

Support *Document*

July 25, 2024

1 How to compile and run

Clone the benchmarking repository

```
git clone git@github.com:Dinis-Lei/datacompression_benchmark.git
```

Run benchmarking process (optional)

```
mkdir input/$TEST_FOLDER
./run.sh -r $N_RUNS -t $TEST_FOLDER -f $COMPRESSOR_INFO

./run_decompress.sh -r $N_RUNS -t $TEST_FOLDER -f $COMPRESSOR_INFO
```

View plots by running `plot.ipynb`

2 Figures

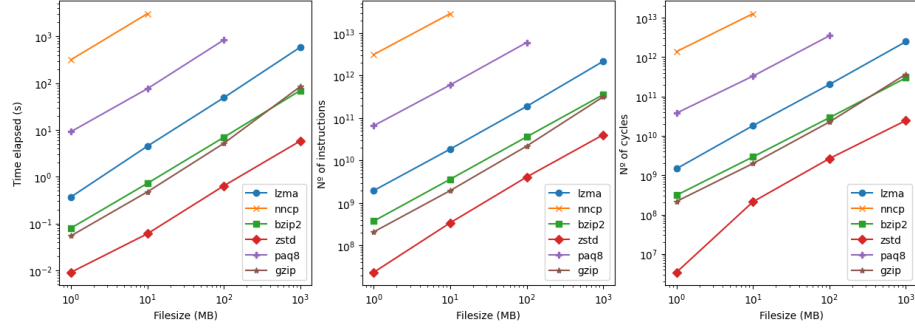


Figure 1: Metrics of the general purpose algorithms when compressing text files. The plots display the time elapsed, the number of instructions and the number of cycles executed by the algorithms respectively.

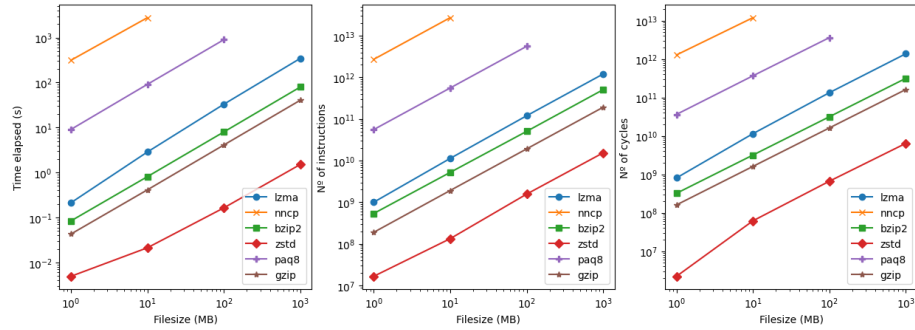


Figure 2: Metrics of the general purpose algorithms when compressing computer generated text files. The plots display the time elapsed, the number of instructions and the number of cycles executed by the algorithms respectively.

