**Artificial Intelligence in Multiple Sequence Alignment**

**Leon Edmiidz (1303038)**

**Bioinformatics Master’s Project (BINF\*6999)**

**Biological Co-advisor: Dr. Jinzhong Fu**

**Computational Co-advisor: Dr. Ayesha Ali**

**External Co-advisor: Dr. Kwokyin Hui**

**August 13, 2024**

**1 University of Guelph, Department of Integrative Biology, 2 University of Guelph, Department of Mathematics and Statistics**

[Abstract 2](#_Toc496418183)

[1. Introduction 3](#_Toc1641176475)

[2. Materials and Methods 6](#_Toc684756608)

[2.1 Compute Nodes 6](#_Toc1354027511)

[2.2 MSA Programs 6](#_Toc1502857941)

[2.3 Data Collection 6](#_Toc235623891)

[2.4 MSA Methods Comparison 7](#_Toc1982830289)

[3. Results 8](#_Toc1568073333)

[3.1 MSA Methods Comparison 9](#_Toc1210216014)

[3.2 Max Episodes Hyperparameter Effects 11](#_Toc896665861)

[3.3 Alignments 14](#_Toc2020964523)

[4. Discussion 17](#_Toc1364370974)

[4.1 DPAMSA Limitations 17](#_Toc205911042)

[4.2 Max Episodes Parameter 18](#_Toc1012662808)

[4.3 Model Training 19](#_Toc334847503)

[4.4 Limitations and Future Directions 20](#_Toc1085508439)

[4.5 Potential in AI-based Techniques 20](#_Toc647574336)

[4.6 Conclusion 21](#_Toc1018545775)

[Acknowledgements 21](#_Toc1118227954)

[Supplementary Links 22](#_Toc459695505)

[References 22](#_Toc1876030984)

# Abstract

Multiple sequence alignment (MSA) is a fundamental tool in bioinformatics, playing a key role in various analyses (Edgar & Batzoglou, 2006). While highly valuable, MSA presents significant computational challenges, particularly with the exponential growth of sequence data generated by advancements in DNA sequencing technology (Zhang *et* al., 2019). Traditional MSA methods often struggle to maintain accuracy and efficiency when dealing with large datasets, especially in regions of low sequence homology (Kemena & Notredame, 2009; Sievers & Higgins, 2017). However, recent progress in artificial intelligence (AI) has opened up new possibilities for improving MSA through AI-based approaches that enhance alignment quality and computational speed. The objective of this project was to analyze the performance of a novel AI-based MSA technique, DPAMSA (Liu *et* al., 2023). We ran DPAMSA and three traditional MSA methods, ClustalW, MAFFT, and T-COFFEE on several datasets and measured the quality of the alignments by calculating the Sum-of-Pairs (SP) score and the Column Score (CS). The findings show that DPAMSA was memory intensive for even small datasets with short sequences, ran magnitudes longer, and was unable to reliably produce high-quality alignment compared to the traditional methods. Although DPAMSA was lackluster in its performance, AI-based MSA is in its very early stages with new technologies that promise faster performance (Dotan *et al*., 2024).

# 1. Introduction

Next-generation sequencing techniques have dramatically increased the availability of genomic data to the point of some researchers calling the modern era of biological research, “the genomic era” (Kemena & Notredame, 2009; Buerki & Baker, 2015; Doyle, 2013). The unprecedented availability of biological data presents a tremendous opportunity for bioinformatics research; however, processing that vast amount of data is computationally challenging. AI can drastically impact research efforts: it can efficiently process large swaths of data, speed up data analyses, and reduce errors during processing. There is lots of potential for AI to heavily impact various areas of research, particularly in life sciences (Holzinger et al., 2023). Multiple sequence alignment (MSA), a key technique in the analyses of biological sequence data, is one area of bioinformatics that could substantially benefit from the AI.

MSA is a cornerstone technique in bioinformatics as it is often one of the early steps in the research process for various types of studies, such as functional gene annotation and vaccine development (Edgar & Batzoglou, 2006; Vaishnav *et* al., 2014). A growing number of biological modelling methods rely on the construction of high-quality MSAs (Kemena & Notredame, 2009). Despite its great benefits, MSA is highly computationally demanding and complex, and its computational requirements and complexity increases with larger datasets (Zhang et al., 2019). The most common MSA techniques to date will often fail to produce high-quality alignments with datasets containing large volumes of sequences, with low-homology regions (Sievers & Higgins, 2017). This has led researchers to investigate whether AI technology could improve MSA beyond traditional techniques (Liu *et al.*, 2023; Dotan *et al*., 2024).

