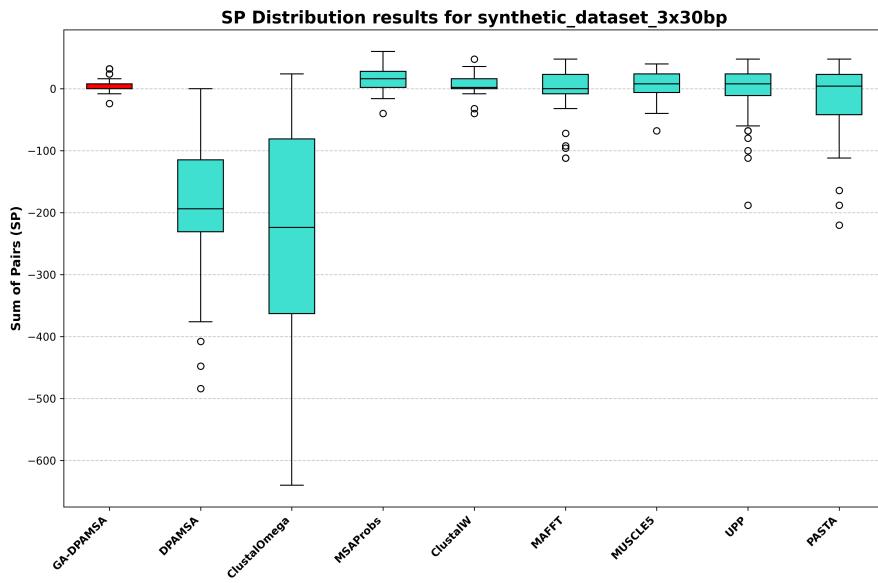


Figure 4: Comparison between the distributions of SP scores in 3x30 dataset.

Also the distribution of the Sum of Pairs (Fig. 4) score is quite similar.

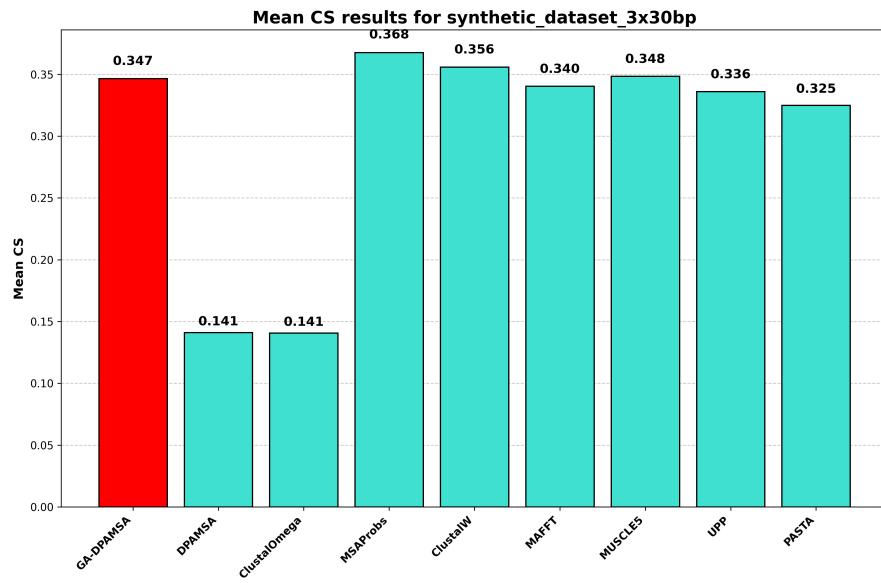


Figure 5: Barplot of the mean CS score results in 3x30 dataset.

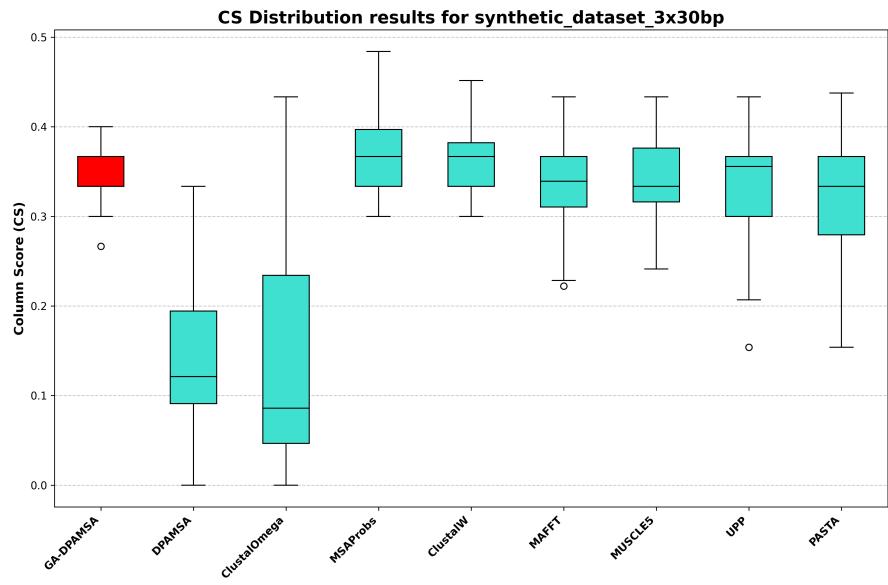


Figure 6: CS distribution results in 3x30 dataset.

