Benchmarking Study: Evaluation of Multiple Sequence Alignment Algorithms

Likita Gangireddy[1], Neeharika Kotimreddy[2], Manvir Chahal[3], and Rhea Kak[4]

[1]Computer Science, 2024, lg425
[2]Computer Science, 2024, nrk59
[3]Computer Science, 2024, mkc222
[4]Computer Science, 2024, rkk66

ABSTRACT

Multiple Sequence Alignment on biological sequences such as DNA, RNA, and proteins, is a fundamental step in determining things such as protein structure prediction, protein family identification, and phylogeny estimation amongst various biological sequences [1]. There are several algorithms that currently exist to determine these sequence alignments. However, each of these algorithms have individual strengths and weaknesses, and can vary in performance depending on the input sequence. We have conducted this benchmark study to provide scientists with better insights as to which algorithm to use when aligning certain sequences. The algorithms that we focus on in this study include Multiple Sequence Comparison by Log-Expectation (MUSCLE), multiple alignment using fast Fourier transform (MAFFT), Clustal Omega, and Kalign. This benchmark uses sequence data of varying lengths from the BAliBASE Reference 1 and Reference 9. We examine computational speed, alignment accuracy when compared to the reference alignment, and Sum-of-Pairs scores which indicates the quality of the pairwise alignments within the sequence. Our results indicate that on shorter sequences MAFFT is the fastest algorithm, MUSCLE is the most accurate, and MAFFT has the highest quality of pairwise alignments. For longer sequences, our results indicate that MUSCLE is the fastest and has the best quality of pairwise alignments, and Kalign is the most accurate. An overall weighted scored based on these metrics indicates that MUSCLE may be the most optimal for short sequences, whereas MAFFT can be the most optimal for longer sequences.

Keywords (minumum 5): Multiple Sequence Alignments, MUSCLE, MAFFT, Clustal Omega, Kalign, Progressive Alignment

Project type: Benchmark

Project repository: https://github.com/likitag/benchmarkStudyCS4775.git

1 Introduction

In the world of computational biology, the alignment of biological sequences plays a pivotal role in unraveling evolutionary relationships among diverse biological entities. Thus, Multiple Sequence Alignment (MSA) is a very important process in bioinformatics as the complexity of these sequences necessitates the use of sophisticated algorithms to align them accurately. This benchmarking study aims to evaluate and compare the performance of four prominent MSA algorithms: MUSCLE, MAFFT, Kalign, and Clustal Omega.

While the existing algorithms demonstrate diverse strengths and weaknesses, their effectiveness can vary based on the characteristics of the input sequences. In this study, we conduct a comprehensive assessment, considering crucial aspects such as computational speed, alignment accuracy concerning reference sequences, and Sum-of-Pairs (SP) scores. Our primary objective is to provide valuable insights to the scientific community, guiding the selection of the most appropriate algorithm for specific sequence alignment scenarios. An important step of all the algorithms is progressive alignment. Progressive Alignment is an important heuristic used in many multiple sequence alignments. Progressive alignment works by building the full alignment progressively, by initially computing pairwise alignments using methods such as the Needleman-Wunsch algorithm and Smith-Waterman algorithm, and then these sequences are clustered together using methods such as k-means [2]. While progressive alignment is a valuable approach for MSA, it cannot guarantee the identification of the global optimal alignment due to its heuristic nature. Nonetheless, it remains a widely adopted strategy in the field of bioinformatics.

Progressive alignment serves as the fundamental building block for all four algorithms examined in this study: MAFFT, MUSCLE, Clustal Omega, and Kalign. These algorithms leverage progressive alignment as a critical component of their alignment processes, facilitating the accurate alignment of multiple sequences, a pivotal task in various biological and computational applications.

Benchmarking MSA algorithms serves a pivotal role in ensuring that the plethora of computational sequence alignment tools available today align with the ever-evolving demands of modern biological research. By carefully evaluating the advantages and disadvantages in various MSA algorithms, researchers can make informed choices when selecting the most suitable algorithm tailored to their specific needs. Additionally, the results of benchmarking aids in the interpretation of alignment results, shedding light on the alignment quality and its biological implications.

