EPFL

MPEG-G Reference-Based Compression of Unaligned Reads Through Ultra-Fast Alignments

U. Ozturk
S. Casale-Brunet
P. Ribeca
M. Mattavelli

DCC 2022

Genetic Data is Big Data [1]

- Genomic data grows at the same rate of other big data domains
- Next-generation, high throughput sequencing (NGS) produces short-read genetic sequencing data at a rate of 130 MB/s
- 50x redundant coverage of the original genetic sequence is typical
- Indexing and compression of this data is necessary for its storage and transport

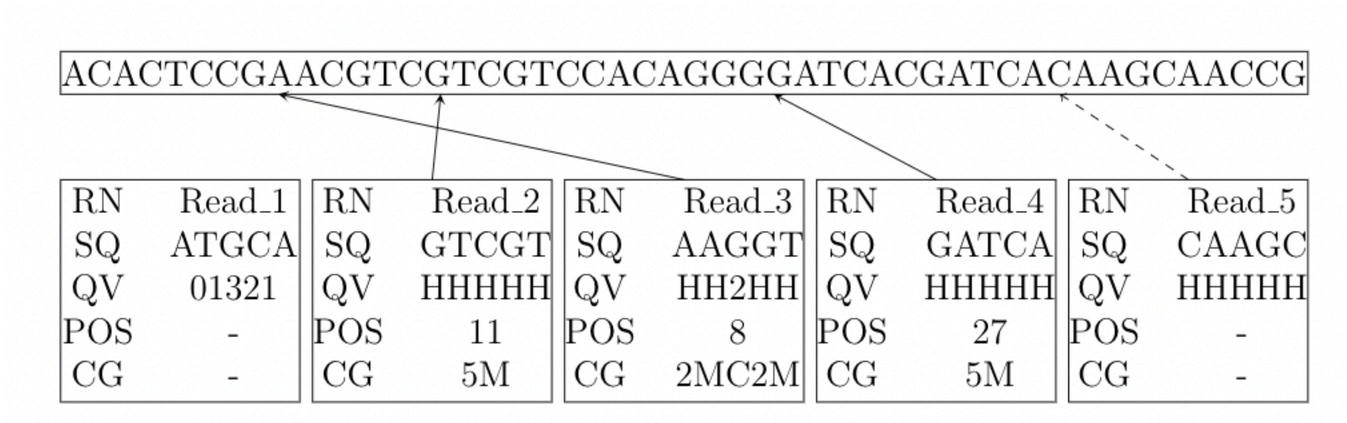
Compression of Sequencing Reads

- Compression of sequencing reads are handled through:
  - Generic data compression algorithms
  - Domain-specific compression strategies and file formats
- Generic data compression algorithms:
  - gzip, bzip2, LZMA…
  - Focus on compression, and not on indexing or selective access
- Domain specific algorithms:
  - Exploit the statistics and characteristics of the underlying process producing the data
  - Reference-based (CRAM, LW-FQZip), assembly-based (Quip), local assembly (DeeZ), reference-free (SPRING), and many more [2,3,4,5,6]…

Sequencing Reads & Descriptors

- Sequencing reads have three fields:
  - Read names
  - Nucleotide sequence (A-C-G-T & N)
  - Quality values
- Sequencing reads may also be mapped to a reference sequence:
  - Relevant descriptors are:
    - Mapping position
    - CIGAR (Compact Idiosyncratic Gapped Alignment Report)
    - Alignment quality

Sequencing Reads & Descriptors



ACACTCCGAACGTCGTCGTCCACAGGGGATCACGATCACAAGCAACCG

RN	Read_1	Read_2	Read_3	Read_4	Read_5
SQ	ATGCA	GTCGT	AAGGT	GATCA	CAAGC
QV	01321	HHHHH	HH2HH	HHHHH	HHHHH
POS	-	11	8	27	-
CG	-	5M	2MC2M	5M	-

Example logical representation of sequencing reads and alignments to a reference sequence. Read 1 is a poor quality read and is unmapped, Reads 2-3-4 map to the reference, Read 5 potentially maps to the reference, however is unaligned.

