Is it time to replace gzip? Comparison of modern compressors for molecular sequence databases

Kirill Kryukov*, Mahoko Takahashi Ueda, So Nakagawa, Tadashi Imanishi
Department of Molecular Life Science, Tokai University School of Medicine,
Isehara, Kanagawa 259-1193, Japan.
*Correspondence: kkryukov@gmail.com

Abstract
Nearly all molecular sequence databases currently use gzip for data compression. Ongoing rapid accumulation of stored data calls for more efficient compression tool. We systematically benchmarked the available compressors on representative DNA, RNA and Protein datasets. We tested specialized sequence compressors 2bit, BLAST, DNA-COMPACT, DELIMINATE, Leon, MFCompress, NAF, UHT and XM, and general-purpose compressors brotli, bzip2, gzip, lz4, lzop, lzturbo, pbzip2, pigz, snzip, xz, zpaq and zstd. Overall, NAF and zstd performed well in terms of transfer/decompression speed. However, checking benchmark results is necessary when choosing compressor for specific data type and application. Benchmark results database is available at: http://kirr.dyndns.org/sequence-compression-benchmark/.

Keywords: compression; benchmark; DNA; RNA; protein; genome; sequence; database.

Molecular sequence databases store and distribute DNA, RNA and protein sequences as compressed FASTA files. Currently, nearly all databases universally depend on gzip for compression. This incredible longevity of the 26-year-old compressor probably owes to multiple factors, including conservatism of database operators, wide availability of gzip, and its generally acceptable performance. Through all these years the amount of stored sequence data kept growing steadily (Karsch-Mizrach et al., 2018), increasing the load on database operators, users, storage systems and network infrastructure. However, someone thinking to replace gzip invariably faces the question: Which of the numerous available compressors to choose? And will the resulting gains be even worth the trouble of switching? Previous attempts at answering these questions (Zhu et al., 2013; Hosseini et al., 2016; Sardaraz and Tahir, 2016; Biji and Achuthsankar, 2017) are limited by testing too few compressors and by using restricted test data. Therefore we set out to benchmark a broader selection of available compressors on a variety of relevant test data. Specifically, we focused on the most common task of compressing DNA, RNA and protein sequences, stored in FASTA format, without using reference sequence.

Biological sequence compression was first proposed in 1986 (Walker and Willett, 1986), and the first practical compressor was made in 1993 (Grumbach and Tahi, 1993). A lively field emerged that produced a stream of methods, algorithms, and software tools for sequence compression. In this study we tested all sequence compressors that we could find and make to work: XM (Cao et al., 2007), DELIMINATE (Mohammed et al., 2012), DNA-COMPACT (Li et al., 2013), MFCompress (Pinho and Pratas, 2014), Leon (Benoit et al., 2015), UHT (Al-Okaily et al., 2017) and NAF (Kryukov et al., 2019). We also included the relatively compact among homology search database formats: BLAST (Altschul et al., 1990) and 2bit - a database format of BLAT (Kent, 2002). In addition, other than gzip, we tested a comprehensive array of general purpose compressors: brotli, bzip2, lz4, lzop, lzturbo, pbzip2, pigz, snzip, xz, zpaq and zstd. We also included a null-compressor, represented by the "cat" command, as a control. See Supplementary Table 1 for the list of compressors we used.

For the test data, we used a variety of commonly used sequence datasets in FASTA format. We used four kinds of test datasets: (1) Individual genomes of various sizes, as example of non-repetitive data, (2) Repetitive DNA datasets, including collections of mitochondrial genomes, influenza virus sequences, bacterial genomes, and human EST data, (3) RNA datasets consisting of 16S rRNA gene sequences, and (4) Standard protein datasets. See Supplementary Table 2 for the list of test data.