Furthermore, the practice of benchmarking encourages the continual refinement of alignment methods over time. The results of these sequence alignments have a wide array of applications that are critical in advancing our comprehension of biological structures. Notable examples include precise protein structure prediction, the identification of potential drug targets, and the comprehensive understanding of protein functions within biological systems.

Exisiting research in MSA evaluation has primarily focused on improving alignment accuracy and speed. More recent studies have highlighted the importance of considering memory efficiency when dealing with larger data sets and resource-constrained environments; for the purposes of this study, we will be looking at accuracy and speed.

2 Methods

2.1 Metrics

In our benchmarking study, we evaluated multiple sequence alignment algorithms by examining three fundamental metrics: computational speed, which determines the efficiency; alignment accuracy, crucial for ensuring reliability; and sum-of-pairs (SP) scores, vital for understanding the alignment's quality across sequence pairs. This assessment approach provided a holistic understanding of each algorithm's strengths and weaknesses.

Speed is a very crucial metric to study when evaluating MSA algorithms because many scenarios of multiple sequence alignment involve a substantial amount of sequences which require aligning. This can be a computationally intensive task to perform. Thus, in order to compute alignments efficiently, it is essential to use an MSA algorithm which optimizes speed. To calculate the computational speed of each algorithm, we used the built in python time module to keep track of the start time and end time of each algorithm, and note the difference.

Accuracy is another key metric that can help determine the quality of a certain algorithm. Accuracy is a vital component in benchmarking since it provides an assessment of the reliability of the MSA algorithms' outputs, which is crucial for proper interpretation of evolutionary relationships. These interpretations can be especially crucial when computing alignments for critical medical and biological research. In order to measure the accuracy of each algorithm, we implemented a custom accuracy function which compares the alignment outputted by the algorithm, with the reference alignment provided in the dataset. The accuracy function does an element wise comparison by finding the number of matching characters with the reference alignment, and dividing it by the total length of the sequence.

Sum-of-pairs is a fundamental method used in evaluating multiple sequence alignments. The sum-of-pairs (SP) score provides a measurement of the quality of a particular alignment by calculating a score for each pairwise alignment in a sequence, and summing up these scores. This provides important insights as to how well each pair of sequences aligns at each position, and maximizing this SP score can prove to be an important guideline when selecting an appropriate algorithm. We have implemented a custom function to calculate the sum-of-pairs scores. This function calculates a score based on the number of matching pairs in each column of a two-dimensional array which represents the sequence alignments. It checks each column and compares every possible pair of elements within that column. The score is incremented whenever a pair of matching elements is found.

2.2 Algorithms

All four of the algorithms we have chosen to utilize in this study are key analysis tools used in the field of genetics. Clustal Omega utilizes HMM profile-profile techniques, MUSCLE makes use of a log expectation algorithm, MAFFT uses the Fast Fourier Transform algorithm, and Kalign utilizes optimal global alignment.

Clustal Omega is known to be particularly effective when handling substantial datasets. It starts by constructing a distance matrix through k-tuple matching, which is necessary to assess sequence differences. Then, it uses this matrix to create a guide tree that allows the algorithm to methodically align sequences, starting from most similar to least similar. Specifically, the mBed

algorithm in Clustal Omega streamlines the process of guide tree construction for sequence alignment by comparing sequences to a subset of representative seed sequences, significantly reducing computational time and memory. This efficient approach, which includes a bisecting k-means clustering and tree-building routines, facilitates the alignment of large datasets, but the resulting guide tree is intended only for alignment guidance, not for phylogenetic analysis [3]. The progressive alignment approach taken by Clustal Omega allows for accuracy in results to be retained.

MAFFT is recognized for being good at balancing speed and precision, with its use of the Fast Fourier Transform (FFT) for initial sequence comparisons and progressive alignment. Utilizing FFT speeds up the alignment process, and once the guide tree is constructed, the algorithm aligns the sequences progressively. Depending on the characteristics of the dataset, the approach to alignment is tailored. The algorithm accepts unaligned sequences in FASTA format, outputting alignments in either FASTA or CLUSTAL formats. It implements several methods for sequence alignment, including the progressive methods (FFT-NS-1 and FFT-NS-2), iterative refinement (FFT-NS-i), and consistency-based iterative refinement methods (G-INS-i, L-INS-i, and E-INS-i), each catering to different alignment needs with varying balances of speed and accuracy [4]. This unique approach makes MAFFT a widely-used tool for large-scale sequence alignments.