Figure 6 exhibits a similar pattern, with the GA slightly lagging behind MUSCLE5 in this instance.

### 5.3.2 Synthetic Dataset 6x30

Significant changes occur as sequence length increases. The scalability of the GA enables obtaining notably distinct results, often surpassing other tools.

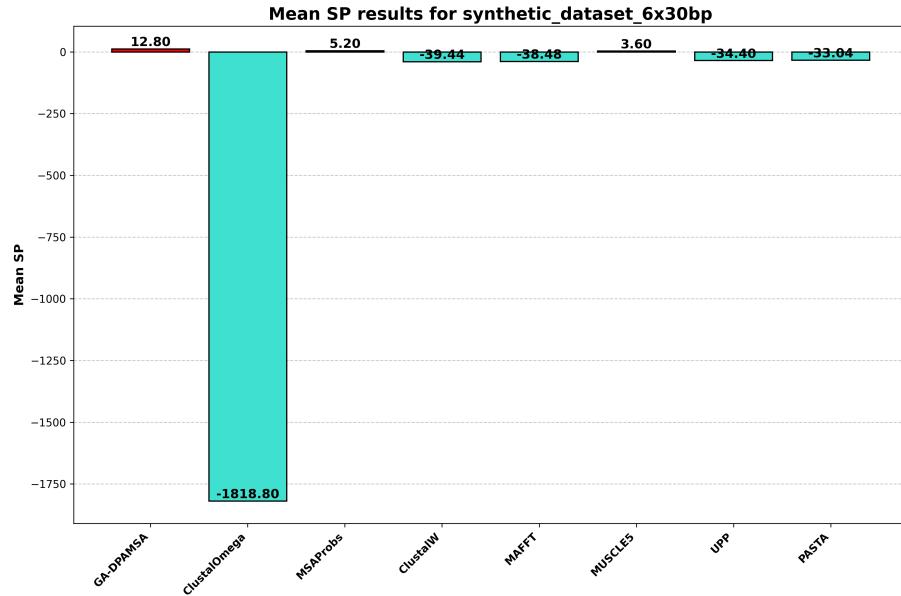


Figure 7: Barplot of the mean SP score results in 6x30 dataset.

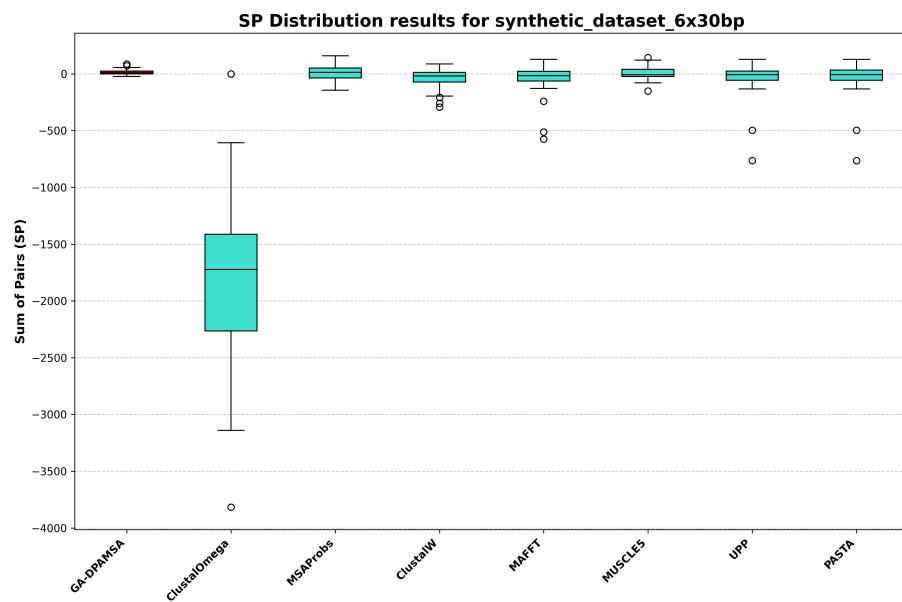


Figure 8: Comparison between the distributions of SP scores in 6x30 dataset.