In 2023, an AI-based MSA technique, Deep reinforcement learning with Positional encoding and self-Attention for MSA (DPAMSA) was developed by Lu et al. (2023). DPAMSA combines deep reinforcement learning with the self-attention mechanism, a machine learning (ML) technique that is crucial in natural language processing (NLP). DPAMSA is primarily based on progressive column alignment and calculates the sub-alignment of each column successively (Liu et al*.*, 2023). The combination of the sub-alignments in the appropriate order forms the complete alignment. Once a column is aligned, the state of the sequences is updated (Liu *et al.*, 2023). To align the next column, thereby updating the state, the current state of the sequences is embedded into a vector. During this embedding process, the sequences are concatenated into one string and the nucleotides are converted into numbers. The partition between the sequences is also represented by a number. This state vector is the format required for processing by the Deep Q Network (DQN, a type of deep reinforcement learning model).

Reinforcement learning relies on an environment that is primarily made up of an agent, with state, action, and reward functions (Liu *et al.*, 2023). For DPAMSA, the deep reinforcement learning agent makes decisions based on its interactions with the rest of the environment (Liu *et al.*, 2023) and the vectorized and embedded sequences are the state. The action in this case is inserting a gap or not inserting a gap. The reward function is represented by the SP score of the aligned positions, which is influenced by the scoring matrix, a parameter for MSAs (Liu *et al.*, 2023).

The DQN's Q network consists of three main components: positional encoding, self-attention mechanism, and a multi-layer perceptron. Positional encoding gives the model an understanding of the position of each element in the sequence. This is crucial for the self-attention mechanism to perform properly (Liu *et al.*, 2023). Self-attention allows the model to weigh the importance of different parts of the sequence when making decisions and allows for features to be extracted based on the state of the sequences. The multi-layer perceptron, a type of neural network used for learning complex patterns in data, is used to process all the extracted features to decide on the action to take by calculating the Q value (Liu et al., 2023). This combination enables the DQN to learn effective strategies for inserting gaps into the sequence, thereby improving the overall quality of the alignment (Liu et al., 2023).

This project delves into the capabilities of DPAMSA by evaluating its performance on several datasets containing DNA sequences from the genus of Enteroviruses, including Rhinoviruses. We hypothesize that DPAMSA will produce higher-quality alignment compared to the traditional methods but require more computational power and execution time. The aim is to compare the performance of DPAMSA to those of three classic MSA programs, ClustalW, MAFFT, and T-COFFEE.

# 2. Materials and Methods

## 2.1 Compute Nodes

The programs and scripts executed for this project were all run on High-Performance Computing (HPC) clusters accessed through The Digital Research Alliance of Canada (the Alliance). DPAMSA was run on a 12.8 GB GPU node (Tesla P100-PCIE-12GB @ 1190MHz). The traditional MSA methods were allocated 4 GB CPU (Intel E5-2683 v4 Broadwell @ 2.1GHz) memory when running.

## 2.2 MSA Programs

DPAMSA was pulled from the git report of Lui et al (2023). Mafft (version: 7.471) was provided as a loadable module on the Alliance’s Cedar clusters. T-COFFEE (version: 13.46.0.919e8c6b\_linux\_x64) was downloaded from the T-COFFEE website. ClustalW (version: 2.1-linux-x86\_64) was downloaded from the ClustalW website. Please refer to the supplementary links section for the websites.

## 2.3 Data Collection

Full genome sequences from enterovirus A, B, C and D and rhinovirus A, B and C were downloaded from the NCBI nucleotide database. Two sequences were downloaded for each species, with each sequence composed of about 7 kilobases. This dataset was then aligned by the traditional MSA method, T-Coffee, to locate regions of high, medium, and low homology between the sequences. From this, sub-sequences of varying lengths and levels of homology were extracted for use in the comparative analyses.

## 2.4 MSA Methods Comparison

To ensure that DPAMSA is being run properly, we tested DPAMSA on a small example dataset (Dataset Test) provided in the original paper. The dataset contains 3 sequences with the average length of the sequences being 23 base pairs long. Once DPAMSA was validated, we tested the size of dataset that it could handle. Initial tests of DPAMSA found that the program would crash or generate an error message when the dataset was too large. After multiple test runs with varying dataset sizes, it was decided that the datasets used in this comparative analysis would consist of 14 sequences of 100 base pairs each. To assess performance on sequences of varying homology, three different datasets were selected: a high homology dataset (Dataset A), a medium homology dataset (Dataset B), and a low homology dataset (Dataset C). These three datasets were then aligned by the four MSA techniques.

The performances of the techniques were analyzed by the qualities of the alignments and the run times. The alignment qualities were quantified by two scoring metrics, the SP score and the CS. SP score is a commonly used scoring mechanism to quantify the qualities of MSAs (Hamada *et al*., 2009). It is based on an adjustable scoring matrix and is the sum of scores of all the possible pairwise alignments. The CS score measures the fraction of columns in the alignment where all the sequences match (Thompson *et al*., 1999).

A key hyperparameter in DPAMSA that affects alignment quality and run time, is Max Episodes. This parameter adjusts the number of alignment iterations performed by the model to find the most optimal alignment, analogous to epochs in ML (Liu et al*.*, 2023). A range of values were used to test the effects of this parameter on the performance of DPAMSA.

