Genomic data compression

S. Cenk Sahinalp

Bogaz'da Yaz Okulu
2018

Genome storage and communication: the need

- Research: massive genome projects (e.g. PCAWG) require to exchange 10000s of genomes.
- Need to cover >200 cancer types, many subtypes.
- Within PCAWG TCGA to sequence 11K patients covering 33 cancer types. ICGC to cover 50 cancer types >15PB data at The Cancer Genome Collaboratory
- Clinic: $100s/genome (e.g. NovaSeq) enable sequencing to be a standard tool for pathology
- PCAWG estimates that 250M+ individuals will be sequenced by 2030

Current needs

- Typical technology: Illumina NovaSeq 100-400 bp reads, 250-500GB uncompressed data for high coverage human genome, high redundancy
- 40% of the human genome is repetitive (mobile genetic elements, centromeric DNA, segmental duplications, etc.)
- Upload/download: 55 hrs on a 10Mbit consumer line; 5.5 hrs on a 100Mbit high speed connection
- Technologies under development: expensive, longer reads, higher error (PacBio, Nanopore) – lower redundancy and higher error rates limit compression

File formats

- Raw read data: FASTQ/FASTA – dominant fields: (read name, sequence, quality score)
- Mapped read data: SAM/BAM – reads reordered based on mapping locus on a reference genome (not suitable for metagenomics, organisms with no reference)
- Key decisions to be made:
- Each format can be compressed through specialized methods – should there be a standardized format for compressed genomes?
- Better file formats based on mapping loci on sequence graphs representing common variants in a pan-genomic reference?

Genome Compression: Towards an International Standard

- Collaboration with MPEG to evaluate the current state of HTS data compression towards an International Standard
- Standard Benchmark DataSet: 2+ TB sequence data:
  - 7 FASTQ samples and 8 SAM samples, covering 6 species, 6 technologies, various use-cases (high and low coverage data, cancer cell lines, WGS, RNA-Seq, metagenomics etc.)
- 15 FASTQ tools and 10 SAM tools evaluated
- Available at https://github.com/sfu-compbio/compression-benchmark

Comparison of high-throughput sequencing data compression tools
[Numanagić et al., *Nat. Meth.*, Dec 2016]

FASTA/Q compression

General purpose compressors used in genomics

- LZ77 tools (gzip, pigz)
- BWT tools (bzip2, pbzip2)
- LZMA (7z)
- Context mixing (zpaq, lpaq)
- NCBI Toolkit (used at SRA for storing samples)

- General compressors do not take into account redundancies specific to FASTQ format (FASTQ files are treated as ordinary text files)

Compressor (on 53.8 GB human g. 6.5x coverage)	**Size (total)**	**Size (field by field)**	**Size (sequence)**
pigz	18.5 GB	16.1 GB	5.9 GB
pbzip2	14.8 GB	14.1 GB	5.4 GB
NCBI SRA	~ 14.2 GB		

Specialized FASTA/Q compressors

- Goals:
  - Read name tokenization
  - Separate sequence and quality score modeling

- Examples:
  - DSRC and DSRC2 [1] (Huffman coding)
  - fastqz, fqzcomp [2] and Slimfastq [3] (context mixing with arithmetic coding)
  - FQC [4] and LFQC [5] (LZMA, paq and ppmd as compression engine)

Compressor (on 53.8 GB)	**Size (total)**	**Size (sequence)**
DSRC2	13.2 GB	5.2 GB
Slimfastq	11.0 GB	4.4 GB
FQC	11.4 GB	N/A

[1] Roguski S, Deorowicz S. **DSRC 2--Industry-oriented compression of FASTQ files.** Bioinformatics, 2014
[2] Bonfield JK, Mahoney MV. **Compression of FASTQ and SAM Format Sequencing Data.** PLoS ONE, 2013
[3] Ezra J. https://github.com/Infinidat/slimfastq
[4] Dutta A, Haque MM, Bose T, Reddy CV, Mande SS. **FQC: A novel approach for efficient compression, archival, and dissemination of FASTQ datasets.** J Bioinform Comput Biol., 2015
[5] Nicolae M, Pathak S, Rajasekaran S. **LFQC: a lossless compression algorithm for FASTQ files.** Bioinformatics, 2015