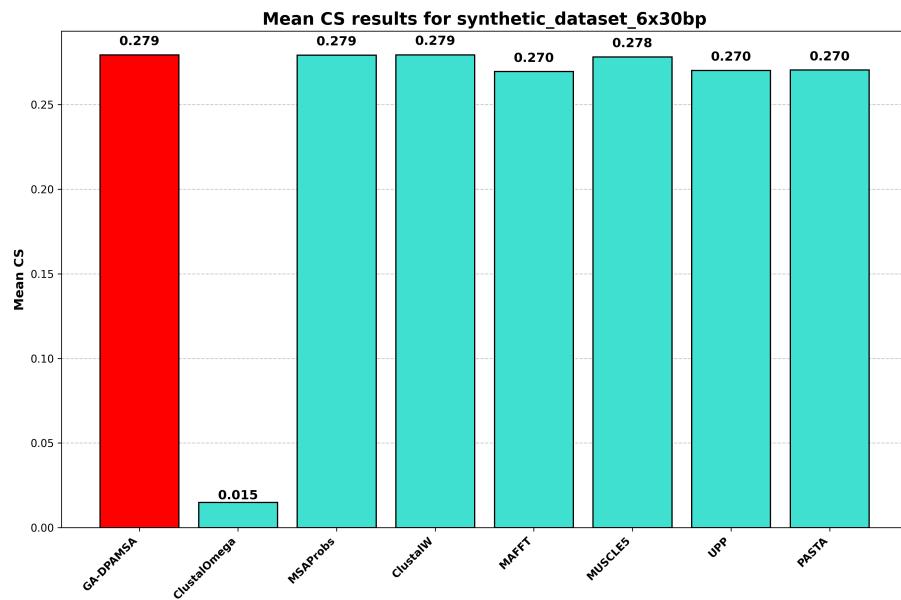


Figure 9: Barplot of the mean CS score results in 6x30 dataset.

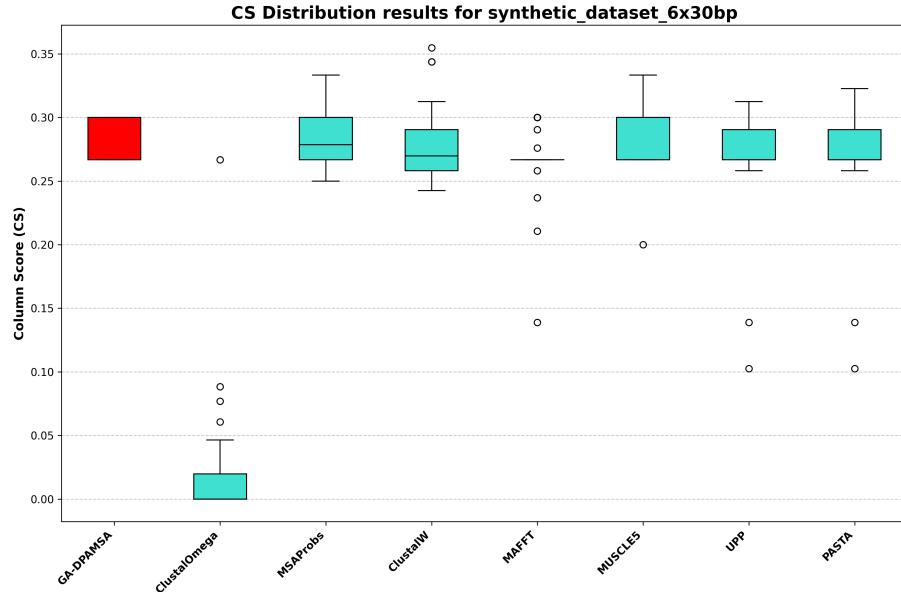


Figure 10: CS distribution results in 6x30 dataset.

As observed, GA-DPAMSA achieves the highest SP score, and the CS is comparable to other top-performing tools. Furthermore, the SP score distribution is more stable and focused, indicating consistent strong performance across all sequences by the model, unlike the variable behavior seen in other tools. ClustalOmega confirms it's worst performance for this dataset.

### 5.3.3 Synthetic Dataset 6x60

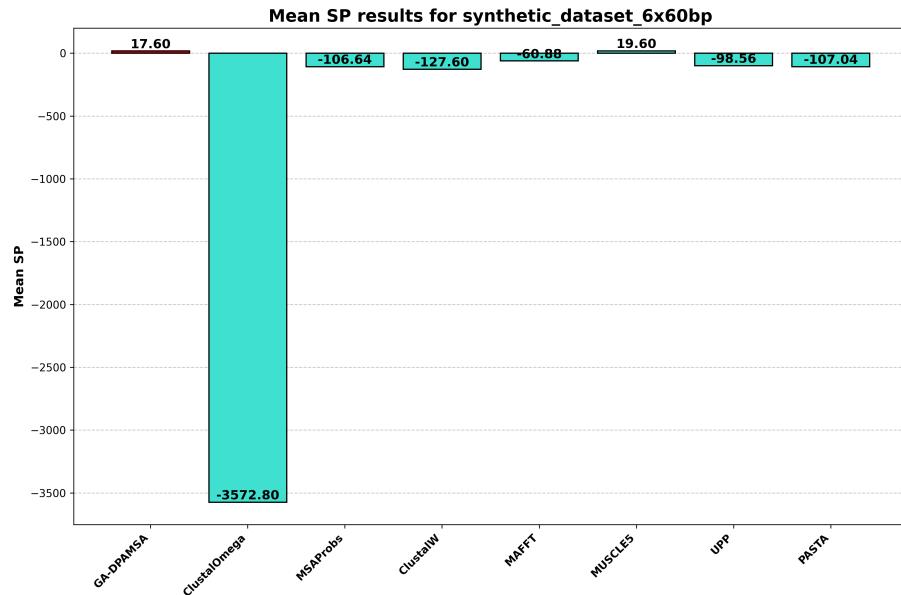


Figure 11: Barplot of the mean SP score results in 6x60 dataset.

