Stanford University

Compression of raw genomic data

Shubham Chandak

Advisor: Tsachy Weissman

Department of Electrical Engineering, Stanford University

Apr 19, 2021

Outline

- Introduction and Motivation
- SPRING: a compressor for FASTQ data
- Lossy compression of nanopore raw signal data

Introduction and Motivation

What is the genome?

What is genome sequencing?

Why compression?

Raw data and downstream analysis

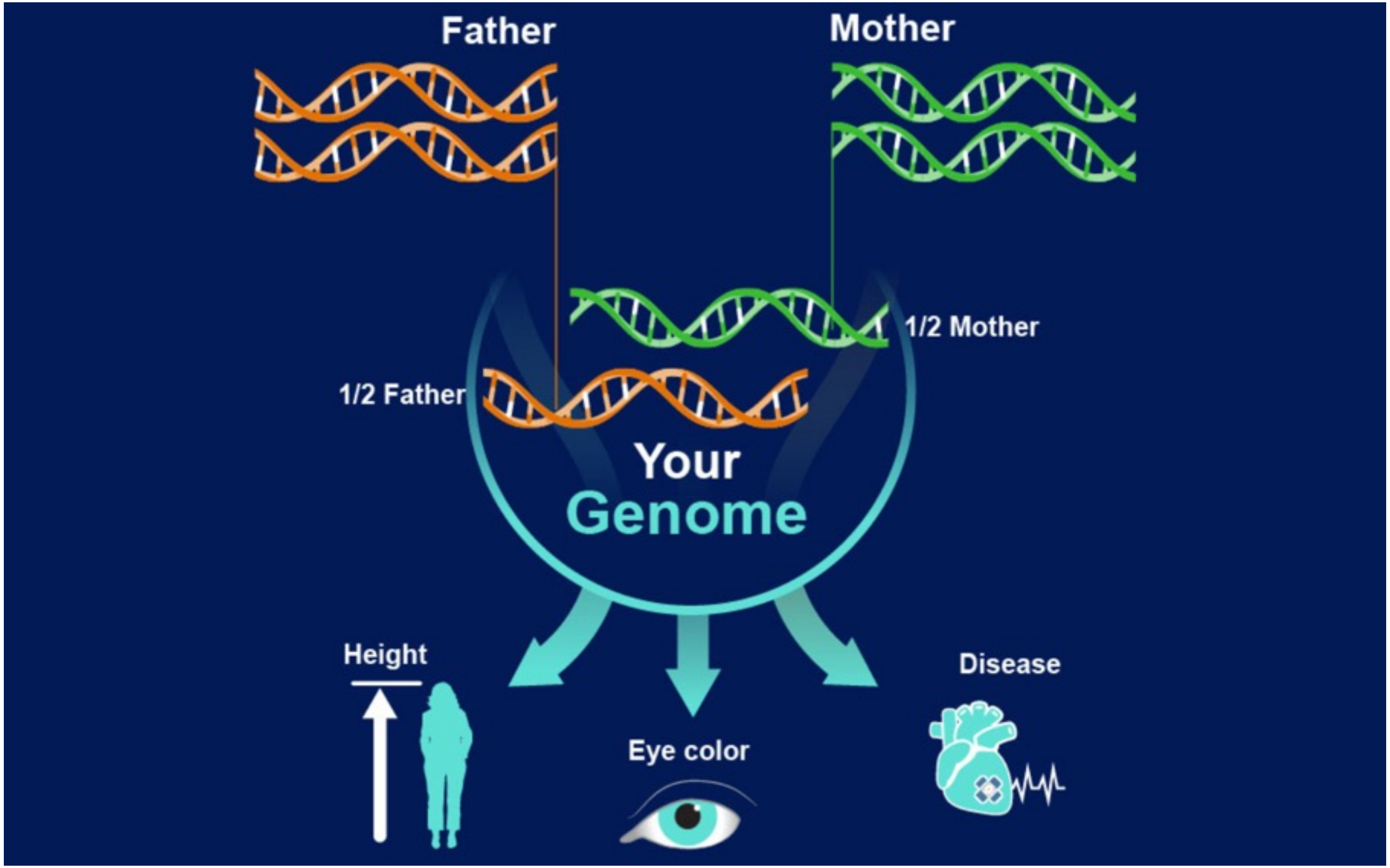


Image source: https://www.genome.gov/About-Genomics/Introduction-to-Genomics

What is the genome?