FASTA/Q compressors based on read reordering

- Goals:
  - Reorder reads to improve locality of reference

Compressor (on 53.8 GB)	Size (total)	Size (sequence)
SCALCE	10.8 GB	3.0 GB
ORCOM	N/A	1.7 GB
Mince	N/A	6.0 GB
LW-FQZip	N/A	

- Examples:
  - SCALCE [1] (uses locally consistent parsing for read reordering/clustering)
  - ORCOM [2] (uses lexicographically smallest k-mers for clustering)
  - Mince [3] (similar to ORCOM)
  - LW-FQZip [4] (uses implicit mapping to a reference)

[1] Hach F, Numanagić I, Alkan C, Sahinalp SC. **SCALCE: boosting sequence compression algorithms using locally consistent encoding.** Bioinformatics, 2012
[2] Grabowski S, Deorowicz S, Roguski L. **Disk-based compression of data from genome sequencing.** Bioinformatics, 2014
[3] Patro R, Kingsford C. **Data-dependent Bucketing Improves Reference-free Compression of Sequencing Reads.** Bioinformatics, 2015
[4] Zhang Y, Li L, Yang Y, Yang X, He S, Zhu Z. **Light-weight reference-based compression of FASTQ data.** BMC Bioinformatics, 2015

FASTA/Q compressors based on read assembly

- Goals:
  - Assemble the underlying genome and map reads to the assembly

- Examples:
  - Quip [1] (Bloom filters, assembles clusters of 1 million reads)
  - Leon [2] (probabilistic de Bruijn graph)
  - k-Path [3] (probabilistic de Bruijn graph)

Compressor (on 53.8 GB)	**Size (total)**	**Size (sequence)**
Quip	11.3 GB	4.5 GB
Leon	13.6 GB	4.7 GB
k-Path	N/A	2.0 GB

[1] Jones DC, Ruzzo WL, Peng X, Katze MG. **Compression of next-generation sequencing reads aided by highly efficient de novo assembly**. Nucleic Acids Res. 2012
[2] Benoit G, Lemaitre C, Lavenier D, Drezen E, Dayris T, Uricaru R, Rizk G.. **Reference-free compression of high throughput sequencing data with a probabilistic de Bruijn graph.** BMC Bioinformatics, 2105.
[3] Kingsford C, Patro K. **Reference-based compression of short-read sequences using path encoding**. Bioinformatics, 2015