We benchmarked each compressor on each test dataset, except in cases of incompatibility (DNA compressors cannot compress protein data) or excessive time requirement (some compressors are so slow

that they would take weeks on larger datasets). For compressors with adjustable compression level, we tested the relevant range of levels. We also tested both 1 and 4-thread variants of compressors that support multi-threading. In total, we used 274 settings of 22 compressors. From each run we recorded compressed size, compression time and decompression time. The details of the method and raw benchmark data are available in Supplementary Methods and Supplementary Data, respectively. Furthermore, we created a web interface to allow convenient browsing and visualizing of the benchmark results: http://kirr.dyndns.org/sequence-compression-benchmark/. It allows selecting any combination of test datasets and compressors, and displaying the results in the form of tables, column graphs (Fig.1), and scatterplots (Fig.2).

The choice of measure for evaluating compressor performance depends on data use pattern. For data archival, compactness is the single most important criterion. For public database scenario, the key measure is how long time it takes from initiating the download of compressed files until accessing the decompressed data. This time consists of transfer time plus decompression time (TD-Time). Corresponding transfer-decompression speed (TD-Speed) can be computed as Original Size / TD-Time. In this case compression time is relatively unimportant, since compression happens only once, while transfer and decompression times affect every user of the database. For one-time data transfer, all three steps of compression, transfer and decompression are timed (CTD-Time), and used for computing the resulting overall speed (CTD-Speed).

A total of 15 measures, including the above-mentioned ones, can be computed on our benchmark website. Any of these measures can be used for selecting the best setting of each compressor and for sorting the list of compressors. These measures can be then shown in a table and visualized in column charts and scatterplots. This allows tayloring the output to answer specific questions, such as what compressor is better at compressing particular kind of data, or which setting of each compressor performs best at particular task. The link speed that is used for estimating transfer times is configurable. We used the default link speed of 100 Mbit/sec, because this is the most common speed of a fixed broadband internet connection, but other speeds can be entered in the website interface.

Fig.1 compares the performance of best settings of all 22 compressors on a set of four small genomes. This set, consisting of genomes smaller than 10 MB, is the only selection among our current test data that allows comparing all compressors, because some compressors are too slow to work with larger data. Fig.1 shows that specialized sequence compressors achieve excellent compression ratio on genome sequences. However, when total TD-Speed or CTD-Speed is considered (measures that are important in practical applications), most of sequence compressors (except NAF) fall far behind the general-purpose ones. Figs S1-S3 show similar charts for other test datasets.

Benchmark results of all settings were visualized as scatterplots (Figs 2, S4-S7), which revealed the complex relationship between compressors and between their settings, based on various measures. Charts similar to Figs 1 and 2 can be produced on our website for any selection of test data, compressors, and performance measures. The available options include log vs linear scale and custom axis ranges. All values can be shown as relative to a specific reference compressor.

Based on our results, we can make the following recommendations for choosing the compressor: For archival (maximum compactness) of genome sequences, "mfc-3", "naf-22" and "zpaq-5" can be preferred (number in each compressor name indicates the setting). However, care has to be taken since "zpaq-5" is extremely slow, and "mfc-3" has slow decompression, leaving "naf-22" as the more practical choice. For public genome databases (optimal transfer and decompression speed), "naf-22" is the clear choice. For one-time transfer, "naf-1" gives the best performance. The relative advantages over "gzip-9" (current de-facto standard) are: 1.4 times in compactness (by "naf-22"), 1.6 times in TD-Speed (by "naf-22") and 16 times in CTD-Speed (by "naf-1").

In case of repetitive DNA/RNA data, "naf-22" is the optimal choice for archival and database applications (5.7 times better than "gzip-9" in compactness, and 4.9 times better than "gzip-9" in TD-Speed). For one-time transfer, "zstd-4-4t" and "naf-3" give the best performance (26 and 25 times better than "gzip-9", respectively).

When applied to protein sequences, "naf-22", "lzturbo-49" and "xz-e9" provide good compactness for data archival (1.8, 1.8 and 1.7 times better than "gzip-9", respectively). For database scenario, "naf-21" performs best (2.0 times better than "gzip-9"). For one-time transfer, "zstd-2-4t" and "naf-1" provide the best performance (2.4 times better than "gzip-9").

Our results show that continued use of gzip is far from an optimal choice. Transitioning from gzip to a better compressor is especially beneficial with repetitive DNA/RNA datasets; however, for genome and protein data the gains are also significant. Overall, we recommend using "naf-22" as the default compressor for sequence databases and "naf-1" - for one-time data transfer. However, it is best to check the results for specific data type and application.