MUSCLE, prioritizing accuracy, begins with a draft alignment before progressing to guide tree construction using the Unweighted Pair Group Method with Arithmetic Mean (UPGMA). This method clusters sequences based on average distances, a strategy that outperforms neighbor-joining in aligning profiles with fewer differences at each node of the guide tree. MUSCLE employs two distance measures for sequence alignment: the k-mer distance for quick evaluation of unaligned pairs based on common subsequences, and the Kimura distance for aligned pairs, adjusting for multiple substitutions at a site [5]. This comprehensive approach enables MUSCLE to achieve high accuracy, especially for medium-sized datasets.

Kalign is known to do well with time efficiency and uses the Wu-Manber algorithm to efficiently match strings and create a distance matrix, which is used to build the guide tree. Once the progressive alignment is performed, more robust but quick refinement is performed with time in mind. Kalign uses advanced scanning methods for estimating pairwise sequence distances, comparing the first 256 characters of shorter sequences across longer ones, and employs a reduced alphabet for protein sequences to improve accuracy with distantly related sequences. It adopts Clustal Omega's guide tree construction, clustering sequences using a bi-secting k-means algorithm based on distances to seed sequences, a process accelerated by AVX (Advanced Vector Extensions) instructions for efficiency. To ensure optimal clustering, Kalign repeats this algorithm 50 times with randomly chosen seed sequences, enhancing the accuracy of its groupings [6]. Kalign's method makes it a popular choice for sequence alignment when dealing with large-scale alignments where time efficiency is vital.

2.3 Computational Pipeline

To perform this benchmark study, we have followed several key steps to attain our results. The first step in our pipeline was to study various prior benchmark studies, and select a few key metrics that we think would be most suitable and feasible for this project. Initially, we considered examining memory usage in addition to the other three metrics which are speed, accuracy, and SP scores. However, we had decided to not include memory as a measure due to the complex nature, and decided to instead prioritize the other three metrics.

The next stage in our pipeline involved selecting data to use from the two reference sets. We selected 10 data sets of short sequences, and also 10 data sets of medium to long sequences, and separated these two. In the BAliBASE dataset, each file of sequences included and input .tfa file with the unaligned sequence, and a reference .msa file. The first step to use both the input an reference files in out benchmarking algorithm was to convert them to .fasta format, which was done using the AlignIO module from the Biopython library.

Before we could begin constructing our Python script, we initially had to go through all the installation steps for the 4 MSA algorithms: MSUCLE, MAFFT, Kalign, and Clustal Omega. Once the installation was complete, we began implementation of our benchmarking algorithm within the Python script. The core functionality of this algorithm takes as input an algorithm to run, the path to the input sequence, and the path for where to write the respective output alignment of the specified MSA algorithm. Within the main function, we calculate computation time, accuracy score, and SP scores. For each input sequence, we run our benchmark algorithm 4 times - once using each MSA algorithm.

Results from the above algorithm were collected for both the set of larger sequences and for the set of short sequences, and were written to a .txt file. Once these results were acquired, we then plotted two sets of graphs using Google Sheets. One set of graphs was for the shorter sequences from Reference Set 1, and another set of graphs for the longer sequences from Reference Set 9. Each graph compares the 4 algorithms based on a specified metric. The final step of our pipeline involves examining these results, evaluating the advantages and disadvantages of each MSA algorithm, and providing an overall ranking of each algorithm using a custom weighted score that takes into account each observed metric.

2.4 Datasets

The primary data source we chose for our project was from the BAliBASE 4 database, which is an Multiple Sequence Alignment benchmarking reference set that contains various sequences and motifs. The link to the database is as follows: http://www.lbgi.fr/balibase/. The creators of the BAliBASE data set have based their test cases on 3D structural superpositions, which are refined to ensure correct alignment, and sorted their alignments into various reference sets that represent real multiple alignment problems [7]. We utilized Reference 9 (linear motifs) for medium to long lengths of input sequences and Reference 1 (variability, length) for short input sequences. In particular, Reference 9 consists of four subsets of linear motifs from various protein families, and the majority of these linear motifs have a length of 3 to 10 amino acids [3]. We ran all the chosen algorithms on both the medium to long and short sequences from both references so that we could gain a better understanding of how algorithm performance increases or decreases based on the length of the sequences.