Compression results on raw (FASTA/Q) read data

Sample	SRR554369		SRR327342		MH0001.081026		SRR1284073		SRR870667		ERR174310		ERR174324	
Organism	*P.aeruginosa*		*S.cerevisiae*		*H.sapiens Gut*		*E.coli*		*T.cacao*		*H.sapiens*		*H.sapiens*	
Technology	Illumina GAIIx		Illumina GAII		Illumina GA		PacBio		Illumina GAIIx		HiSeq		HiSeq	
Coverage	105x		Unknown		Unknown		5x		65x		25x		335x	
Original	**550**		**3,881**		**1,880**		**1,309**		**22,944**		**53,869**		**2,717,029**	
	165		*947*		*512*		*649*		*7,463*		*20,966*		*1,059,387*	
pigz	**158**	1.00	**1,020**	1.00	**501**	1.00	**547**	1.00	**6,943**	1.00	**18,597**	1.00	**305,690**	1.00
	48	1.00	*277*	1.00	*149*	1.00	*188*	1.00	*2,108*	1.00	*5,982*	1.00	*104,927*	1.00
pbzip2	**125**	1.19	**831**	1.41	**390**	1.33	**463**	0.74	**5,577**	1.07	**14,887**	0.80	**242,834**	0.21
	44	6.12	*251*	6.80	*139*	6.10	*176*	6.99	*1,879*	3.59	*5,473*	3.08	*95,969*	1.23
DSRC2	**105**	0.21	**668**	0.26	**312**	0.25	N/A		**4,761**	0.23	**13,214**	0.20	N/A	
	41	2.15	*257*	3.22	*128*	2.06			*1,865*	1.45	*5,239*	1.25		
Fqzcomp	**89**	0.35	**559**	0.37	**280**	0.41	N/A		**4,028**	0.34	**11,320**	0.31	N/A	
	37	N/A	*203*	7.39	*120*	N/A			*1,556*	N/A	*4,623*	3.38		
Fastqz	N/A		N/A		N/A		N/A		N/A		**10,955**	3.45	N/A	
											N/A	N/A		
Slimfastq	**94**	0.54	**507**	0.48	**266**	0.54	N/A		**4,280**	0.52	**11,045**	0.47	**178,092**	0.49
	30	11.62	*149*	9.93	*104*	10.94			*1,416*	5.82	*4,426*	4.89	*77,629*	5.94
FQC	**76**	1.04	**494**	1.23	**268**	1.51	**413**	0.98	**3,912**	1.20	**11,409**	1.22	N/A	
	N/A	12.16	N/A	13.42	N/A	18.66	N/A	12.12	N/A	6.34	N/A	5.87		
LFQC	**69**	9.24	**490**	8.67	**266**	10.44	**407**	18.03	**2,412**	8.53	N/A		N/A	
	17	159.86	*129*	146.15	*103*	162.94	*156*	386.25	N/A	N/A				
SCALCE	**76**	0.38	**487**	0.29	**297**	0.40	**421**	0.67	**3,699**	0.35	**10,827**	0.30	**161,067**	0.57
	17	4.59	*68*	3.87	*71*	5.83	*161*	9.78	*998*	2.88	*3,017*	2.36	*28,452*	1.94
LW-FQZip	**117**	1.10	**790**	0.60	N/A		N/A		**5,038**	2.16	N/A		N/A	
	45	5.60	*320*	5.25					*1,735*	2.56				
Quip	**89**	0.39	**537**	0.48	**272**	0.49	**420**	0.36	**3,914**	0.50	**11,312**	0.46	**184,051**	0.38
	37	7.58	*181*	8.51	*114*	11.00	*159*	10.59	*1,462*	5.74	*4,556*	5.39	*79,771*	4.64
Leon	**87**	3.61	**544**	2.66	**291**	3.66	**479**	2.81	**4,518**	3.93	**13,623**	3.55	**220,397**	1.13
	19	16.95	*89*	17.02	*87*	14.22	*170*	34.31	*1,360*	10.49	*4,739*	9.90	*83,539*	4.66
KIC	**95**	5.75	**613**	7.40	**307**	5.32	**451**	9.40	**4,498**	6.74	**13,006**	6.19	N/A	
	32	6.89	*188*	7.88	*122*	7.38	*168*	9.37	*1,594*	3.60	*4,915*	3.30		
Orcom		0.25		0.22		0.43	N/A			0.49		0.33		0.12
	11	0.77	*36*	0.43	*51*	0.90			*825*	0.72	*1,798*	0.43	*6,921*	0.23
BEETL		4.09		3.14		2.46	N/A			3.97		4.39	N/A	
	23	37.52	*117*	32.22	*114*	29.83			*1,200*	20.68	*3,912*	21.76		
k-Path		1.01		0.81		6.47	N/A			1.35		1.62	N/A	
	14	15.25	*45*	9.49	*62*	71.69			*660*	9.05	*2,088*	8.55		

SAM/BAM compression

General purpose compressors for SAM files

- LZ77 tools (gzip, pigz)
- BWT tools (bzip2, pbzip2)
- Current standard: LZ77-based BAM (Samtools [1], Sambamba [2], Picard [3])

- None of those methods treat differently separate SAM columns.
- Clearly, simple stream separation without any additional post-processing increases significantly the overall compression rate

[1] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, 1000 Genome Project Data Processing Subgroup. **The Sequence Alignment/Map format and SAMtools.** Bioinformatics, 2009

[2] Tarasov A, Vilella AJ, Cuppen E, Nijman IJ, Prins P. **Sambamba: fast processing of NGS alignment formats.** Bioinformatics, 2015

[3] Broad Institute. http://broadinstitute.github.io/picard/

Compressor (on human cancer g. sample 427 GB)	**Size**	**Size (separate streams)**
pigz	119 GB	103 GB
pbzip2	100 GB	94 GB
Samtools	131 GB	102 GB

