Improving Performance of Multiple Sequence Alignment through Maximal Exact Match Identification

Tim Wehning*
BSc Advanced Technology
University of Twente
Enschede, The Netherlands

***Abstract*—Multiple sequence alignment is an integral part in the field of DNA analysis and genomics, and it is necessary in order to properly identify evolutionary patterns as well as functional motifs. However, one of its biggest drawbacks is scalability. Execution times increase rapidly with larger numbers of sequences to be aligned. In this paper a new approach is presented, that takes the concept of seed-and-extend algorithms from pairwise sequence alignment and applies it to multiple sequences. The result is an alignment tool called MEMSA (MEM Extracting Multiple Sequence Aligner), which applies multiple pre-processing steps in order to reduce the search space of alignment. It shows promising results for data sets with a high homology but struggles with genomic sequences that are too divergent. For a data set of 500 MERS genomes, the tool of this paper was able to reduce the execution time for alignment by a factor of 27 while even improving alignment quality slightly.**

The source code of the developed tool is available online at https://github.com/timweh/MEMSA

***Keywords*—**genomics, multiple sequence alignment, maximal exact matches

I. Introduction

A. Motivation

The analysis of biological sequences plays a crucial role in various domains of bioinformatics, aiding in the understanding of genetic information. Multiple sequence alignment (MSA) is a fundamental technique used for aligning and comparing multiple sequences simultaneously, revealing conserved regions, evolutionary patterns, and functional motifs. However, one of the MSA limitations is its high computational demand when processing large sequence databases. Efficiently aligning large sets of sequences remains a significant challenge due to the non-linear asymptotic time complexity of traditional alignment algorithms like MAFFT [1] or MUSCLE [2]. These algorithms exhaustively compare every possible combination of characters, resulting in computational bottlenecks when dealing with extensive datasets. Consequently, the need for novel approaches to enhance the computational efficiency of MSA algorithms has become increasingly apparent.

This paper aims to address the computational inefficiencies of MSA by proposing a methodology that uses the power of maximal exact match (MEM) identification. MEMs are defined as subsequences that appear across different sequences, representing conserved regions. By identifying these seeds and only selectively aligning the subsequences between them, the search space for alignments can be reduced significantly and computational efforts focussed. Thus, redundant comparisons are minimized and the overall scalability of MSA algorithms is enhanced, enabling more effective analysis of large-scale sequence data sets. In this paper, we focused on the implementation of this suggested method, which will be presented under the name MEM Extracting Multiple Sequence Aligner, or MEMSA.

B. Purpose of the Research

The key objective of this research is to develop an innovative approach that incorporates MEM identification into the MSA process, aiming to achieve substantial computational efficiency improvements without compromising the quality of sequence alignments.

To achieve this objective, there are two main goals that will be targeted in this paper:

- Firstly, we will explore existing algorithms for seed identification, investigate their strengths and limitations, and propose modifications or novel strategies to optimize their utility in the context of MSA.
- Secondly, we will also evaluate the impact of the approach proposed in this paper on computational efficiency by comparing it to an existing state-of-the-art MSA algorithm using real-world sequence data sets.

By improving the computational efficiency of MSA, this research has the potential to accelerate the analysis of biological sequences, allowing researchers to more efficiently compare large quantities of genomes. This can lead to a better understanding of evolutionary relationships and identification of functional motifs, which are key parts of biological research.

In summary, this paper aims to present a comprehensive investigation into the integration of MEM identification techniques within the MSA framework with the goal of increasing computational efficiency while maintaining alignment accuracy and preserving valuable biological insights.

*Research conducted at the Computer Architecture for Embedded Systems (CAES) group of the Faculty of Electrical Engineering, Mathematics, and Computer Science (EEMCS) at the University of Twente, under supervision of Dr. ir. Nikolaos Alachiotis. [July 2023]

II. Background

A. DNA Analysis

A key part of the field of bioinformatics is genomics, which deals with the analysis of the genomic sequences of DNA. In bioinformatics, genomic sequences are represented by strings of characters that correspond to the four different nucleobases that make up the DNA: A for adenine, T for thymine, G for guanine, and C for cytosine. A few more characters exist that are ambiguous and may represent two or more nucleobases.

When DNA is sequenced, the machines can only read small parts of the genome. Due to inaccuracies, these reads may have errors. When trying to read a whole genome, a lot of these reads will therefore be collected and then one can try to reconstruct the full genome by correctly putting these reads together. This can be achieved by using different sequence alignment techniques.

Sequence alignment techniques are not only used to arrange small reads but also used to align whole genomes, in order to allow them to be compared. By aligning whole sequences, similar and dissimilar regions can be identified easily and it can be seen how these sequences are related by evolution. From some well-conserved regions that are called motifs, one can even deduce information about the characteristics and functionality of different species.

