Date of publication xxxx xx, xxxx, date of current version 31 10 2023.

Digital Object Identifier 10.1109/ACCESS.2023.DOI

Impact of sorting in DNA sequence compression

TIAGO FONSECA[1,2], ARMANDO J. PINHO [1,2] AND DIOGO PRATAS[1,2,3]
[1]IEETA/LASI, Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro, Portugal
[2]DETI, Department of Electronics, Telecommunications and Informatics, University of Aveiro, Portugal
[3]DoV, Department of Virology, University of Helsinki, Finland

Corresponding author: Tiago Fonseca (e-mail: tiagorfonseca@ua.pt).

ABSTRACT **Background:** The increase in the production of genomic data led to a growing need to find efficient methods that could store and analyze this type of data. Due to its redundant structure and the storage space needed, data compression is seen as an essential strategy to deal with this kind of problem. There are various tools available, from those created with a more general use intent to those created to deal specifically with this type of data. The tool present here is FASTA ANALYSIS and allows the sorting of FASTA sequences (a genomics sequence representation type) according to different criteria, taking advantage of puting closer similar blocks.

Findings: The tests performed had as an objective to compare the impact of sorting two distinct file types. On one hand, there were generated nearly-synthetic sequences with the AlcoR toolkit, who permits to build sequences from specific input data. On the other hand, there were used groups of specific genomes and collections of genomes organized according to their type (bacteria, viral, etc). The results here presented show a set of specific cases which document how the compression of sorted files is affected by the compressor and the file composition.

Conclusions: FASTA ANALYSIS is a tool which allows the improvement of compression efficiency of FASTA sequences, by sorting the sequences present on Multi-FASTA files, according to different criteria. The use of this tool is easy and intuitive. FASTA ANALYSIS is distributed under GPLv3 and it's available for free download at https://github.com/tiagof1993/FASTA-ANALYSIS.

INDEX TERMS Data Compression, FASTA files, DNA sequences, Bioinformatics, Block-sorting

I. INTRODUCTION

THE compression and storage of genomic data is a crucial part of biology and medicine studies. The storage of this type of data has been discussed since the early 90s and a lot of research and developments have been done in this field. One of the main problems faced is that with the growth of the available information, the capacity to save it becomes rather limited. To deal with this question, the use of lossless compression is an effective method who permits the conservation of substantial quantities of data in a reduced space. During the last decades, several compression algorithms and implementations have been developed to tackle this obstacle.

A. OBJECTIVES

The work displayed here has a set of objectives being the most relevant of them, the analysis of the impact of sorting DNA sequences on compression efficiency. Other relevant objectives are: conduct sorting tests on FASTA [2] sequences and verify if they bring any improvement to the compression of multi-FASTA files, determine how compressors deal with the data they receive, how the compressor used affects compression ratio and time, discern how the content of the file affects the compression efficiency and how the existent genomes also shape the compression process.

II. METHODS

This section will describe the developed tool in addition to the process behind the creation of the Simulated nearly-synthetic sequences.

A. FASTA ANALYSIS

FASTA ANALYSIS is a tool capable of sorting the DNA sequences inside a multi-FASTA file using a set of different criteria.

To achieve this, FASTA ANALYSIS carries out a set of operations which are reported here:

- Read the sequences present in the input file and store the positions that define their beginning and end in a vector;
- Sort the positions vector based on the content present within those positions;
- Re-read the sequences on the input file in the order defined by the sorted positions vector and write that content directly on the output file.

An illustration of this process can be seen in Figure 1.

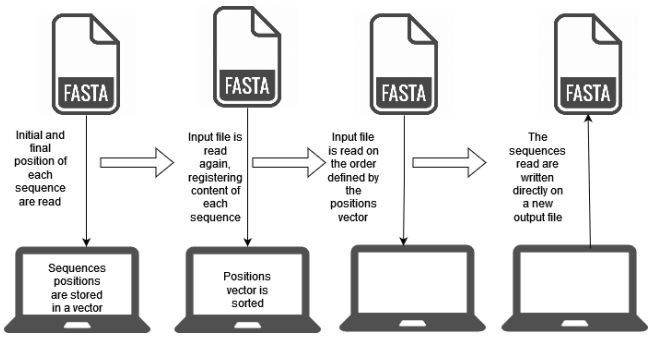

FIGURE 1: Methodology of the FASTA ANALYSIS tool.