Specialized SAM tools

- Separate fields into different compression streams
- Use reference to store sequence information, if possible

Compressor (on human cancer sample; 427 GB)	**Size**
Cramtools	95 GB
Scramble	82 GB
Scramble (without reference)	86 GB
Quip (without reference)	98 GB
sam_comp * does not support all SAM fields	42 GB*

- Primarily reference based:
  - CRAM format (Scramble [1], Cramtools [2])
- Assembly and reference based:
  - Quip [3]
- Statistical modeling and arithmetic encoding:
  - sam_comp [4]

[1] Bonfield JK.**The Scramble conversion tool.** Bioinformatics, 2014
[2] Hsi-Yang Fritz M, Leinonen R, Cochrane G, Birney E. **Efficient storage of high throughput DNA sequencing data using reference-based compression.** Genome Res., 2011
[3] Jones DC, Ruzzo WL, Peng X, Katze MG. **Compression of next-generation sequencing reads aided by highly efficient de novo assembly**. Nucleic Acids Res. 2012
[4] Bonfield JK, Mahoney MV. **Compression of FASTQ and SAM Format Sequencing Data.** PLoS ONE, 2013

Local assembly based SAM tools

- Avoid redundant storing of SNPs and other small SVs
- Find the underlying genome via local assembly, and encode SNPs and small SVs only once

- Examples:
  - DeeZ [1]
  - CBC [2]

Compressor (on human cancer sample; 427 GB)	**Size**
DeeZ	78 GB

Sequence only without assembly	**Sequence only with assembly**
4,169 MB	4,120 MB

[1] Hach F, Numanagić I, Sahinalp SC. **DeeZ: reference-based compression by local assembly.** Nat. Methods, 2014
[2] Ochoa I, Hernaez M, Weissman T. **Aligned genomic data compression via improved modeling.** Journal of bioinformatics and computational biology, 2014

DeeZ: DeeNA Zeep

Motivation

- BAM (the most common format for storage and communication) misses some opportunities in SAM format, particularly common SNV loci in reads

- Alternative SAM/BAM compression tools, based on arithmetic coding (AC) and other data modeling methods, like Quip and Samcomp, provide superior compression, but not random-access capability

- DeeZ locally assembles reads and represents each SNV once, on the contig.

DeeZ: Quality scores

- Quality scores account for majority of the space in almost any format
  - minor improvement in quality score compression is more beneficial than improvement in other areas

DeeZ on human cancer sample **427 GB**	**Size**	**Gain**
Sequence only without local assembly (5% of compressed file)	4,169 MB	
Sequence only with local assembly (5% of compressed file)	4,120 MB	49 MB
Quality scores only with order-1 AC model (42% of compressed file)	33,516 MB	
Quality scores only with sam_comp model (41% of compressed file)	31,010 MB	2,506 MB