There are a lot of different alignment algorithms that tackle this problem in different ways. The most fundamental method is pairwise sequence alignment.

B. Pairwise Sequence Alignment

Pairwise sequence alignment describes the alignment of one sequence with another. The standard for pairwise sequence alignment was set with the Needleman-Wunsch [3] algorithm in 1970. It is a dynamic programming algorithm that is used for global (also end-to-end) alignments. This method was developed further and made more universal by Smith-Waterman [4] which allows local alignments. Both algorithms use a so-called scoring matrix that assigns scores to potential alignments by positively considering matches between characters and negatively accounting for mismatches and the insertion of gaps. The highest score in the matrix will be determined in order to create the optimal alignment by tracing back. These dynamic programming algorithms produce very exact alignments but run very slowly due to their computational complexity.

A faster alternative method called BLAST [5] was published in 1990 that uses an approximation to create alignments. The algorithm decreases the search space by taking small segments from the sequence, so-called seeds, and finding matches in the reference sequence that serve as anchors for the alignments. This concept serves as the foundation for more modern alignment algorithms of the seed-and-extend principle, such as BWA-MEM [6] and Bowtie2 [7] which are commonly used for the alignment of reads [8]. The backbone of this method is a heuristic that identifies seeds in both sequences, which are usually exact matches of a certain size. These seeds are then used to reduce the alignment to an extension phase between the seeds. The identification of exact matches varies between different algorithms but is usually either maximum exact matches (MEMs) or k-mers, that have a fixed size. MEMs are seeds that cannot be further extended to either side, meaning they are adjacent to mismatches on both sides. These alignment algorithms are a lot faster than the aforementioned traditional approaches, however they cannot guarantee an optimal alignment. One of the most commonly used tools for pairwise alignment, MUMmer [9], also uses this principle. Its alignment generator for nucleotides, NUCmer, is optimized for globally aligning large-scale sequences.

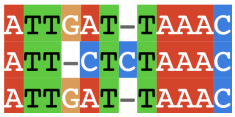

Fig. 1: Example of an MSA, created with MView [10]. Gaps (represented by a "–" dash) are inserted to maximize matching columns.

C. Multiple Sequence Alignment

In order to properly draw conclusions about the evolutionary relationship between sequences and to find motifs in order to predict biological function, multiple sequences need to be aligned at once. Multiple Sequence Alignment (MSA), thus, was developed for this purpose. It is a bioinformatics technique that is able to align three or more biological sequences, such as DNA, RNA, or protein sequences, in order to identify similarities and differences among them. By aligning multiple sequences, MSA provides valuable insights into evolutionary relationships, functional motifs, conserved regions, and structural properties. Conserved regions often correspond to functional elements, such as protein domains or regulatory motifs. Because MSA enables the detection of evolutionary relationships, it provides a framework for understanding the evolutionary history of organisms. MSA also plays a crucial role in comparative genomics, as it enables for the identification of genetic variations and the detection of mutations associated with certain diseases or phenotypic traits.

MSA aligns sequences in a manner that maximizes the overall similarity while considering gaps and insertions. The alignment is typically represented as a matrix, where each row corresponds to a sequence and each column represents a position in the alignment (see Figure 1). The alignment process involves identifying conserved residues across sequences and optimizing their alignment based on a scoring scheme that considers sequence similarity and gap penalties.

Several computational algorithms have been developed to perform MSA. Some of the most prominent ones are ClustalW [11] and T-Coffee [12], which are progressive alignment methods. Progressive alignment algorithms construct the alignment progressively by building a guide tree based on pairwise sequence similarities and then aligning sequences accordingly. Other methods that combine this progressive technique with an iterative method are MAFFT [13] and MUSCLE [2]. Iterative methods refine the alignment iteratively by employing profile-based strategies, where a profile of previously aligned sequences is used to align new sequences.

III. Related Work

There has been a lot of development in the past to identify seeds. The most popular tool currently is MUMmer [9], a method identifying maximal unique matches (MUMs) which are MEMs that are unique in both sequences. Another notable open-source tool is slaMEM [14]. While both tools make use of variable-size seeds, A. Kutzner et al. [15] proposed a combination of fixed-size and variable-size seeding, which showed promising results in the context of PacBio reads. Furthermore, there have been attempts to accelerate the seeding process, specifically for supramaximal exact matches (SMEMs) [16] and for very large genomes [17].

For MSA, MAFFT [13] is a progressive tool that uses a fast Fourier transform approximation. Its FFT-NS-2 algorithm consistently performs as one of the best in benchmark studies [18]. Due to its fast execution and still accurate alignment, it is therefore one of the best tools available. One weakness of MAFFT and most other MSA tools, however, is that they do not recognize homologies within the input sequences that could simplify the alignment.

