Fragmenting Genome Sequences by Coding Regions to Improve Performance of the AmpliDiff Algorithm for Large Genomes

Samuel Karskens[1]

Supervisor(s): Jasmijn A. Baaijens, Jasper van Bemmelen

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 28, 2024

Name of the student: Samuel Karskens
Final project course: CSE3000 Research Project
Thesis committee: Jasmijn A. Baaijens, Jasper van Bemmelen, Chirag Raman

Abstract

Abundance estimation with the use of environmental samples has been used during the SARS-CoV-2 pandemic to identify the abundances of different lineages. AmpliDiff [14], an algorithm that tries to find parts of DNA that can differentiate between different input genomes was used on a SARS-CoV-2 dataset to find these amplicons. The AmpliDiff algorithm was able to run on the SARS-CoV-2 set but seemed infeasible for datasets that contain larger or more complex genomes because of the computational requirements and runtime. We introduce a new pre-processing strategy based on selecting the most differentiable coding regions and show the modifications done to AmpliDiff to make AmpliDiff work following this new method. Based on the results we conclude that the approach is promising but still requires more research to be used optimally.

1 Introduction

Using environmental samples to estimate abundances of different strains, variants or species can be useful. This was especially true during the SARS-CoV-2 pandemic, where wastewater samples were used to identify abundances of different variants [7]. Genome sequencing, a technique to determine the content of DNA, is often used in abundance estimation to identify the strain, variant, or species of a genome from a sample. Whole Genome Sequencing and Target Sequencing are two ways you can do genome sequencing. One amplifies the whole genome and later on sequences the genome, and the other only amplifies a specific target sequence and sequences this target sequence.

Before sequencing is possible, the whole genome or part of a sequence needs to be amplified. This is done with a technique called Polymerase Chain Reaction (PCR). For PCR to be able to amplify a sequence, a primer is required. Primers are small parts of DNA that can bind to a single DNA strand and serve as a starting point for replication in PCR. Tools that are used to find primers are mostly focused on finding primers for a region that is already specified by the user [13]. AmpliDiff [14] is an algorithm that combines finding amplicons with the finding of corresponding primers where amplicons are defined as parts of DNA that will be replicated by PCR. The amplicons are the parts of DNA that are getting amplified by PCR, while the primers are the parts that are used as starting points for the PCR.

In the AmpliDiff [14] paper, the authors stated that one part of the algorithm that can be improved is the pre-processing phase, where all input genomes need to be pre-processed using Multiple Sequence Alignment (MSA). MSA is a technique that is used to align multiple input genomes with each other. In AmpliDiff, genomes with different lengths and DNA are used as input, which does not make it directly clear which parts of DNA correspond with each other. The alignment is essential for AmpliDiff to be able to compare all the input genomes and discover discriminatory amplicons. However, MAFFT [8], the multiple sequence alignment algorithm used in AmpliDiff, is a bottleneck and becomes infeasible for very large genomes such as E.coli [15]. Not only does the preprocessing become infeasible, but the AmpliDiff algorithm itself can also not deal with very large or complex genomes. Since the algorithm needs to go over all reference genomes in each phase, it does not scale well.

Here, we introduce a modification of AmpliDiff that allows the algorithm to scale better to larger and more complex genomes. We look at improving the runtime of the AmpliDiff algorithm while at the same time minimizing the loss of potential primers and discriminatory amplicons. We introduce a new pre-processing strategy for AmpliDiff that fragments the input genomes based on the coding regions and uses a ranking algorithm to select the most differentiating coding regions to use as input for AmpliDiff. Based on three different datasets, all containing different-sized genomes, we show that AmpliDiff becomes feasible to run and has a lower runtime. The amplicons are benchmarked by using simulated reads and the VLQ pipeline [3] to perform abundance estimation. By comparing the effectiveness of the amplicons found by using only the most differentiating regions to whole genome sequencing (WGS), we conclude that this new method is promising but still requires more research to be used optimally.

2 Methodology

2.1 Background literature

MSA

The AmpliDiff algorithm uses MAFFT [8] to align multiple sequences with each other. MAFFT is an algorithm that is widely used and does give decent performance in general but does not work very well for very large genomes. This has been shown in a paper where an algorithm called SaAlign [15] was introduced. SaAlign is a tool that shows a substantial performance increase compared to MAFFT and does seem to be able to align 100 large Mitochondrion genomes in 20.2h, while the MAFFT algorithm was infeasible to run. Another multiple sequence alignment algorithm that was introduced in 2021 was FMAlign [9], which also showed major improvements compared to MAFFT. FMAlign was able to align 4 E. coli sequences in 23 minutes and 53 seconds. However, this does not show that this is also the case for larger datasets. So, even though many efforts have been made, there is no guarantee that MSA remains feasible for very large datasets. Showing the need to look at other ways.

FMAlign [9] works by using vertical division to divide sequences. The different subsequences are then aligned with MAFFT and, in the end, concatenated back together to create an entire alignment. The new method proposed in this paper also uses vertical division but in another way. Namely, fragmenting input genomes by their coding regions. While FMAlign first needs to find how to divide sequences, the new method can use existing annotations to fragment the input genomes, reducing runtime. Vertical division also allows parallelization of the alignments, as already shown in FMAlign. This is also applied in the proposed pre-processing algorithm.

