Nucleotide Archival Format (NAF) enables efficient lossless reference-free compression of DNA sequences

Kirill Kryukov[1*], Mahoko Takahashi Ueda[2], So Nakagawa[1,2], Tadashi Imanishi[1]

[1]Department of Molecular Life Science, Tokai University School of Medicine, Isehara, Kanagawa 259-1193, Japan.
[2]Micro/Nano Technology Center, Tokai University, Hiratsuka, Kanagawa 259-1292, Japan.

*To whom correspondence should be addressed.

Abstract

Summary: DNA sequence databases use compression such as gzip to reduce the required storage space and network transmission time. We describe Nucleotide Archival Format (NAF) - a new file format for lossless reference-free compression of FASTA and FASTQ-formatted nucleotide sequences. NAF compression ratio is comparable to the best DNA compressors, while providing dramatically faster decompression. We compared our format with DNA compressors: DELIMINATE and MFCompress, and with general purpose compressors: gzip, bzip2, xz, brotli, and zstd.

Availability and implementation: NAF compressor and decompressor, as well as format specification are available at https://github.com/KirillKryukov/naf. Format specification is in public domain. Compressor and decompressor are open source under the zlib/libpng license, free for nearly any use.

Contact: kkryukov@gmail.com

Introduction

DNA sequence databases are growing exponentially, owing to the continuing advances in sequencing technologies. Data compression is typically used for all stored DNA sequence data to save storage space and network transmission times. In 1993 the first specialized DNA compressor was proposed (Grumbach and Tahi, 1993). Since then, numerous DNA compressors were developed (e.g., Cao et al., 2007, Li et al., 2013, Benoit et al., 2015, Al-Okaily et al., 2017). In our experience only two compressors pass the practicality threshold: DELIMINATE (Mohammed et al., 2012) and MFCompress (Pinho and Pratas, 2014). They are stable, support commonly used features of FASTA format, and are efficient enough to be able to handle practical tasks such as compressing (and decompressing) entire vertebrate genomes. Although they achieve impressive compression ratios, both DELIMINATE and MFCompress have very slow decompression, significantly limiting their usefulness with large databases.

Despite the numerous studies on DNA compression, currently the majority of databases continue to rely on gzip (https://www.gzip.org/) for DNA compression. We attribute this enduring popularity to gzip's wide availability, robustness, and speed (especially decompression speed). These qualities appear to be able to outweigh gzip's less than stellar compression ratio. Other popular general purpose compressors have been developed, such as bzip2 (http://www.bzip.org/) and lzma/xz (https://tukaani.org/xz/format.html). In addition, recent years have seen the emergence of a new generation of advanced general purpose compressors, most notably brotli (https://github.com/google/brotli) and zstd (https://github.com/facebook/zstd). These compressors improve upon gzip performance, but still cannot touch specialized DNA compressors in compression strength.

In this work we describe a new DNA compression format called Nucleotide Archival Format (NAF). NAF provides a state of the art compression ratio, on par with DELIMINATE and slightly behind MFCompress. At the same time, it provides 30 to 80 times faster decompression. NAF compresses and decompresses from/to FASTA and FASTQ formats. NAF supports masked sequence and ambiguous IUPAC nucleotide codes. NAF has no restrictions on sequence length, and does not require reference sequences.

Methods

Analogously to many previous methods, including DELIMINATE and MFCompress, NAF compression operates by splitting the input into headers, nucleotide sequences, mask (in case of masked sequence), and qualities (in case of FASTQ input), which are processed separately. Nucleotide sequences are concatenated together, with lengths stored separately. The combined nucleotide sequence is then converted into 4-bit encoding, which stores 2 nucleotides per byte. This representation is extremely fast, for both encoding and decoding, and allows natively representing ambiguous IUPAC nucleotide codes (NC-IUB, 1985), including 'N', 'Y', 'R', etc. All the resulting data streams are then compressed with the general purpose compressor zstd (https://facebook.github.io/zstd/).

The decompression consists of decompressing those separate streams, and re-assembling them together into FASTA or FASTQ-formatted output. NAF's high decompression speed owes to these factors: 1) Using zstd compressor, which itself is designed for fast decompression. 2) In NAF, zstd works with 4-bit encoded sequence, which means that it deals with data half the size of FASTA-format sequence. 3) Decoding of 4-bit sequence is very fast using a simple table lookup for pairs of nucleotides.

NAF implementation provides interface that is friendly to automated sequence analysis workflows. NAF compressor reads data from standard input stream, enabling on-the-fly compression of data originating from other process. Similarly, NAF decompressor allows piping decompressed sequences directly into the next analysis step. In addition, NAF allows decompressing only headers or only sequences, as well as 4-bit encoded sequence. This will allow applications such as sequence search or composition analysis to work directly with 4-bit encoded sequence.

Results

We compared NAF with DNA compressors DELIMINATE and MFCompress, as well as general purpose compressors: gzip, bzip2, xz, brotli, and zstd. Figs. 1 and S1, and Table S1 show their results on the human genome. Fig. S2 and Table S2 show average compression ratios on larger set of genomes. Table S3 shows the result on FASTQ data. Table S4 compares availability and features of these compressors.