- Sequence of DNA bases in {A, C, G, T}
- Two complementary strands
- For humans:
  - 3 billion bases (x2)
  - Across 23 (x2) chromosomes

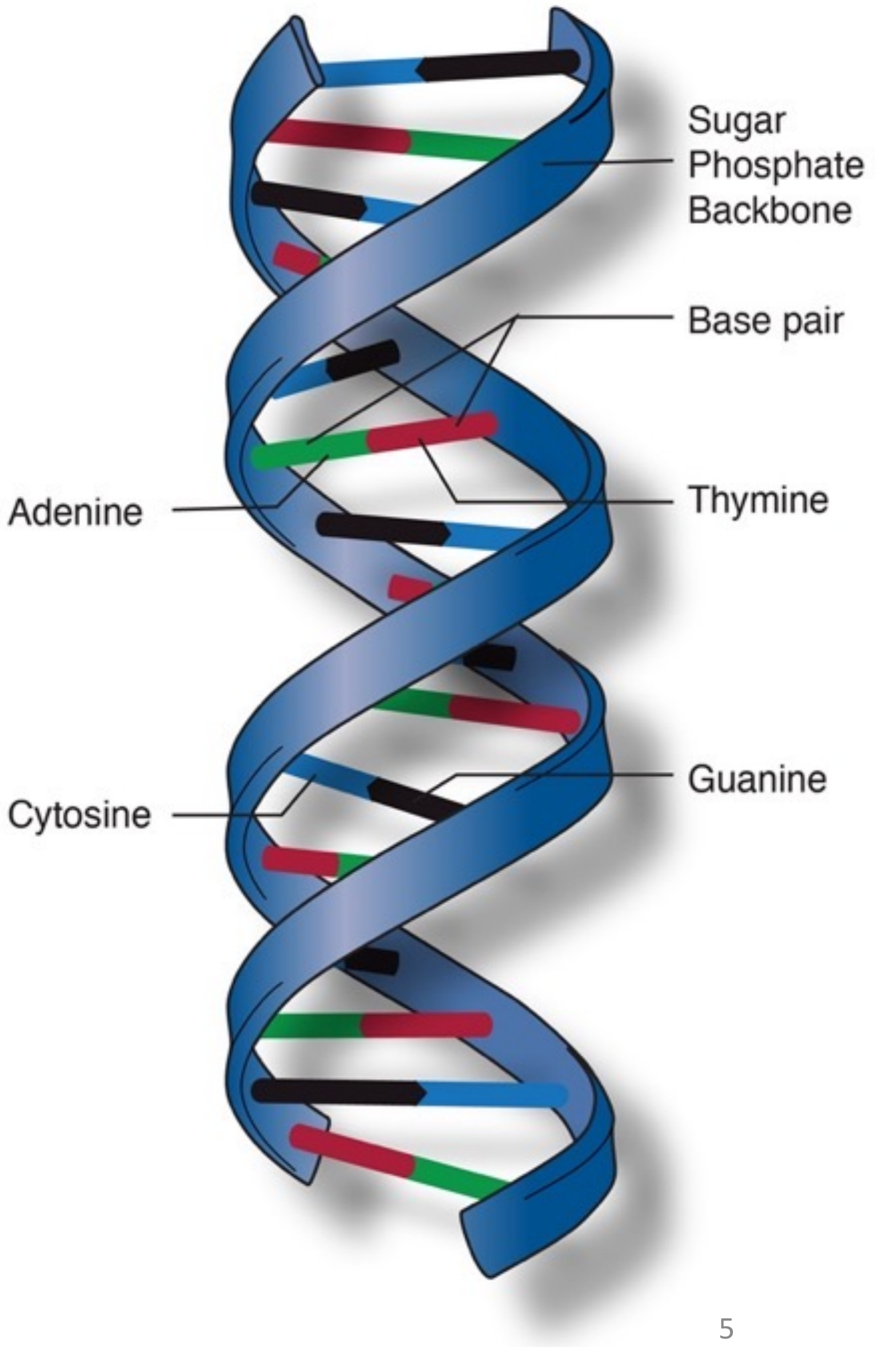
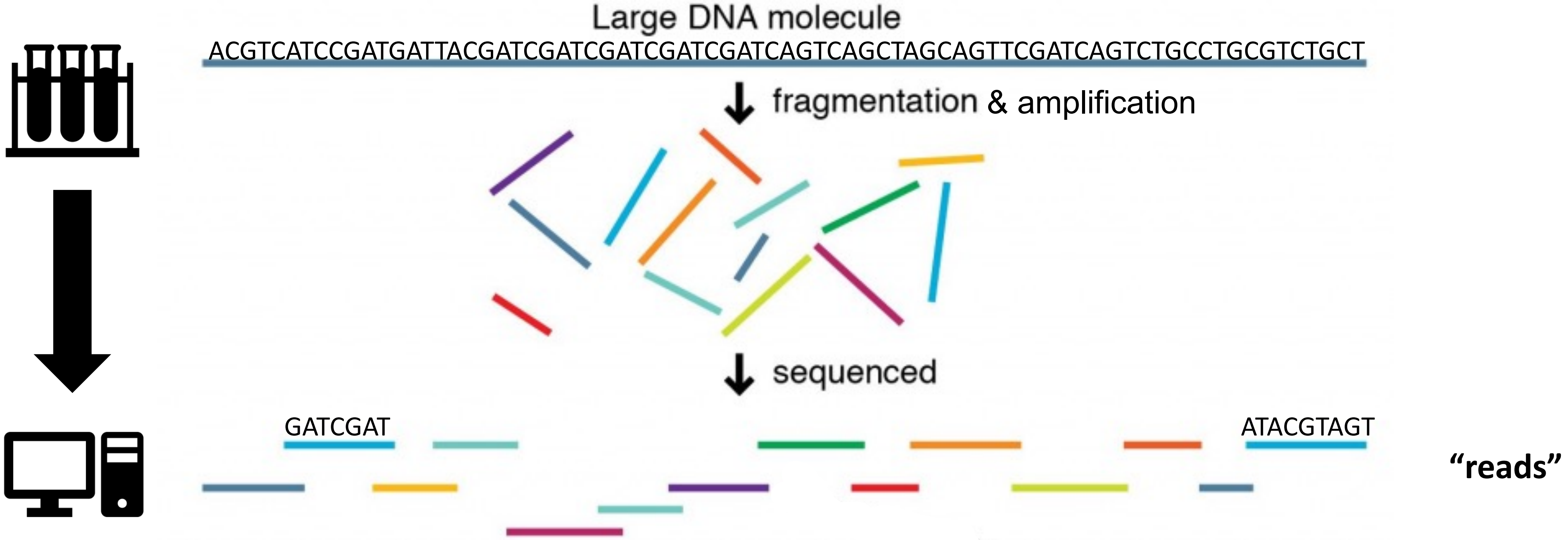