Ranking

For AmpliDiff to find an amplicon in a DNA region, there needs to be sufficient difference between lineages in that region. However, in reality, not all regions of a genome contain the same amount of difference, some may even contain the same DNA for all sequences depending on the dataset. For example, a study done on SARS-CoV-2 mutations based on 10287271 sequence samples [1] showed that there are regions in the genome where no mutation occurred in more than 90 percent of the regions. These regions include but are not limited to nsp11, nsp7 and nsp10.

For E.coli specifically, it has also been shown that core genes evolve rapidly in a long-term evolution experiment [10]. They showed that the most evolving genes tended to be core genes. This could be a great sign that by using only core genes that existed in all 60 strains used in the study, there is enough difference to differentiate between all different strains.

Based on these observations and findings, a ranking algorithm is proposed that will rank all coding regions based on the amount of difference there is between the sequences. This distance metric is calculated using the Mash algorithm [11]. Only the selected coding regions need to be aligned, reducing the input size of MAFFT [8] quite substantially depending on the number of coding regions used.

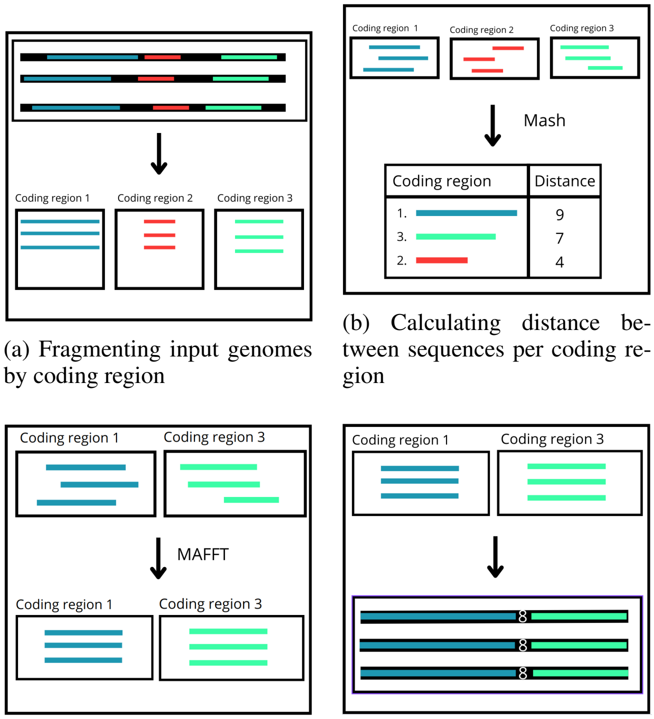
AmpliDiff

The AmpliDiff algorithm consists of 3 phases: creating a set of candidate primers, finding amplicons, and selecting amplicons with a greedy algorithm while checking for the existence of corresponding primers. As shown in the paper of AmpliDiff the run of 2749 input sequences all around 30000 base pairs on a High-Performance Computing (HPC) cluster with 200GB RAM and 12 CPU cores takes 7 hours and 56 minutes. Based on the fact that the algorithm needs to go over all reference genomes in each phase we can conclude that the algorithm will not scale well to genomes that are 1000 times larger than SARS-CoV-2 genomes.

2.2 The pre-processing algorithm

The pre-processing strategy requires all input genomes to have annotations of the coding regions or genes, if they cannot be downloaded directly from a database like NCBI [12] one has to annotate the sequences first to be able to run the pre-processing algorithm. However, this is not taken into account in this research.

First, the input genomes are fragmented using the location of the coding regions, having each fragment consisting of one coding region, resulting in other non-coding DNA being removed (Fig. 1a). Second, the Mash algorithm [11] is run on every coding region in parallel, and the most differentiating coding regions are selected to process further (Fig. 1b). Selecting the most differentiating coding regions was done by using the distance metric that was calculated based on the result from Mash [11]. Mash returns the distance between every pair of sequences. The distance metric was calculated by adding all distances to each other and dividing it by the number of sequences that were included in the specific coding region. The number of coding regions is an input parameter for AmpliDiff. Third, MAFFT [8] is run on every selected coding region separately in parallel (Fig. 1c). MAFFT makes sure that in every coding region all sequences are aligned with each other. Fourth, the multiple sequence aligned coding regions are put back together (Fig. 1d). Between every coding region, a specific character is placed, in this case, an ”8”, that would let AmpliDiff know that at that place, a split exists.



(a) Fragmenting input genomes by coding region

(b) Calculating distance between sequences per coding region

(c) Running multiple sequence alignment on the two most differentiating coding region

(d) Combining the two coding regions back to get one sequence per genome

Figure 1: The pre-processing algorithm. Selecting the two most differentiating coding regions.

For the E. coli and MonkeyPox dataset an extra step was added to the pre-processing algorithm to deal with missing annotations and duplicate coding regions or genes. The step is added before step 1 as shown in Figure 1a. The method works as follows. A list is made of regions that adhere to the following requirements. First, the region should never occur more than once in a sequence. Second, the region should be annotated in all the input sequences. After, this list is used as input for step 1 (Fig. 1a) to make sure that it only selects regions from that list.