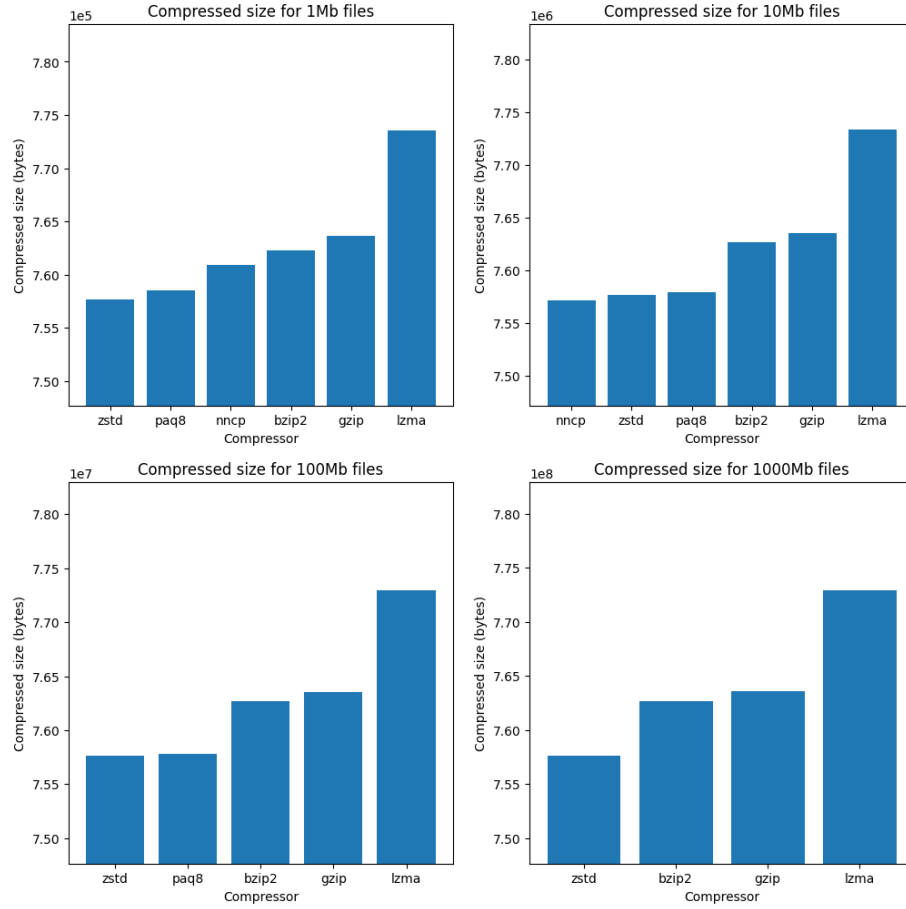


Figure 3: Size of the compressed files generated by the general purpose algorithms when compressing computer generated text files. Each plot is associated to one of the four text files used in the benchmark.

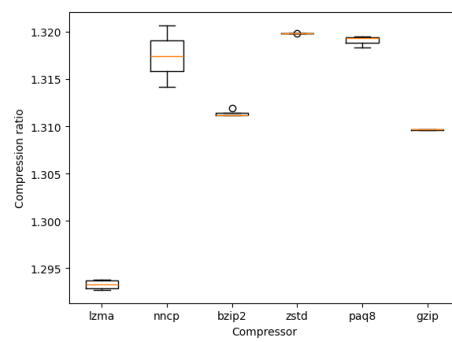


Figure 4: Compression ratio of each general purpose compression algorithm for the computer generated text file benchmark.

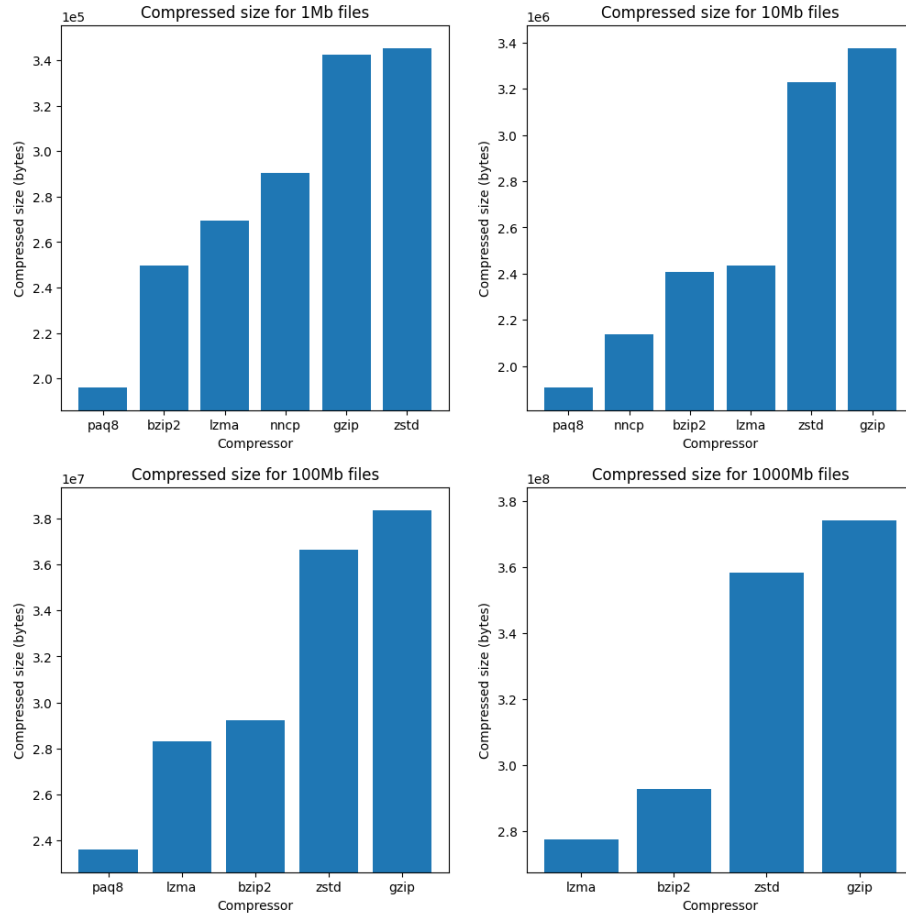


Figure 5: Size of the compressed files generated by the general purpose algorithms when compressing text files. Each plot is associated to one of the four text files used in the benchmark.

