Comparison of high-throughput sequencing data compression tools

Ibrahim Numanagić[1,7], James K Bonfield[2,7], Faraz Hach[1,3], Jan Voges[4], Jörn Ostermann[4], Claudio Alberti[5], Marco Mattavelli[5] & S Cenk Sahinalp[1,3,6]

High-throughput sequencing (HTS) data are commonly stored as raw sequencing reads in FASTQ format or as reads mapped to a reference, in SAM format, both with large memory footprints. Worldwide growth of HTS data has prompted the development of compression methods that aim to significantly reduce HTS data size. Here we report on a benchmarking study of available compression methods on a comprehensive set of HTS data using an automated framework.

Current trends in HTS data generation indicate that storage, transmission and bandwidth costs will soon surpass the costs of sequencing and become the main bottleneck in genomics as well as in the application of HTS data to precision medicine. One way to reduce the burden of HTS data on storage, bandwidth and transmission is the use of high-performance compression methods developed specifically for HTS data. In the last 25 years the Moving Picture Experts Group (MPEG), also known as the ISO/IEC JTC1/SC29/WG11 committee, has developed a methodology that has yielded compression standards extensively adopted by the digital media industry. A growing number of experts in genome data processing have joined MPEG experts in a working group (which we will refer to as the 'MPEG HTS compression working group') to explore how data compression expertise from the multimedia world can help improve the performance of existing genomic data compression tools. The ISO Technical Committee 276 (Biotechnology) (ISO TC 276) Working Group 5 (Data Processing and Integration) has also joined the effort with its specific biotechnology expertise. The ultimate goal of the activity of the MPEG HTS compression working group is to design and specify genomic data compression and transport technology by means of an open standard and interoperability among systems.

As a first step toward developing an open standard, MPEG and ISO TC 276 have issued a call to the international community to evaluate the effectiveness of available compression methods on a common set of genome data. For this purpose, the MPEG HTS compression working group compiled an HTS data set with a wide spectrum of characteristics for ensuring statistically meaningful results: raw (FASTQ) and aligned (SAM, or BAM) data with both deep and shallow coverage; fixed-length and variable-length reads obtained by sequencing technologies from leading manufacturers (Illumina, Pacific Biosciences, Oxford Nanopore, Ion Torrent); genome, exome and transcriptome data from various organisms (*Homo sapiens*, bacteria, plants, insects, etc.); and several sample types (metagenomic, cancer cell lines, etc.) as well as simulated human data are included in the final data set of 4 terabytes. The data set was reviewed and approved by all members of the MPEG HTS compression working group for benchmarking purposes. It is expected that the data set will grow further to accommodate future technologies and additional requirements.

As members of the MPEG HTS compression working group, we have conducted a comparative study of all available lossless HTS compression tools on the MPEG benchmarking data set, significantly expanding some recent comparative studies and surveys[1–3]. We developed an open-source, publicly available framework specifically tailored for HTS compression evaluation, placing a special emphasis on fairness and reproducibility of the benchmarking process (https://github.com/sfu-compbio/compression-benchmark). Together with the data diversity provided by the MPEG benchmarking data set, this framework is also suitable for the review and comparison of future tools.

Data set selection criteria, complete framework description, detailed usage information and comprehensive benchmarking results are available in **Supplementary Notes 1–6** and **Supplementary Figure 1**.

We aimed to evaluate all available approaches used for HTS data compression. These approaches include both industry-scale tools as well as research-oriented prototypes. Compression performance, running times, memory usage and parallelization capabilities were measured for each tool.

Most HTS data are maintained either as raw sequencing information in a FASTQ file or as reference-aligned (mapped) data in SAM or BAM formats. The FASTQ and SAM schemata describe different data fields with similar properties (e.g., the sequence field consists only of DNA nucleotides, while mapping loci are usually represented as a nondecreasing sequence of integers), generating large files. It is common to use general-purpose compression tools on these files, which treat them as simple plain text files and thus produce suboptimal compression rates because they are not able to exploit the underlying data schemata.

All HTS data compression tools are built on a standard set of general compression algorithms, which are surveyed in **Supplementary Note 2**.