Reference-based Compression

- Nucleotide sequences in sequencing reads are (approximate) substring samples from a longer sequence

- Store the offset and length of the sequence in relation to a reference sequence and the reference sequence itself instead of the actual sequencing read
  - Saves space
  - Exploited by CRAM

- **Requires alignment mapping information…**

Fast Alignment

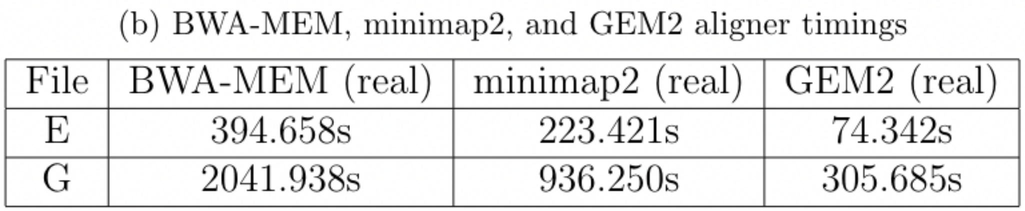
- Our contribution:
    - Improve compression rates for unaligned sequencing data by aligning reads to a reference sequence
    - The alignment process is merely meant for compression:
        - No regard to biological accuracy
        - Emphasis on alignment mapping speed rather than quality
    - Executed under the MPEG-G Framework:
        - MPEG-G supports reference-based compression
        - Aligner outputs transcoded to data structures defined in ISO/ IEC 23092-2

Fast Alignment

- Three steps of the procedure:
  - Fast alignment:
    - An aligner is used to align sequencing reads to a reference sequence
    - Require that 80%> of sequencing reads to be aligned
  - Transcoding:
    - Aligner output is transcoded into MPEG-G records
    - Information extraneous to compression is thrown away
  - Encoding:
    - An MPEG-G compliant codec is used to encode MPEG-G records

Fast Alignment - Choice of Aligner

- Three aligners were benchmarked: BWA-MEM, minimap2, and GEM2
- Parameters were tuned to map sequencing reads as fast as possible
- Two input files from the MPEG-G data repository were used
- GEM2 outperformed BWA-MEM and minimap2 in terms of speed under the given constraints, and was used for the rest of the experiments, and parameter space (-m,-e) is sampled for compression rate vs. time tradeoffs



(b) BWA-MEM, minimap2, and GEM2 aligner timings

File	BWA-MEM (real)	minimap2 (real)	GEM2 (real)
E	394.658s	223.421s	74.342s
G	2041.938s	936.250s	305.685s