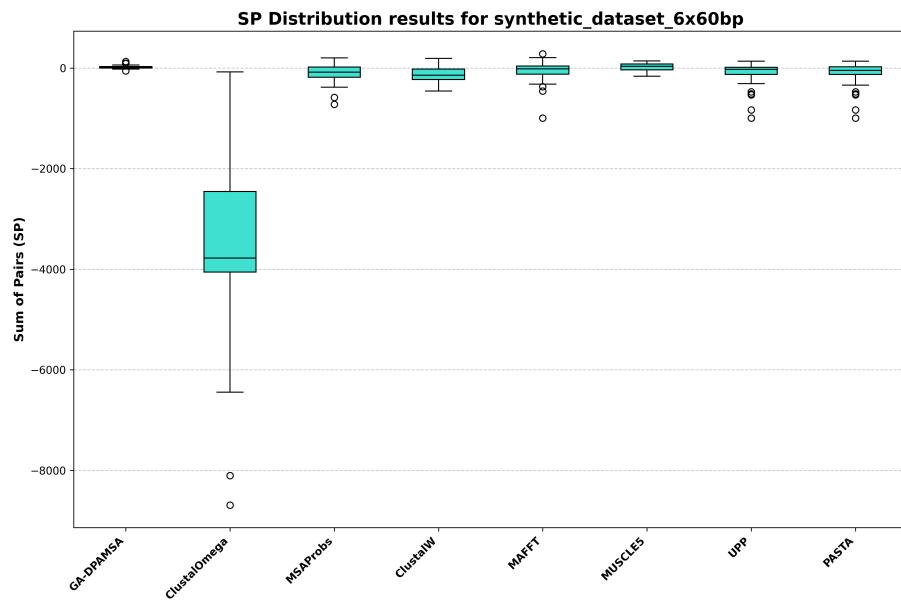


Figure 12: Comparison between the distributions of SP scores in 6x60 dataset.

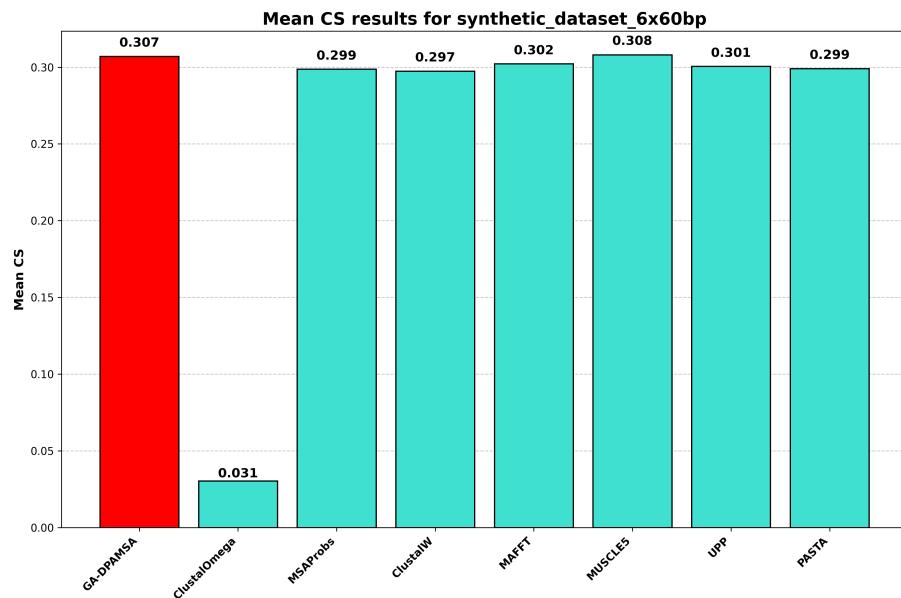


Figure 13: Barplot of the mean CS score results in 6x60 dataset.