B. Morgenstern et al. [19] made use of the ability to give user-defined anchor points as constraints for MSA with DIALIGN [20]. This requires some knowledge or assumptions about the homology of the sequences. The results were mixed: for some sets of input sequences, they were able to enforce getting meaningful alignments, whereas, for some other inputs, the anchors had a negative impact. The requirement of meta knowledge about the sequences is a major drawback because obtaining this information requires additional steps so it would negatively impact alignment speed.

F. Pitschi et al. [21] further developed the idea of using anchor points in sequences to put restrictions on the alignment algorithm. The detection of anchor points was automatized so this method does not require any prior knowledge about homologies in the data set anymore. The proposed method managed to achieve improved accuracy when applying these anchors to ClustalW. For T-Coffe and DIALIGN, it did not achieve any improvements. Its MS4 column partial detection scheme purely aims to maximize alignment quality, which happens at the expense of alignment speed due to the complexity of the scheme.

IV. Methodology

A. Approach

This new method takes the concept of seed-and-extend alignment and applies it to MSA. When aligning multiple sequences we expect that there will be certain motifs that are well-conserved across all the sequences. After identifying all these matching regions, they can be used as anchors, and only the parts in between need to be aligned. Therefore, this approach should be viewed as an improvement to established MSA algorithms through preprocessing and applying an additional heuristic, rather than a full MSA itself.

In the first step, an arbitrary sequence from the input sequences is chosen and used as a reference to find all pairwise MEMs between that reference and the rest of the sequences. In the second step, these pairwise generated seeds are compared to each other in order to find the intersections between them. This is done in a pairwise recursive fashion until only seeds that are present in every sequence are left.

```
Seq_1: TGCCGTGACGACTGTACGCTTACTGCATGCGCGG
Seq_2: GGTGCTCGTGACGCTGCTTCTGCATGCGCGAT
Seq_3: CTCGTGACGACTGGCTGCATGCGAGTT
```

Fig. 2: Example sequences with MEMs (colored) of minimum size 5 identified

```
Seq_1: --TGC-CGTGACGACTGTACGCTTACTGCATGCGCGG-
Seq_2: GGTGCTCGTGACG-CTG----CTT-CTGCATGCGCGAT
Seq_3: ----CTCGTGACGACTG---G----CTGCATGCGAGTT
```

Fig. 3: Example sequences after aligning subsequences not contained in seeds

Due to the nature of genomic mutations, a lot of mismatches that bound seeds consist of only a single character as a result of a point mutation. The resulting gaps between the seeds are trivial to align because they are of the same length and optimally aligned already, so they do not require any gap insertions. Therefore, in order to avoid unnecessary calls of the MSA algorithms, gaps between the seeds will be merged in the third step. Since there might be multiple consecutive point mutations within a distance less than the minimum seed size chosen for the MEM identification, these gaps in between seeds, which are trivial to align, might be longer than just one character. Therefore, the maximum size of gaps to be merged can be set as an input parameter.

After the previous step, all the subsequences that are not contained in the seeds (example shown in Figure 2) get extracted and MSA is performed on them (the result can be seen in Figure 3). This results in n+1 executions of MSA, where n is the number of seeds. However, the sequences that need to be aligned are significantly smaller than the full sequence and because traditional MSA algorithms have a non-linear time complexity, that means that execution will be faster.

In the last step, the results of all the individual alignments are recombined with the seeds, and the full alignment is recreated. The general procedure is shown in Figure 4.

B. Implementation

A C++ tool for this study was created which takes a FASTA file as input and produces an aligned FASTA file with inserted gaps as output. These files contain the strings that represent the nucleobases of the genome. This tool is integrated with slaMEM for the MEM identification and MAFFT, and its FFT-NS-2 algorithm, for the MSA process due to these algorithms' public accessibility. The default minimum seed size is 20 when calling slaMEM, which is the default value for most seed-and-extend algorithms. The index file generated by slaMEM is read by the tool and converted into an internal seed data structure. The seeds will be checked to see if they are in the wrong order or redundant, and if either is the case, the program terminates. This implies that either the chosen minimum seed size was too small or that the input sequences are too divergent. In the former case, the code can be run

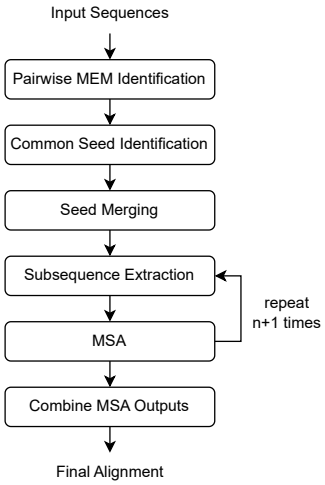

Fig. 4: MEMSA pipeline with n seeds

again with a higher minimum seed size to check if this fixes the issue. If the latter is the case, the input data set has to be chosen more carefully or sequences that are too divergent need to be removed manually. In case seeds overlap, the overlapping part is excluded from both seeds. For every MAFFT call on the resulting subsequences, a FASTA file, which contains the subsequences, is created and the output FASTA file from MAFFT is read. The partial alignments are then recombined into a single output FASTA file containing the final alignment. The name of our developed tool is called MEMSA, a recursive acronym for MEM Extracting Multiple Sequence Aligner.