FASTA ANALYSIS is able to sort files based on 5 different criteria. Those are: the sequence's size, the number of AT (Adenine and Thymine) occurrences, the number of CG (Cytosine and Guanine) occurrences, the percentage of AT bases present in the sequence and the percentage of CG bases present in the sequence. There's also a method which shuffles the sequences present on the file. Although it does not bring compression improvements, it can be useful to benchmark the other methods.

This tool is used as an executable, compiled directly from C++ source code. Typically, for its execution there are utilized a group of arguments, namely an input multi-FASTA file, a sorting type, and the seed value used for shuffle operations.

B. SIMULATED NEARLY-SYNTHETIC SEQUENCES

The term "Simulated nearly-synthetic Sequences" defines a set of multi-FASTA files created with the help of the AlcoR tookit [1]. This creation process is divided into two stages. In the first, there is created a set of sequences written on different FASTA files. In the second stage, versions of these sequences are generated and organized on a single multi-FASTA file.

III. RESULTS

The results obtained using this benchmark show a wide variety of cases in terms of compression gains and compression times. Table 1 reports in a concise form the most relevant results. The results showed above depict the best and worst scenarios for every compressor that was utilized.

Table 1: Best and worst cases obtained on this benchmark.

File	Compressor	Sorting Type	Gain(%)
Shuffled Viral-Bacterial	JARVIS3	CG	23,90%
Bacterial	MBGC	ATP	20,35%
Shuffled Viral-Bacterial	NAF	size	16,75%
Shuffled Viral-Bacterial	LZMA	size	15,45%
Shuffled Viral-Bacterial	MFCompress 8	size	14,85%
Shuffled Viral-Bacterial	MFCompress 4	CG	12,83%
Shuffled Viral-Bacterial	bzip2	AT	5,05%
Shuffled Viral-Bacterial	Gzip	size	0,85%
Viral	Gzip	CG	-0,10%
Bacterial	LZMA	ATP	-0,55%
Viral	bzip2	CG	-2,20%
Viral	MFCompress 8	ATP	-2,80%
Viral	JARVIS3	CGP	-2,90%
Viral	NAF	AT	-5,83%
Viral	MFCompress 4	ATP	-6%
Viral-Bacterial	MBGC	CGP	-28,78%

Among them the most relevant are the top two on the list. The first one, not only is the best case overall, but it also shows the combination of the compressor [3] and the dataset with the best results overall. The second case on the list represents a more specific scenario where the bacteria dataset is used by a compressor specialized in those type of genomes, as it is the case of MBGC [4]. Each sorting strategy was applied to 5 different files using 7 compressors. For every file, a total of 40 different scenarios were benchmarked, due to the fact that MFCompress [5] was tested with 2 partition modes. Each scenario comprises a group of distinct executions on a pre-selected group of levels.

IV. CONCLUSIONS

The results shown above permit us to draw some conclusions regarding the compressors, the sorting strategies used, and the cases where they best fit. One particularly noticeable aspect is the fact that on average the DNA-specific data compressors outperform the remaining ones. Furthermore, it can be said that the sorting strategy adopted also impacts the efficiency of each compressor. An example of that are the improvements obtained by DNA-specific compressors on files sorted by number of bases. Due to a high redundancy of their content, the simulated nearly-synthetic files achieve higher compression ratios on the lower compression levels but fall dramatically on the higher ones.

V. FUTURE WORK

There are still some aspects which deserve further work. The first of them is the use of larger files on the benchmark tests which could prove useful in assessing the compression efficiency with a higher assurance degree. It's also important to study further how the content of a file influences the compression. That can be done by performing a benchmark with larger sequences or sequences representing species from different biological families. FASTA ANALYSIS can also be improved through a more diverse set of sorting options and the optimization of those already existent.

REFERENCES

[1] Jorge M Silva, Weihong Qi, Armando J Pinho, and Diogo Pratas. "AlcoR: alignment-free simulation, mapping, and visualization of low-complexity regions in biological data," in GigaScience (2023).
[2] William R Pearson and David J Lipman. "Improved tools for biological sequence comparison," in PNAS (1988).
[3] Armando J Pinho and Diogo Pratas. "JARVIS3 - genome data compressor" URL: https://github.com/cobilab/jarvis3
[4] Szymon Grabowski and Tomasz M. Kowalski. "MBGC: multiple bacteria genome compressor," in GigaScience (2022).
[5] Armando J Pinho and Diogo Pratas. "MFCompress: a compression tool for FASTA and multi-FASTA data," in Bioinformatics (2014).