2.3 Modifications of AmpliDiff

Two main modifications were made to the AmpliDiff algorithm to make it able to handle the fragmented input. First, AmpliDiff is not allowed to select primers or amplicons that are on overlapping regions between two coding regions. Since AmpliDiff does not know about the information in between coding regions, this could result in amplicons or primers that do not correspond to the real sequence. Second, AmpliDiff is only allowed to select primers in the same

coding region as the amplicon. This prevents the distance between the primer and amplicon from getting so large that the amplicon will not be amplified correctly by PCR.
To accommodate these changes, the primer feasibility check in AmpliDiff is changed to also check if the primer does not contain any special character. If this is the case, the primer will be deleted from the primer index. This is the first step of the AmpliDiff algorithm after pre-processing. To ensure that primers are in the same coding region as the amplicons, the algorithm ensures that the amplicon and the search width around the amplicon do not contain any special characters. The search width is the range that the AmpliDiff algorithm uses to find primers and is important to ensure that the primers and amplicons are not too far away from each other. Otherwise, they will not be amplified by PCR.

3 Experimental Setup and Results

3.1 Experimental setup

To be able to answer the research questions this project consists of three experiments using three different datasets. All experiments were run on a High-Performance Computing (HPC) cluster with 185GB RAM and 12 CPU cores. The first set consisted of 480 Sars-CoV-2 genomes, the second set consisted of 359 E. coli genomes, and the last set out of 485 MonkeyPox genomes.

AmpliDiff

All AmpliDiff runs were done with the following input parameters: the amplicon width is set to 400bp, the number of cores used in multi-processing is set to 12, and the number of amplicons that AmpliDiff should generate is set to 10. The other input parameters are equal to the default values. The AmpliDiff algorithm is not added in the comparison for the E. coli dataset since the required pre-processing was not able to finish within the 24-hour time limit of the HPC cluster.

Abundance estimation

All outputs are benchmarked by using simulated reads done by ART [6], which are used in the VLQ pipeline [3] to estimate abundances. The VLQ pipeline uses Kallisto [4] to pseudoalign the reads to the different lineages or strains from the reference set. Two datasets were made for the benchmark. A simulation set and a reference set. The simulation set contained genomes that were used to simulate reads. While the reference set contained genomes that were used in the VLQ pipeline. Kallisto pseudoaligns the reads to the reference set. Subsequently, the VLQ pipeline can use that output to estimate abundances. The reference set was used as input for AmpliDiff, while the simulation set was only used to simulate reads from. Therefore, the simulation set did not have any overlap with the reference set, no sequences appeared in both, but both sets contained the same lineages.
Mean absolute prediction errors (MAPE) were calculated with the same formula that was used in the AmpliDiff paper [14]. The absolute difference between the real abundances and estimated abundances is summed and divided by the total amount of lineages that occurred in the simulation set. The reference set being the set that was used to run AmpliDiff with. The MAPE are used to compare the quality of the amplicons found by AmpliDiff, the modification of AmpliDiff and whole genome sequencing.
The abundance estimation setup is kept the same as the one described in the simulation study section of the AmpliDiff [14] paper. However, the read simulation are only done once for each setting in contrast to the 20 times in the AmpliDiff paper. The specific random seeds and parameters can be found in the Appendix. For every dataset and simulationset the list of accession numbers can be found in the Appendix.

Cumulative differentiability

For every dataset, the differentiability of the found amplicons is assessed by showing the cumulative differentiability of the amplicons. The cumulative differentiability is calculated by getting the number of sequence pairs the specific amplicon can differentiate between, and dividing this by the total number of sequence pairs that need to be differentiated. A sequence pair is only counted when the lineage or strain is different.

SARS-CoV-2

To test the new pre-processing strategy and ranking system, 500 Sars-CoV-2 complete annotated genomes were selected by first downloading a set of accession numbers of all complete sequences using the NCBI [12] web interface. The random sampling was excluded, and the filter named proteins was set to include the following set ”ORF1ab polyprotein”, ”ORF6 protein”, ”ORF1a polyprotein” and ”ORF7a protein”. This resulted in 338754 sequences. This CSV file with accession numbers is used to download the sequences and coding regions from NCBI. The 500 genomes dataset was created using proportionate stratified sampling. Only sequences that contained all annotations were selected.
Instead of using the overlapping ORF1ab, we used the more specific annotations of the non-structural proteins. Sequences that contained the truncated ORF8 protein were filtered out to avoid missing data, which could potentially influence the distances in the ranking algorithm. Two sequences containing an annotation named ”orf1ab polyprotein” were also removed from the dataset because no other sequences had this annotation. Since ORF1ab overlapped with ORF1a, ORF1a was not taken into account. This also resulted in NSP11 not being used as a coding region in the algorithm, all to prevent overlapping parts. In the end, this resulted in a dataset of 480 genomes.
The simulation set was created by deleting all accessions from the original CSV file containing 338754 sequences and removing all lineages that did not exist in the reference set. By using proportionate stratified sampling, we obtain a simulation set of 601 sequences.