C. Evaluation

In order to evaluate the performance of this method and to compare it to a simple MSA without any preprocessing, multiple tests are being performed. Since the main goal of this new approach is to improve execution time, this is the first metric that is being measured and the main point for comparison. The second metric is the quality of the alignment, because improved speed only matters if the quality of the alignment does not suffer too much. There are a lot of different ways for assessing the quality of an alignment, such as sum of pairs (SP) and column score (CS), but they can only be used if there is a reference alignment of the same sequences available [22]. The most common reference for alignments is BAliBASE [23]. However, there have recently been some criticisms regarding their non-transparent benchmark calculations [24].

Since we want to be able to test our tool with any input sequences, we do not use reference alignments but instead, introduce our own two indicators. Both of them range from 0 (completely misaligned) to 1 (perfectly aligned), thus a high score is desirable. The first one is the mismatching-to-matching columns ratio (MMCR). A column counts as a match if and only if the whole column matches across all input sequences. A mismatched column occurs as soon as there is at least one mismatching character (gaps are ignored).

This quality indicator is defined as

$$MMCR = \frac{matches}{matches + mismatches} \tag{1}$$

and ranges from 0 (not a single matching column) to 1 (not a single mismatch). If there are only matching columns that contain gaps it will not be defined, but that is an unrealistic situation because a gap insertion usually has a higher penalty than a mismatch.

The other metric is average matches per column (AMPC) and measures how many characters are matching in each column. In order to do that, it counts how many occurrences of each character there are in each column and takes the maximum value, so the number of occurrences of the character that occurs the most in said column. This value is being calculated for all columns and then reduced by one. After that, the average between all columns is taken and normalized (divided by the amount of sequences that were aligned minus one).

It is defined as

$$AMPC = \frac{\sum_{i=0}^{l} max(\#a_i, \#c_i, \#g_i, \#t_i) - 1}{l(n-1)} \tag{2}$$

with l being the length of the alignment and n being the amount of sequences. It ranges from 0 (no column contains 2 or more same characters) to 1 (every columns contains only identical characters). These metrics are used in combination because they focus on slightly different aspects of the alignment. The drawback of MMCR is that it assigns the same score to all columns that contain a mismatch or a gap, no matter how many matches or mismatches they contain. AMPC takes these into account. The drawback of AMPC, however, is that it is sensitive to gap insertions, which can lower its score. Therefore, both metrics are considered to make sure that certain alignment tendencies are not favoured over the other.

To evaluate the performance of this program, it will be tested with a collection of genomes from different SARS-CoV-2 and MERS strains obtained from the National Center for Biotechnology Information (NCBI) Virus Variation Resource. [25]

V. Results

A. MERS Virus

The first tests were performed with a set of 500 MERS genomes. For these runs, the default parameters (minimum seed length = 20, maximum merge gap = 1) were used and the execution time and quality of alignment were measured as a function of the number of input sequences. The results were compared to the equivalent alignment performed by MAFFT FFT-NS-2.

As can be seen in Figure 5, the execution time takes a lot longer with just MAFFT than the preprocessed one for larger numbers of sequences. At n = 500 MAFFT took 34:11 minutes whereas MEMSA took only 1:15 minute. That corresponds to a factor of 27.