Our comprehensive benchmark will help in navigating the complex landscape of data compression. With dozens of compressors available, making an informed choice is not an easy task and requires careful analysis of the project requirements, data type and compressor capabilities. Our benchmark is the first resource providing a detailed practical evaluation of various compressors on a wide range of biological datasets. Using our Sequence Compression Benchmark database users can analyze compressor performances on variety of metrics, and construct custom reports for answering project-specific questions.

In contrast to previous studies that showed their results in static tables, our project is dynamic in two important senses: (1) The result tables and charts can be dynamically constructed for a custom selection of test data, compressors, and measured performance numbers, and (2) Our study is not a one-off benchmark, but marks the start of a project where we will continue to add compressors and test data.

Making an informed choice of compressor with the help of our benchmark will lead to increased compactness of sequence databases, with shorter time required for downloading and decompressing. This will reduce the load on network and storage infrastructure, and increase speed and efficiency in biological and medical research.

References

Al-Okaily, A., et al. (2017) "Toward a Better Compression for DNA Sequences Using Huffman Encoding" *J. Comp. Biol.*, 24(4), 280–288. doi:10.1089/cmb.2016.0151

Altschul, S.F., et al. (1990) "Basic local alignment search tool" *J. Mol. Biol.*, 215(3), 403-410. doi:10.1016/S0022-2836(05)80360-2

Benoit, G., et al. (2015) "Reference-free compression of high throughput sequencing data with a probabilistic de Bruijn graph" *BMC Bioinformatics*, 16:288. doi:10.1186/s12859-015-0709-7

Biji, C.L. and Achuthsankar, S.N. (2017) "Benchmark Dataset for Whole Genome Sequence Compression" *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 14(6), 1228-1236. doi:10.1109/TCBB.2016.2568186

Cao, M.D., et al. (2007) "A simple statistical algorithm for biological sequence compression" *Data Compression Conference*, DCC '07, Snowbird, UT, IEEE Computer Society, pp. 43-52. doi:10.1109/DCC.2007.7

Grumbach, S. and Tahi, F. (1993) "Compression of DNA sequences" Data Compression Conference, DCC '93, Snowbird, Utah. IEEE Computer Society, pp. 340-350. doi:10.1109/DCC.1993.253115

Hosseini, M., et al. (2016) "A Survey on Data Compression Methods for Biological Sequences" *Information*, 7(4), 56. doi: 10.3390/info7040056

Karsch-Mizrachi, I., et al. (2018) "The international nucleotide sequence database collaboration" *Nucleic Acids Res.*, 46(Database issue), D48–D51. doi:10.1093/nar/gkx1097

Kent, W.J. (2002) "BLAT - The BLAST-Like Alignment Tool" *Genome Research*, 12(4), 656-664. doi:10.1101/gr.229202

Kryukov, K., et al. (2019) "Nucleotide Archival Format (NAF) enables efficient lossless reference-free compression of DNA sequences" *Bioinformatics* (in press), btz144. doi:10.1093/bioinformatics/btz144

Li, P., et al. (2013) "DNA-COMPACT: DNA COMpression Based on a Pattern-Aware Contextual Modeling Technique" *PLoS ONE*, 8(11), e80377. doi:10.1371/journal.pone.0080377

Mohammed, M.H., et al. (2012) "DELIMINATE — a fast and efficient method for loss-less compression of genomic sequences" *Bioinformatics*, 28, 2527–2529. doi:10.1093/bioinformatics/bts467

Pinho, A.J. and Pratas, D. (2014) "MFCompress: a compression tool for FASTA and multi-FASTA data" *Bioinformatics*, 30, 117-118. doi:10.1093/bioinformatics/btt594

Sardaraz, M. and Tahir, M. (2016) "Advances in high throughput DNA sequence data compression" *J. Bioinform. Comput. Biol.*, 14(3), 1630002. doi:10.1142/S0219720016300021

Walker, J.R. and Willett, P. (1986) "Compression of nucleic acid and protein sequence data" *Comput. Appl. Biosci.*, 2(2), 89-93.