Image source: https://www.genome.gov/genetics-glossary/Double-Helix

Genome sequencing



Large DNA molecule

ACGTCATCCGATGATTACGATCGATCGATCGATCGATCAGTCAGCTAGCAGTTCGATCAGTCTGCCTGCGTCTGCT

↓ fragmentation & amplification

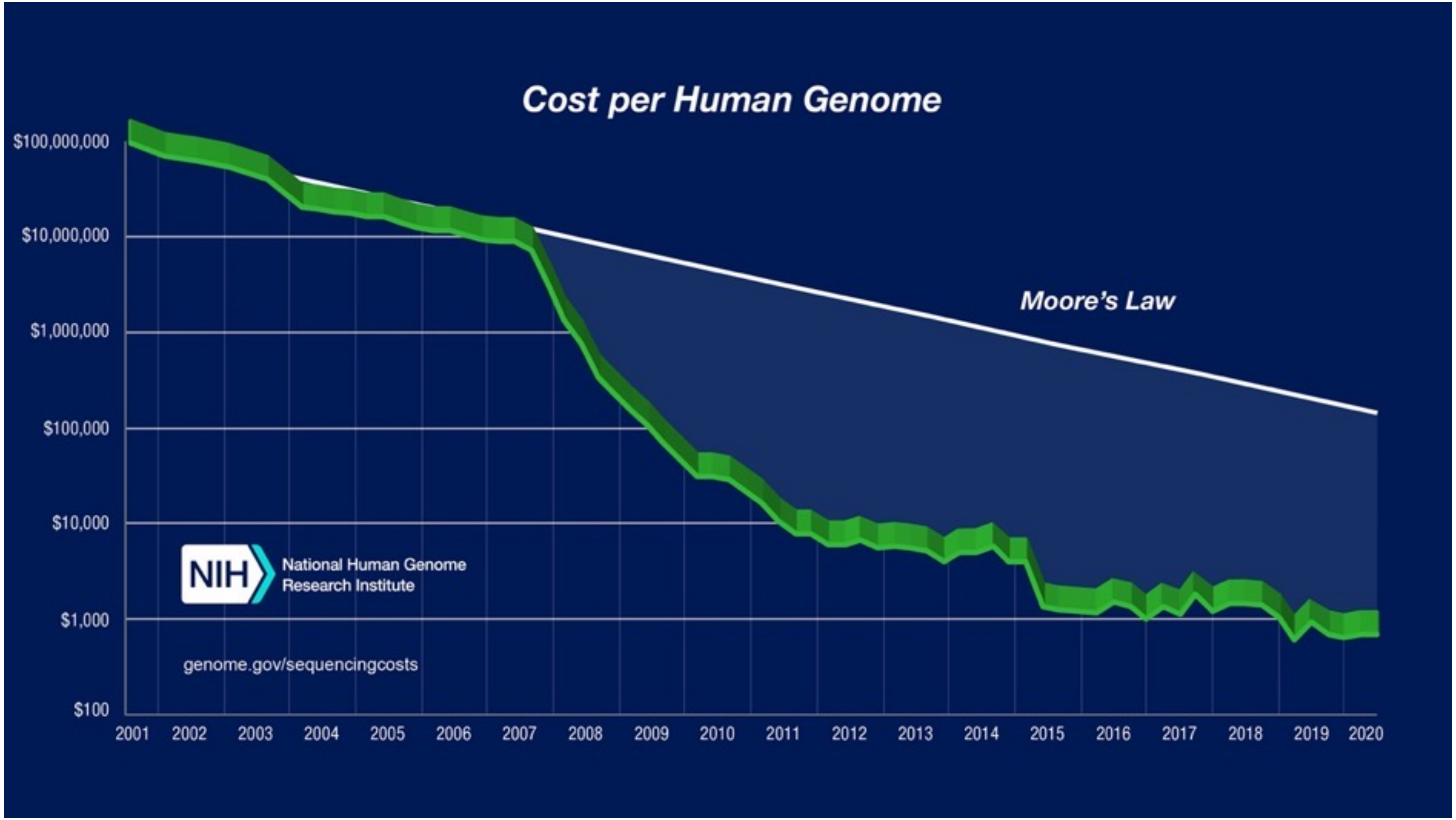
↓ sequenced

GATCGAT

ATACGTAGT

“reads”

Image source: https://www.genengnews.com/uncategorized/first-nanopore-sequencing-of-human-genome/


Cost per Human Genome
$100,000,000
$10,000,000
$1,000,000
$100,000
$10,000
$1,000
$100
Moore's Law
NIH
National Human Genome Research Institute
genome.gov/sequencingcosts
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020

nature

THE INTERNATI… WEEKLY JOURNAL OF SCIENCE

UK BIOBANK

Genetic and health data from half a million people in the United Kingdom

PAGES 194, 203 & 210

500K human genomes

The international journal of science / 5 December 2019

Volume 576 Issue 7785

nature

GENOMES FROM ASIA

Initial sequence data set from the GenomeAsia 100K project

MVP

EARTH BIOGENOME PROJECT

sequencing life for the future of life

~1.5M eukaryote species

How big is 40 exabytes?

Genomics projects will generate 40 exabytes of data in the next decade.