To test if DPAMSA would run better on datasets containing shorter sequences, Dataset A was replicated and altered. Dataset A30 and A60 were created by shortening all the sequences in Dataset A to 30 and 60 base pairs long, respectively. Further testing was done to analyze whether DPAMSA would perform better when the sequences in the dataset slightly varied in length, like Dataset Test. Instead of having all the sequences in Datasets A, A30 and A60 be the same length, the sequences were altered so that the average sequence lengths are 100 for Dataset A, 30 for Dataset A30, and 60 for Dataset A60. These new datasets with slightly varying sequence lengths were titled Datasets AX, A30X, and A60X, respectively.

# 3. Results

## 3.1 MSA Methods Comparison

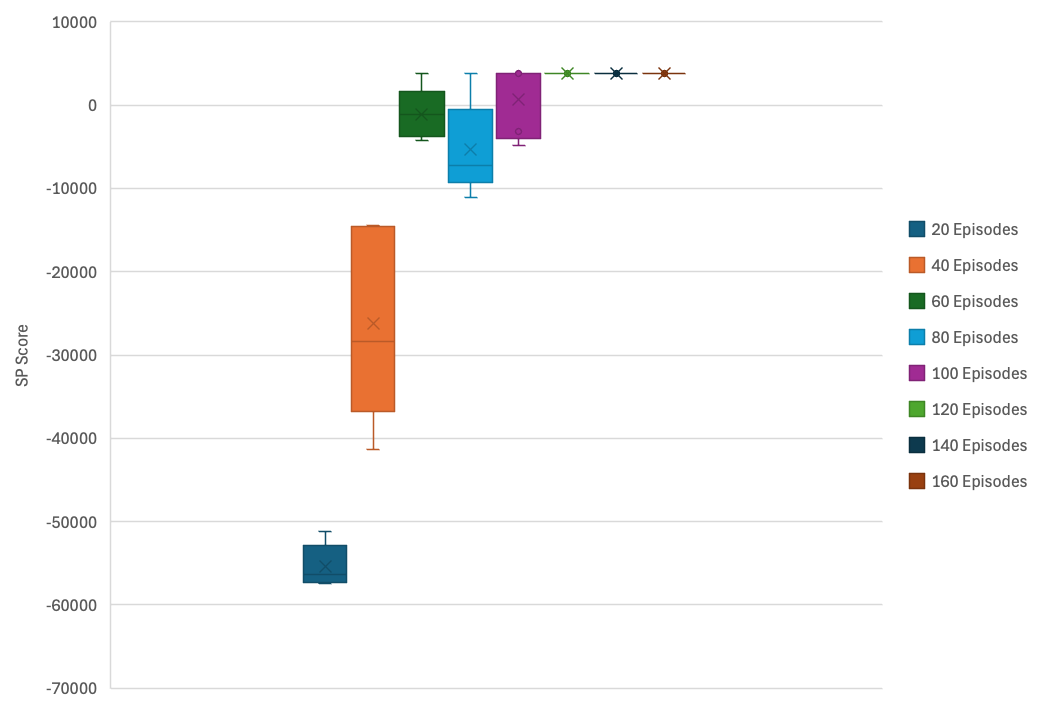
DPAMSA was successfully validated on Dataset Test and performed better than the traditional methods, as it did in the original paper. However, DPAMSA consistently performed worse compared to the traditional methods for all the other datasets. Table 1 shows that the SP and CS scores for the alignments produced by DPAMSA are lower than those of the traditional methods for most datasets. SP scores for DPAMSA were often negative while the traditional methods achieved positive scores in the thousands. In addition to the lower quality alignments, DPAMSA took magnitudes longer to run.

**Table 1**: Results table to compare the performance of the four MSA techniques. Higher SP score and CS score indicate higher quality alignments. Lower run time indicates computational efficiency.