FASTQ files are typically compressed with general-purpose Gzip and bzip2 tools. Sequence archives commonly use NCBI's

[1]School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada. [2]Wellcome Trust Sanger Institute, Hinxton, UK. [3]Vancouver Prostate Centre, Vancouver, British Columbia, Canada. [4]Institut für Informationsverarbeitung, Leibniz Universität, Hannover, Germany. [5]École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. [6]School of Informatics and Computing, Indiana University, Bloomington, Indiana, USA. [7]These authors contributed equally to this work. Correspondence should be addressed to S.C.S. (cenksahi@indiana.edu).

RECEIVED 12 MAY; ACCEPTED 1 SEPTEMBER; PUBLISHED ONLINE 24 OCTOBER 2016; DOI:10.1038/NMETH.4037

SRA format, which is loosely based upon the LZ-77 scheme. Specialized FASTQ compressors initially perform a form of transformation (read-identifier tokenization or 2-bit nucleotide encoding) followed by statistical modeling and entropy coding. Examples of such approaches are DSRC2 (ref. 4), FQC[5], Fqzcomp and Fastqz[6], Slimfastq, and LFQC[7].

Because the read order within a FASTQ file is arbitrary, reordering the reads in a manner that brings similar reads together can significantly boost compression rates[8]. This is especially true if the underlying genome is repetitive, or if the coverage of the data is high; in such cases, schemes like LZ-77 can benefit significantly from the improvement of data locality. Tools like SCALCE[8], Orcom[9], Mince[10] and BEETL[11] use this approach as a preprocessing step in order to improve compression performance.

Alternative approaches aim to achieve compression by replacing each read with a pointer to the underlying reference genome, provided such a reference genome is available. LW-FQZip[12] is one such example that relies on sequence mapping to obtain a list of pointers. If the reference genome is not available, it can be constructed *de novo* by assembling the reads into contigs, usually through the use of de Bruijn graphs. Subsequently, a read can be represented as a pointer to an assembled contig or as a path within a de Bruijn graph. Primary tools that use assembly for data compression are Quip[13], Leon[14], k-Path[15] and KIC[16]. Note that both sequence mapping and assembly are computationally intensive tasks; as a result, most of the tools mentioned above sacrifice speed for maintaining high compression rates.

SAM files are mostly stored in their compressed equivalent, the BAM format. Commonly used tools for BAM manipulation are Samtools[17], Picard, and Sambamba[18]. All BAM-based tools support arbitrary ordering of the reads and do not require a reference during compression or decompression. None of them treat various streams in a BAM file differently.

An alternative to the SAM format is CRAM, a reference-based format that separates different fields in the reads and applies a variety of compression techniques on each. CRAM is implemented in Cramtools[19] and Scramble[20]; and has recently been incorporated in Samtools[17] and Picard.

In both SAM and CRAM, reads covering the same sequence variant are encoded independently. As such, the same variant is redundantly encoded across the reads. In order to eliminate this redundancy, DeeZ, a newer alternative[21] to SAM and CRAM, implicitly assembles the underlying donor genome in order to encode these variants only once. CBC[22] and TSC[23] follow a similar path, only encoding variants one time. All of these tools treat each SAM field independently, and they apply a variety of

Table 1 | A summary of evaluated tools and their performance in a single-thread mode.