E. coli

The E. coli dataset contains 500 genomes from NCBI [12], created using proportionate stratified sampling. Further processing selected only genomes that contained 3500 gene annotations or more, resulting in 359 genomes. This set contained 358 strains, one to two different sequences per strain. Since E. coli contains a lot of duplicate genes, in the pre-processing, only the genes that all sequences contained and had no duplicates were selected before the new pre-processing strategy even began, see section 2.2. In this case, gene annotations were used because they were using the same naming scheme in all selected sequences. The simulation set was constructed in the same way as the one for SARS-CoV-2. The set contained 387 genomes.

MonkeyPox

The procedure for creating the MonkeyPox dataset differed from the other two datasets since the new proposed classification system [5] for MonkeyPox had not been used yet in the NCBI [12] dataset. All complete and annotated genomes were downloaded from NCBI first. The set of 1775 genomes were assigned a clade using a tool called Nextclade [2]. Nextclade checks which mutations occur in the sequence based on a reference sequence. With these mutations, it searches for the closest clade, and that one gets assigned to the sequence. After, the set was obtained by using proportionate stratified sampling. Sequences with more than 177 coding regions were deleted to maximize the set of coding regions that were in all sequences. This resulted in a set of 485 sequences and contained 24 clades out of 30 clades that exist. The simulation set was constructed the same way as the ones for SARS-CoV-2 and E. coli and consisted of 618 sequences and the same 24 lineages.

3.2 Results

SARS-CoV-2

In Figure 2, one sees the cumulative differentiability of every amplicon. It can be seen that the cumulative differentiability of every amplicon is almost equal in all experiments. The AmpliDiff amplicons seem to be slightly more differentiable compared to the other experiments. The 15 most differentiating regions experiment seems slightly better in terms of differentiability than the 5 and 10 most differentiating regions.

In Figure 3, the mean absolute prediction error is displayed that was obtained after the abundance estimation. It can be seen that only using 5, 10 or 15 regions is not enough to match the quality of the amplicons found by AmpliDiff. However, the mean absolute prediction error when only using 5 or 10 regions is close to the error of AmpliDiff. One striking observation is that the mean absolute prediction error increased in the 15 most differentiating regions compared to using only 5 or 10.

Also, looking at the abundance estimation in Figure 3, the top-10 and top-15 perform worse than AmpliDiff, and the top-5 is slightly better. All of them perform immensely worse than whole genome sequencing.

In Table 2, you can see the runtime of AmpliDiff and the modified AmpliDiff on the 5, 10, and 15 most differentiating coding regions. Most of the time was spent on the greedy algorithm. Interestingly, the pre-processing phase takes more time when only a number of regions are selected.

E.coli

In Figure 4, one sees the cumulative differentiability of the 10 amplicons for the 5, 10, and 15 most differentiable genes. One sees that after adding the second amplicon, the differentiability does not change much anymore. An observation is

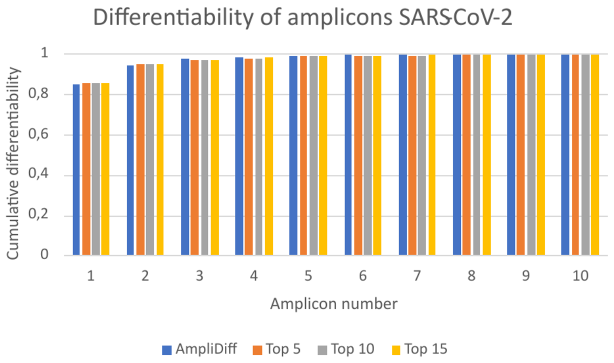
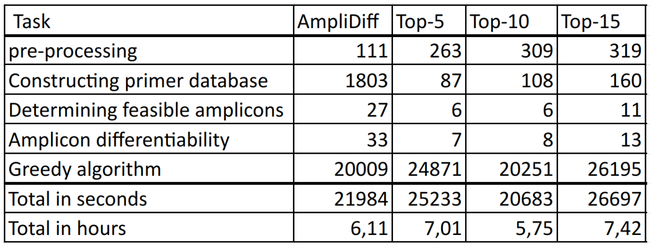


Figure 2: Cumulative relative differentiability of 10 amplicons found by using the 5, 10 and 15 most differentiating coding regions as well as AmpliDiff. Amplicons generated with SARS-CoV-2 dataset containing 480 genome sequences.

Coding region name	Difference
ORF7b	1.84
surface glycoprotein	1.06
nucleocapsid phosphoprotein	0.95
ORF3a	0.81
membrane glycoprotein	0.80
ORF6	0.77
nsp6	0.65
ORF7a	0.64
ORF8	0.63
envelope protein	0.57
nsp4	0.50
nsp9	0.39
3C-like proteinase	0.33
leader protein	0.31
3'-to-5' exonuclease	0.29
endoRNAse	0.29

Table 1: Most differentiating coding regions in the SARS-CoV-2 dataset. Sorted from most to least differentiating. Rounded to two decimals.



Task	AmpliDiff	Top-5	Top-10	Top-15
pre-processing	111	263	309	319
Constructing primer database	1803	87	108	160
Determining feasible amplicons	27	6	6	11
Amplicon differentiability	33	7	8	13
Greedy algorithm	20009	24871	20251	26195
Total in seconds	21984	25233	20683	26697
Total in hours	6,11	7,01	5,75	7,42

Table 2: Runtime comparison using the SARS-CoV-2 dataset. Comparing 5, 10, and 15 of the most differentiating regions to AmpliDiff. All numbers displayed are in seconds, rounded to whole seconds, except for the last row that shows hours, rounded to two decimals.