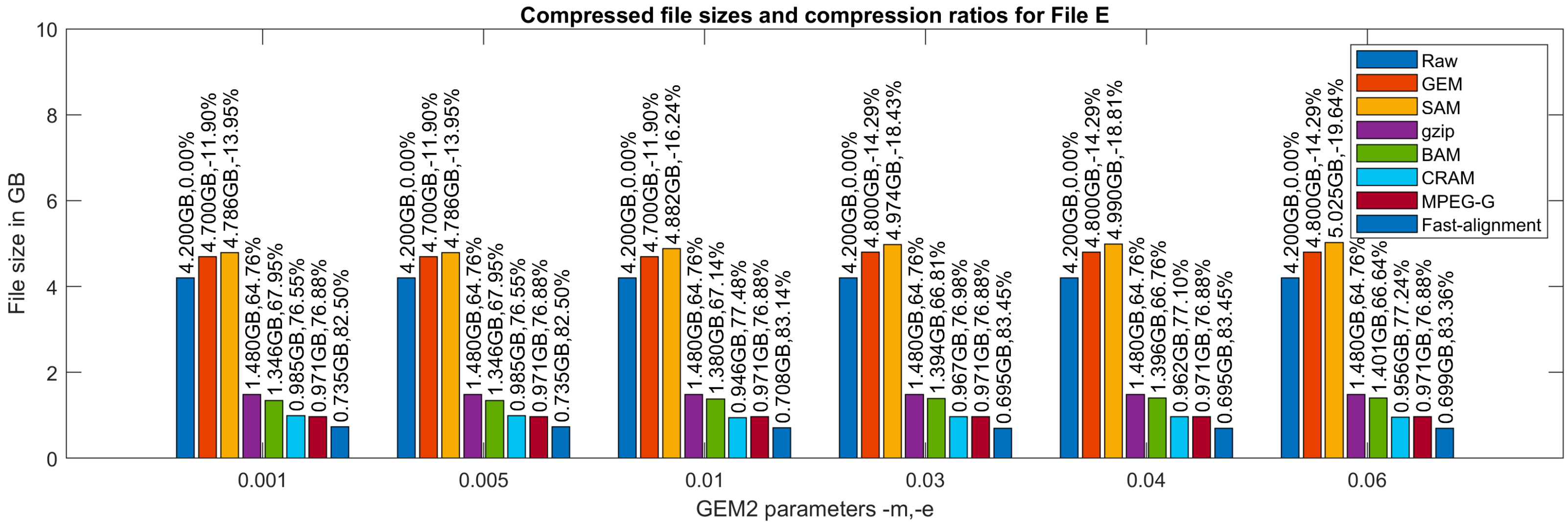
Fast Alignment - Transcoding & Encoding

- A GEM2 to MPEG-G records transcoder (g2tc) was implemented:
  - Only one alignment is kept per sequencing read
  - Paired-end reads are re-named carefully to save space
  - Template paired-end reads are grouped together into the same record as much as possible
  - Alignment qualities are discarded and reads are sorted in terms of their mapping positions
  - Resulting records are encoded using a proprietary encoder

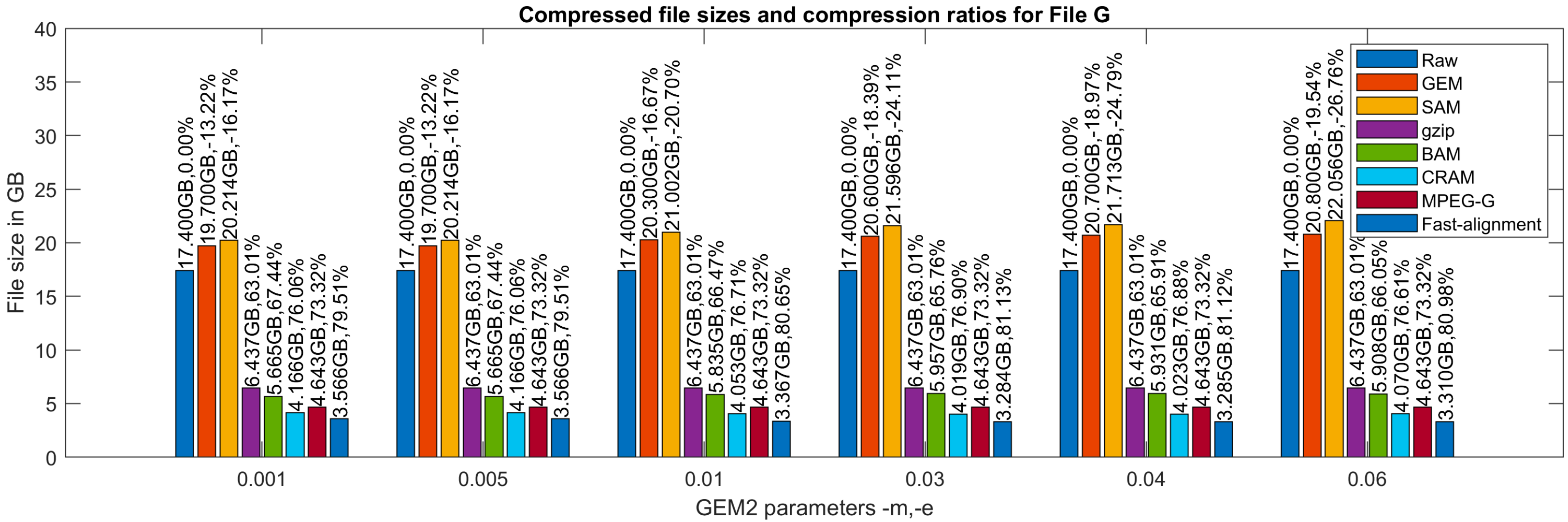
Fast Alignment - Experimental Setup

- The Fast Alignment pipeline was compared to:
  - gzip (commonly used to compressed unaligned reads)
  - BAM
  - CRAM (compressed SAM files with reference based compression)
  - MPEG-G without alignment descriptors
- On two files present in the MPEG-G file repository:
  - ERR174310.chr9 (file E, 4.2 GB)
  - G15511.HCC1143_BL.1.chr9 (file G, 17.6 GB)
  - hs37d5 as the reference sequence
- With different GEM2 parameters (-m,-e 0.001, 0.005, 0.01, 0.03, 0.04, 0.06)

