BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 15 2014, pages 2213–2215
doi:10.1093/bioinformatics/btu208

Sequence analysis

Advance Access publication April 18, 2014

DSRC 2—Industry-oriented compression of FASTQ files

Łukasz Roguski[1] and Sebastian Deorowicz[2,*]

[1]Polish-Japanese Institute of Information Technology, 02-008 Warszawa and [2]Institute of Informatics, Silesian University of Technology, 44-100 Gliwice, Poland

Associate Editor: Michael Brudno

ABSTRACT

Summary: Modern sequencing platforms produce huge amounts of data. Archiving them raises major problems but is crucial for reproducibility of results, one of the most fundamental principles of science. The widely used gzip compressor, used for reduction of storage and transfer costs, is not a perfect solution, so a few specialized FASTQ compressors were proposed recently. Unfortunately, they are often impractical because of slow processing, lack of support for some variants of FASTQ files or instability. We propose DSRC 2 that offers compression ratios comparable with the best existing solutions, while being a few times faster and more flexible.

Availability and Implementation: DSRC 2 is freely available at http://sun.aei.polsl.pl/dsrc. The package contains command-line compressor, C and Python libraries for easy integration with existing software and technical documentation with examples of usage.

Contact: sebastian.deorowicz@polsl.pl

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

Received on December 3, 2013; revised on April 1, 2014; accepted on April 16, 2014

1 INTRODUCTION

Genome sequencing has growing impact on medicine. There are emerging projects like the Personal Genome Project (Ball *et al.*, 2012) or the Million Veteran Program (Roberts, 2013), in which hundreds of thousands of human genomes are to be sequenced. Illumina Company offers whole genome sequencing for clinical purposes for a few thousand US dollars. The companies like Ion Torrent promise the whole human genome sequencing in hours for <1000 dollars to be available soon. It seems that personalized medicine for the masses will be available in the near future.

The low cost of pure sequencing is not, however, everything, as the data must be stored and transferred. The IT costs were not treated seriously in the past, when the sequencing was expensive and slow. Nowadays, the costs of storage and transfers counted in hundreds of dollars for a single genome per year are no longer negligible. Moreover, the improvements in this field are far behind what is present in the sequencing (Deorowicz and Grabowski, 2013).

An obvious solution to the data deluge is data compression and many specialized compressors appeared in the recent years (see the survey, Deorowicz and Grabowski, 2013). These tools are, however, rather 'experimental' and tend to suffer from one or more of the following drawbacks: (i) they focus mainly on the compression ratio, and as a consequence (de)compression is slow (sometimes even comparable with the speed of sequencing), (ii) they are available as external tools, so the compressed formats cannot be directly used by other software, (iii) they have no support for some types of FASTQ files, e.g. in color space or variable-length reads and (iv) they are unstable and crash frequently.

The focus on the compression ratio can sometimes be justified, especially, when the goal is just storage for archival purposes. In many situations the data are, however, stored locally and decompressed many times (for various analyses), so the decompression speed could be a significant factor of the processing speed of a whole pipeline. Thus, in practice the well-known but rather inefficient gzip program is still in wide use.

We think that it is time for industry-oriented solutions. Thus, we introduce DSRC 2, supporting any variant of FASTQ files. Its compression ratios are much better than gzip/bzip2 and only moderately worse than the best existing programs, but the speed of (de)compression is high. DSRC 2 also supports Illumina's plan of quality resolution reduction (http://res.illumina.com/documents/products/whitepapers/whitepaper_datacompression.pdf, 2012) and can read/write to pipes, for easy integration with pipelines.

2 METHODS AND IMPLEMENTATION

DSRC 2 is a multithreaded tool written in C using Boost libraries. In the compression mode, a single thread reads the input FASTQ file in blocks (typically of tens of MBs size) and puts them into an input queue. Several threads perform the compression of the blocks, storing the results in an output queue. Finally, a single thread writes the compressed blocks in a single file. The decompression is organized in the same fashion.

At the high level, the processing of a single block is similar to those of the existing compressors. The reads are split into three streams: IDs, sequences in base space (color space data are converted to base space) and quality values. The non-ACGT symbols from the sequence stream are transferred to the quality stream, just as in DSRC 1 (Deorowicz and Grabowski, 2011).

The processing of IDs was slightly improved compared with DSRC 1. Additionally, some parts of IDs can be optionally removed, which could be helpful, as often most of the IDs data are irrelevant for further analysis.

DNA symbols can be encoded in three ways. In the first one, each base is stored in 2 bits (just as in DSRC 1). In the second one, the Huffman coder on these symbols is used. The last method uses arithmetic coder (Salomon and Motta, 2010), with contextual probability estimation of orders up to 9.

Quality values can be stored in two ways. In the first one (similar as in DSRC 1), the compressor uses one of the following modes (after estimation of which one should give better ratio): the order-1 Huffman coder

*To whom correspondence should be addressed.