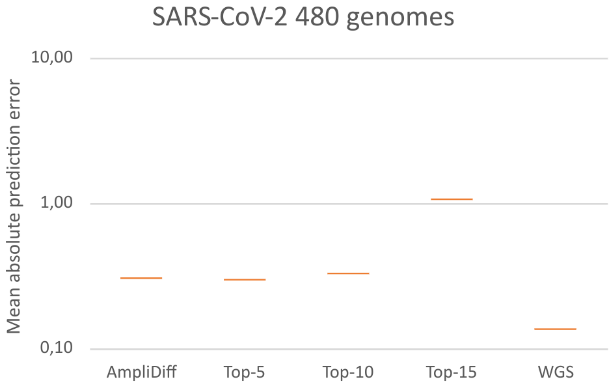


Figure 3: The mean absolute prediction error for abundance estimation of the SARS-CoV-2 dataset. Comparing the 5, 10 and 15 most differentiating coding regions to whole genome sequencing and AmpliDiff. The vertical axis is logarithmically scaled.

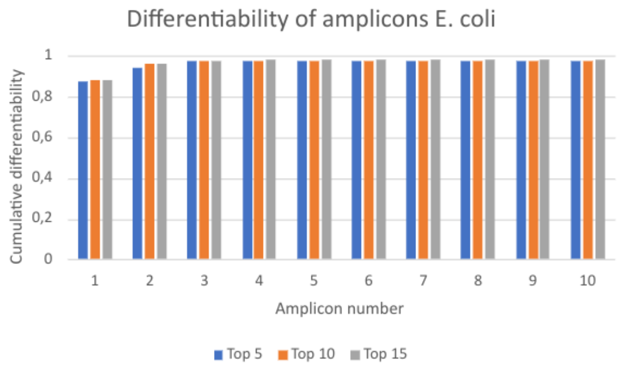


Figure 4: Cumulative differentiability of 10 amplicons found by using the 5, 10 or 15 most differentiable genes. Amplicons generated with the E.coli dataset containing 359 genome sequences.

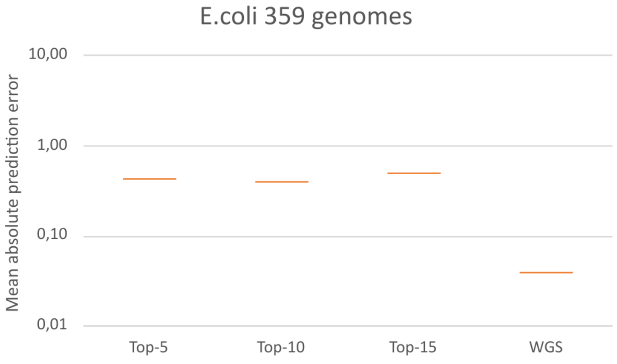


Figure 5: The mean absolute prediction error for abundance estimation in the E. coli dataset. Comparing the 5,10, and 15 most differentiating genes to whole genome sequencing. The vertical axis is logarithmically scaled and starts at 0,01 instead of 0,10 shown in the other MAPE figures.

Task	Top-5	Top-10	Top-15
pre-processing	1361	2408	2435
Constructing primer database	25	99	266
Determining feasible amplicons	3	6	8
Amplicon differentiability	3	4	6
Greedy algorithm	601	481	404
Total in seconds	1993	2998	3118
Total in hours	0,55	0,83	0,87

Table 3: Runtime comparison using the E. coli dataset. Comparing the runtimes of the 5, 10, and 15 most differentiating regions. All numbers displayed are in seconds, rounded to whole seconds, except for the last row that shows hours, rounded to two decimals.

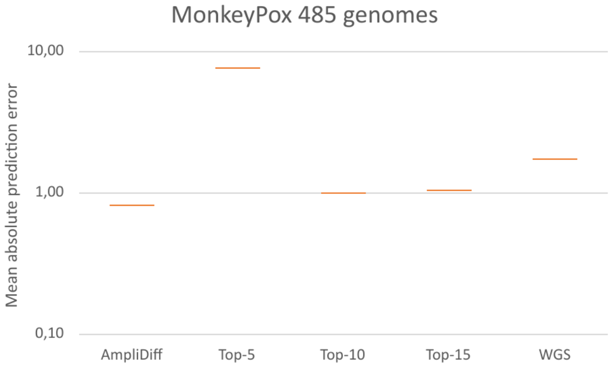


Figure 6: The mean absolute prediction error for abundance estimation in the MonkeyPox dataset. Comparing the 5, 10, and 15 most differentiable coding regions to whole genome sequencing and AmpliDiff. The vertical axis is logarithmically scaled.

that all amplicons that were found by using more genes were always equally or more differentiable. The differentiability is very close to each other for every number of genes. Only by adding amplicon 2 can a clear difference in differentiability between the 5 most differentiating genes and the 10 and 15 most differentiating genes be noticed. It also shows that there is not much difference between the differentiability when using 5, 10, or 15 genes. This is a good sign since this gives some evidence that not only the ranking algorithm selected the most differentiable genes, the most differentiable genes also allowed differentiating amplicons to be found that can be amplified.