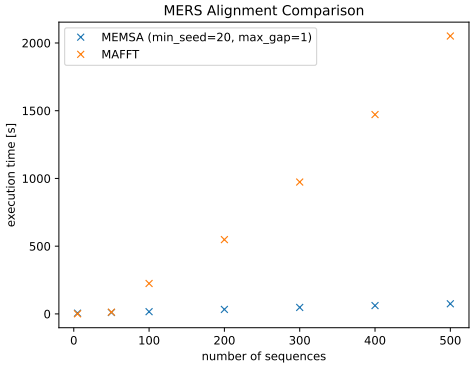

Fig. 5: Comparison of alignment speed between MEMSA and MAFFT

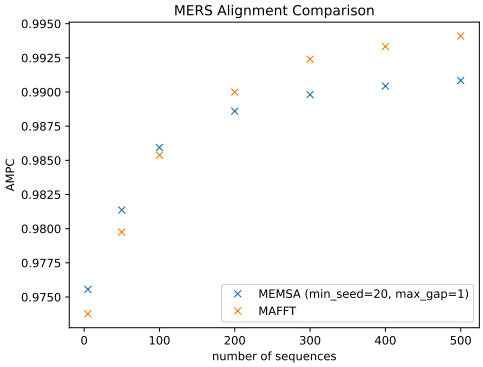

Fig. 7: AMPC indicator of alignment quality

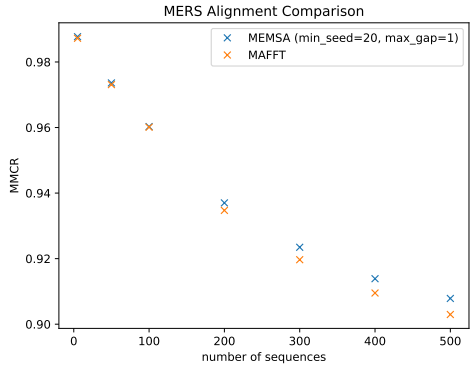

Fig. 6: MMCR indicator of alignment quality

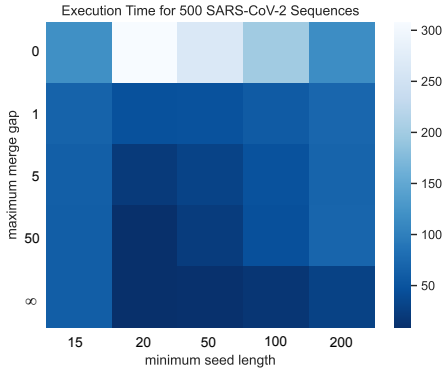

Fig. 8: Execution time [s] as function of seeding parameters

When it comes to alignment accuracy (see Figure 6), the amount of matching columns decreases with more input sequence in similar fashion for both MEMSA and MAFFT. The MMCR of the alignments performed by MAFFT decreases just slightly faster.

In terms of matches per column (see Figure 7), they again follow a similar trajectory, however, the AMPC increases with a larger amount of sequences. Also, the MAFFT alignment performs slightly better in this regard.

B. SARS-CoV-2

For the next tests, we switched to a larger data set of 2000 random SARS-CoV-2 genomes from the NCBI database [25]. In order to see the effect of the input parameters for the seeding process (minimum seed length and maximum merge gap), 500 of those genomes were picked and the execution time of MEMSA was measured for different combinations of parameters. As can be seen in Figure 8, not merging any seeds significantly slows down the process compared to the default parameter of 1. The higher the limit for merging, the faster the alignment, meaning that the execution is the fastest if all gaps of equal size get merged (no upper limit). As for the minimum seed length, decreasing it below the default value 20 also slows down the alignment, however, alignment speed peaks at the default value and then gets slower again for larger minimum seed sizes. The quality of alignment in terms of MMCR was the highest for seed size 20. For merging gaps less or equal to 1 it was MMCR = 92.8% and for 5 and higher it dropped to 92.4%. The AMPC increased continuously from 97.8% to 99.3% with increasing minimum seed sizes and maximum merge gaps (see Table I).

In the next step, MAFFT and MEMSA are compared once more but this time using the new data set of SARS-CoV-2 genomes. For MEMSA the optimal minimum seed size of 20 that we determined in the prior step and that coincides with the default value is being used. Both small values and no limit at all are being considered for the maximum merge gap and compared in order to quantify the trade-off between alignment speed and quality.

In Figure 9 and Figure 10 we can see a clear trade-off between accuracy and execution time for the different parameter settings. MAFFT and the MEMSA alignment with 1 as the maximum merge gap take a lot longer than the ones with higher maximum merge gaps. However, they have a higher MMCR instead. Figure 11 shows that there is a drop-off of the AMPC for a maximum merge gap of 1 whereas it remains approximately constant for the other alignments.

An overview of the exact values for the most important results can be seen in Table I.

input		seed parameters		MEMSA output			MAFFT output		
virus	sequences	min_seed	max_gap	time [s]	MMCR	AMPC	time [s]	MMCR	AMPC
MERS	500	20	1	75	0.9079	0.9908	2051	0.9030	0.9941
SARS-CoV-2	500	15	0	119	0.9243	0.9780	74	0.9286	0.9928
SARS-CoV-2	500	20	0	308	0.9278	0.9901	74	0.9286	0.9928
SARS-CoV-2	500	200	0	115	0.9241	0.9929	74	0.9286	0.9928
SARS-CoV-2	500	15	∞	62	0.9237	0.9784	74	0.9286	0.9928
SARS-CoV-2	500	20	∞	9	0.9240	0.9928	74	0.9286	0.9928
SARS-CoV-2	500	200	∞	29	0.9240	0.9930	74	0.9286	0.9928
SARS-CoV-2	2000	20	1	667	0.8235	0.9847	520	0.8239	0.9924
SARS-CoV-2	2000	20	2	297	0.8166	0.9908	520	0.8239	0.9924
SARS-CoV-2	2000	20	3	209	0.8156	0.9915	520	0.8239	0.9924
SARS-CoV-2	2000	20	∞	83	0.8151	0.9919	520	0.8239	0.9924