Zhu, Z., et al. (2013) "High-throughput DNA sequence data compression" *Brief. Bioinform.*, 16(1), 1-15. doi:10.1093/bib/bbt087

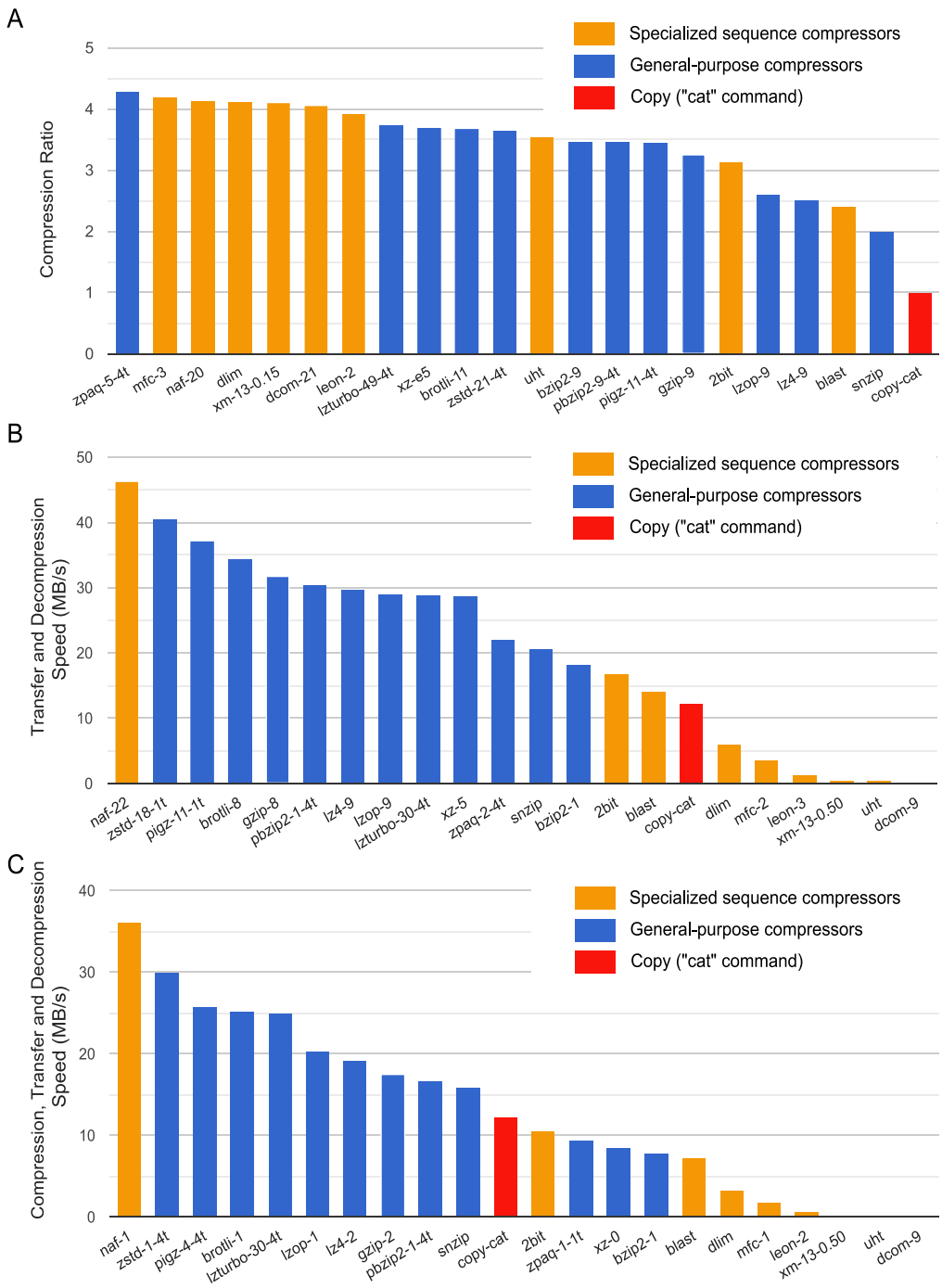

Fig. 1. Comparison of 22 compressors on a set of 4 genomes. Best settings of each compressor are selected based on different aspects of performance: A - compressed size, B - transfer and decompression speed, and C - compression, transfer and decompression speed. Specialized sequence compressors are shown in orange color, and general-purpose compressors are shown in blue. The copy-compressor ("cat" command), shown in red color, is included as a control. The selected settings of each compressor are shown in their names, after hyphen. Multi-threaded compressors have "-1t" or "-4t" at the end of their names to indicate the number of threads used. Test data consists of 4 bacterial and