Each shark = 100,000,000 GB of data

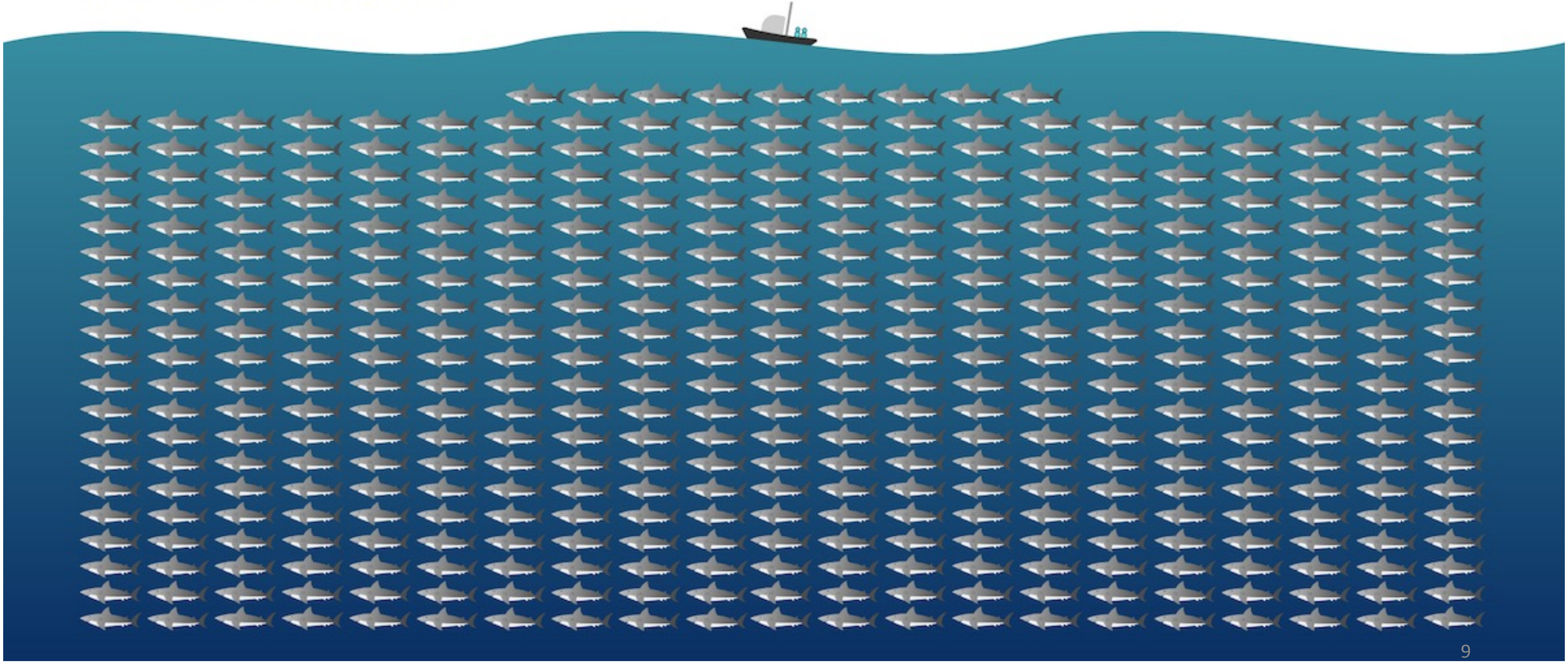


Image source: https://www.genome.gov/about-genomics/fact-sheets/Genomic-Data-Science

Sequencing & downstream analysis

- Aim: learn about the genome from the sequenced reads

Sequencing & downstream analysis

- Aim: learn about the genome from the sequenced reads
- Two major analysis pipelines:
  - Assembly
  - Alignment + Variant Calling