TABLE I: Most important results from test runs

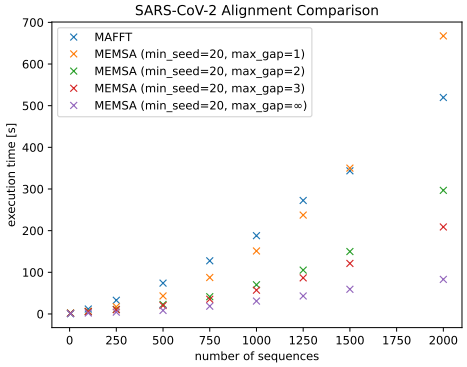

Fig. 9: Comparison of alignment speed between MAFFT and MEMSA with different maximum merge gaps

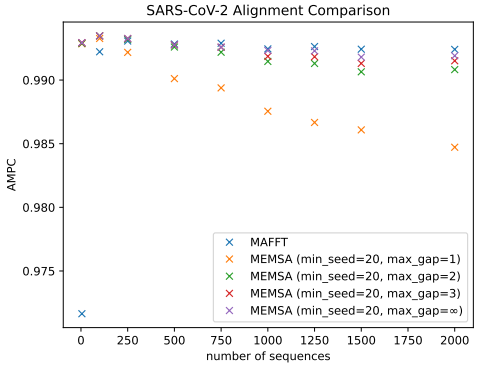

Fig. 11: Comparison of AMPC indicator for alignment quality with different maximum merge gaps

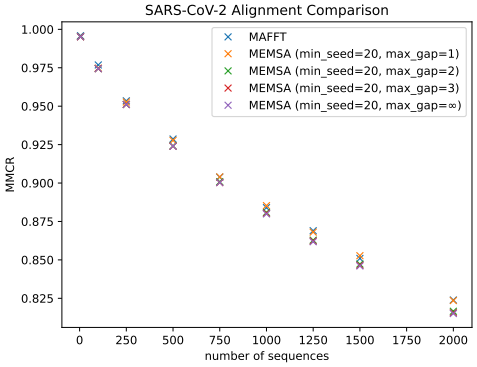

Fig. 10: Comparison of MMCR indicator for alignment quality with different maximum merge gaps

VI. Discussion

A. Input Data

The first requirement for MEMSA to work properly is that there is at least one well-conserved region across all input sequences. Besides that, the seeds identified in the pairwise MEM search must be in the same order as in the reference. This means that this method requires a data set that is homologous. If the sequences of the data set are too divergent, this approach will not yield better results than just MAFFT. In case these assumptions are violated, the MSA algorithm has to be called on the whole sequence, thus making this approach slightly slower due to the additional preprocessing.

All the sequences from the NCBI database [25] that were used showed a high homology. Therefore, these results should be considered as the optimal conditions for the method. Aligning a pandoravirus data set was also considered to see how well MEMSA scales with longer genome sequences. However, the available sequences on NCBI were too divergent (see Figure 12) and could not be aligned using our heuristic. As can be seen, the pairwise seeds of the different genomes are not in order and thus cannot be used.

B. Seeding Parameters

The seeding parameters (minimum seed size and maximum gap merge) have a great impact on both the speed and accuracy of the alignment. Generally, having larger, merged seeds reduces the amount of MSA calls that need to be performed. This has a direct effect on the execution time. However, the larger the gaps between the seeds are, the more likely it is that the subsequences in these gaps are not trivial anymore but instead misaligned. This risk increases with larger gaps, so it is a trade-off between execution time and accuracy. As for the minimum seed size, the optimal length is around 20, which is the value used by all common seed-and-extend aligners. If it gets increased more, fewer seeds will be found and thus the search space will be bigger. If a minimum seed size smaller