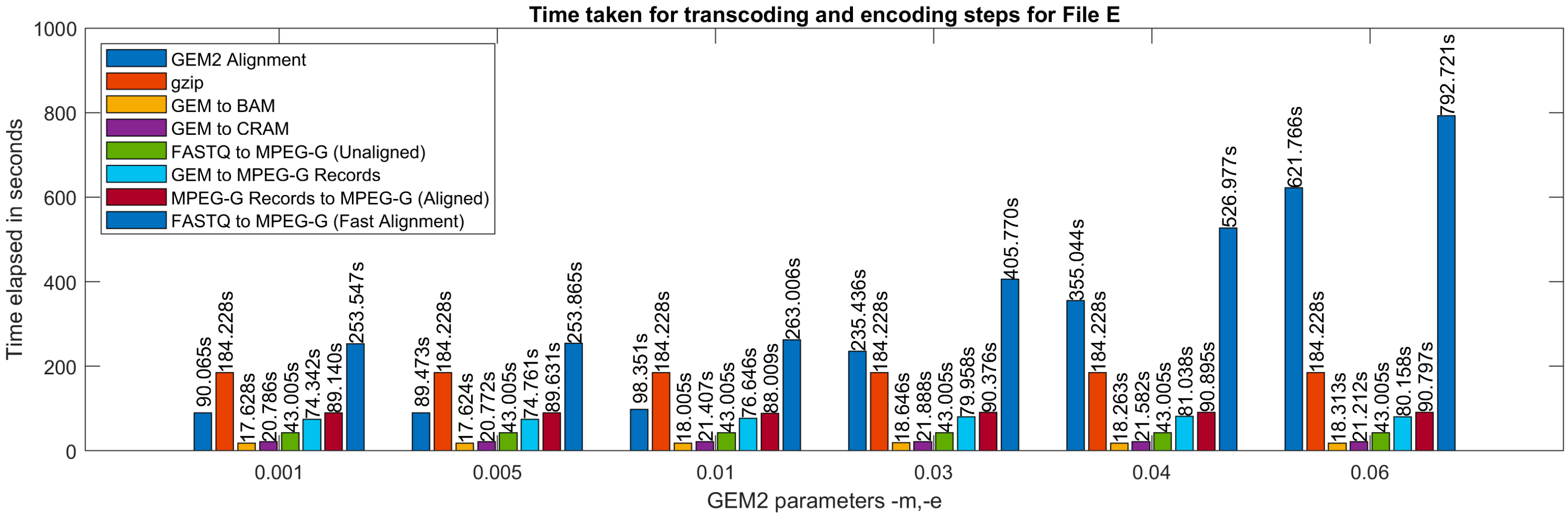
Fast Alignment - Results - File Size



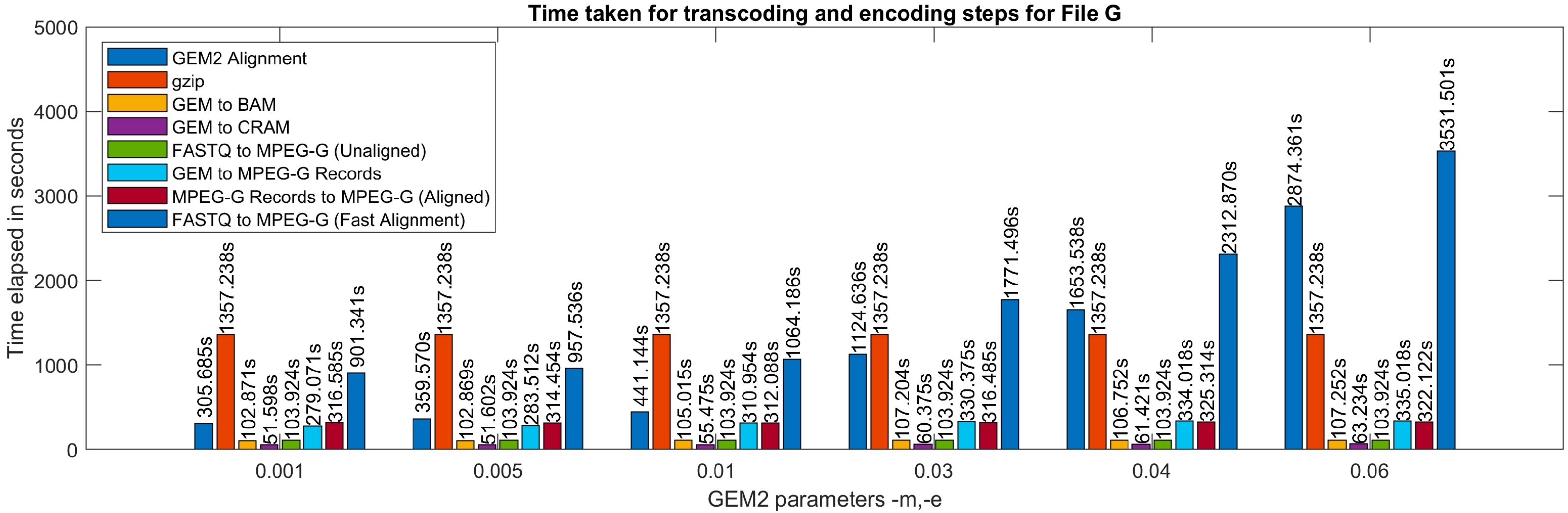
Fast Alignment - Results - File Size



Fast Alignment - Results - Times



Fast Alignment - Results - Times



Fast Alignment - Discussion & Improvements

- The alignment step is the bottleneck of the process
- Provides better compression than gzip for the same compression time
- Fast alignment outperforms other methods in terms of compression rates … at the cost of higher compression times

- The alignment bottleneck could be reduced:
  - Run multiple GEM2 aligners by loading the index many times in memory to reduce
  - Streaming between alignment, transcoding, and encoding instead of communication via disk I/O

References

[1] Zachary D. Stephens, Skylar Y. Lee, Faraz Faghri, Roy H. Campbell, Chengxiang Zhai, Miles J. Efron, Ravishankar Iyer, Michael C. Schatz, Saurabh Sinha, and Gene E. Robinson, “Big data: Astronomical or genomical?,” PLoS biology, vol. 13, no. 7, pp. e1002195–e1002195, Jul 2015, 26151137[pmid].