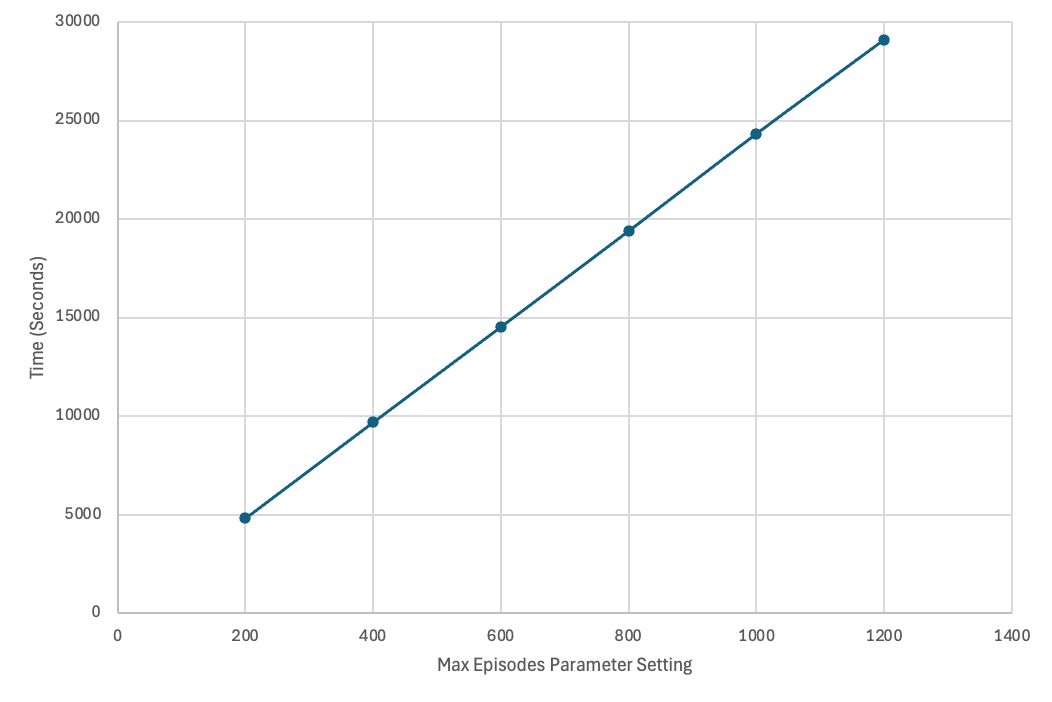
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software** | **Dataset** | **SP score** | **CS score** | **Time (seconds)** |
| DPAMSA | A | 3800 | 0.15 | 1995.000 |
| ClustalW | A | 19064 | 0.54 | 0.343 |
| MAFFT | A | 19064 | 0.54 | 0.381 |
| T-COFFEE | A | 18916 | 0.51 | 3.308 |
| DPAMSA | B | 10104 | 0.29 | 2918.820 |
| ClustalW | B | 10236 | 0.30 | 0.308 |
| MAFFT | B | 10104 | 0.29 | 0.098 |
| T-COFFEE | B | 10348 | 0.30 | 3.911 |
| DPAMSA | C | -14552 | 0.00 | 3770.230 |
| ClustalW | C | -14676 | 0.02 | 0.127 |
| MAFFT | C | -27112 | 0.01 | 0.098 |
| T-COFFEE | C | -20096 | 0.01 | 3.786 |
| DPAMSA | AX | -2712 | 0.01 | 18124.000 |
| ClustalW | AX | 15916 | 0.45 | 0.241 |
| MAFFT | AX | 15916 | 0.44 | 0.285 |
| T-COFFEE | AX | 14548 | 0.42 | 3.021 |
| DPAMSA | A30 | -8524 | 0.02 | 26172.500 |
| ClustalW | A30 | 3796 | 0.45 | 0.100 |
| MAFFT | A30 | 3796 | 0.45 | 0.247 |
| T-COFFEE | A30 | 3244 | 0.39 | 1.867 |
| DPAMSA | A60 | -33324 | 0.00 | 778.540 |
| ClustalW | A60 | 8008 | 0.40 | 0.094 |
| MAFFT | A60 | 8008 | 0.40 | 0.262 |
| T-COFFEE | A60 | 7300 | 0.34 | 2.538 |
| DPAMSA | A30X | -12800 | 0.00 | 23725.580 |
| ClustalW | A30X | 1060 | 0.13 | 0.348 |
| MAFFT | A30X | 1076 | 0.13 | 0.276 |
| T-COFFEE | A30X | 732 | 0.10 | 2.328 |
| DPAMSA | A60X | -15944 | 0.02 | 76524.720 |
| ClustalW | A60X | 4484 | 0.21 | 0.026 |
| MAFFT | A60X | 4156 | 0.19 | 0.256 |
| T-COFFEE | A60X | 3776 | 0.19 | 2.655 |
| DPAMSA | Test | 152 | 0.73 | 395.150 |
| ClustalW | Test | 92 | 0.56 | 0.065 |
| MAFFT | Test | 100 | 0.60 | 0.276 |
| T-COFFEE | Test | 76 | 0.52 | 1.585 |

## 3.2 Max Episodes Hyperparameter Effects

The Max Episodes hyperparameter was positively correlated with alignment quality, except at 80 episodes as shown in Figure 1. A higher setting for the parameter also increased the run time. For Datasets A, B, and C, the quality of the alignments, as measured by SP score, plateaued near 100 Max Episodes. Figure 1 shows that DPAMSA could not achieve a better SP score than 3800 for Dataset A.

