Bioinformatics, 35(15), 2019, 2674–2676
doi: 10.1093/bioinformatics/bty1015
Advance Access Publication Date: 7 December 2018
Applications Note

OXFORD

Sequence analysis

SPRING: a next-generation compressor for FASTQ data

Shubham Chandak [1,*], Kedar Tatwawadi[1], Idoia Ochoa[2], Mikel Hernaez [3,*] and Tsachy Weissman[1]

[1]Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA, [2]Department of Electrical and Computer Engineering and [3]Carl R. Woese Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

*To whom correspondence should be addressed.

Associate Editor: Bonnie Berger

Received and revised on October 12, 2018; editorial decision on December 3, 2018; accepted on December 6, 2018

Abstract

Motivation: High-Throughput Sequencing technologies produce huge amounts of data in the form of short genomic reads, associated quality values and read identifiers. Because of the significant structure present in these FASTQ datasets, general-purpose compressors are unable to completely exploit much of the inherent redundancy. Although there has been a lot of work on designing FASTQ compressors, most of them lack in support of one or more crucial properties, such as support for variable length reads, scalability to high coverage datasets, pairing-preserving compression and lossless compression.

Results: In this work, we propose SPRING, a reference-free compressor for FASTQ files. SPRING supports a wide variety of compression modes and features, including lossless compression, pairing-preserving compression, lossy compression of quality values, long read compression and random access. SPRING achieves substantially better compression than existing tools, for example, SPRING compresses 195 GB of 25× whole genome human FASTQ from Illumina's NovaSeq sequencer to less than 7 GB, around 1.6× smaller than previous state-of-the-art FASTQ compressors. SPRING achieves this improvement while using comparable computational resources.

Availability and implementation: SPRING can be downloaded from https://github.com/shubhamchandak94/SPRING.

Contact: schandak@stanford.edu or mhernaez@illinois.edu

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

1 Introduction

There has been a tremendous increase in the amount of genomic data produced in the past few years, mainly driven by the improvements in High-Throughput Sequencing technologies and the reduced cost of sequencing a genome. A single genome sequencing experiment on humans typically results in hundreds of millions of short reads (of length 100–150 bp), which are (possibly corrupted) substrings of the same underlying genome sequence. These raw sequencing data is typically stored in the FASTQ format, which consists of the reads along with the quality values which indicate the confidence in the read sequence and read identifiers which consist of metadata related to the sequencing process. In most cases, the reads are sequenced in pairs from short fragments of the genome, resulting in paired-end FASTQ files. A typical FASTQ dataset for a human genome sequencing experiment requires hundreds of GBs of storage space (for a typical sequencing coverage of 30×). Due to the huge sizes involved, compression of the FASTQ files is of utmost importance for their storage and distribution.

There is significant amount of recent work on FASTQ compression (Numanagić *et al.*, 2016), including SCALCE (Hach *et al.*, 2012),

Table 1. Compressed sizes (in MB) for selected datasets

Organism	Technology	Coverage	Uncompressed	Lossless mode				Recommended lossy mode		
			Size	pigz	FaStore	SPRING	Improvement	FaStore	SPRING	Improvement
Pseudomonas aeruginosa	GAIIx	50	768	279	145	**115**	1.26×	88	**62**	1.41×
Metagenomic	HiSeq 2000	—	19 284	6911	3602	**3206**	1.12×	1935	**1736**	1.11×
H.sapiens	HiSeq 2000	28	227 246	74 250	35 662	**28 901**	1.23×	17 417	**13 460**	1.29×
H.sapiens	NovaSeq	25	195 748	36 131	11 101	**6971**	1.59×	9927	**5657**	1.75×
H.sapiens	NovaSeq	100	787 616	144 927	33 734	**25 883**	1.30×	28 846	**20 316**	1.42×

Notes: Improvement is reported with respect to FaStore. Best results for each mode are bold-faced.

Fqzcomp (Bonfield and Mahoney, 2013), DSRC 2 (Roguski and Deorowicz, 2014) and FaStore (Roguski *et al.*, 2018). Since the reads are sub-strings of the underlying genome, there is much redundancy to be exploited for compression. Specialized compressors, which explicitly utilize the structure present in the reads, can achieve a compression gain of more than 10× as compared to generic universal compressors such as Gzip (Numanagić *et al.*, 2016). The quality values, on the other hand, have less structure and thus can take up a more significant fraction of the storage space in the compressed domain. Recent work (Ochoa *et al.*, 2017; Roguski *et al.*, 2018) has shown that the quality values can be lossily compressed without adversely affecting the performance of variant calling, one of the most widely used downstream application in practice. Moreover, newer technologies such as Illumina's Novaseq are using quality values with fewer levels (4 levels instead of the previous 8 or 40 levels), hence supporting the claim that the precision in the quality values can be reduced with no impact on variant calling performance.