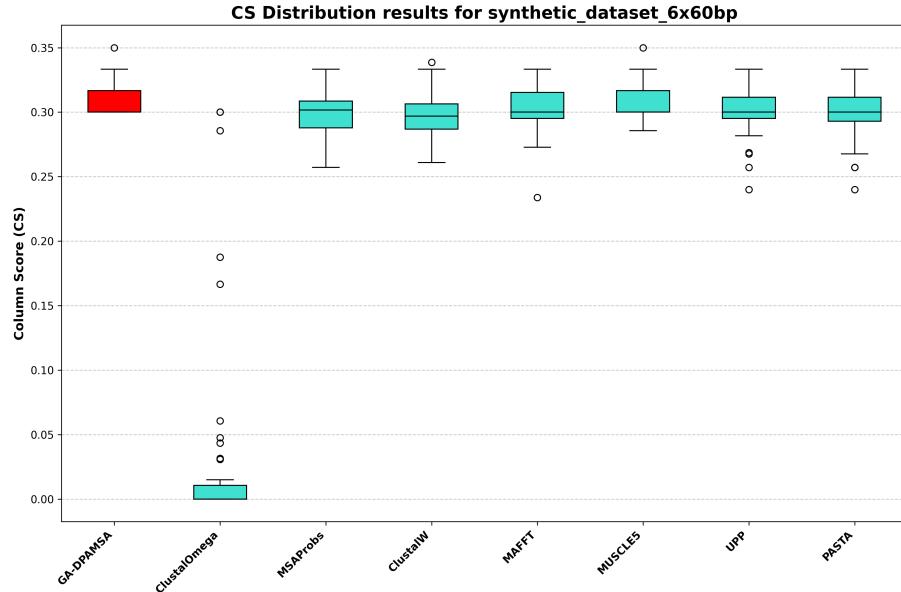


Figure 14: CS distribution results in 6x60 dataset.

As sequence length increases, the performance of standard alignment tools often deteriorates, leading to negative Sum op pairs scores. Notably, MUSCLE5 demonstrates resilience under these conditions, achieving a robust SP score of 19.60, the only standard tool to maintain a positive score. In comparison, GA-DPAMSA attains a commendable SP score of 17.60, indicating its effectiveness in handling longer sequences without training the model with such sequences.

### 5.3.4 Real Dataset 4x101

The dataset was derived from the original *ENCFF222FVM*<sup>2</sup> dataset available on encodeproject.org. It comprises 10 FASTA files, each containing 4 sequences of 101 nucleotides in length.

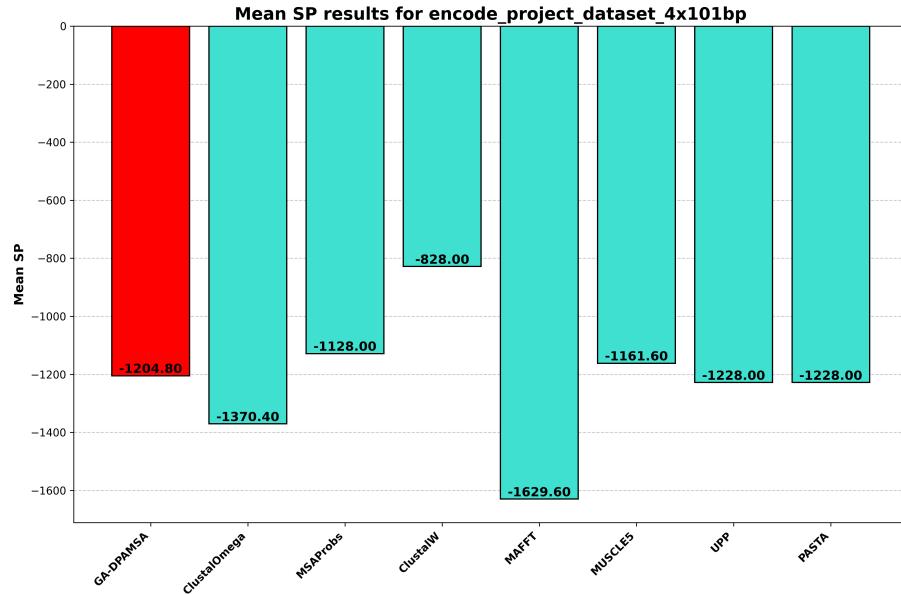


Figure 15: Barplot of the mean SP score results in 4x101 dataset.

<sup>2</sup><https://www.encodeproject.org/files/ENCFF222FVM/>

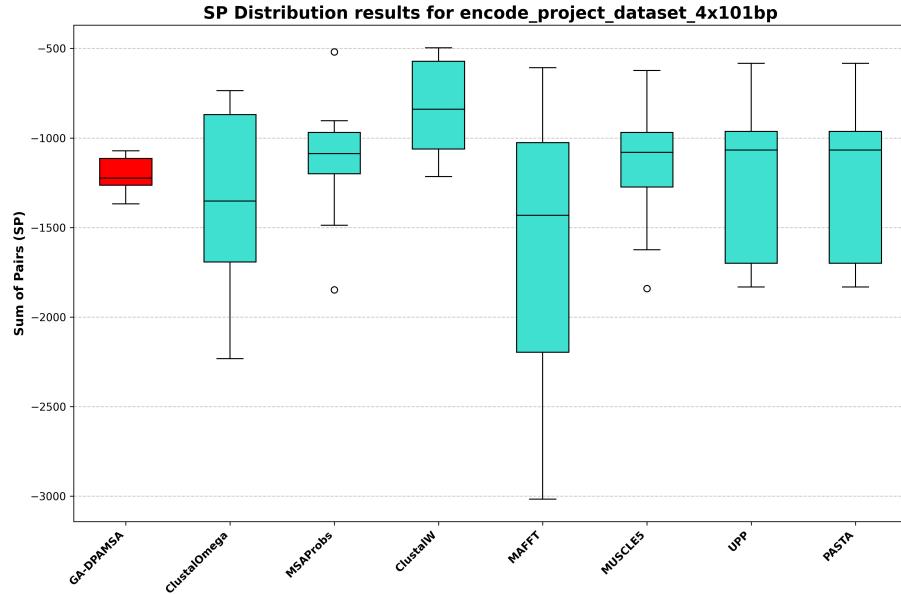


Figure 16: Comparison between the distributions of SP scores in 4x101 dataset.