In Table 3, different runs of the modified AmpliDiff algorithm on 5, 10 and 15 of the most differentiating genes are compared with each other. Adding more genes in the input, results in a higher runtime as expected. Figure 5 show the MAPE for the E. coli dataset. Here, not much difference can be seen between the 5, 10, and 15 regions. However, the same striking observation as for the SARS-CoV-2 dataset can be done relating to the increase in error when instead of 10 regions, 15 regions were considered.

MonkeyPox

In Table 4, the runtime comparison using the MonkeyPox dataset can be seen. As expected, all runs that focused on specific regions were significantly faster than AmpliDiff when

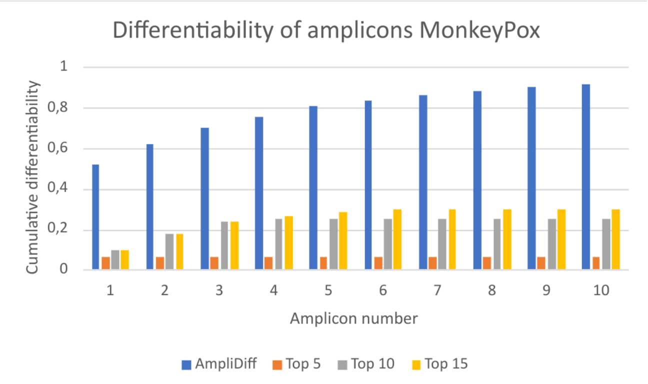
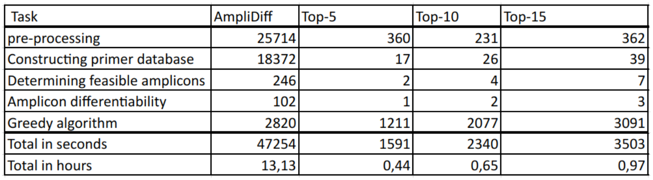


Figure 7: Cumulative relative differentiability of 10 amplicons found by using the 5,10 and 15 most differentiating coding regions as well as AmpliDiff. Amplicons generated with MonkeyPox dataset containing 485 genome sequences.



Task	AmpliDiff	Top-5	Top-10	Top-15
pre-processing	25714	360	231	362
Constructing primer database	18372	17	26	39
Determining feasible amplicons	246	2	4	7
Amplicon differentiability	102	1	2	3
Greedy algorithm	2820	1211	2077	3091
Total in seconds	47254	1591	2340	3503
Total in hours	13,13	0,44	0,65	0,97

Table 4: Runtime comparison using the MonkeyPox dataset. Comparing 5,10, and 15 of the most differentiating regions to AmpliDiff. All numbers displayed are in seconds, rounded to whole seconds, except for the last row that shows hours, rounded to two decimals.

we include the pre-processing. However, looking specifically at the greedy algorithm task, the run shows that including the 15 regions, is slower than AmpliDiff itself. In Figure 6, the MAPE is shown in the same way as the previous datasets. In this figure, it is very clear that only using five regions is not enough to find good amplicons. Only using 10 or 15 regions seems to have a lower MAPE than WGS. However, AmpliDiff still shows the lowest MAPE.

Not all AmpliDiff runs were able to find 10 amplicons. When using 10 regions, the algorithm could only find four. When using 15 regions, the algorithm could only find six. This can also be seen in Figure 7. As expected, more regions result in amplicons that are more differentiable. However, it is important to note that the differentiability is very low in all experiments. Certainly, when compared to the differentiability of the amplicons found by AmpliDiff. Another interesting observation is that even though the amplicons found when using only the 10 and 15 most differentiating regions were substantially worse in terms of differentiability, they still showed a lower MAPE than WGS.

4 Responsible Research

The following steps were taken to ensure reproducibility. All the source code used during this project is publicly available on GitHub. The GitHub repository URL can be found in the Appendix. The code is written in Python, and all libraries used are defined in the repository. All the libraries that are used are widely available. Also, in section 3.1, the experimental setup is explained in great detail to allow anyone to reproduce the results. In addition, all the data that was used during this research is from a public database called NCBI [12] that anyone can access. All the specific sequences used are defined in the Appendix. Furthermore, during simulating reads with ART [6], the generations of the reads were done with specific random seeds to allow anyone to create the same reads. The random seeds are specified in the Appendix. An important aspect to take into account is that reproducing this work without the use of a High-Performance Cluster is more difficult because of the high computational requirements. However, since the datasets are relatively small, fast desktop computers might be able to run the algorithms within a reasonable time frame.

5 Discussion

5.1 Differentiability of coding regions

In Table 1, one can see which coding regions were most differentiating based on outputs from the Mash algorithm [11]. A study published in 2023 showed comparable results [1]. This study observed the highest frequency of mutations in 'membrane glycoprotein', 'envelope protein', and 'nucleocapsid phosphoprotein'. On the contrary, our results did not contain the coding regions in the same order but still showed that these coding regions belonged to the most differentiable. The study also observed no mutations in 90 percent of nsp7, nsp8, nsp9, nsp10, nsp11, and nsp16. However, in our results, nsp9 belongs to the 15 most differentiating coding regions. This can potentially be explained by the size of our dataset, which is very small compared to the one they used in the study [1]. Additionally, the distance metric considers the difference between sequences from the same strain or lineage. This results in counting differences between sequences that do not have to be differentiated. Even though there should only be a small difference between sequences from the same strain or lineage, it could be part of why the order in terms of differentiability is different compared to the previously mentioned study [1].