Although there has been a lot of work on designing FASTQ compressors, most of them lack in support of one or more crucial properties, such as support for variable length reads (Roguski *et al.*, 2018), scalability to high coverage datasets, pairing-preserving compression (Roguski and Deorowicz, 2014) and lossless compression (Hach *et al.*, 2012). Partly due to these factors, Gzip is still the prevalent FASTQ compressor, even though it provides worse compression ratios (Numanagić *et al.*, 2016).

In this work, we present the next-generation compressor SPRING, which supports all the crucial properties, while achieving significantly better compression as compared with state-of-the-art FASTQ compressors. SPRING is also eminently practical in terms of its memory/time requirements, and supports selective access to the compressed data.

2 Methods and results

SPRING supports the following recommended modes of FASTQ compression:

i. **Lossless mode (default):** In this mode, the FASTQ file is compressed so that it can be exactly reconstructed, i.e. the reads, quality, read identifiers and the read order information can be perfectly recovered.
ii. **Recommended lossy mode:** In this mode, the information relevant for most of the genomic applications (such as alignment, assembly, variant calling, etc.) is preserved. This includes the reads along with pairing information and binned quality values. The quality values are subjected to the Illumina's standardized 8-level binning (https://www.illumina.com/documents/products/whitepapers/whitepaper_datacompression.pdf) before compression (Novaseq qualities are left unchanged). The read identifiers and the order of the pairs is discarded (i.e. the decompressed FASTQ file contains the read pairs in an arbitrary order). The relative ordering of the first and the second read in each pair is still preserved.

Although we advocate for these default modes, SPRING can be highly customized based on the user needs, and provides additional capabilities such as custom binning of quality values using QVZ (Malysa *et al.*, 2015) and binary thresholding.

For short reads (up to 511 bp), the read compression in SPRING is based on HARC (Chandak *et al.*, 2018), with significant improvements and added support for variable-length reads. SPRING also supports long read compression, where BSC (https://github.com/IlyaGrebnov/libbsc/) is used as the read compressor. Furthermore, SPRING compresses the streams in blocks, allowing for fast decompression of a subset of reads (random access). More details and results for these features are provided in the Supplementary Material.

Table 1 shows the compression results for the two recommended modes for selected datasets. We compare SPRING to FaStore (Roguski *et al.*, 2018), the best performing FASTQ compressor and pigz (parallelized Gzip), the most commonly used FASTQ compressor in practice. We observe that SPRING achieves significant compression gains with respect to FaStore in both modes, especially for human NovaSeq datasets, while being comparable in computational resources (Supplementary Material). For example, running with eight threads in the lossless mode, SPRING requires 2 h and 31 GB RAM for compressing the 25× NovaSeq *Homo sapiens* dataset, which is competitive with FaStore (2.5 h and 41 GB). For decompressing the dataset, SPRING requires 26 min and 6.1 GB RAM. This is slower than FaStore (12 min), but with significantly lower memory consumption (23 GB for FaStore). In comparison to pigz, SPRING achieves 2×–5× better compression ratios but requires higher computational resources (see Supplementary Material for further details and more extensive results).

In conclusion, this work presents the FASTQ compressor SPRING, which outperforms existing tools, offering 1.3×–1.8× improvement in compression over the next best performing tool on data sequenced on Illumina's latest sequencer, NovaSeq. SPRING supports a wide variety of modes and features and is competitive in terms of computational requirements. Furthermore, the streams generated by SPRING can be easily transformed to streams compatible with the upcoming standard developed by the MPEG-G group for genomic information representation (Alberti *et al.*, 2018). Future work includes integration of SPRING into the standard and developing specialized read compressors for long read technologies.

Funding

This work was partially supported by NIH Grant 5U01CA198943-03, grant numbers 2018-182798 and 2018-182799 from the Chan Zuckerberg Initiative DAF, an advised fund SVCF and an SRI grant from UIUC.

Conflict of Interest: none declared.

References

Alberti,C. *et al*. (2018) An introduction to MPEG-G, the new ISO standard for genomic information representation. https://www.biorxiv.org/content/early/2018/10/08/426353.

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

Chandak,S. *et al*. (2018) Compression of genomic sequencing reads via hash-based reordering: algorithm and analysis. *Bioinformatics*, **34**, 558–567.

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

Malysa,G. *et al*. (2015) QVZ: lossy compression of quality values. *Bioinformatics*, **31**, 3122–3129.

Numanagić,I. *et al*. (2016) Comparison of high-throughput sequencing data compression tools. *Nat. Methods*, **13**, 1005.

Ochoa,I. *et al*. (2017) Effect of lossy compression of quality scores on variant calling. *Brief. Bioinform.*, **18**, 183–194.

Roguski,L. and Deorowicz,S. (2014) DSRC 2-industry-oriented compression of FASTQ files. *Bioinformatics*, **30**, 2213–2215.

Roguski,L. *et al*. (2018) Fastore: a space-saving solution for raw sequencing data. *Bioinformatics*, **34**, 2748–2756.