protist genomes (accession numbers: GCA_000398605.1, GCA_000211355.2, GCA_000988165.1, GCA_000165345.1). Benchmark CPU: Intel Xeon E5-2643v3 (3.4 GHz). Link speed of 100 Mbit/s was used for estimating the transfer time.

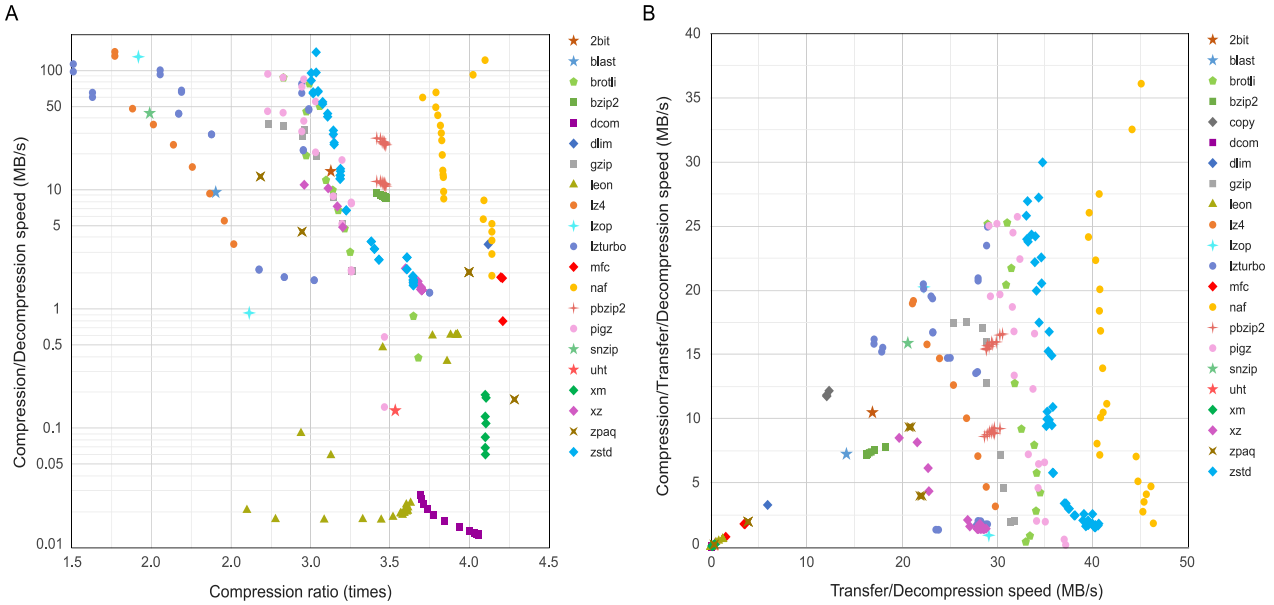

Fig. 2. Comparison of 274 settings of 22 compressors on a set of 4 genomes. Each point represents a particular setting of some compressor. Panel A shows the relationship between compression ratio and compression + decompression speed. Panel B shows the transfer + decompression speed plotted against compression + transfer + decompression speed. Test data consists of 4 bacterial and protist genomes (accession numbers: GCA_000398605.1, GCA_000211355.2, GCA_000988165.1, GCA_000165345.1). Benchmark CPU: Intel Xeon E5-2643v3 (3.4 GHz). Link speed of 100 Mbit/s was used for estimating the transfer time.