Genome assembly

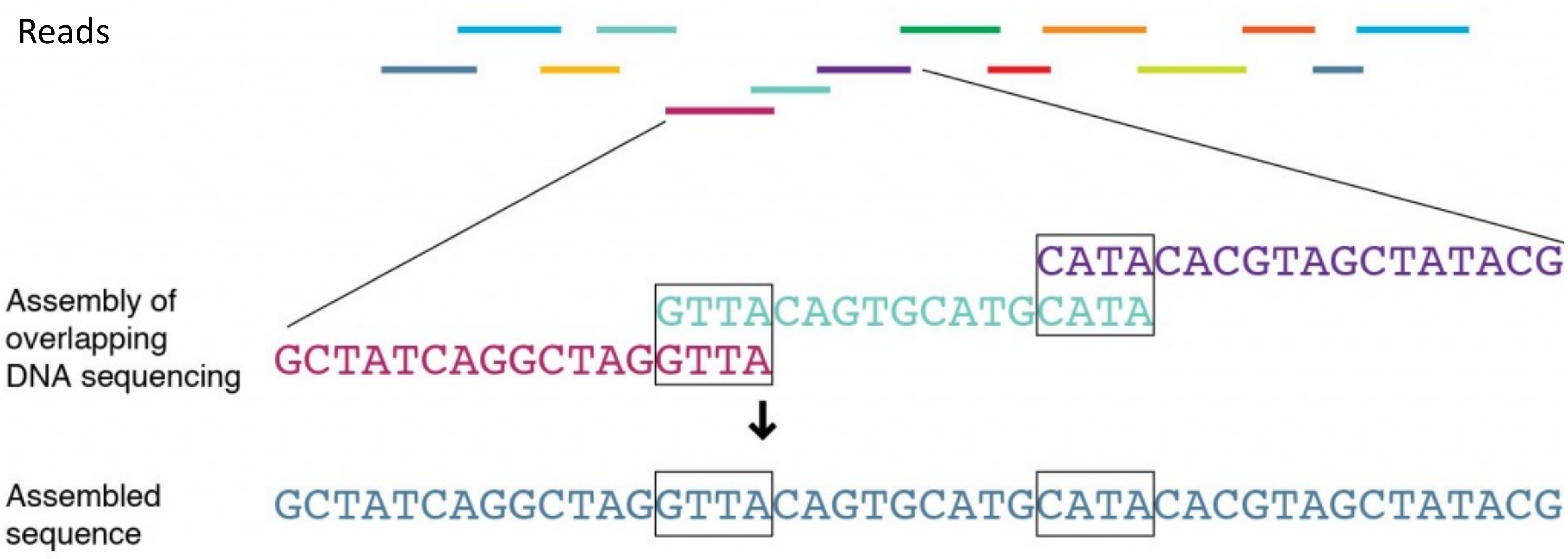

Reads
Assembly of overlapping DNA sequencing
CATACACGTAGCTATACG
GTTACAGTGCATGCATA
GCTATCAGGCTAGGTTA
Assembled sequence
GCTATCAGGCTAGGTTACAGTGCATGCATACACGTAGCTATACG