3 Results

We have created multiple graphical representations to assess the performance of each algorithm across three fundamental metrics: computational speed, accuracy, and memory usage. Furthermore, to better grasp the scalability of these algorithms, we have generated graphs that contrast the algorithms' performance when handling short sequences from Balibase Reference Set 1 with their performance on medium to long sequences from Balibase Reference Set 9. Additionally, we have conducted a comparison of Sum-of-Pairs (SP) scores for each algorithm using both types of data. The SP function is a method of assessing alignment quality which has been shown to provide a reasonable trade-off between structural correctness and computability [8]. We have constructed 2 graphs for each metric. The red column graph represents data collected from BAliBASE Reference Set 1 and the blue column graph represents data collected from BAliBASE Reference Set 9. We have also constructed two additional graphs comparing a computed overall score for each algorithm on both data sets.

Figures 1 and 2 compare the average computational speed in milliseconds among the 4 algorithms. For the MUSCLE algorithm, the average speed was 14.25 ms for Ref Set 1 and 10981.8 ms for Ref Set 9. For the MAFFT algorithm, the average speed was 84.37 ms for Ref Set 1 and 536.03 ms for Ref Set 9. For the Clustal Omega algorithm, the average speed was 4.82 ms for Ref Set 1 and 1746.4 ms for Ref Set 9. For the Kalign algorithm, the average speed was 3.25 milliseconds for Ref Set 1 and 133.85 ms for Ref Set 9.

Figures 3 and 4 compare the average accuracy scores among the various algorithms. For the MUSCLE algorithm, the average accuracy was 0.49 for Ref Set 1 and 0.31 for Ref Set 9. For the MAFFT algorithm, the average accuracy was 0.15 for Ref Set 1 and 0.297 for Ref Set 9. For the Clustal Omega algorithm, the average accuracy was 0.24 for Ref Set 1 and 0.33 for Ref Set 9. For the Kalign algorithm, the average accuracy was 0.37 for Ref Set 1 and 0.418 for Ref Set 9.

Figures 5 and 6 compare the average SP scores among the various algorithms. For the MUSCLE algorithm, the average SP score was 345.22 for Ref Set 1 and 1606793 for Ref Set 9. For the MAFFT algorithm, the average SP score was 407 for Ref Set 1 and 986978 for Ref Set 9. For the Clustal Omega algorithm, the average SP score was 322 for Ref Set 1 and 802931 for Ref Set 9. For the Kalign algorithm, the average SP score was 295.6 for Ref Set 1 and 476771 for Ref Set 9.

In Figures 7 and 8, we have computed an overall score using the following scoring equation:

$$0.4 \cdot \text{accuracy} + 0.3 \cdot \text{normalized SP score} - 0.3 \cdot \text{normalized computational time}$$

For the MUSCLE algorithm, the overall score was 0.221 for Ref Set 1 and 0.124 for Ref Set 9. For the MAFFT algorithm, the overall score was 0.061 for Ref Set 1 and 0.262 for Ref Set 9. For the Clustal Omega algorithm, the overall score was 0.134 for Ref Set 1 and 0.22 for Ref Set 9. For the Kalign algorithm, the overall score was 0.148 for Ref Set 1 and 0.167 for Ref Set 9. Below, we have included each of the 6 graphs for your reference.

4 Discussion

4.1 Analysis

Our results have given us several insights into the advantages and disadvantages of each algorithm. These insights can help aid in decision-making when assessing which algorithm to use for a particular sequence. Our calculation of computational time show that the Kalign takes the least amount time to run for the BAliBASE Reference 1, whereas the MAFFT algorithm takes significantly longer to run on this data set compared to the other 3 algorithms. This can indicate that in terms of computational speed on sequences of shorter lengths, Kalign may be the best algorithm to use, whereas MAFFT would not be preferable. Calculations of computational speed on BAliBASE Reference Set 9 also indicate that Kalign is the fastest. This indicates that the Kalign algorithm is consistently an optimal algorithm in terms of speed for both short sequences and large sequences.

However, when the MAFFT algorithm was run on Reference Set 9, it was the second fastest algorithm, with a speed fairly similar to that of the Kalign algorithm. This indicates that despite MAFFT's inefficiency in terms of speed on short sequences, MAFFT can prove to be a very time efficient algorithm to use on larger sequences.