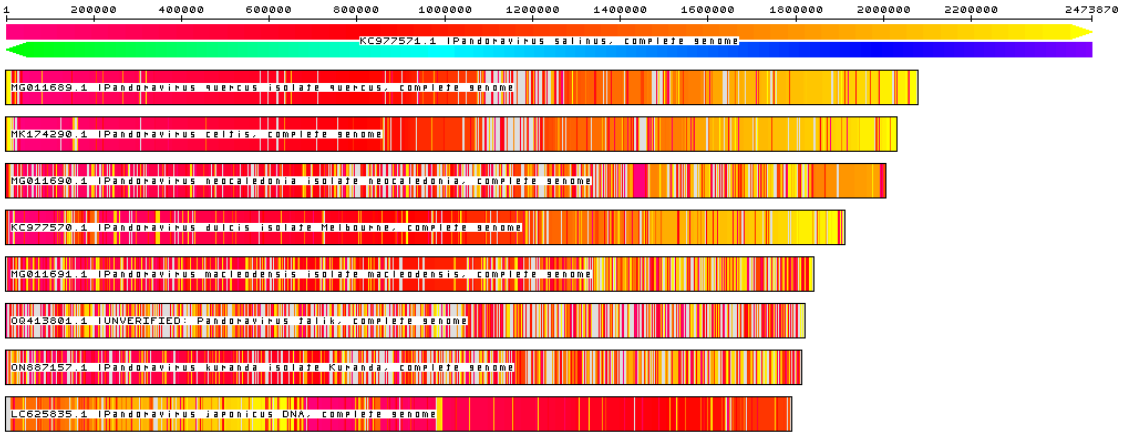

Fig. 12: Pairwise seed visualization for pandoravirus data set, generated with slaMEM [14]

than 20 is chosen, more and more ”random” seeds will be found just by pure chance, which might disturb the alignment process. Furthermore, choosing a smaller minimum seed size slows down the MEM identification drastically.

C. Quality Metrics

Assessing the quality of an MSA is a complicated problem and there are many different approaches. Both metrics introduced in Section IV-C are heavily dependent on the homology of the sequences and thus only suited for comparing two alignments of the same sequences with each other. If used in absolute terms, they do not just show how well the alignment tool worked but even more so, how similar and thus ”alignable” these sequences are. The results of the alignments show that MEMSA tends to insert more gaps than MAFFT. This is due to the fact that MAFFT automatically applies some heuristics when the input data becomes too large. However, for the alignment of the shorter subsequences between the seeds, the heuristics are not in place, which tends to favor gap insertions over mismatches. Since the AMPC is heavily affected by these excessive insertions, it tends to be lower for the results of MEMSA than those of MAFFT. Due to this bias towards gap insertion, AMPC is not as effective as MMCR for comparing the alignment quality of the two tools. These insertions of gaps should neither have a positive nor a negative impact on the alignment quality factor, which is why MMCR is suited a lot better for this comparison.

VII. Conclusions

A tool for MSA has been developed and some first tests have been performed to assess its execution time and alignment quality in comparison to MAFFT.

It has been shown that the input sequences need to show a certain amount of homology in order to see an improved execution time. However, since just one single seed across all sequences suffices in order to be able to apply the heuristic, it can be viewed as a low-risk-high-reward situation. If the sequences generate usable seeds, the alignment process will be significantly faster (27 times faster for 500 sequences of MERS genomes) whereas if the method fails it will only be slightly slower than traditional alignment because the MEM identification just takes a fraction of the actual alignment time. There are still some challenges to overcome, in order to make this procedure feasible for broader applications, but it has shown some promising first results. One of these limitations, that could be addressed in further research, is to find a method that eliminates identified seeds that hinder the alignment procedure because they for example do not occur in the same order across different sequences.

Future research could also investigate the feasibility of using MEMSA for side-channel trace alignment since those traces by nature have well-conserved regions between them, which correspond to specific CPU operations.

References

[1] K. Katoh and H. Toh, “Recent developments in the MAFFT multiple sequence alignment program,” *Briefings in Bioinformatics*, vol. 9, no. 4, pp. 286–298, 03 2008. [Online]. Available: https://doi.org/10.1093/bib/bbn013

[2] R. C. Edgar, “Muscle: a multiple sequence alignment method with reduced time and space complexity,” *BMC Bioinformatics*, vol. 5, no. 1, p. 113, Aug 2004. [Online]. Available: https://doi.org/10.1186/1471-2105-5-113

[3] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[4] T. Smith and M. Waterman, “Identification of common molecular subsequences,” *Journal of Molecular Biology*, vol. 147, no. 1, p. 195 – 197, 1981.

[5] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, 1990.

[6] M. Vasimuddin, S. Misra, H. Li, and S. Aluru, “Efficient architecture-aware acceleration of bwa-mem for multicore systems,” in *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2019, pp. 314–324.

[7] B. Langmead and S. L. Salzberg, “Fast gapped-read alignment with bowtie 2,” *Nature Methods*, vol. 9, no. 4, pp. 357–359, Apr 2012. [Online]. Available: https://doi.org/10.1038/nmeth.1923

[8] N. Ahmed, K. Bertels, and Z. Al-Ars, “A comparison of seed-and-extend techniques in modern dna read alignment algorithms,” in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2016, pp. 1421–1428.