Compression strength of NAF is close to DELIMINATE and slightly behind MFCompress. All three DNA compressors achieve markedly better compression ratio than the general purpose compressors. However what sets NAF apart is its high speed of decompression. In the human genome example, NAF decompression is faster by a factor of 35 and 78 than DELIMINATE and MFCompress, respectively. NAF compression speed is average (Table S1), but since compression is typically performed once, it's less important than decompression, which is executed many times by the users of the data.

When considering the typical application of compression for distributing data from sequence databases, we can estimate the total time that it takes from initiating download until accessing the decompressed data, for different compressors. Figs. 1B and S1 compare access times for 8 compressors, as well as for the uncompressed FASTA format, while assuming link speeds of 100 Mbit/s and 1000 Mbit/s, respectively. It can be seen that NAF enables the fastest distribution of data over network, allowing to reduce waiting time and bandwidth cost (in addition to reducing the required storage space).

Conclusion

NAF offers significant advantages over both general purpose and specialized DNA compressors. NAF's combination of compactness and decompression speed enables rapid distribution of database sequences to users. NAF's fast decompression allows storing NAF-compressed databases, decompressing them on-the-fly when necessary. We believe that our new format will help saving network bandwidth, time, and disk space, and thus contribute greatly to both operators and users of DNA sequence databases.

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

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

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

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

Nomenclature Committee of the International Union of Biochemistry (NC-IUB) (1985) "Nomenclature for incompletely specified bases in nucleic acid sequences" *Eur. J. Biochem.*, 150, 1-5.

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

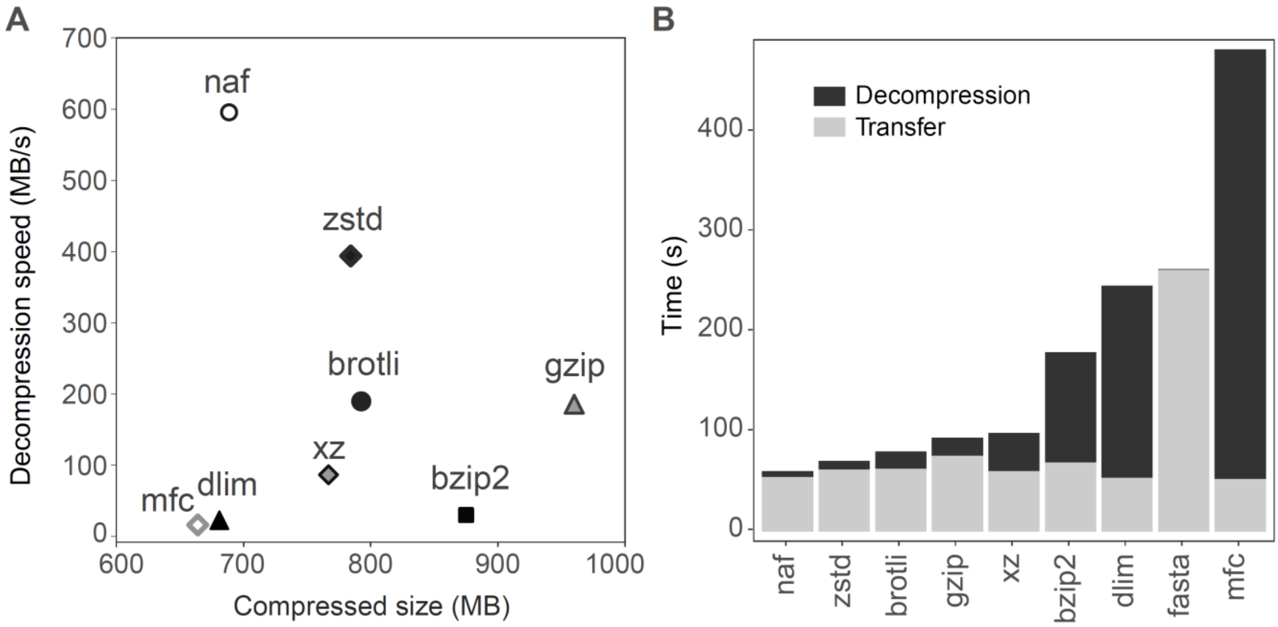

Fig. 1. Comparison of compressors on the human genome (GRCh38, 3.3 GB). (A) Compressed size vs decompression speed. "mfc" and "dlim" represent MFCompress and DELIMINATE, respectively. Compressor options used: 'gzip -9', 'bzip2 -9', 'brotli -Z', 'zstd --ultra -22', 'xz -e9', 'ennaf', 'delim a', 'MFCompressC -3'. CPU used: Intel Xeon E5-2643v3 (3.4 GHz). (B) Time of accessing the genome stored on a remote server, in various formats (including the uncompressed FASTA). Total time consists of network transfer time (estimated for a link speed of 100 Mbit/s) and decompression time (measured).