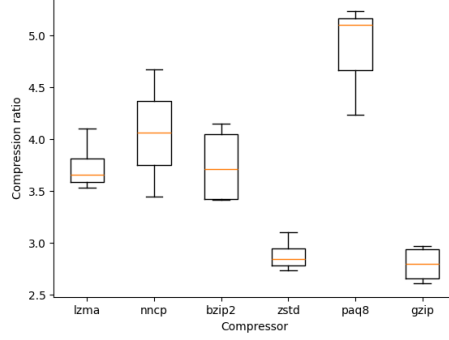


Figure 6: Compression ratio of each general purpose compression algorithm for the text file benchmark.

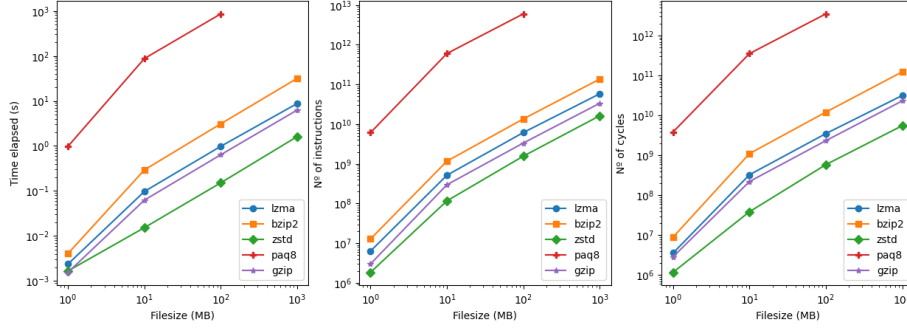


Figure 7: Metrics of the general purpose algorithms when decompressing text files. The plots display the time elapsed, the number of instructions and the number of cycles executed by the algorithms respectively.

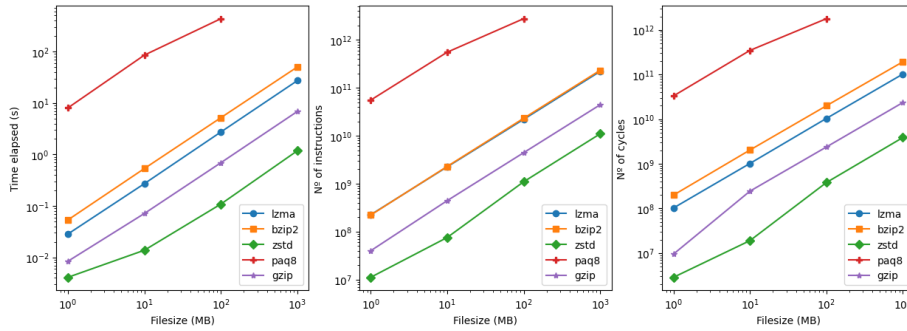


Figure 8: Metrics of the general purpose algorithms when decompressing computer generated text files. The plots display the time elapsed, the number of instructions and the number of cycles executed by the algorithms respectively.

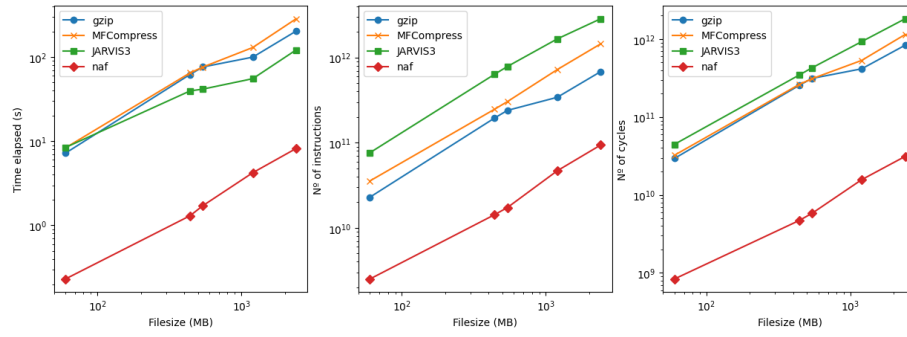


Figure 9: Metrics of the specific algorithms when compressing FASTA files. The plots display the time elapsed, the number of instructions and the number of cycles executed by the algorithms respectively.