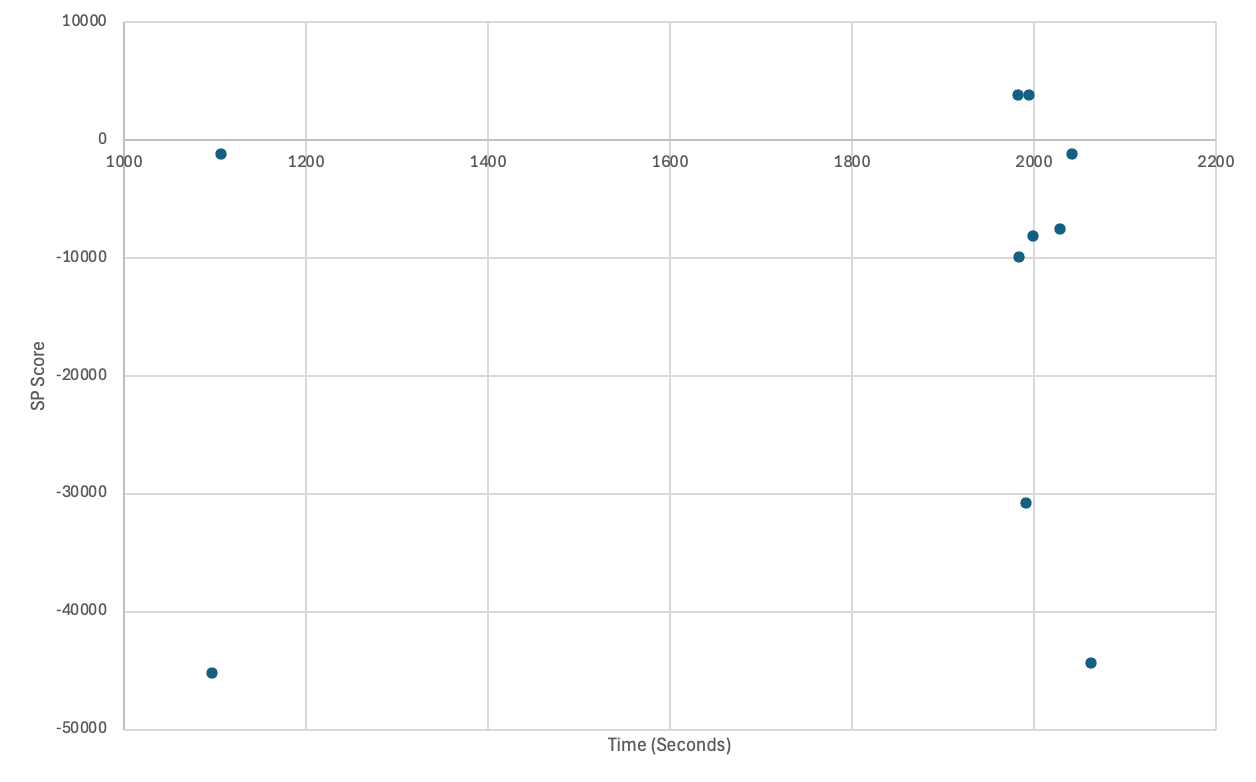


**Figure 1**: Max Episodes vs. SP score from running DPAMA runs on Dataset A.

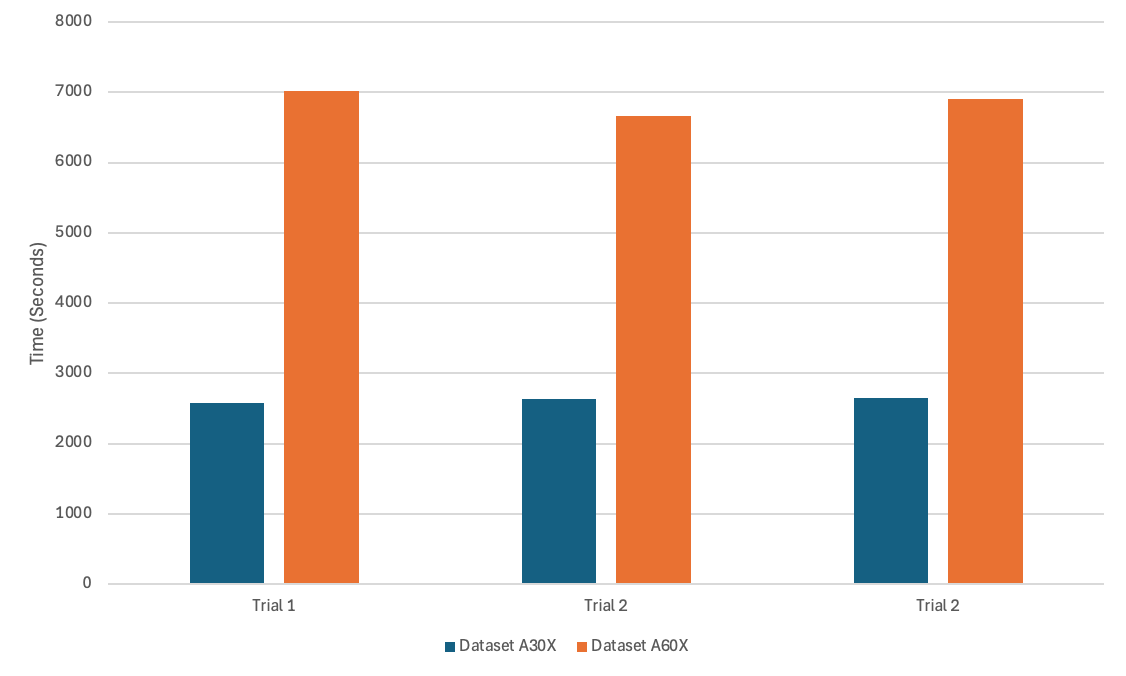


**Figure 2**: Linear relationship between the Max Episodes Parameter and the run time for DPAMSA performed on Dataset C.

There is some degree of randomness in the alignments, therefore alignments can differ between runs on the same dataset with the same parameter settings. We found that at the same parameter settings, larger datasets take longer to align compared to smaller datasets, as shown in Figure 4.



**Figure 3**: Run**-**to**-**run variation in **SP score** between same dataset with **identical** Max Episodes. This data was collected from DPAMSA runs on Dataset A with Max Episodes set to 50.



**Figure 4**: Runtimes for datasets of different sizes at the same Max Episodes setting (1000). Dataset A60X contains sequences that are double the length of those in Dataset A30X.

## 3.3 Alignments

The alignments for Datasets A, B, and C for DPAMSA and the best performing traditional methods are shown below. These alignments are the best results produced by DPAMSA, yet did not include any gaps. Check the supplementary alignments PDF for the complete set of alignments for Datasets A, B, and C.