Compression results on mapped (SAM/BAM) read data

Sample	DH10B		9827.2.49		sample-2-1		K562.LID8465		HCC1954		NA12878.S1	
Organism	*E.coli*		*H.sapiens*		*H.sapiens*		*H.sapiens*		*H.sapiens*		*H.sapiens*	
Technology	MiSeq		HiSeq		IonTorrent		RNASeq		Cancer Cell		HiSeq	
Coverage	490x		2x		0.7x		7x		35x		60x	
Original	5,579		21,059		5,924		75,915		427,028		589,083	
pigz	1,336	0.77	6,021	1.55	1,378	1.48	12,785	1.06	119,839	1.40	113,462	0.13
		0.63		0.82		0.49		0.70		0.91		0.60
pbzip2	1,074	1.65	5,243	1.93	1,127	4.04	10,251	3.57	100,280	1.62	89,598	0.46
		3.16		3.39		3.72		2.46		3.23		0.59
Samtools	1,407	1.00	6,499	1.00	1,469	1.00	13,757	1.00	131,566	1.00	121,710	1.00
		1.00		1.00		1.00		1.00		1.00		1.00
Picard	1,425	1.42	6,517	1.04	1,474	1.82	13,818	1.48	132,861	1.18	N/A	
		2.76		1.52		2.10		2.44		1.91		
Sambamba	1,407	1.05	6,499	0.93	1,469	1.12	13,757	1.05	131,566	1.39	121,710	0.13
		1.08		1.13		0.97		0.97		1.12		0.53
Cramtools	1,066	0.93	3,778	1.42	1,170	2.12	10,344	1.70	95,442	1.28	N/A	
		1.71		1.67		4.93		2.00		1.50		
Scramble	863	0.23	3,297	0.29	1,030	0.62	9,261	0.38	82,041	0.27	66,632	0.10
		0.76		0.66		1.58		0.67		0.71		0.50
Scramble without reference	899	0.29	4,236	1.18	1,113	0.45	9,839	0.43	86,914	0.37	72,407	0.10
		0.74		0.63		1.06		0.78		0.79		0.47
Scramble with bzip2	851	0.76	3,262	0.62	998	1.50	8,611	1.27	80,094	0.60	N/A	
		0.89		0.66		1.72		0.81		0.82		
DeeZ	823	0.56	3,221	0.78	1,028	1.81	8,120	0.92	78,473	0.91	62,966	0.26
		3.90		2.46		5.51		3.35		2.94		1.00
DeeZ with bzip2	730	0.91	2,734	1.23	918	3.49	7,266	2.01	74,509	1.66	53,497	0.41
		10.11		5.60		9.86		7.91		6.39		1.90
TSC	1,105	2.21	7,939	0.80	1,193	2.55	20,864	3.17	164,627	0.50	N/A	
		9.05		2.24		6.75		6.27		2.65		
Quip	1,103	0.67	4,419	0.94	1,230	0.96	11,186	1.19	98,303	0.83	97,165	0.44
		10.69		7.81		3.37		8.27		9.05		2.18
Quip with reference	803	0.67	N/A		N/A		8,743	1.17	N/A		64,493	0.43
		10.06						8.20				2.20
sam_comp	700	0.68	2,649	0.76	891	1.20	7,023	0.71	42,522	0.62	53,263	0.37
		3.36		2.95		6.54		3.56		3.25		2.00

Optimal Compressed Representation of High Throughput Sequence Data via Light Assembly

Cenk Sahinalp

Based on joint work with

Kaiyuan Zhu, Tony Ginart, Joseph Hui, Ibrahim Numanagić,

Thomas Courtade, David Tse

Current FASTQ Compression Schemes

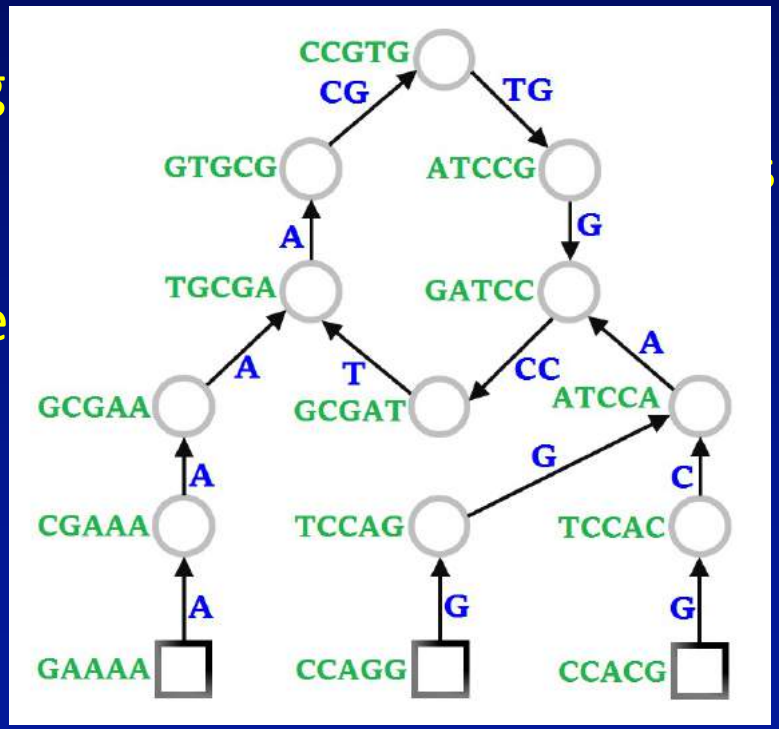
- General purpose compressors (FASTQ files are treated as ordinary text files)
  - gzip (parallel gzip--pigz), bzip2 (parallel bzip2--pbzip2)
- Alignment e underlying reference g
  - use de-n raph
    - Quip,
  - use an us ng
    - LW-FQ
- Reordering reordering of the reads ly boost the compression rates while avoiding information loss.
  - SCALCE, Orcom, Mince