When evaluating the performance of various alignment tools on this dataset, a general decline in accuracy was observed, with all tools yielding negative SP scores.

Notably, ClustalW exhibited the least degradation, securing the highest SP score among the tools tested. GA-DPAMSA delivered a respectable performance, ranking fourth behind ClustalW, MsAProbs, and MUSCLE5.

As illustrated in Plot 16, GA-DPAMSA consistently produced stable results across all sequence blocks, whereas other tools demonstrated significant variability among the 4x101 sequence sets.

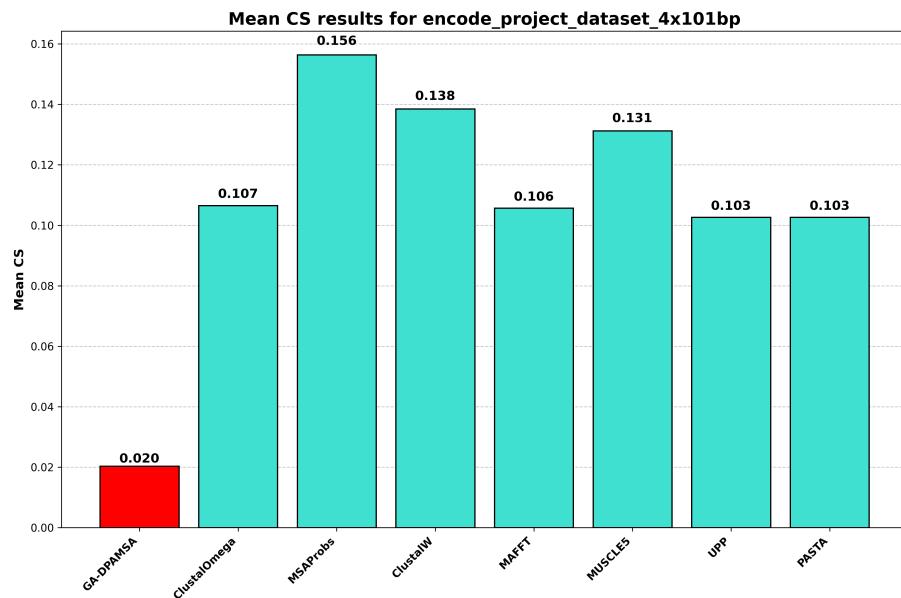


Figure 17: Barplot of the mean CS score results in 4x101 dataset.

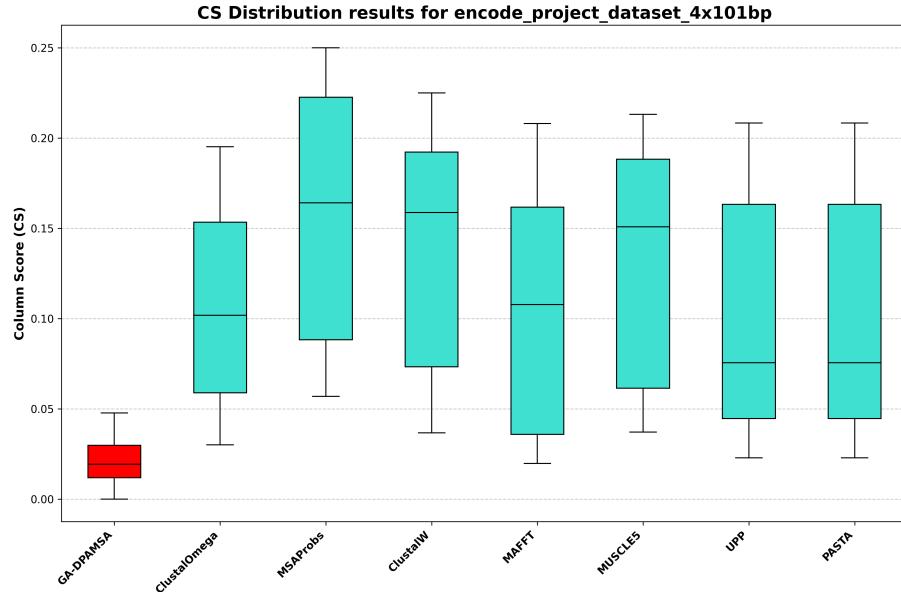


Figure 18: CS distribution results in 4x101 dataset.

In evaluating column scores, GA-DPAMSA consistently achieved lower results compared to other alignment tools. This suggests that while GA-DPAMSA performs well in aligning partial sequences, it encounters challenges in achieving optimal full-column alignments, particularly when applied to real datasets. Additionally, standard tools exhibited high variance in performance, whereas GA-DPAMSA demonstrated consistent results across different sequence blocks.