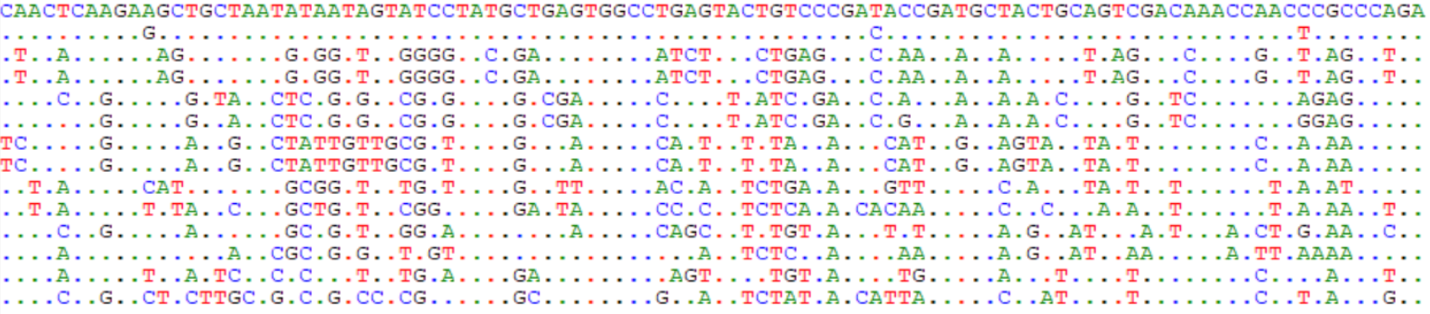
**DPAMSA Dataset A**



**MAFFT Dataset A**



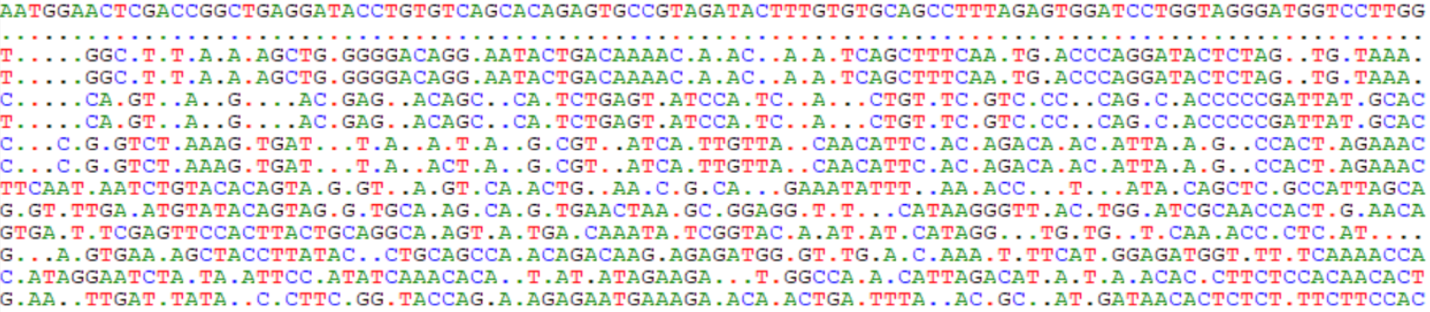
**DPAMSA Dataset B**



**T-COFFEE Dataset B**



**DPAMSA Dataset C**



**ClustalW Dataset C**



# 4. Discussion

## 4.1 DPAMSA Limitations

DPAMSA exhibits multiple limitations that hinder its effectiveness and practicality compared to traditional MSA methods. A major limitation of DPAMSA that came up early in this project was its ability to only work with small datasets. When working with datasets containing 14 sequences, the average length of the sequences could not exceed 100 base pairs long on a 12.8GB GPU. The complete dataset containing the whole genome of the Enteroviruses caused the program to prematurely terminate, meanwhile, the traditional methods had no issues with the complete genomes. While a more powerful GPU would likely have allowed DPAMSA to work with larger datasets, the traditional methods of MSA have no problems running alignments on datasets containing around 100 sequences of several thousand base pairs on desktop computer systems (Katoh & Toh, 2008). DPAMSA’s reliance on large amounts of GPU memory is certainly a limitation for its performance but also in terms of accessibility as many people who want to run MSA may not have access to powerful GPU nodes.

Accessibility of DPAMSA is a limitation not only in terms of GPU access but also in ease of running the program. The GitHub for DPAMSA contains multiple scripts with no documentation to explain how the scripts are to be used. The original code also appears to accept an unspecified format. For this project, the code had to be edited to allow for the input of FASTA files. The code must be read and analyzed before users can get the program to run.

As shown in Figure 1, the quality of DPAMSA, as measured by the SP score, plateaued for most of the datasets even as the Max Episodes was increased. The highest quality alignments had unaltered sequences or gaps at the ends. This suggests that the DPAMSA was unable to insert beneficial gaps. When gaps were inserted in the middle, they ended up being deleterious as these alignments had extremely low SP scores. This is a major flaw, as inserting gaps in the appropriate locations to improve alignments is the purpose of MSA.

When comparing the performance of DPAMSA to those of the traditional methods, it is evident that it often fails to outperform them in various types of datasets. The traditional methods outperformed DPAMSA in alignment quality for most of the datasets tested in this study. Not only did DPAMSA produce poor alignments with low SP and CS scores, but it also took magnitudes longer to run compared to the traditional methods. This is strong evidence which points to the fact that DPAMSA is not an improvement compared to traditional techniques of MSA.