[2] The SAM/BAM Format Specification Working Group, “Cram format specification (version 3.0),” http://samtools.github.io/hts-specs/CRAMv3.pdf, 2021.

[3] Zhi-An Huang, Zhenkun Wen, Qingjin Deng, Ying Chu, Yiwen Sun, and Zexuan Zhu, “LW-FQZip 2: a parallelized reference-based compression of FASTQ files,” BMC Bioinformatics, vol. 18, no. 1, pp. 179, Mar 2017.

[4] Daniel C. Jones, Walter L. Ruzzo, Xinxia Peng, and Michael G. Katze, “Compression of next-generation sequencing reads aided by highly efficient de novo assembly,” Nucleic acids research, vol. 40, no. 22, pp. e171–e171, Dec 2012, 22904078[pmid].

[5] Faraz Hach, Ibrahim Numanagic, and S. Cenk Sahinalp, “Deez: reference-based compression by local assembly,” Nature Methods, vol. 11, no. 11, pp. 1082–1084, Nov 2014.

[6] Shubham Chandak, Kedar Tatwawadi, Idoia Ochoa, Mikel Hernaez, and Tsachy Weissman, “SPRING: a next-generation compressor for FASTQ data, Bioinformatics (Oxford,England), vol. 35, pp. 2674–2676, 08 2019.