(**a**) FASTQ tools

Sample	**SRR554369**		**SRR327342**		**MH0001.081026**		**SRR1284073**		**SRR870667**		**ERR174310**		**ERP001775**	
Organism	Pseudomonas *aeruginosa*		Saccharomyces *cerevisiae*		*Homo sapiens gut*		*Escherichia coli*		*Theobroma cacao*		*H.sapiens*		*H.sapiens*	
Technology	Illumina GAIIx		Illumina GAII		Illumina GA		PacBio		Illumina GAIIx		HiSeq		HiSeq	
Coverage	25×		80×		Unknown		140×		20×		7×		120×	
	File size	Time	File size	Time	File size	Time	File size	Time	File size	Time	File size	Time	File size	Time
Uncompressed —full file	550		3,881		1,880		1,309		22,944		53,869		2,717,029	
Uncompressed —reads only	165		947		512		649		7,463		20,966		1,059,387	
pigz	158	1	1,020	1	501	1	547	1	6,943	1	18,597	1	305,690	1
	48	1	277	1	149	1	188	1	2,108	1	5,982	1	104,927	1
pbzip2	125	1.19	831	1.45	390	1.29	463	0.74	5,577	0.99	14,887	0.81	242,834	0.21
	44	5.97	251	6.85	139	6.35	176	6.99	1,879	3.61	5,473	2.83	95,969	1.23
DSRC2	105	0.22	668	0.26	312	0.24	N/A		4,761	0.21	13,214	0.2	N/A	
	41	2.11	257	3.09	128	1.91			1,865	1.39	5,239	1.22		
DSRC2 (extra)	95	0.71	595	0.7	287	0.8	N/A		4,246	0.7	11,598	0.67	N/A	
	39	11.81	230	11.33	125	12.96			1,636	6.25	4,773	5.78		
Fqzcomp	89	0.34	559	0.37	280	0.41	N/A		4,028	0.33	11,320	0.32	N/A	
	37	N/A	203	7.54	120	N/A			1,556	N/A	4,623	3.29		
Fqzcomp (extra)	94	0.39	589	0.39	286	0.41	N/A		4,228	0.35	11,673	0.34	N/A	
	41	N/A	234	7.5	128	N/A			1,796	N/A	5,167	3.47		
Fastqz	N/A		N/A		N/A		N/A		N/A		10,955	3.45	N/A	
											4,312	N/A		
Slimfastq	94	0.55	507	0.47	266	0.54	N/A		4,280	0.51	11,045	0.47	178,092	0.49
	30	11.46	149	9.55	104	11.32			1,416	5.8	4,426	4.76	77,629	5.94
FQC	76	1.05	494	1.18	268	1.39	413	0.98	3,912	1.16	11,409	1.22	N/A	
	N/A		N/A		N/A	17.07	N/A	12.12	N/A	5.93	N/A	5.74		
LFQC	69	18.63	490	18.54	266	21.06	407	18.03	2,412	14.46	N/A		N/A	
	17	315.41	129	310.81	103	339.93	156	386.25	N/A	N/A				
SCALCE	76	0.77	487	0.63	297	0.8	421	0.67	3,699	0.6	10,827	0.59	161,067	0.57
	17	9.05	68	8.23	71	12.17	161	9.78	998	4.89	3,017	4.57	28,452	1.94
LW-FQZip	117	1.13	790	0.6	N/A		N/A		5,038	2.27	N/A		N/A	
	45	5.62	320	5.16					1,735	2.5				
Quip	89	0.5	537	0.53	272	0.47	420	0.36	3,914	0.48	11,312	0.46	184,051	0.38
	37	10.7	181	11.53	114	11.37	159	10.59	1,462	5.57	4,556	5.22	79,771	4.64
Leon	87	3.43	544	2.92	291	3.91	479	2.81	4,518	4.15	13,623	3.43	220,397	1.13
	19	16.84	89	16.7	87	14.84	N/A	34.31	1,360	10.91	4,739	9.67	83,539	4.66
KIC	95	5.81	613	7.29	307	4.73	451	9.4	4,498	6.5	13,006	6.25	N/A	
	32	6.65	188	7.72	122	6.35	N/A	9.37	1,594	3.44	4,915	3.33		
Orcom		0.5		0.46		0.87	N/A			0.83		0.66		0.12
	11	1.51	36	0.91	51	1.87			825	1.22	1,798	0.83	6,921	0.23
BEETL		4.12		2.82		2.46	N/A			4.44		4.38	N/A	
	23	36.81	117	30.24	114	31.02			1,200	22.11	3,912	20.95		
Mince		4.52		4.26		5.19	N/A			2.42		2.57	N/A	
	10	2.38	37	2.24	50	3.25			685	0.86	1,955	0.9		
k-Path		2.03		1.73		13.04	N/A			2.29		3.22	N/A	
	14	30.08	45	20.19	62	149.57			660	15.39	2,088	16.57		

(continued)

Table 1 | A summary of evaluated tools and their performance in a single-thread mode. (continued)