## 4.2 Max Episodes Parameter

The Max Episodes Parameter is undoubtedly a key parameter that heavily influences the performance of the program. Although there is a general positive correlation between the number of episodes and the performance of DPAMSA, a greater number of episodes leads to longer run times. It was found that even with large numbers of episodes, the program would occasionally “glitch” to produce poor alignments after running for hours or days.

## 4.3 Model Training

The unique design of DPAMSA, which includes real-time model training, significantly impacts its performance and speed. As DPAMSA runs and the sequence state is updated, the information regarding the environment is stored in the replay memory. Once the replay memory contains enough data (set by the batch size parameter), the state and action data are transferred to a verification Q network, meanwhile the state and action data from the following step are transferred to the target Q network (Liu et al., 2023). Both Q networks have the same network structure and calculate their respective Q values (Liu et al., 2023). The following loss function is used to calculate the discrepancy between the verification Q value and the target Q value,

*Loss* = (At - Qv)2

where At is the largest action value from the target Q network and Qv is the verification Q value. This allows for the optimization of the parameters of the verification Q network through backpropagation, improving upon the verification Q network with each iteration. This allows the model to learn from its previous iterations, continuously working to find the most optimal alignment (Liu et al., 2023). This design for DPAMSA requires the model to be trained on the dataset it is working with as it aligns the sequences. This dramatically slows DPAMSA down as model training is a part of the MSA process. This is untypical for AI models as they are usually trained beforehand on a dataset to ensure that the predictions made by the model can be calculated quickly. This, however, explains why DPAMSA took significantly longer to run than the traditional MSA methods.

## 4.4 Limitations and Future Directions

The greatest limitation of this project was the short duration. More comprehensive analyses could have been performed if more time was available. Given the effects of the Max Episodes parameter, it would be interesting to see if the program would successfully break through the plateaus that it encountered if it was given a much higher number of episodes to run. If the Max Episodes parameter is set to hundreds of thousands, perhaps DPAMSA would be able to produce higher quality alignments at the expense of a much longer run time. DPAMSA is currently only capable of utilizing one GPU node. If the code is altered to give it the ability to run on multiple GPU nodes to perform different tasks in parallel, DPAMSA could potentially run significantly faster and even work with larger datasets.