Compressor (on 53.8 GB human with 6.5x coverage)	Size (total)	Size (field by field)	Size (sequence)
pigz	18.5 GB	16.1 GB	5.9 GB (2.251)
pbzip2	14.8 GB	14.1 GB	5.4 GB (2.060)
SRA	~ 14.2 GB		

Assembltrie: Our New Compressed Representation

- Combine the advantages of reordering and alignment based compressors.
- The input reads are organized into a *forest* of compact trie-like data structures called *read forest*.
  - Each node v represents a read (a string
  - Each directed edge (u, v) represents
    covered by its parent, v
  - May contain a single cycle acting as the

not



An example trie-like structure

Combinatorial Optimization Formulation

- Among all possible read forests, our objective is to find the one contains minimum number of symbols, i.e.

$$T^* = \arg\min_{T} \sum_{\tau \in T} \sum_{v \in V_\tau} w[v, \pi(v)]$$

τ: a trie in the forest T, with the corresponding vertex set V_τ;
$w[v,u]$: the length of the shortest suffix of v that can not be covered by a prefix of u;
$\pi(v)$: the parent node of v in T

- The greedy algorithm to build the desired read forest
  - Pick for each read u an already processed read v with minimum $w[u,v]$, set its parent $\pi(u)$ to v
  - If no such v exists, set $\pi(u) = u$
  - Identify each already processed read v with $w[v,v] < K$ (resp. $w[v,v] \geq K$) to trie with only u (can assume $\pi(u)$ = NIL)

Theorem: The greedy algorithm computes the optimal T^* with minimum overlap K.

Information Theoretic Upper Bound for HTS Data Compression

- Assembltrie achieves combinatorial optimality
  - For any finite collection $\mathcal{R}$ of reads to be compressed with any explicit or implicit (overlap graph) assembly based compressor, it produces the smallest number of symbols to be encoded for reads.
- Is it possible to obtain better compression performance by a fundamentally different data structure (i.e. representation of reads)?
  - NO. The minimum number of bits needed by any algorithm to describe the reads $\mathcal{R}$ is given by $H(\mathcal{R})$.
  - Assembltrie approaches this lower bound asymptotically for low-error
- ≠ Optimal compression in practice
  - The proof does not consider read errors
  - Need to account for reverse complements, paired reads etc.

$$H(\mathcal{R}) \approx NL\log(3)h_2(p) + |G| \cdot H\left(\text{Poisson}\left(N/|G|\right)\right) + LZ(G)$$

Sequencing errors Read sampling process Reference genome

$$h_2(p) = -p\log p\ -(1-p)\log 1-p$$

N: number of reads; *L*: read length; *G*: reference genome of length $|G|$

Compression Performance (8 Threads, in bit per base)

MPEG HTS FASTQ Dataset — Sampled S.cerevisiae

Sample	Read L. / Cov.	Assembltrie	Orcom	Mince	K-Path	SCALCE
P.aeruginosa	100 / 25	0.345	0.518	0.484	0.673	0.821
S.cerevisiae	63 / 80	0.271	0.304	0.312	0.384	0.578
H.sapiens gut	44 / NA	0.757	0.804	0.786	2.545	1.104
T.cacao	108 / 20	1.733	0.884	0.735	0.707	1.070
Sim. *T.cacao*	108 / 19	0.479	0.667	N/A	N/A	N/A
H.sapiens 1	101 / 7	0.570	0.686	0.746	0.797	1.151
H.sapiens 2	101 / 20	0.322	0.364	N/A	N/A	N/A

