Genomic Data Compression and Processing for Large and Growing Databases

Science of Information

EE 25N

Tsachy Weissman
thanks

Idoia Ochoa

Mikel Hernaez

Shubham Chandak

Kedar Tatwawadi

Dmitri Pavlichin
a slide we’ve seen (multiple times..)
why compression of genomic data?

• proxy for reduced cost of storage, communication, computation, processing, etc.
• compression as modelling
2 modes

• lossless
• lossy
(true) story of human genome compression
**FASTA Compression**: Compression of a single genome

- Human genome can be represented using 2 bits/bp, compressed size \( \approx 1\text{GB} \)

- Specialized compressors: MFCompress\(^1\) : \( \approx 1.6 \text{ bits/bp} \)

- \( H(\text{human genome}) \): "Entropy" of the human genome
  \[
  H(\text{human genome}) \lessapprox 2 \text{ bits/bp} \sim 1 \text{ GB}
  \]

---

Using a reference

- \( H(\text{human genome}) \lesssim 1 \text{ GB} \)
- We can do better if we know another genome (reference)
- **Using a reference:** GenomeZip\(^2\) compresses James Watson's genome using: \( \approx 2.5\text{MB} \)

\[
H \left( \begin{array}{c|c}
\text{human} \\
\text{genome} \\
\hline
\text{another} \\
\text{human} \\
\text{genome}
\end{array} \right) \lesssim 2.5\text{MB}
\]

Using a collection

- $H(\text{human genome}) \lesssim 1 \text{ GB}$

- $H \left( \frac{\text{human genome}}{\text{another human genome}} \right) \lesssim 2.5 \text{ MB}$

- **GTRAC$^3$** compressor:

  $H \left( \frac{\text{human genome}}{\text{1K other genomes}} \right) \lesssim 1 \text{ MB}$

- **GTC$^4$** compressor:

  $H \left( \frac{\text{human genome}}{\text{27K other genomes}} \right) \lesssim 200 \text{ KB}$

---


Full Genome Compression

- $H(\text{human genome}) \lesssim 1 \text{ GB}$
- $H \left( \begin{array}{c} \text{human genome} \\ \text{another human genome} \end{array} \right) \lesssim 2.5 \text{ MB}$
- $H \left( \begin{array}{c} \text{human genome} \\ \text{1K other genomes} \end{array} \right) \lesssim 1 \text{ MB}$
- $H \left( \begin{array}{c} \text{human genome} \\ \text{27K other genomes} \end{array} \right) \lesssim 200 \text{ KB}$
File Size per Genome Vs Database Size

H(human genome | Sl)

# of Sl genomes

File size

Human genome

1GB

2MB

1MB

160KB

1KB

1

1K

27K

$10^{10}$
Information Theoretic Perspective

$X \xrightarrow{\text{Compressor}} \text{Compressor} \xrightarrow{\text{bits } \approx H(X|SI)} \text{Decompressor} \xrightarrow{\text{Decompressor}} X$

SI
Information Theoretic Perspective

\[ X \rightarrow \text{Compressor} \rightarrow \text{Decompressor} \rightarrow X \]

\#bits = ?

SI
Information Theoretic Perspective

why lossy compression?

LOSSLESS

Original → Compressed → Restored

LOSSY

Original → Compressed → Restored
why *lossy* compression of *genomic* data?
genome sequencing

Reads

Genome

Coverage
Reads + alignment information

Identifiers

Quality scores
“QualComp: a new lossy compressor for quality scores based on rate distortion theory”

“QVZ: lossy compression of quality values”
how does lossy compression of quality scores affect the inference?

“Effect of lossy compression of quality scores on variant calling”

with Idoia Ochoa, Mikel Hernaez, Rachel Goldfeder and Euan Ashley

*Briefings in Bioinformatics, 2017*
**FASTQ file**

- **Alignment**
  - **BWA** doesn’t use the Quality Scores

**Reference genome**

**SAM file**

**Variant caller**

**VCF file**

**Extract Quality Scores**

**Lossy Compression**

**Lossy Decompression**
performance metrics

- sensitivity: $\frac{\text{T.P.}}{\text{T.P.} + \text{F.N.}}$
- precision: $\frac{\text{T.P.}}{\text{T.P.} + \text{F.P.}}$
- F-score: Harmonic mean of sensitivity and precision
NA12878, Chr. 11

- **Sensitivity**
  - GATK (Lossy)
  - GATK (Lossless)
  - Platypus (Lossy)
  - Platypus (Lossless)
  - Samtools (Lossy)
  - Samtools (Lossless)

- **Precision**

- **F-score**

**Bits per quality score**
zooming in

*GATK variant caller*

- **Original**
- **QVZ**

E-Score

Distortion level at the lossy compressor (MSE)
denoising via lossy compression

- “Occam filters”
- “Compresstimation”
- etc.
violating the data processing principle?

suggests need to lift hood of:

- variant callers
- assemblers
- etc.
thanks