## 4.5 Potential in AI-based Techniques

Despite the current limitations of DPAMSA, there is potential for significant improvement in AI-based MSA techniques. DPAMSA is one of the earliest versions of AI-based MSA, released only last year. It is important to keep in mind that AI-based MSA is still in its earliest stages of development. The technology is relatively new and will likely only continue to improve (Holzinger et al., 2023). Some AI-based MSA techniques focus on protein sequences and predicting amino acid folding patterns such as AlphaFold and DeepMSA (Jumper *et* al., 2021; Zhang *et al*., 2019). Many of these promising novel MSA techniques are still in the works and are not yet available for open-source use. Another example is BetaAlign, introduced by Dotan et al. (2024). BetaAlign is a transformer model-based MSA technique that uses an ensemble of transformers to perform MSA. The model is pre-trained on various types of datasets and uses a majority voting technique for the transformers to predict which alignment is likely the most accurate. Although the program has not yet been released, it does seem to show promising preliminary results.

## 4.6 Conclusion

While the novel AI-based MSA technique, DPAMSA, holds promise for enhancing sequence alignments, our study revealed its current limitations compared to traditional methods like ClustalW, MAFFT, and T-COFFEE. DPAMSA's reliance on substantial GPU memory and prolonged run times, coupled with its inability to produce high-quality alignments, underscore the need for further refinement and optimization in the technique. Further exploration of DPAMSA with extensive Max Episodes and parallel GPU processing could yield improvements. Despite the drawbacks of DPAMSA, AI-based MSA is in its infancy, and future advancements hold great potential. The emergence of innovative techniques like BetaAlign underscores the rapid progress in this domain. As AI-based MSA matures, it is poised to revolutionize bioinformatics, enabling more efficient and accurate analyses of vast biological sequence data.

# Acknowledgements

Thank you very much to the computational coadvisor, Dr. Ayesha Ali, the biological coadvisor, Dr. Jinzhong Fu, and the external coadvisor, Dr. Kwokyin Hui. Their contributions to this project make its completion possible.

# Supplementary Links

<https://github.com/ZhangLab312/DPAMSA>

<http://www.clustal.org/clustal2/>

<https://tcoffee.org/Projects/tcoffee/index.html#DOWNLOAD>

# References

Dotan, E., Wygoda, E., Ecker, N., Alburquerque, M., Avram, O., Belinkov, Y., & Pupko, T. (2024). Betaalign: A deep learning approach for multiple sequence alignment. *bioRxiv*. https://doi.org/10.1101/2024.03.24.586462

Edgar, R. C., & Batzoglou, S. (2006). Multiple sequence alignment. *Current Opinion in Structural Biology*, *16*(3), 368–373. <https://doi.org/10.1016/j.sbi.2006.04.004>

Hamada, M., Sato, K., Kiryu, H., Mituyama, T., & Asai, K. (2009). CentroidAlign: Fast and accurate aligner for structured RNAS by maximizing expected sum-of-pairs score. *Bioinformatics*, *25*(24), 3236–3243. <https://doi.org/10.1093/bioinformatics/btp580>

Holzinger, A., Keiblinger, K., Holub, P., Zatloukal, K., & Müller, H. (2023). AI for Life: Trends in Artificial Intelligence for Biotechnology. *New Biotechnology*, *74*, 16–24. https://doi.org/10.1016/j.nbt.2023.02.001

Katoh, K., & Toh, H. (2008). Recent developments in the MAFFT multiple sequence alignment program. *Briefings in Bioinformatics*, *9*(4), 286–298. https://doi.org/10.1093/bib/bbn013

Kemena, C., & Notredame, C. (2009). Upcoming challenges for multiple sequence alignment methods in the high-throughput era. *Bioinformatics*, *25*(19), 2455–2465.<https://doi.org/10.1093/bioinformatics/btp452>

Liu, Y., Yuan, H., Zhang, Q., Wang, Z., Xiong, S., Wen, N., & Zhang, Y. (2023). Multiple sequence alignment based on deep reinforcement learning with self-attention and positional encoding. *Bioinformatics*, *39*(11). https://doi.org/10.1093/bioinformatics/btad636

Russell, D. J. (Ed.). (2014). *Multiple sequence alignment methods*. Totowa, NJ: Humana Press.

Sievers, F., & Higgins, D. G. (2017). Clustal Omega for making accurate alignments of many protein sequences. *Protein Science*, *27*(1), 135–145.<https://doi.org/10.1002/pro.3290>

Thompson, J. D., Plewniak, F., & Poch, O. (1999). A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*, *27*(13), 2682–2690. <https://doi.org/10.1093/nar/27.13.2682>

Vaishnav, N., Gupta, A., Paul, S., & John, G. J. (2014). Overview of computational vaccinology: Vaccine development through information technology. *Journal of Applied Genetics*, *56*(3), 381–391.<https://doi.org/10.1007/s13353-014-0265-2>

Zhang, C., Zheng, W., Mortuza, S. M., Li, Y., & Zhang, Y. (2019). DEEPMSA: Constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics*, *36*(7), 2105–2112.<https://doi.org/10.1093/bioinformatics/btz863>