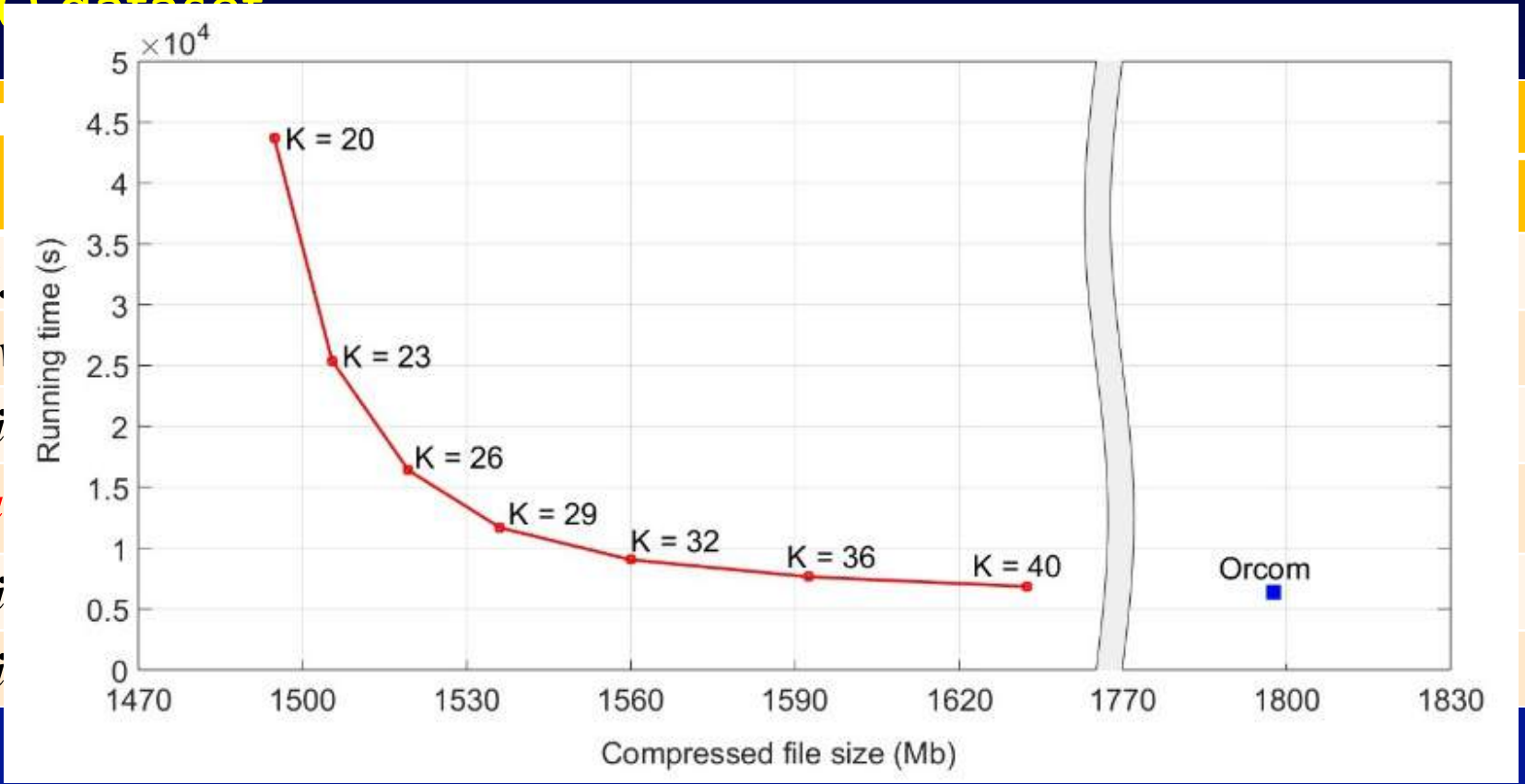
Coverage

Ref: Numanagić, I., Bonfield, J. K., Hach, F., Voges, J., Ostermann, J., Alberti, C., ... & Sahinalp, S. C. (2016). Comparison of high-throughput sequencing data compression tools. *Nature methods*.