[9] S. Kurtz, A. Phillippy, A. L. Delcher, M. Smoot, M. Shumway, C. Antonescu, and S. L. Salzberg, “Versatile and open software for comparing large genomes.” *Genome biology*, vol. 5, no. 2, 2004.

[10] N. P. Brown, C. Leroy, and C. Sander, "MView: a web-compatible database search or multiple alignment viewer," *Bioinformatics*, vol. 14, no. 4, pp. 380–381, 1998.

[11] F. Sievers *et al.*, "Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega," *Mol Syst Biol*, vol. 7, p. 539, Oct. 2011.

[12] C. Notredame, D. G. Higgins, and J. Heringa, "T-coffee: a novel method for fast and accurate multiple sequence alignment," *Journal of Molecular Biology*, vol. 302, no. 1, pp. 205–217, 2000. [Online]. Available: https://doi.org/10.1006/jmbi.2000.4042

[13] K. Katoh, K. Misawa, K. Kuma, and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform," *Nucleic Acids Research*, vol. 30, no. 14, pp. 3059–3066, 07 2002. [Online]. Available: https://doi.org/10.1093/nar/gkf436

[14] F. Fernandes and A. T. Freitas, "slaMEM: efficient retrieval of maximal exact matches using a sampled LCP array," *Bioinformatics*, vol. 30, no. 4, pp. 464–471, 12 2013. [Online]. Available: https://doi.org/10.1093/bioinformatics/btt706

[15] A. Kutzner, P.-S. Kim, and M. Schmidt, "A performant bridge between fixed-size and variable-size seeding," *BMC Bioinformatics*, vol. 21, no. 1, 2020.

[16] M.-C. F. Chang, Y.-T. Chen, J. Cong, P.-T. Huang, C.-L. Kuo, and C. H. Yu, "The smem seeding acceleration for dna sequence alignment," in *2016 IEEE 24th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2016, pp. 32–39.

[17] N. Khiste and L. Ilie, "E-mem: efficient computation of maximal exact matches for very large genomes," *Bioinformatics*, vol. 31, no. 4, pp. 509–514, 10 2014. [Online]. Available: https://doi.org/10.1093/bioinformatics/btu687

[18] M. T. Pervez, M. E. Babar, A. Nadeem, M. Aslam, A. R. Awan, N. Aslam, T. Hussain, N. Naveed, S. Qadri, U. Waheed, and M. Shoaib, "Evaluating the accuracy and efficiency of multiple sequence alignment methods," *Evol Bioinform Online*, vol. 10, pp. 205–217, Dec. 2014.

[19] B. Morgenstern, S. J. Prohaska, D. Pöhler, and P. F. Stadler, "Multiple sequence alignment with user-defined anchor points," *Algorithms for Molecular Biology*, vol. 1, no. 1, p. 6, Apr 2006. [Online]. Available: https://doi.org/10.1186/1748-7188-1-6

[20] B. Morgenstern, "Dialign: multiple dna and protein sequence alignment at bibiserv," *Nucleic acids research*, vol. 32, no. suppl_2, pp. W33–W36, 2004.

[21] F. Pitschi, C. Devauchelle, and E. Corel, "Automatic detection of anchor points for multiple sequence alignment," *BMC Bioinformatics*, vol. 11, no. 1, p. 445, Sep 2010. [Online]. Available: https://doi.org/10.1186/1471-2105-11-445

[22] V. Ahola, T. Aittokallio, M. Vihinen, and E. Uusipaikka, "A statistical score for assessing the quality of multiple sequence alignments," *BMC Bioinformatics*, vol. 7, no. 1, p. 484, Nov 2006. [Online]. Available: https://doi.org/10.1186/1471-2105-7-484

[23] A. Bahr, J. D. Thompson, J.-C. Thierry, and O. Poch, "BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations," *Nucleic Acids Research*, vol. 29, no. 1, pp. 323–326, 01 2001. [Online]. Available: https://doi.org/10.1093/nar/29.1.323

[24] P. W. Jacek Błażewicz, Piotr Formanowicz, "Some remarks on evaluating the quality of the multiple sequence alignment based on the balibase benchmark," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 4, pp. 675–678, 2009. [Online]. Available: http://eudml.org/doc/207965

[25] E. L. Hatcher *et al.*, "Virus Variation Resource – improved response to emergent viral outbreaks," *Nucleic Acids Research*, vol. 45, no. D1, pp. D482–D490, 11 2016. [Online]. Available: https://doi.org/10.1093/nar/gkw1065