Table 1. Compression results

Compressor	Illumina dataset						ABI Solid dataset						LS454/IonTorrent dataset					
	Fast comp. mode			Best comp. mode			Fast comp. mode			Best comp. mode			Fast comp. mode			Best comp. mode		
	Ratio	c-sp	d-sp	ratio	c-sp	d-sp	Ratio	c-sp	d-sp	ratio	c-sp	d-sp	Ratio	c-sp	d-sp	ratio	c-sp	d-sp
pigz	3.04	48	127	3.11	27	127	3.30	74	128	3.39	54	114	2.85	43	123	2.91	19	120
pbzip2	3.81	50	141	3.81	48	**142**	4.10	39	156	4.10	39	**148**	3.52	58	141	3.52	**58**	**135**
Quip	**5.09**	20	16	4.12	11	12	—	—	—	—	—	—	4.28	18	13	4.29	4	4
FQZcomp	4.87	49	36	**5.52**	9	9	**5.40**	50	38	**5.63**	24	22	**4.32**	42	32	**4.87**	7	6
DSRC 1	4.19	31	40	4.60	14	38	4.87	26	39	4.94	21	40	3.93	40	40	4.09	18	40
SeqDB	2.39	205	169	—	—	—	—	—	—	—	—	—	1.90	137	150	—	—	—
DSRC 2	4.25	**279**	**379**	4.95	**60**	55	5.02	**246**	**359**	5.27	71	71	3.94	**342**	**433**	4.35	52	45

Note: Ratio is expressed as the original file size divided by the compressed file size. Columns 'c-sp' and'd-sp' denote compression and decompression speeds (in MB/s), respectively. The programs were run for the maximal number of threads they allow, but not >8. Empty cells mean lack of support such type of files. The best results are in bold. comp. denotes compression.

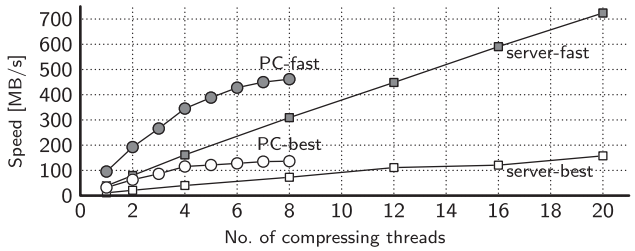

Fig. 1. Scalability of DSRC 2 for SRR065390_1 file in the lossless mode

with context being the position in the read, or the order-1 Huffman coder of the run-length-encoded quality stream. In the second method, the quality values are compressed arithmetically with context lengths up to 6.

3 RESULTS

To evaluate the proposed compressor, we collected datasets for three different technologies: Illumina (fixed-length reads, base space), ABI SOLiD (fixed-length reads, color space), LS454/IonTorrent (variable-length reads, base space). Majority of experiments were performed on a four 8-core AMD Opteron ™ 6136 2.4 GHz CPUs server with RAID-5 disk matrix containing 6 HDDs. In one test, we also used the PC machine containing 4-core (with hyperthreading) i7 4770 3.4 GHz CPU and SSD disk.

For the comparison, we used two popular universal tools, pigz (parallel gzip) and p-bzip2 (parallel bzip2), and the best existing FASTQ compressors, Quip (Jones *et al*., 2012), FQZcomp (Bonfield and Mahoney, 2013), SeqDB (Howison, 2013) and DSRC 1 (Deorowicz and Grabowski, 2011). The compressors were run in two modes: the best ratio and fast compression, not necessarily the fastest possible, but with a 'reasonable' ratio (Table 1). DSRC 2 consumes <400 MB of main memory in the fast mode and <6.5 GB in best mode. These values are much higher than of gzip (<10 MB in parallel variant), but we think they are still acceptable.

In the lossless mode, the best ratios were obtained by FQZcomp, but its low speed makes it rather impractical. The compression ratio of DSRC 2 is ~10–15% smaller, but its speed is an order of magnitude (or more) higher. In the fast mode, the speed of DSRC 2 is sometimes I/O-limited, while the compression ratio is still much better than of the de facto standards gzip/bzip2. What is important is that DSRC 2 is similarly fast in both compression and decompression. This allows the application in storage of the intermediary results in the processing pipelines, potentially with total improvement of the complete processing speed because of I/O transfer reduction.

The relative results for the Illumina's quality-reduced data are presented in Table 1 of the Supplementary File S1. Figure 1 shows how DSRC 2 speed scales up for growing number of threads.

4 CONCLUSION

We propose a specialized FASTQ compressor, DSRC 2, for industry-oriented and research purposes. Its compression ratios are much better than that of the widely used gzip program and only moderately worse than the best existing (in terms of ratio) FASTQ compressors. DSRC 2 is, however, usually a few times faster than all the competitors. We believe our software, containing command-line (de)compressor and libraries for popular programming languages, could replace gzip in real pipelines and repositories.

ACKNOWLEDGEMENT

The authors thank Szymon Grabowski for his helpful comments after reading the preliminary versions of the article.

Funding: This work was partially supported by the European Union from the European Social Fund within the INTERKADRA project UDAPOKL-04.01.01-00-014/10-00 and by the Polish National Science Centre under the project DEC-2012/05/B/ST6/03148. The work was performed using the infrastructure supported by POIG.02.03.01-24-099/13 grant: 'GeCONiI—Upper Silesian Center for Computational Science and Engineering'.

Conflict of interest: none declared.

REFERENCES

Ball,M.P. *et al.* (2012) A public resource facilitating clinical use of genomes. *Proc. Natl Acad. Sci. USA*, **109**, 11920–11927.

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

Deorowicz,S. and Grabowski,S. (2011) Compression of DNA sequence reads in FASTQ format. *Bioinformatics*, **27**, 860–862.

Deorowicz,S. and Grabowski,S. (2013) Data compression for sequencing data. *Algorithms Mol. Biol.*, **8**, 25.

Howison,M. (2013) High-throughput compression of FASTQ data with SeqDB. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **10**, 213–218.

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

Roberts,J. (2013) Million veterans sequenced. *Nat. Biotechnol.*, **31**, 470.

Salomon,D. and Motta,G. (2010) *Handbook of Data Compression*. Springer, London.