When examining the accuracies of each algorithm, we observed that MUSCLE was most accurate and MAFFT was the least accurate for shorter sequences. For the longer sequences, Kalign was the most accurate, whereas the other three algorithms had accuracy scores that were slightly lower than KAlign, but not by a significant amount.

The SP score calculations demonstrated that MAFFT has the highest SP score for shorter sequences. However, all the algorithms had fairly similar SP scores. For larger sequences MUSCLE had a notably higher SP score and Kalign had a notably smaller SP score compared to the other algorithms. Thus, in situations where local alignment quality is emphasized, MUSCLE may be the optimal choice and Kalign may not be preferable.

The overall score for each algorithm, taking into account all of the factors listed above, indicate that MUSCLE is the most optimal algorithm to use on shorter sequences, and MAFFT is the most optimal algorithm to use on larger sequences.

4.2 Further Research

In discussing further research, we can first consider the use of MAFFT and Kalign MSA research in emerging methodologies. Both the MAFFT and Kalign approaches have established themselves as important tools in bioinformatics, which is known for distinctive approaches and algorithms in sequence alignment. However, as the field evolves, we also need to revisit and reassess these tools for the evolved standards and datasets. Further research can analyze and compare further performance metrics of MAFFT and Kalign. Although far more nuanced than our conducted research, discussing new, more descriptive metrics in relation to algorithm evaluation can give us a more comprehensive understanding of performance and efficiency in relation to specific data sets. For example, we chose to omit memory as a metric in our study due to its complexity and the scope of our project; however, looking further into memory could prove useful when considering data set size, sequence length, multi-core processing, and resource constraints.

On top of new advances and evolution in MSA approaches, expanding our research on MSA algorithm evaluation means we need to consider new metrics and refine existing metrics to fit a larger, more applicable scope. In another recent study titled "MAGUS+eHMMS: improved multiple sequence alignment accuracy for fragmentary sequences", researchers focused on addressing MSA in data sets consisting of both full-length and fragmentary sequences, which is a challenge frequently encountered by various biological sequence analyses [9]. The UPP (Ultra-large alignments using Phylogeny-aware Profiles) method used to tackle this issue employs PASTA (Progressive Alignment with Scoring Trees and Approximate Bayesian Inference) for the backbone alignment of full-length sequences and integrates fragmentary sequences using an eHMMs (ensemble Hidden Markov Model) technique. This study, however, introduces a novel approach called MAGUS+eHMMs, which replaces PASTA with MAGUS. This approach demonstrated superior alignment accuracy when compared to UPP, which marks a significant advancement in the field [10]. This study introduced additional metrics for evaluation beyond the traditional metrics of alignment speed and accuracy. These supplementary metrics encompassed aspects such as robustness to sequence heterogeneity, the ability to handle datasets with varying levels of fragmentation, and the impact of evolutionary rates on alignment precision; these metrics were more detailed, specific, and comprehensive for comparison and evaluation of MSA [9]. This multifaceted approach utilized in this study underscores the importance of a nuanced evaluation when selecting MSA algorithms by taking into account the unique attributes of the dataset and challenges it presents. This is very necessary when expanding such research to real-world applications where ongoing research and innovation require more nuance in the dynamic field of biological sequence analysis.

Furthermore, it is vital that we go beyond the scope of this project and understand the real-world implications of our work and how other researchers are building and expanding upon MSA algorithms and evaluation. One such example would be the recent study titled "Recursive MAGUS: Scalable and accurate multiple sequence alignment" in PLOS Computational Biology, which introduces a significant advancement in the field of multiple sequence alignment (MSA), especially in the context of large-scale datasets [10]. This study introduces MAGUS (Multiple Sequence Alignment using Graph Clustering), an innovative divide-and-conquer alignment method that utilizes the Graph Clustering Merger (GCM) technique. This recent development marks a significant evolution from traditional MSA approaches. Originally capable of aligning up to 40,000 sequences, this study documents the extension of MAGUS to handle data sets significantly larger, showcasing its ability to align up to one million sequences; scalability is significantly increased while still maintaining speed and accuracy [10]. This extension of MSA could be groundbreaking for phylogenetics, genomic analysis, functional genomics, drug discovery, and various fields of biomedical research involving analysis of gene families, mutations, diseases.