(**b**) SAM tools

Sample	DH10B		9827.2.49		sample-2-1		K562.LID8465		dm3		NA12878.PB		HCC1954		NA12878.S1	
Organism		*E. coli*		*H.sapiens*		*H.sapiens*		*H.sapiens*		*Drosophila melanogaster*		*H.sapiens*		*H.sapiens*		*H.sapiens*
Technology		MiSeq		HiSeq		IonTorrent		RNASeq		PacBio		PacBio		cancer cell		HiSeq
Coverage		420×		2×		0.6×		6×		75×		15×		30×		50×
Uncompressed	5,579		21,059		5,924		75,915		30,081		126,545		427,028		589,083	
	File size	Time	File size	Time	File size	Time	File size	Time	File size	Time	File size	Time	File size	Time	File size	Time
pigz	1,336	0.77	6,021	1.55	1,378	1.48	12,785	1.06	12,315	1.39	52,914	1.37	119,839	1.4	113,462	0.13
		0.63		0.82		0.49		0.7		0.79		0.7		0.91		0.6
pbzip2	1,074	1.65	5,243	1.93	1,127	4.04	10,251	3.57	9,717	0.72	43,128	0.94	100,280	1.62	89,598	0.46
		3.16		3.39		3.72		2.46		2.93		3.94		3.23		0.59
Samtools	1,407	1	6,499	1	1,469	1	13,757	1	12,853	1	57,090	1	131,566	1	121,710	1
		1		1		1		1		1		1		1		1
Picard	1,425	1.42	6,517	1.04	1,474	1.82	13,818	1.48	12,837	0.74	57,316	0.55	132,861	1.18	N/A	
		2.76		1.52		2.1		2.44		1.09		1		1.91		
Sambamba	1,407	1.05	6,499	0.93	1,469	1.12	13,757	1.05	12,859	2.48	57,090	0.93	131,566	1.39	121,710	0.13
		1.08		1.13		0.97		0.97		1.92		1.12		1.12		0.53
Cramtools (CRAM v2)	1,066	0.93	3,778	1.42	1,170	2.12	10,344	1.7	7,577	0.93	38,266	1.01	95,442	1.28	N/A	
		1.71		1.67		4.93		2		2.05		2.39		1.5		
Scramble (CRAM v3)	863	0.23	3,297	0.29	1,030	0.62	9,261	0.38	6,551	0.14	34,425	0.31	82,041	0.27	66,632	0.1
		0.76		0.66		1.58		0.67		0.58		0.84		0.71		0.5
Scramble (CRAM v3 w/o reference)	899	0.29	4,236	1.18	1,113	0.45	9,839	0.43	10,562	0.21	44,843	0.46	86,914	0.37	72,407	0.1
		0.74		0.63		1.06		0.78		1.14		1.54		0.79		0.47
Scramble (CRAM v3 with bzip2)	851	0.76	3,262	0.62	998	1.5	8,611	1.27	6,469	0.17	33,921	0.48	80,094	0.6	N/A	
		0.89		0.66		1.72		0.81		0.67		1.63		0.82		
DeeZ	823	0.56	3,221	0.78	1,028	1.81	8,120	0.92	6,681	0.51	34,639	0.64	78,473	0.91	62,966	0.26
		3.9		2.46		5.51		3.35		1.77		1.86		2.94		1
DeeZ (with bzip2 and sam_comp qualities)	730	0.91	2,734	1.23	918	3.49	7,266	2.01	6,585	0.71	34,172	1.22	74,509	1.66	53,497	0.41
		10.11		5.6		9.86		7.91		4.86		6.67		6.39		1.9
TSC	1,105	2.21	7,939	0.8	1,193	2.55	20,864	3.17	8,397	3.14	45,452	1.46	164,627	0.5	N/A	
		9.05		2.24		6.75		6.27		6.46		6.43		2.65		
Quip	1,103	0.67	4,419	0.94	1,230	1.15	11,186	1.19	9,024	0.44	42,642	0.67	98,303	0.83	97,165	0.44
		10.69		7.81		4.43		8.27		7.52		9.87		9.05		2.18
Quip (with reference)	803	0.67	N/A		N/A		8,743	1.17	6,461	0.41	N/A		N/A		64,493	0.43
		10.06						8.2		7.19						2.2
sam_comp*	700	0.68	2,649	0.76	891	1.2	7,023	0.71	8,356	0.51	32,670	0.59	42,522	0.62	53,263	0.37
		3.36		2.95		6.54		3.56		5.49		5.42		3.25		2