Image source: https://knowgenetics.org/whole-genome-sequencing/

Alignment and Variant Calling

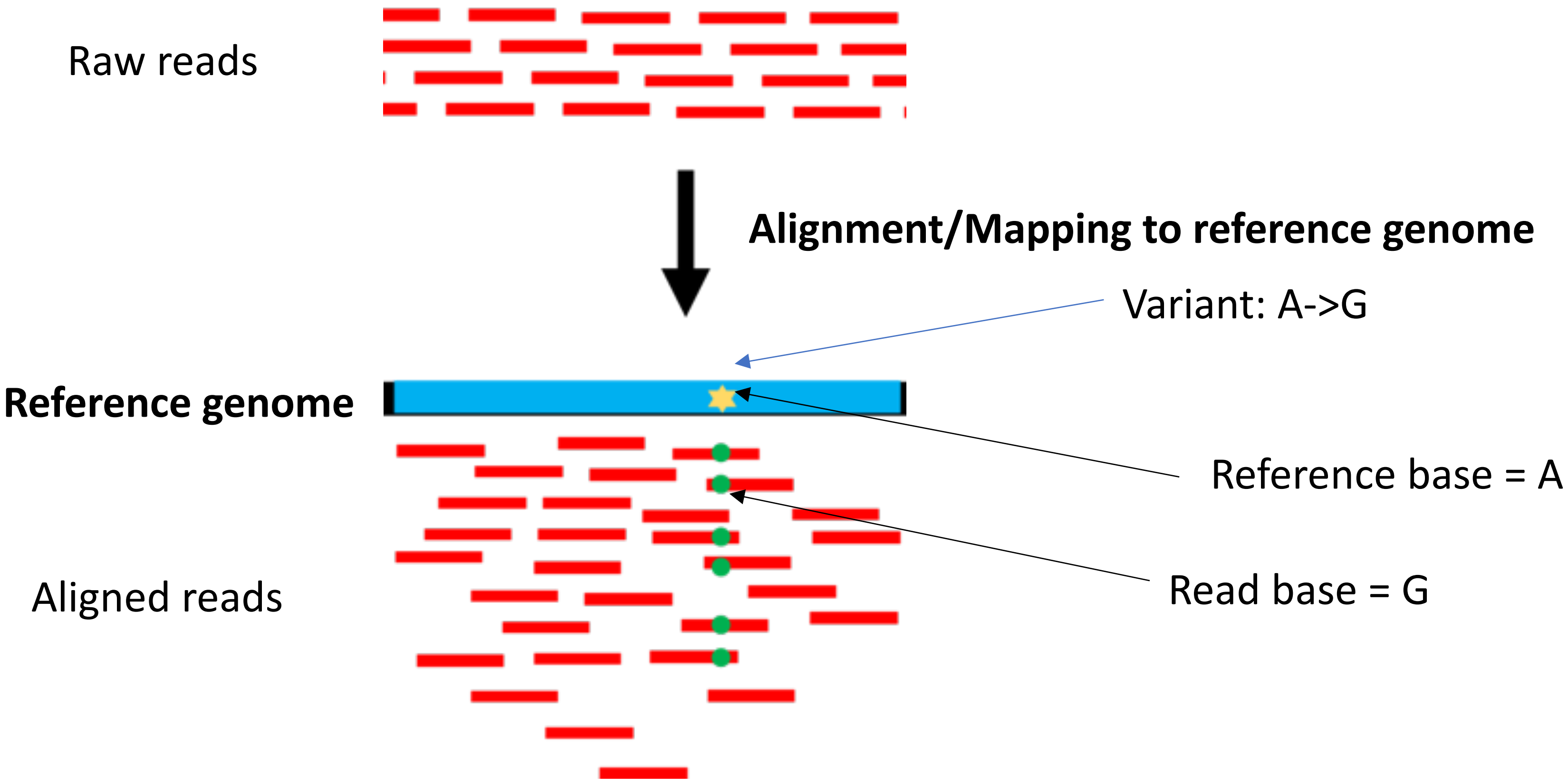


Image source: https://training.galaxyproject.org/archive/2019-02-07/topics/proteomics/tutorials/proteogenomics-dbcreation/tutorial.html/