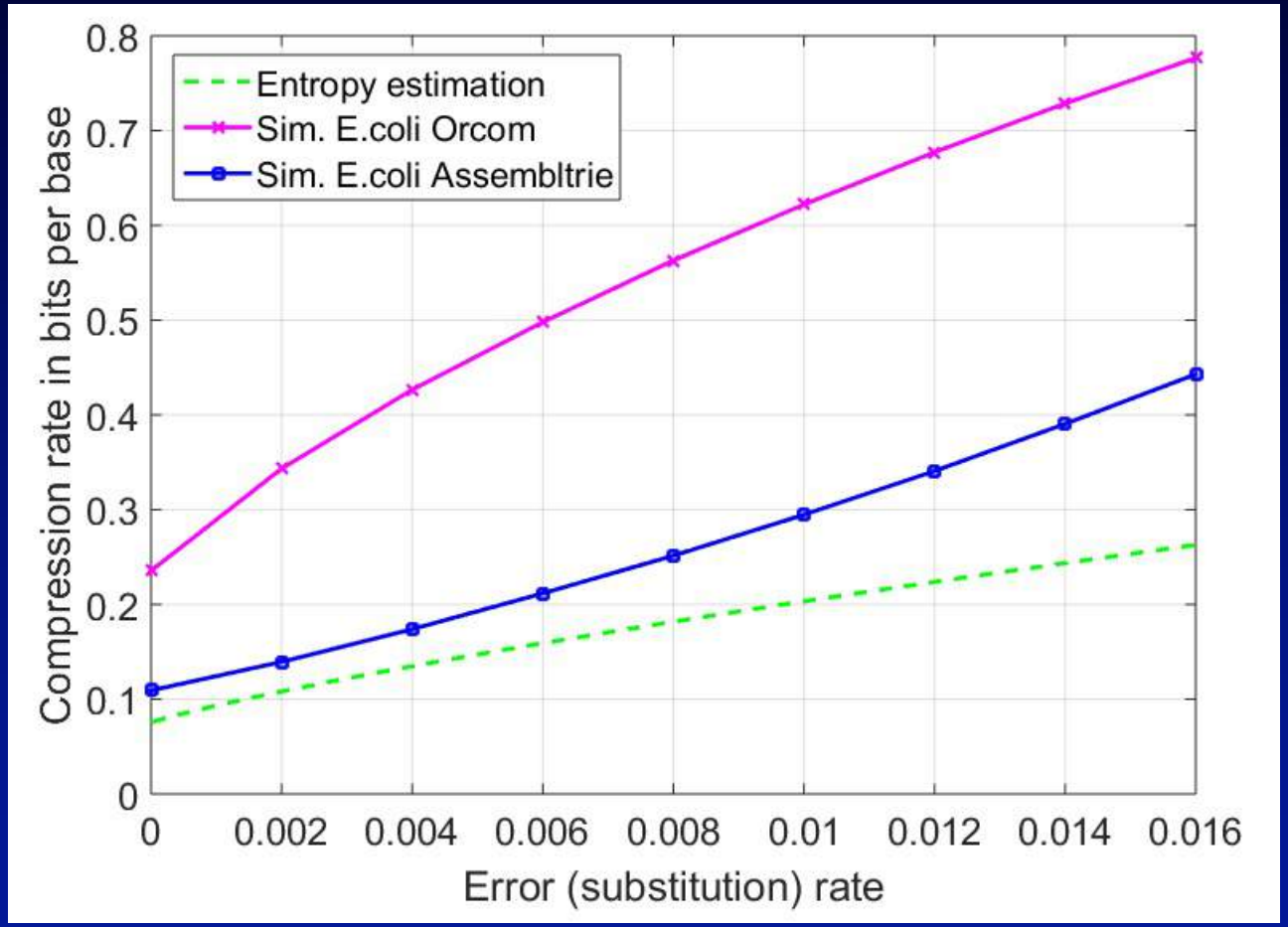
Running Time (8 Threads, in seconds)

- Default running time to generate the compression rates in MPEG FASTQ dataset

Sampl						
P.aeru						
S.cere						
H.sapi						
T.caca						
H.sapi						
H.sapi						



Assembltrie's Performance vs Information theoretic upper bound on compression



Acknowledgements

- National Science Foundation (NSF) CCF-1619081, CCF-1528132 and CCF-0939370 (Center for Science of Information)
- National Institutes of Health (NIH) GM108348
- The Cancer Genome Collaboratory
- Indiana University Precision Health Initiative