(a) For each FASTQ tool and sample, the left two columns indicate the overall compressed size and the sequence-only compressed size, respectively. Right column indicates the compression and decompression times relative to Gzip (pigz), with lower values indicating better running time. The last three rows show the performance of sequence-only tools. In the ERP001775 sample, all tools were run with four threads, with the exception of pigz and Slimfastq. KIC and k-Path also required running with four threads. (b) For each SAM tool and sample, the left column indicates the overall compressed data size. Right column indicates the compression and decompression times relative to Samtools, with lower being better. Last row denotes sam_comp, which does not support all SAM fields. In the NA12878.S1 sample, all tools were run with four threads, with the exception of Samtools and sam_comp. Dark green color indicates the best tool in the given category (compression rate or time), while light green indicates second-best tool. Analogously, orange denotes second-worst tool, while magenta indicates the worst performance. Asterisk indicates that the tool does not support all SAM fields. Missing data points (caused by either crashes or compatibility issues) are marked with N/A, and further explained in **Supplementary Note 5**.

compression techniques to each field. Finally, Quip[13] and sam_comp[6] employ highly optimized statistical models for various SAM fields, which puts them among the best performing tools in terms of pure compression rate. The full description of the evaluated tools and the infrastructure used for the evaluation is available in **Supplementary Notes 3** and **5**.

An overview of our results is presented in **Table 1**, while more detailed results are available in **Supplementary Tables 1–7**.

Some of the available tools achieve significant gains over Gzip and bzip2, both in terms of compression rate and speed. The best compression rates are offered by tools that reorder reads; these tools are especially effective for sequence compression. Alternative tools such as LFQC may also provide good compression rates, but they come with a high running time overhead. It should be noted that reordering-based tools also perform very well in terms of speed. Their memory usage is slightly higher but not unreasonable, and it can be user configured in most cases. Many tools significantly improve their performance through parallelization, even though the performance improvement is not always proportional to the number of processors. The best trade-off between the number of processors and running time is typically achieved with four threads.

The majority of the available tools are optimized for Illumina-style short, fixed-length reads—many tools do not provide an option to compress long, variable-length read collections, such as data obtained with sequencers from Pacific Biosciences.

It is possible to obtain better compression rates than those achieved by Samtools, even with the simple use of Gzip. However, unlike Samtools, Gzip does not provide random-access capability. Among the available tools, only BAM and CRAM-family tools, DeeZ and TSC provide a random-access facility. Interestingly, Scramble and DeeZ also improve upon the performance of sam_comp and Quip in most cases, both in terms of compression rate and speed, while providing random-access capability. In most cases, Scramble can also decompress files faster than Samtools can.

Our evaluation of all compression tools currently available in the literature on a wide variety of data sets resulted in no overall winner that can perform well on each data type and under every performance measure we used. We conclude that an integrated solution that chooses the specific approach which performs the best on the input data type(s) with respect to the performance measure most important for the specific application would yield the best outcome, both for raw and aligned sequence data. Many of the tools we benchmarked improve not only the compression rate but also the compression time of the most commonly used methods; i.e., Gzip or pigz for FASTQ files and Samtools for SAM, or BAM, files. Although this is not always the case for decompression time, the time necessary for decompressing pigz-compressed FASTQ files and Samtools-compressed SAM files is insignificant; in fact ≤1/100 of the time necessary for running the most commonly used downstream analysis pipelines (e.g., for read mapping in FASTQ files and for variant calling in SAM files). Future integration of some of the best performing compression tools we tested with commonly used variant calling pipelines such as GATK (which produces multiple BAM files during execution) may significantly improve the overall running time of GATK on account of smaller data footprints and thus improved locality of reference.

MPEG's decision to issue a call for proposals soliciting the submission of technology for genomic data processing and storage was aimed at developing a standard compressed file format in the coming years. Such a standard will likely integrate the best features of the tools and formats evaluated in this study. The potential impact of an international standard for genomic data compression would be groundbreaking in terms of both systems interoperability and efficiency, enabling population-wide scaling of existing genomic applications.