Sequencing & downstream analysis

- Aim: learn about the genome from the sequenced reads
- Two major analysis pipelines:
  - Assembly
  - Alignment + Variant Calling
- Several sequencing methods with different features
  - We focus on two leading technologies

Sequencing technologies

Illumina NextSeq 550

Oxford Nanopore MinION

- High throughput
- Short reads
- Low error rate

- Portable and real-time
- Long reads
- Native DNA & direct RNA sequencing

We will talk about compression techniques for both technologies.

Image source:
https://www.genengnews.com/uncategorized/first-nanopore-sequencing-of-human-genome/
https://www.illumina.com/systems/sequencing-platforms/nextseq.html

Outline

- Introduction and Motivation
- SPRING: a compressor for FASTQ data
- Lossy compression of nanopore raw signal data

GitHub

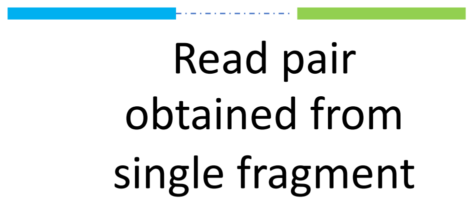
SPRING: a compressor for FASTQ data

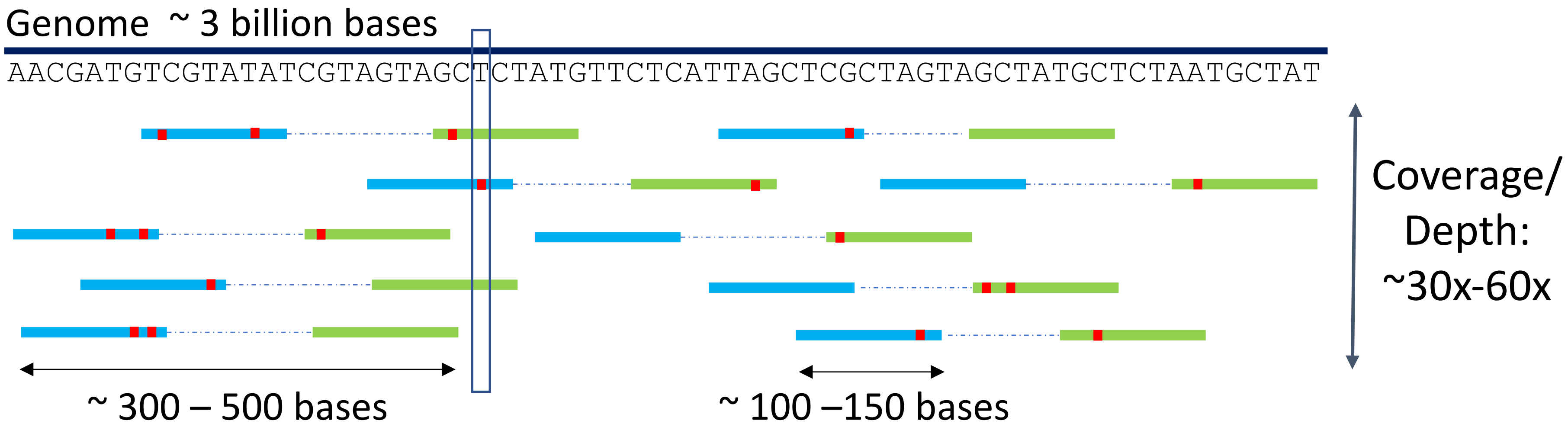
with Kedar Tatwawadi, Idoia Ochoa, Mikel Hernaez, Tsachy Weissman

Chandak, Shubham, et al. "SPRING: a next-generation compressor for FASTQ data." *Bioinformatics* 35.15 (2019): 2674-2676.