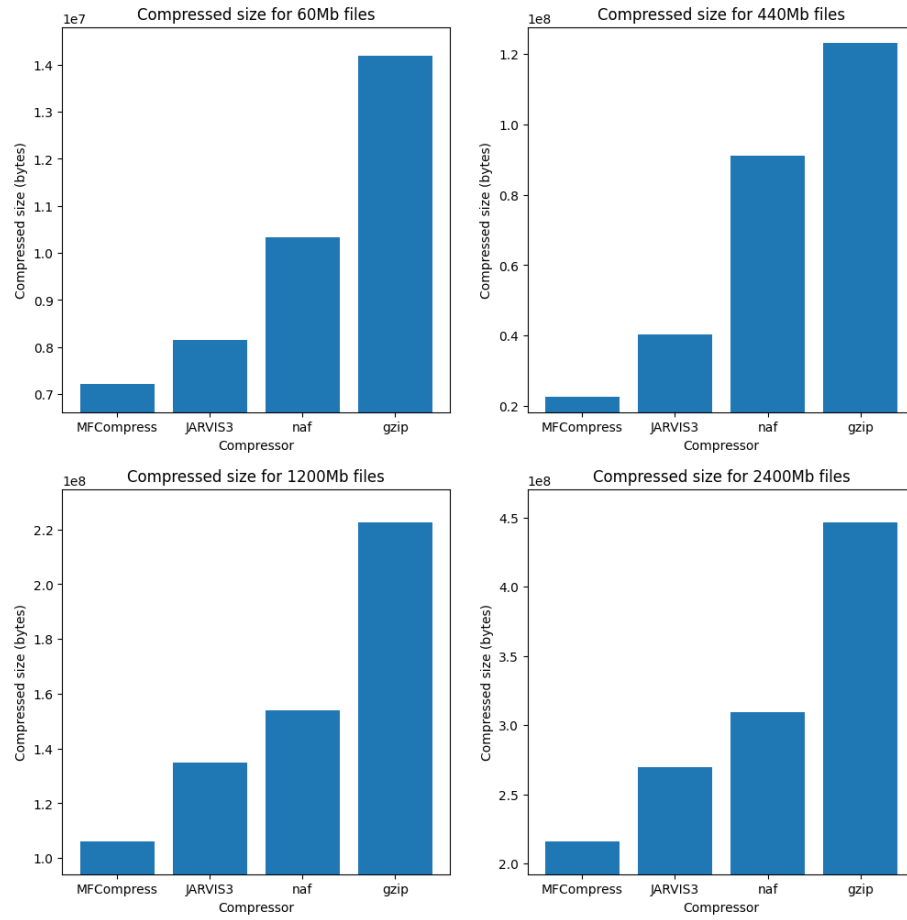


Figure 10: Size of the compressed files generated by the specific algorithms when compressing FASTA files. Each plot is associated to one of the four text files used in the benchmark.

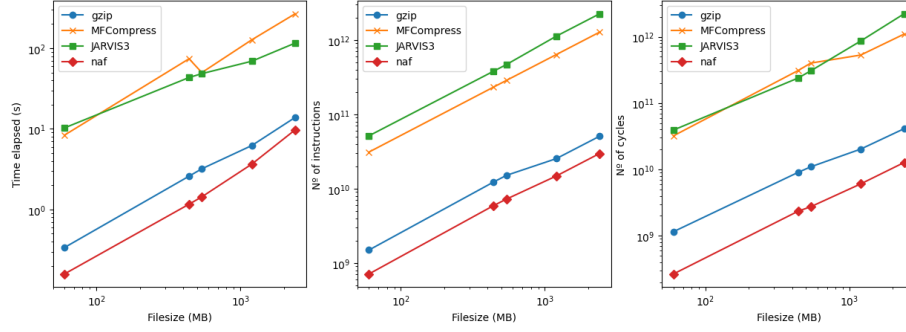


Figure 11: Metrics of the specific algorithms when decompressing FASTA files. The plots display the time elapsed, the number of instructions and the number of cycles executed by the algorithms respectively.

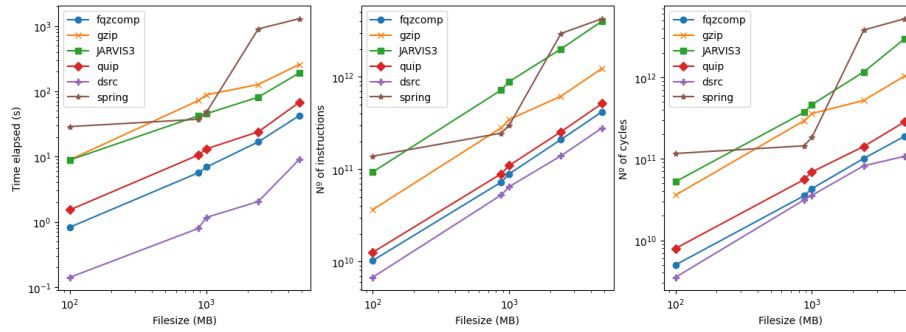


Figure 12: Metrics of the specific algorithms when compressing FASTQ files. The plots display the time elapsed, the number of instructions and the number of cycles executed by the algorithms respectively.