#### 5.4 Final considerations

According to the results obtained, the experimental evaluation of the GA-DPAMSA framework across various synthetic and real datasets has demonstrated its robustness and adaptability in multiple sequence alignment tasks. The integration of a genetic algorithm with the DPAMSA model has notably enhanced performance, particularly evident in datasets with increased sequence length and complexity. While standard tools often exhibit performance degradation under these conditions, GA-DPAMSA maintains consistent and reliable results. However, the framework encounters challenges in achieving optimal full-column alignments in real datasets, as indicated by lower column scores. This suggests potential areas for refinement, such as enhancing the model's capability to handle full-column alignments. Overall, GA-DPAMSA presents a promising approach for multiple sequence alignment, effectively balancing accuracy and consistency across diverse datasets.

## 6 Future Developments

### 6.1 Introduction

This chapter explores a series of potential future advancements, describing new methodologies, technical innovations, and broader applications that can further enhance the performance and ease of use of the framework.

MSA tools have evolved extremely quickly in terms of scalability and accuracy in recent years. Future improvements will most likely result from the combination of sequence alignment with other forms of information, such as homology to more extensive sets of DNA sequences. Choice of parameters for alignment programs is still a current issue, as demonstrated by the sensitivity of benchmarking results to parameter choice. Algorithmically, processing all sequences at once rather than progressive alignment (consistency-based methods being a step in this direction) has been a useful strategy. Finally, better use of phylogenetic relationships and the use of models of DNA sequence evolution also hold promise for improved performance in alignment.

More broadly, sequence space organization will increasingly be important as new sequence and functional information becomes available. For a set of DNA sequences that have just been acquired, with automation, it ought to be feasible to place them into advanced databases that will calculate domain organization, evolutionary relationships, and function.[14]

### 6.2 Mutations during the Mutation Phase

Mutations based on a local DRL policy are crucial for avoiding local optima. Potential future extensions might include:

- **Augmented Local Analysis:** The current strategy splits sequences into overlapping sub-boards and identifies weak regions for targeted mutation. More sophisticated clustering algorithms based on structural and evolutionary similarity could be incorporated, enabling the DRL agent to focus more precisely on the most critical segments.
- **Evolution of the DRL Module:** Adding an ensemble of DRL agents with varying mutation strategies may enhance the system's flexibility and resilience. Techniques such as reward shaping could be employed to balance exploration and exploitation more dynamically, coupled with real-time mutation strategy adaptation.
- **Adaptive Simulated Annealing:** While the current framework employs a simulated annealing function with variable temperature in the acceptance of mutations, future improvements could include an automated temperature control scheme. By adjusting the temperature according to prevailing fitness trends, the algorithm would better explore the solution space.

### 6.3 Enhancements in the Crossover Phase

Crossover is used to merge partial solutions from one or more parents. Future work in this area can involve:

- **Biologically Informed Crossover:** By including evolutionary anchors, the crossover operation can be refined so that recombination events respect functional and structural integrity. A connection to conservation databases might guide which sequences or regions are swapped, minimizing disruption to key domains.
- **Intelligent Post-Crossover Adjustment:** The post-crossover stage currently involves smart gap deletion based on conservation scores. Follow-up work utilizing machine learning tools to predict the effect of gap deletion on overall alignment quality could enable finer, more mindful adjustments, leading to better biological preservation.

### 6.4 Scalability and Computational Efficiency

With increasingly large and complex datasets, computational efficiency is becoming a major concern:

- **Parallelization and GPU Acceleration:** Due to the complexity of the mutation and crossover processes, employing parallel processing architectures, such as GPUs and distributed computing, could considerably reduce processing time. This would allow the processing of larger datasets without sacrificing performance.
- **Intelligent Caching and Pruning:** Applying techniques for caching intermediate values and avoiding unnecessary computations could further refine the evolutionary process. By focusing computational resources on the most probable nodes in the evolutionary tree, the overall efficiency of the system would be significantly improved.

### 6.5 Validation, Benchmarking, and Future Applications

To ensure that the proposed optimizations result in tangible improvements, the following should be performed:

- **Testing on Diverse Datasets:** Expanding experiments to heterogeneous datasets—with varying sequence lengths and complexities—will be crucial in establishing the versatility of the framework. Comparisons with traditional MSA programs and existing high-performance tools will help determine the advantages and limitations of GA-DPAMSA.
- **Parameter Sensitivity Analysis:** Comprehensive studies examining the impact of key parameters (e.g., gap frequency, initial annealing temperature, conservation thresholds) will provide useful guidelines for tuning the system to address specific biological questions.

- **Broader Bioinformatics Applications:** In addition to MSA, GA-DPAMSA methods can be applied to other bioinformatics challenges, such as evolutionary profile comparison and homology-based clustering. Such extensions could utilize the same advances in optimization and computing power to create a universal, high-performance platform.

## 6.6 Conclusions

The GA-DPAMSA system represents a significant contribution to the field of Multiple Sequence Alignment by integrating the efficiency of genetic algorithms with cutting-edge DRL techniques. The future developments outlined in this chapter provide a framework for enhancing all aspects of the system, from mutation to crossover and selection, while simultaneously addressing scalability and validation challenges.

