BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 1 2014, pages 117–118
doi:10.1093/bioinformatics/btt594

Sequence analysis

Advance Access publication October 16, 2013

MFCompress: a compression tool for FASTA and multi-FASTA data

Armando J. Pinho* and Diogo Pratas

IEETA, Department of Electronics, Telecommunications and Informatics, University of Aveiro, 3810–193 Aveiro, Portugal

Associate Editor: John Hancock

ABSTRACT

Motivation: The data deluge phenomenon is becoming a serious problem in most genomic centers. To alleviate it, general purpose tools, such as gzip, are used to compress the data. However, although pervasive and easy to use, these tools fall short when the intention is to reduce as much as possible the data, for example, for medium- and long-term storage. A number of algorithms have been proposed for the compression of genomics data, but unfortunately only a few of them have been made available as usable and reliable compression tools.

Results: In this article, we describe one such tool, MFCompress, specially designed for the compression of FASTA and multi-FASTA files. In comparison to gzip and applied to multi-FASTA files, MFCompress can provide additional average compression gains of almost 50%, i.e. it potentially doubles the available storage, although at the cost of some more computation time. On highly redundant datasets, and in comparison with gzip, 8-fold size reductions have been obtained.

Availability: Both source code and binaries for several operating systems are freely available for non-commercial use at http://bioinformatics.ua.pt/software/mfcompress/.

Contact: ap@ua.pt

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on July 29, 2013; revised on September 20, 2013; accepted on October 13, 2013

1 INTRODUCTION

Saying that the volume of genomic data produced every day is large is clearly a euphemism. With the dramatic drop in price of the sequencing machines (the $1000 limit for sequencing a human genome will be history shortly), virtually everyone will want to sequence everything. Unfortunately, the pace at which storage and communication resources are evolving is not enough, and the genomic data centers are being flooded with data. It is a data deluge (Berger *et al.*, 2013).

The interest for DNA compression was started with the Biocompress algorithm of Grumbach and Tahi (1993). The subsequent two decades have seen the publication of a considerable number of algorithms for compressing DNA sequences and several other forms of genomic data (e.g. Bonfield and Mahoney,2013; Cao *et al.*, 2007; Cox *et al.*, 2012; Hach *et al.*, 2012; Jones *et al.*, 2012; Korodi and Tabus,2007; Matos *et al.*, 2013; Pinho *et al.*, 2011,2012; Popitsch and Haeseler, 2013), although usually aiming more at proving the concept than at providing usable compression tools. For example, the majority of these algorithms assume data drawn from the four-letter alphabet, ACGT, ignoring other letters that can be found in DNA data sequences. No doubt that for the purpose of showing the potentialities of the algorithms, this is often a fair approach, because those letters outside the main alphabet are usually rare and, therefore, represent only a small fraction of the bits required to represent the sequence. However, a usable lossless compression tool needs to be capable of handling every letter that it finds in the file and to reproduce it exactly during decompression. Moreover, genomic files rarely contain only sequence information. Usually, they also include additional data, such as headers, quality scores and alignment information. Therefore, the compression tools need to compress these data as well in an efficient way.

In this article, we describe MFCompress, a tool for compressing FASTA and multi- FASTA files. Recently, Mohammed *et al.* (2012) proposed DELIMINATE, also a compression method for FASTA and multi-FASTA files, which relies on a preprocessing stage, where header and sequence data are separated and transformed, followed by a general purpose compressor (7-Zip).

The MFCompress tool described here provides better compression than DELIMINATE for the large majority of the files used in the benchmarking dataset, at a similar compression and decompression time. This dataset is an extended version of the benchmarking dataset used by Mohammed *et al.* (2012), to which we added some larger files, due to its increasing importance and commonness. MFCompress relies on multiple competing finite-context models and arithmetic coding, a powerful approach for DNA data compression (Pinho *et al.*, 2011).

2 METHODS

The compression tool that we describe in this article relies on probabilistic models (finite-context models) that comply to the Markov property, i.e. that estimate the probability of the next symbol of the information source using the $k > 0$ immediate past symbols (order-k context) to select the probability distribution. MFCompress uses single finite-context models for encoding the header text, as well as multiple competing finite-context models for encoding the main stream of the DNA sequences (Pinho *et al.*, 2011).

The compression algorithm divides the data source into two separate sub-sources: one containing the headers of the FASTA records, the other one the sequences. The sub-source that deals with the sequences may be further divided into two or three streams (Supplementary Fig. S4 of the Supplementary Material): the main stream, the extra stream and the case stream. The *main stream* is a four-symbol information source, conveying

*To whom correspondence should be addressed.

most of the information of the four DNA bases. Both upper and lower case characters representing the four DNA bases are converted to this four-symbol alphabet. If characters other than the four DNA bases are also present, they are all mapped to the '0' symbol in the main stream.

When the sequences contain other characters besides the DNA bases, another coding stream must be present to disambiguate the occurrences of the '0' symbol in the main stream. This *extra stream* is responsible for representing all non-acgt/ACGT characters that have been found in the sequences, as well as to indicate when the '0' in the main stream is an a/A DNA base.

If the sequences contain both DNA bases in upper and lower case, an additional binary symbol is associated to each symbol in the main stream, indicating the respective case type (the *case stream*).

A more detailed description of the methods is provided in the Supplementary Material.