As mentioned in chapter 3.2, when only using 10 or 15 regions with the MonkeyPox dataset, the algorithm could not find 10 amplicons. It could be that the 15 most differentiating regions do not contain enough variability to create highly differentiable amplicons. Or that the variability could make finding primers for the amplicons infeasible.

5.2 Runtime

In the SARS-CoV-2 dataset, the new method does not generate better runtime results than AmpliDiff, except for the run with the 10 most differentiating regions where the results of the new method were better. The results could be explained by the variability of the used computing resource, since only single runs were done and no averages were taken.

In the runtime comparison for the SARS-CoV-2 dataset, the new pre-processing phase takes longer than the pre-processing from AmpliDiff. This can potentially be explained by the need to calculate all the distances between every pair of sequences for every coding region when only some regions

are selected. Since the SARS-CoV-2 genome is relatively small, it shows that the extra calculations do not outweigh the normal pre-processing.
The runtime comparison for the SARS-CoV-2 and MonkeyPox datasets shows that the greedy algorithm takes longer when only a selected number of regions is considered instead of the whole sequences in AmpliDiff. This can potentially be explained by the extra constraints introduced in the AmpliDiff algorithm.
Overall, for larger genomes such as MonkeyPox and E. coli, the pre-processing strategy is quite effective. This is especially seen in the runtimes of the experiments done on MonkeyPox. Using the 15 most differentiating regions has a ten times reduction in runtime compared to AmpliDiff.

5.3 MAPE

The amplicons found in the E. coli dataset showed a significantly higher MAPE than WGS. The difference between the original sequence length and the actual sequence length when only selecting the most differentiating coding regions was tremendously more significant than the differences in the other two datasets. Increasing the number of coding regions used for E. coli could potentially fix this problem.
The MAPE showed an increase in all datasets when comparing the use of 15 to the use of 10 regions. Additionally, this increase was even more prominent in the SARS-CoV-2 dataset. A potential reason for this could be the following. In AmpliDiff, primers that do occur multiple times in a single sequence are removed from the feasible primer set because they could create unwanted byproducts. However, since the same checks are done when only some coding regions are used, this can cause problems. Since AmpliDiff does not know about the missing DNA, it cannot check if primers occur multiple times in a single sequence. It could be that the extra five regions added for the top-15 run contained primer sequences used in the top-10 run. Which could mean that primers occurred more than once in a single sequence. This could have forced AmpliDiff to delete these primers to prevent unwanted amplifications.

5.4 Missing sequence in AmpliDiff

The version of AmpliDiff used in this research did contain a bug that deleted one sequence from the input when the number of input sequences was not explicitly defined. This will probably have a low impact on the results since it was only one and the same sequence that was left out. The only comparison made in this research that could have been influenced was the comparison in MAPE. The difference between the error obtained when using amplicons instead of WGS could potentially be lower since the algorithm could also have accounted for the last sequence to find the most differentiable amplicons. The left-out sequences are mentioned for every set and can be found in the Appendix.

6 Future Work and Conclusion

6.1 Future Work

There are still a lot of parts of the algorithm that can be improved or require more research. One of them is about the constraint that forces primers to be in the same region as the corresponding amplicon. In the currently proposed modification, all primers will be filtered out when there is a split between an amplicon and the area around the amplicon that is searched for primers, called the search width. This prevents primers from being in different coding regions than the amplicons are from. Since the search width during this experiment was 50bp and the primer size 25bp, there could be feasible primers in this region. How many primers we lose in practice by doing this could be researched. As well as the trade-off between the potential increase of the runtime and having more feasible primers.
Another part is about unwanted amplifications. Currently, there is no check for checking if a primer exists somewhere else in the genome outside of the regions used. In the original AmpliDiff algorithm, the whole genome could be checked since the input consisted of the whole genome sequences. However, it could not be used as effectively in the modified version since the AmpliDiff input was reduced to a certain number of input coding regions. The information of the other DNA was missing, which made checking if a primer already existed somewhere infeasible. This could, however, be solved by checking the whole sequence that the primer occurred in for duplicate primers when a primer is found in the algorithm.
Last, some other things that should be considered when using the ranking selection method are the following. First, it is difficult to tell how many coding regions one will need to be able to differentiate between all input sequences without experimenting. Second, this method does not consider that one needs more conserved areas to be able to design primers. Since we only select the most differentiating coding regions, it could be the case that less differentiating coding regions are needed to be able to also find corresponding primers. One way to resolve this problem would be to make the ranking selection iterative. AmpliDiff will start with a low number of selected coding regions and try to find amplicons and primers, adding coding regions iteratively until the amplicons' defined differentiability is met.