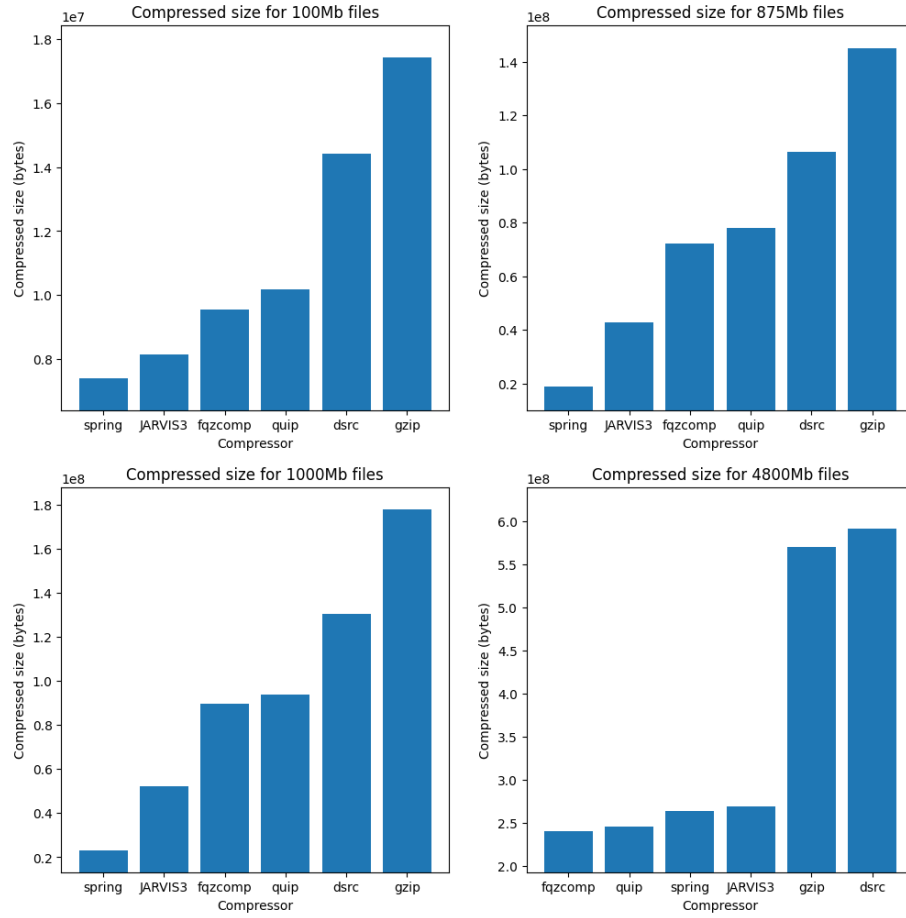


Figure 13: Size of the compressed files generated by the specific algorithms when compressing FASTQ files. Each plot is associated to one of the four text files used in the benchmark.

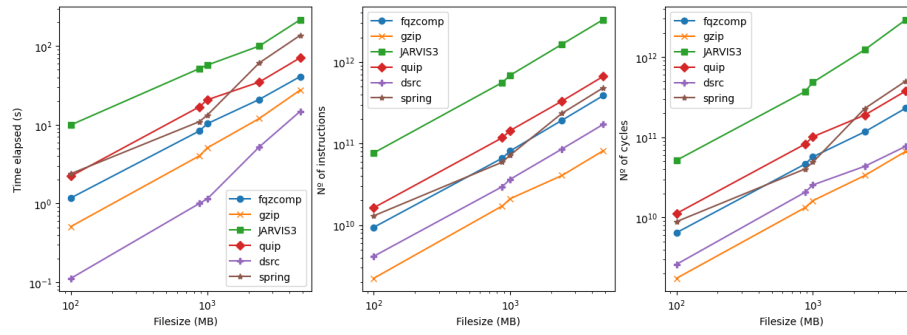


Figure 14: Metrics of the specific algorithms when decompressing FASTQ files. The plots display the time elapsed, the number of instructions and the number of cycles executed by the algorithms respectively.