Author Contributions

This section is required to all project groups with more than a single member. Place group member's name under the appropriate contribution.

- Study design: Manvir, Likita, Neeha, Rhea
- Coding: Likita
- Experiments: Likita, Neeha
- Analyses: Manvir, Rhea
- Writing:
  - *Introduction:* Rhea, Neeha
  - *Methods:* Manvir, Likita
  - *Results:* Likita
  - *Discussion:* Likita, Neeha

References

1. Nute M, Saleh E, Warnow T, 2018. Benchmarking statistical multiple sequence alignment. *bioRxiv*, .

2. Daugelaite J, O' Driscoll A, Sleator RD, 2013. An overview of multiple sequence alignments and cloud computing in bioinformatics. *ISRN Biomathematics*, 2013:615630.

3. Sievers F, Higgins DG, 2014. *Clustal Omega, Accurate Alignment of Very Large Numbers of Sequences*, pages 105–116. Humana Press, Totowa, NJ.

4. Katoh K, Asimenos G, Toh H, 2009. *Multiple Alignment of DNA Sequences with MAFFT*, pages 39–64. Humana Press, Totowa, NJ.

5. Edgar RC, 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797.

6. Lassmann T, 2019. Kalign 3: multiple sequence alignment of large datasets. *Bioinformatics*, 36(6):1928–1929.

7. Thompson JD, Koehl P, Ripp R, Poch O, 2005. Balibase 3.0: Latest developments of the multiple sequence alignment benchmark. *Proteins: Structure, Function, and Bioinformatics*, 61(1):127–136.

8. Chatzou M, Magis C, Chang JM, Kemena C, Bussotti G, Erb I, Notredame C, 2015. Multiple sequence alignment modeling: methods and applications. *Briefings in Bioinformatics*, 17(6):1009–1023.

9. Shen C, Zaharias P, Warnow T, 2021. MAGUS+eHMMs: improved multiple sequence alignment accuracy for fragmentary sequences. *Bioinformatics*, 38(4):918–924.

10. Smirnov V, 2021. Recursive magus: Scalable and accurate multiple sequence alignment. *PLOS Computational Biology*, 17(10):1–17.

Figures

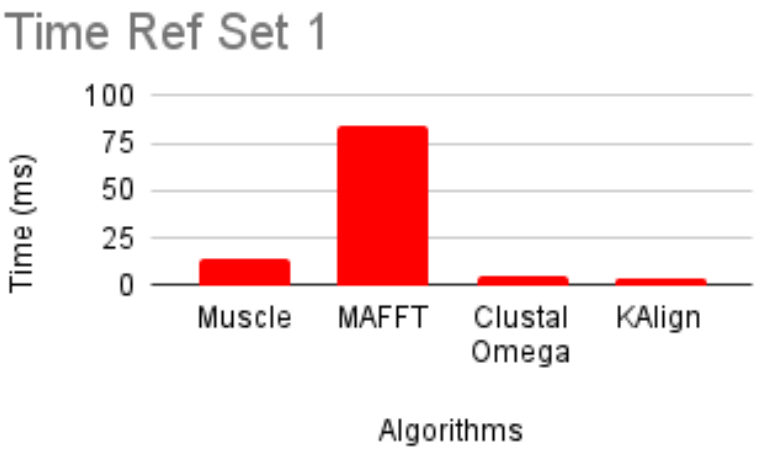

Figure 1. Comparison of computational speed measured in milliseconds amongst the 4 different sequencing algorithms on the input sequences from Balibase Reference Set 1.

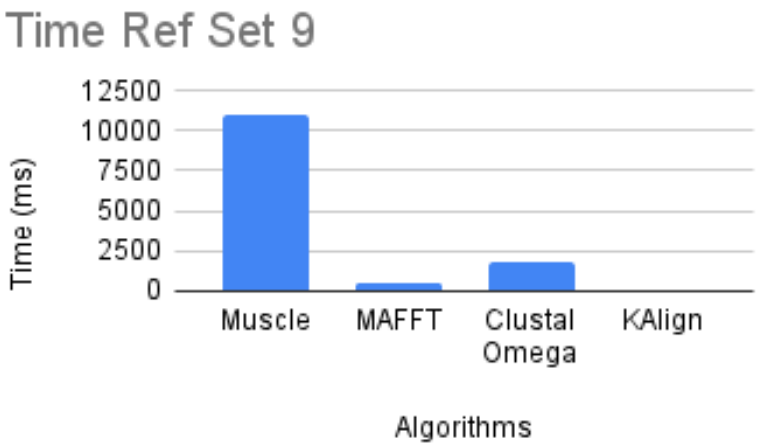

Figure 2. Comparison of computational speed measured in milliseconds amongst the 4 different sequencing algorithms on the input sequences from Balibase Reference Set 9.