6.2 Conclusion

In this research, we looked at improving the runtime of AmpliDiff while minimizing the loss of potential primers and discriminatory amplicons. We showed that only using the ten most differentiating regions in MonkeyPox had better results than whole genome sequencing. However, this could not be shown for E. coli, which could have required the use of more differentiating regions because of the size of the genome. Applying the new method to the SARS-CoV-2 dataset showed comparable mean absolute prediction errors when compared to AmpliDiff, except for using 15 regions, which can potentially be explained by primers occurring more than once in a single sequence. The new method reduced the runtime substantially for larger genomes but not for the smaller genome of SARS-CoV-2. We conclude that the new method looks promising but will require more research to be applied correctly.

7 Acknowledgements

Kevin den Boon was involved in the development of the modifications for AmpliDiff.

A Appendix

The code for the modification of AmpliDiff as well as the data used in this research can be found on GitHub (https://github.com/SamuelKarskens/AmpliDiff). The specific random seeds and parameters that were used for the abundance estimation can also be found on GitHub.

References

[1] Mohammad Hadi Abbasian, Mohammadamin Mahmanzar, Karim Rahimian, Bahar Mahdavi, Samaneh Tokhanbigli, Bahman Moradi, Mahsa Mollapour Sisakht, and Youping Deng. Global landscape of SARS-CoV-2 mutations and conserved regions. *Journal of Translational Medicine*, 21(1):152, February 2023.

[2] Ivan Aksamentov, Cornelius Roemer, Emma B. Hodcroft, and Richard A. Neher. Nextclade: clade assignment, mutation calling and quality control for viral genomes. *Journal of Open Source Software*, 6(67):3773, 2021.

[3] Jasmijn A Baaijens, Alessandro Zulli, Isabel M Ott, Ioanna Nika, Mart J van der Lugt, Mary E Petrone, Tara Alpert, Joseph R Fauver, Chaney C Kalinich, Chantal BF Vogels, et al. Lineage abundance estimation for sars-cov-2 in wastewater using transcriptome quantification techniques. *Genome biology*, 23(1):236, 2022.

[4] Nicolas L. Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5):525–527, May 2016.

[5] Christian Happi, Ifedayo Adetifa, Placide Mbala, Richard Njouom, Emmanuel Nakoune, Anise Happi, Nnaemeka Ndodo, Oyeronke Ayansola, Gerald Mboowa, Trevor Bedford, et al. Urgent need for a non-discriminatory and non-stigmatizing nomenclature for monkeypox virus. *PLoS biology*, 20(8):e3001769, 2022.

[6] Weichun Huang, Leping Li, Jason R. Myers, and Gabor T. Marth. ART: a next-generation sequencing read simulator. *Bioinformatics (Oxford, England)*, 28(4):593–594, February 2012.

[7] Ray Izquierdo-Lara, Goffe Elsinga, Leo Heijnen, Bas B. Munnink, Claudia M.E. Schapendonk, David Nieuwenhuijse, Matthijs Kon, Lu Lu, Frank M. Aarestrup, Samantha Lycett, and et al. Monitoring sars-cov-2 circulation and diversity through community wastewater sequencing, the netherlands and belgium. *Emerging Infectious Diseases*, 27(5):1405–1415, 2021.

[8] Kazutaka Katoh, Kazuharu Misawa, Kei ichi Kuma, and Takashi Miyata. Mafft: A novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic Acids Research*, 30(14), 2002.

[9] Huan Liu, Quan Zou, and Yun Xu. A novel fast multiple nucleotide sequence alignment method based on FM-index. *Briefings in Bioinformatics*, 23(1):bbab519, January 2022.

[10] Rohan Maddamsetti, Philip J. Hatcher, Anna G. Green, Barry L. Williams, Debora S. Marks, and Richard E. Lenski. Core Genes Evolve Rapidly in the Long-Term Evolution Experiment with Escherichia coli. *Genome Biology and Evolution*, 9(4):1072–1083, April 2017.

[11] Brian D. Ondov, Todd J. Treangen, Páll Melsted, Adam B. Mallonee, Nicholas H. Bergman, Sergey Koren, and Adam M. Phillippy. Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biology*, 17(1):132, June 2016.

[12] Eric W Sayers, Evan E Bolton, J Rodney Brister, Kathi Canese, Jessica Chan, Donald;C Comeau, Ryan Connor, Kathryn Funk, Chris Kelly, Sunghwan Kim, and et al. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 50(D1), 2021.

[13] Andreas Untergasser, Ioana Cutcutache, Triinu Koressaar, Jian Ye, Brant C. Faircloth, Maido Remm, and Steven G. Rozen. Primer3—new capabilities and interfaces. *Nucleic Acids Research*, 40(15), 2012.

[14] Jasper van Bemmelen, Davida S Smyth, and Jasmijn A Baaijens. Amplidiff: An optimized amplicon sequencing approach to estimating lineage abundances in viral metagenomes. *bioRxiv*, pages 2023–07, 2023.

[15] Ziyuan Wang, Junjie Tan, Yanling Long, Yijia Liu, Wenyan Lei, Jing Cai, Yi Yang, and Zhibin Liu. SaAlign: Multiple DNA/RNA sequence alignment and phylogenetic tree construction tool for ultra-large datasets and ultra-long sequences based on suffix array. *Computational and Structural Biotechnology Journal*, 20:1487–1493, 2022.