By incorporating adaptive methods, bio-inspired approaches, and advanced computational techniques, GA-DPAMSA has the potential to yield even higher-quality alignments and serve as a robust platform for tackling increasingly complex bioinformatics problems. Continuous refinement of these methodologies holds the promise of not only improving alignment accuracy but also expanding the framework’s applicability to diverse biological challenges, ultimately advancing our understanding of molecular evolution and functional genomics.

## References

- [1] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice”. In: *Nucleic Acids Research* 22.22 (Nov. 1994), pp. 4673–4680. ISSN: 0305-1048. DOI: 10.1093/nar/22.22.4673. eprint: <https://academic.oup.com/nar/article-pdf/22/22/4673/7122285/22-22-4673.pdf>. URL: <https://doi.org/10.1093/nar/22.22.4673>.
- [2] Fabian Sievers and Desmond G Higgins. “Clustal Omega, accurate alignment of very large numbers of sequences”. In: *Multiple sequence alignment methods* (2014), pp. 105–116.
- [3] Kazutaka Katoh and Daron M. Standley. “MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability”. In: *Molecular Biology and Evolution* 30.4 (Jan. 2013), pp. 772–780. ISSN: 0737-4038. DOI: 10.1093/molbev/mst010. eprint: <https://academic.oup.com/mbe/article-pdf/30/4/772/6420419/mst010.pdf>. URL: <https://doi.org/10.1093/molbev/mst010>.
- [4] Robert C. Edgar. “MUSCLE: multiple sequence alignment with high accuracy and high throughput”. In: *Nucleic Acids Research* 32.5 (Mar. 2004), pp. 1792–1797. ISSN: 0305-1048. DOI: 10.1093/nar/gkh340. eprint: <https://academic.oup.com/nar/article-pdf/32/5/1792/7055030/gkh340.pdf>. URL: <https://doi.org/10.1093/nar/gkh340>.
- [5] Yongchao Liu, Bertil Schmidt, and Douglas L. Maskell. “MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities”. In: *Bioinformatics* 26.16 (June 2010), pp. 1958–1964. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq338. eprint: [https://academic.oup.com/bioinformatics/article-pdf/26/16/1958/48854142/bioinformatics\\_26\\_16\\_1958.pdf](https://academic.oup.com/bioinformatics/article-pdf/26/16/1958/48854142/bioinformatics_26_16_1958.pdf). URL: <https://doi.org/10.1093/bioinformatics/btq338>.
- [6] Siavash Mirarab, Nam Nguyen, and Tandy Warnow. “PASTA: ultra-large multiple sequence alignment”. In: *Research in Computational Molecular Biology: 18th Annual International Conference, RECOMB 2014, Pittsburgh, PA, USA, April 2-5, 2014, Proceedings 18*. Springer. 2014, pp. 177–191.
- [7] Nam-phuong D Nguyen et al. “Ultra-large alignments using phylogeny-aware profiles”. In: *Genome biology* 16 (2015), pp. 1–15.
- [8] Cédric Notredame, Desmond G Higgins, and Jaap Heringa. “T-coffee: a novel method for fast and accurate multiple sequence alignment11Edited by J. Thornton”. In: *Journal of Molecular Biology* 302.1 (2000), pp. 205–217. ISSN: 0022-2836. DOI: <https://doi.org/10.1006/jmbi.2000.4042>. URL: <https://www.sciencedirect.com/science/article/pii/S0022283600940427>.

- [9] Ramchalam Kinattinkara Ramakrishnan, Jaspal Singh, and Mathieu Blanchette. “RLALIGN: A Reinforcement Learning Approach for Multiple Sequence Alignment”. In: *2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE)*. 2018, pp. 61–66. DOI: 10.1109/BIBE.2018.00019.
- [10] Yarong Li. “Sequence Alignment with Q-Learning Based on the Actor-Critic Model”. In: *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 20.5 (June 2021). ISSN: 2375-4699. DOI: 10.1145/3433540. URL: <https://doi.org/10.1145/3433540>.
- [11] Aryan Lall and Siddharth Tallur. “Deep reinforcement learning-based pairwise DNA sequence alignment method compatible with embedded edge devices”. In: *Scientific Reports* 13.1 (2023), p. 2773.
- [12] Yuhang Liu et al. “Multiple sequence alignment based on deep reinforcement learning with self-attention and positional encoding”. In: *Bioinformatics* 39.11 (Oct. 2023), btad636. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btad636. eprint: <https://academic.oup.com/bioinformatics/article-pdf/39/11/btad636/52771912/btad636.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btad636>.
- [13] Biswanath Chowdhury and Gautam Garai. “A bi-objective function optimization approach for multiple sequence alignment using genetic algorithm”. In: *Soft Computing* 24 (2020), pp. 15871–15888.
- [14] Robert C Edgar and Serafim Batzoglou. “Multiple sequence alignment”. In: *Current opinion in structural biology* 16.3 (2006), pp. 368–373.