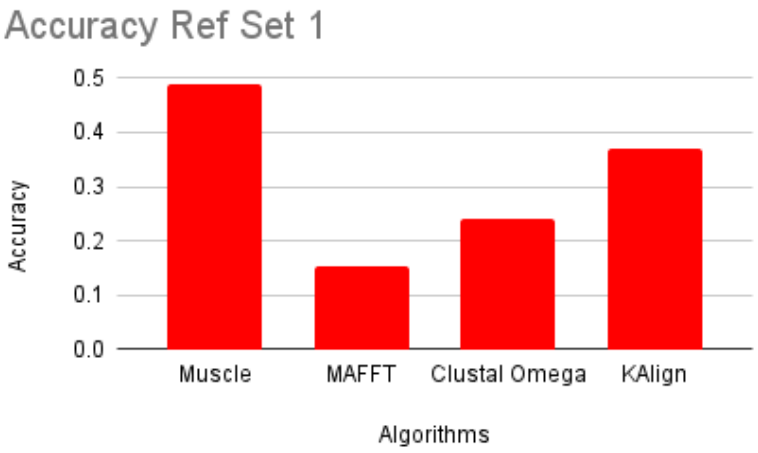

Figure 3. Comparison of accuracy scores amongst the 4 different sequencing algorithms on the input sequences from Balibase Reference Set 1.

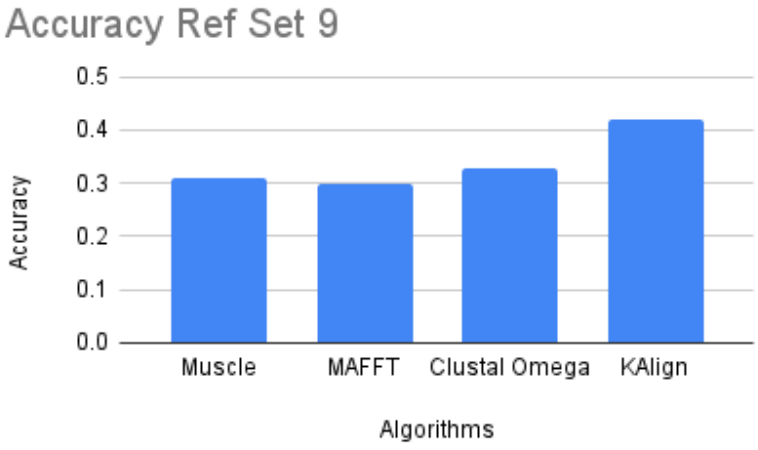

Figure 4. Comparison of accuracy scores amongst the 4 different sequencing algorithms on the input sequences from Balibase Reference Set 9.

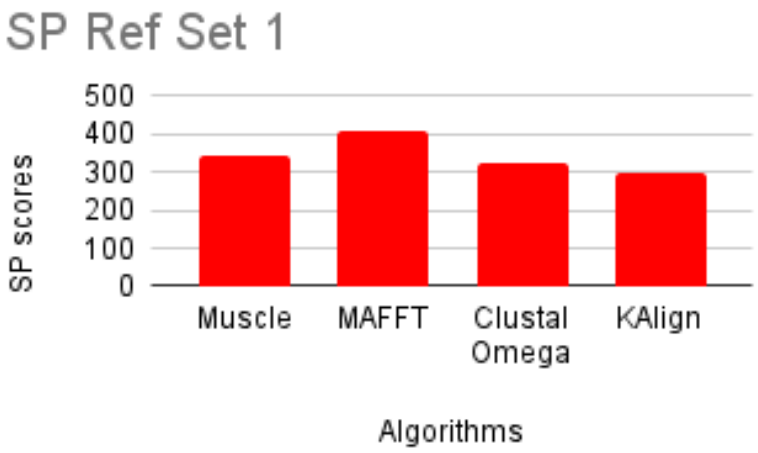

Figure 5. Comparison of SP scores amongst the 4 different sequencing algorithms on the input sequences from Balibase Reference Set 1.