METHODS

Methods, including statements of data availability and any associated accession codes and references, are available in the online version of the paper.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS
This research was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Frontiers program 'Cancer Genome Collaboratory' project (S.C.S., F.H., I.N.); the Vanier Canada Graduate Scholarships program (I.N.); National Institutes of Health (NIH) (R01GM108348 to S.C.S.); National Science Foundation (NSF) (1619081 to S.C.S.); Indiana University Grant Challenges Program Precision Health Initiative (S.C.S.); Wellcome Trust (098051 to J.K.B.); Leibniz Universität Hannover eNIFE grant (J.V. and J.O.); Swiss Platform for Advanced Scientific Computing (PASC) PoSeNoGap project (C.A. and M.M.). We would also like to thank the authors of evaluated compression tools for providing support for their tools and replying to our bug reports.

AUTHOR CONTRIBUTIONS
The study was initiated by I.N., C.A. and M.M. I.N. designed the benchmarking framework and performed the experiments. J.K.B. evaluated the framework. I.N., J.K.B., J.V., J.O., F.H., C.A., M.M. and S.C.S. contributed to writing the manuscript. S.C.S. and F.H. oversaw the project.

COMPETING FINANCIAL INTERESTS
The authors declare no competing financial interests.

1. Giancarlo, R., Rombo, S.E. & Utro, F. *Brief. Bioinform.* **15**, 390–406 (2014).
2. Holland, R.C. & Lynch, N. *GigaScience* **2**, 5 (2013).
3. Deorowicz, S. & Grabowski, S. *Algorithms Mol. Biol.* **8**, 25 (2013).
4. Roguski, L. & Deorowicz, S. *Bioinformatics* **30**, 2213–2215 (2014).
5. Dutta, A., Haque, M.M., Bose, T., Reddy, C.V. & Mande, S.S. *J Bioinform. Comput. Biol.* **13**, 1541003 (2015).
6. Bonfield, J.K. & Mahoney, M.V. *PLoS One* **8**, e59190 (2013).
7. Nicolae, M., Pathak, S. & Rajasekaran, S. *Bioinformatics* **31**, 3276–3281 (2015).
8. Hach, F., Numanagić, I., Alkan, C. & Sahinalp, S.C. *Bioinformatics* **28**, 3051–3057 (2012).
9. Grabowski, S., Deorowicz, S. & Roguski, L. *Bioinformatics* **31**, 1389–1395 (2015).
10. Patro, R. & Kingsford, C. *Bioinformatics* **31**, 2770–2777 (2015).
11. Cox, A.J., Bauer, M.J., Jakobi, T. & Rosone, G. *Bioinformatics* 1415–1419 (2012).
12. Zhang, Y. *et al. BMC Bioinformatics* **16**, 188 (2015).
13. Jones, D.C., Ruzzo, W.L., Peng, X. & Katze, M.G. *Nucleic Acids Res.* **40**, e171 (2012).
14. Benoit, G. *et al. BMC Bioinformatics* **16**, 288 (2015).
15. Kingsford, C. & Patro, R. *Bioinformatics* **31**, 1920–1928 (2015).
16. Zhang, Y., Patel, K., Endrawis, T., Bowers, A. & Sun, Y. *Gene* **579**, 75–81 (2016).
17. Li, H. *et al. Bioinformatics* **25**, 2078–2079 (2009).
18. Tarasov, A., Vilella, A.J., Cuppen, E., Nijman, I.J. & Prins, P. *Bioinformatics* **31**, 2032–2034 (2015).
19. Hsi-Yang Fritz, M., Leinonen, R., Cochrane, G. & Birney, E. *Genome Res.* **21**, 734–740 (2011).
20. Bonfield, J.K. *Bioinformatics* **30**, 2818–2819 (2014).
21. Hach, F., Numanagić, I. & Sahinalp, S.C. *Nat. Methods* **11**, 1082–1084 (2014).
22. Ochoa, I., Hernaez, M. & Weissman, T. *J. Bioinform. Comput. Biol.* **12**, 1442002 (2014).
23. Voges, J., Munderloh, M. & Ostermann, J. Predictive coding of aligned next-generation sequencing data. In *Proc. 2016 Data Compression Conference* 241–250 (IEEE, 2016).

ONLINE METHODS

See **Supplementary Notes 1–6** for details regarding data formats, compression techniques, compression tools, data set and tool selection, and benchmarking.

Data availability. Benchmark data samples are available at https://github.com/sfu-compbio/compression-benchmark/blob/master/samples.md.