3 RESULTS AND DISCUSSION

In the Supplementary Material, we provide compression results obtained using several popular general purpose compression methods, namely gzip, bzip2, ppmd and lzma (the last two using the versions implemented in the 7z archiver), as well as by the recent special purpose compressor DELIMINATE (Mohammed *et al*., 2012) and by the compressor that we describe in this article.

Supplementary Table S1 in the Supplementary Material shows the total compressed file size, in bytes, obtained with gzip in the FFN and FNA datasets (composed of all bacteria in the NCBI), as well as the compressing gains attained by the other methods in relation to gzip. We can see that MFCompress provides a compression gain of ~3.5% in relation to DELIMINATE for the FFN dataset and of ~4% for the FNA dataset. Compared with gzip, the compression gain of MFCompress is ~25%.

In Supplementary Table S2 of the Supplementary Material, we present the compression results for the human genome dataset (HG19). The gain of the default mode of MFCompress is marginal in comparison with DELIMINATE (only 1.8%). For the more complex coding mode, the gain is ~3.3%. In relation to gzip, the gain is >34%.

Regarding the CAMERA dataset, in Supplementary Table S4 of the Supplementary Material, we provide compression results regarding the 26 files that have been used in this dataset, showing significant gains over DELIMINATE for most of them. To give a wide range of examples, we chose files with sizes from $\sim 5 \times 10^5$ to $>3.5 \times 10^{10}$ characters, i.e. covering five orders of magnitude. For three of these files, DELIMINATE was not able to provide reliable results: in two cases the decoded file was different from the original file and in one case the encoder crashed. In relation to gzip, the size was reduced to almost half.

In Supplementary Table S5 of the Supplementary Material, we show the compression results of two highly redundant datasets (all *Escherichia* and *Salmonella* genomes of the FNA dataset). In this case, MFCompress attained an 8-fold file size reduction over gzip and >44% gain in relation to DELIMINATE.

4 CONCLUSION

For daily use, general purpose compression tools, such as gzip, may continue to play an important role in the context of genomic data processing, mainly due to its pervasiveness and relatively good speed. However, as shown in this article, special purpose compression tools can sometimes attain additional file reductions as large as 50% or even more, in relation to gzip. In our opinion, the possibility to virtually double the amount of sequence data that can be stored in a given space, exclusively by means of software compression tools, is an opportunity worthy of consideration by the genomic laboratories. Higher compression can only be obtained using more complex algorithms, often requiring some more time and memory to run. However, these additional requirements are compensated by the relief attained in terms of storage requirements. In conclusion, we believe that the compression tool reported in this article is a relevant contribution to slow down the negative impact of the data deluge that we are facing nowadays.

Funding: European Fund for Regional Development (FEDER) through the Operational Program Competitiveness Factors (COMPETE) and by the Portuguese Foundation for Science and Technology (FCT), in the context of projects FCOMP-01-0124-FEDER-022682 (FCT reference PEst-C/EEI/UI0127/2011) and Incentivo/EEI/UI0127/2013.

Conflict of Interest: none declared.

REFERENCES

Berger,B. *et al*. (2013) Computational solutions for omics data. *Nat. Rev. Genet*., **14**, 333–346.

Bonfield,J.K. and Mahoney,M.V. (2013) Compression of FASTQ and SAM format sequencing data. *PLoS One*, **8**, e59190.

Cao,M.D. *et al*. (2007) A simple statistical algorithm for biological sequence compression. In: *Data Compression Conference, DCC-2007, Snowbird, Utah*. IEEE Computer Society, pp. 43–52.

Cox,A.J. *et al*. (2012) Large-scale compression of genomic sequence databases with the Burrows-Wheeler transform. *Bioinformatics*, **28**, 1415–1419.

Grumbach,S. and Tahi,F. (1993) Compression of DNA sequences. In: *Data Compression Conference, DCC-93, Snowbird, Utah*. IEEE Computer Society, pp. 340–350.

Hach,F. *et al*. (2012) SCALCE: boosting sequence compression algorithms using locally consistent encoding. *Bioinformatics*, **28**, 3051–3057.

Jones,D.C. *et al*. (2012) Compression of next-generation sequencing reads aided by highly efficient de novo assembly. *Nucleic Acids Res*., **40**, e171.

Korodi,G. and Tabus,I. (2007) Normalized maximum likelihood model of order-1 for the compression of DNA sequences. In: *Data Compression Conference, DCC-2007, Snowbird, Utah*. IEEE Computer Society, pp. 33–42.

Matos,L.M.O. *et al*. (2013) A compression model for DNA multiple sequence alignment blocks. *IEEE Trans. Inf. Theory*, **59**, 3189–3198.

Mohammed,M.H. *et al*. (2012) DELIMINATE - a fast and efficient method for loss-less compression of genomic sequences. *Bioinformatics*, **28**, 2527–2529.

Pinho,A.J. *et al*. (2011) On the representability of complete genomes by multiple competing finite-context (Markov) models. *PLoS One*, **6**, e21588.

Pinho,A.J. *et al*. (2012) GReEn: a tool for efficient compression of genome resequencing data. *Nucleic Acids Res*., **40**, e27.

Popitsch,N. and Haeseler,A. (2013) NGC: lossless and lossy compression of aligned high-throughput sequencing data. *Nucleic Acids Res*., **41**, e27.