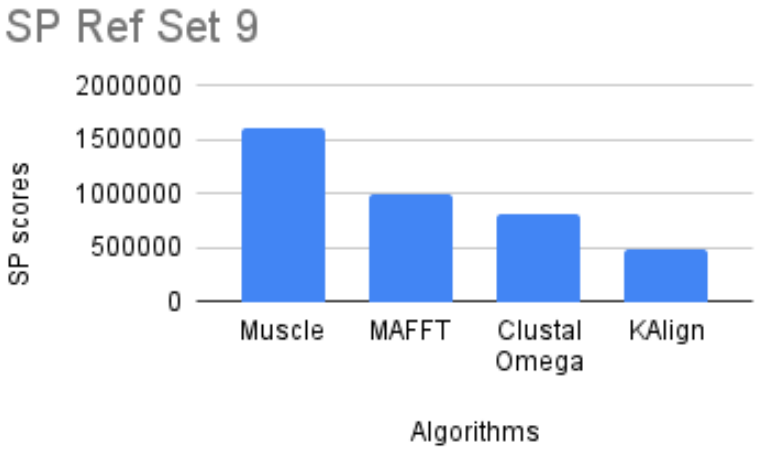

Figure 6. Comparison of SP scores amongst the 4 different sequencing algorithms on the input sequences from Balibase Reference Set 9.

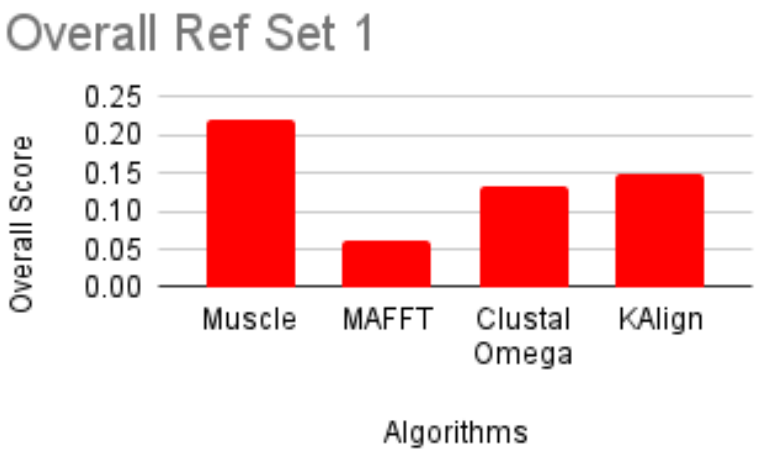

Figure 7. Comparison of overall scores amongst the 4 different sequencing algorithms on the input sequences from Balibase Reference Set 1.

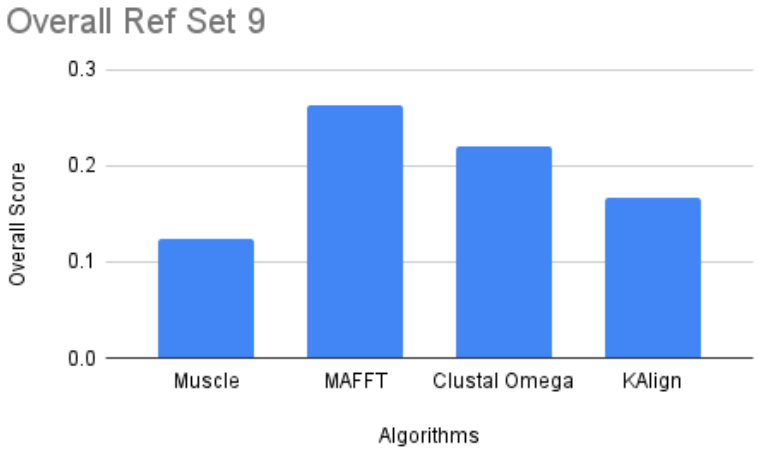

Figure 8. Comparison of overall scores amongst the 4 different sequencing algorithms on the input sequences from Balibase Reference Set 9.