Paired-end genome sequencing

- Genome: long string of bases {A, C, G, T}
- Sequenced as noisy paired substrings (*reads*):



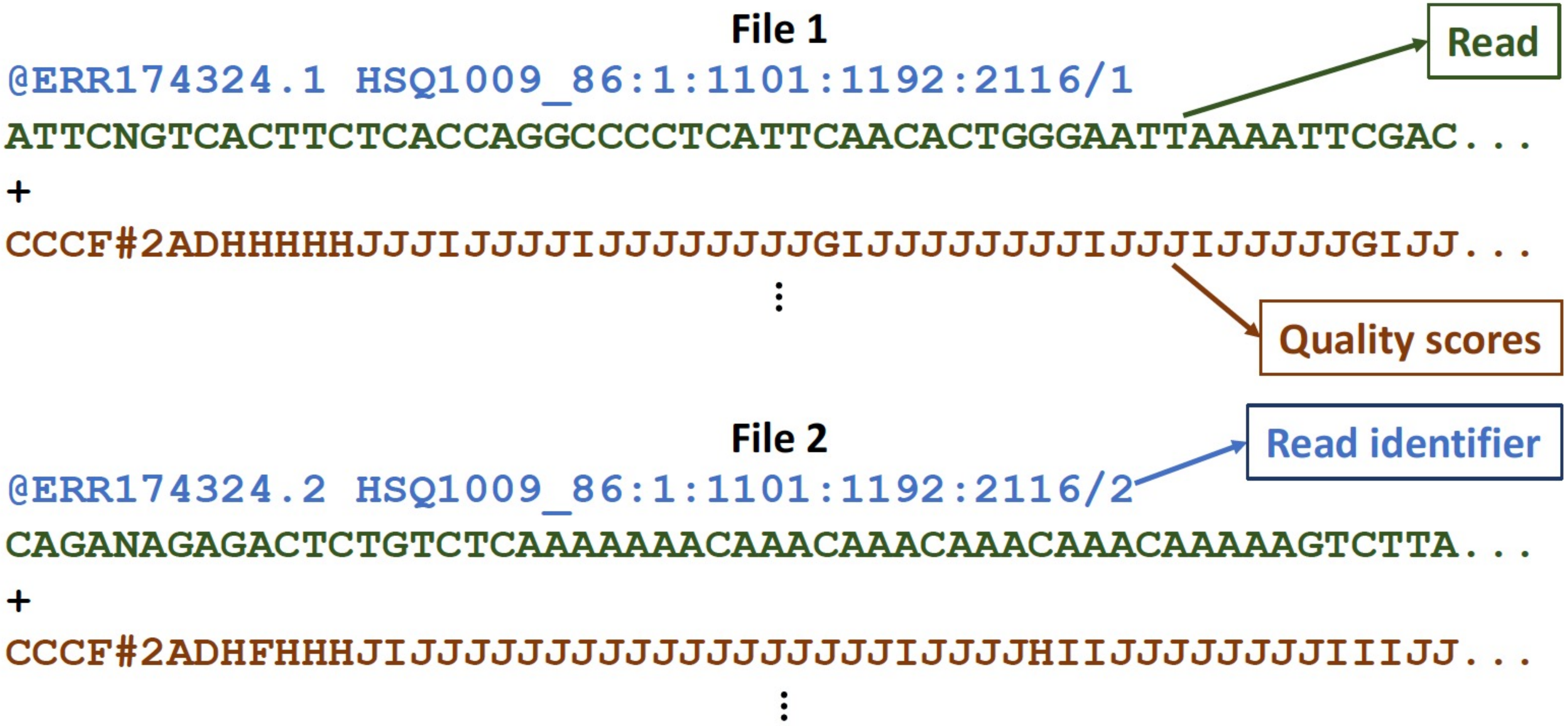


Genome ~ 3 billion bases
AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT
Coverage/ Depth: ~30x-60x
~ 300 – 500 bases
~ 100 –150 bases

Why store raw reads?

- Pipelines improve with time - need raw data for reanalysis
- For temporary storage or regulatory requirements
- When reference genome not available – e.g., de novo assembly or metagenomics

FASTQ format



File 1

```
@ERR174324.1 HSQ1009_86:1:1101:1192:2116/1
ATTCNGTCACTTCTCACCAGGCCCCTCATTCAACACTGGGAATTAAAATTCGAC...
+
CCCF#2ADHHHHHJJJIJJJJIJJJJJJJJGIJJJJJJJJIJJJIJJJJJGIJJ...
⋮
```

File 2

```
@ERR174324.2 HSQ1009_86:1:1101:1192:2116/2
CAGANAGAGACTCTGTCTCAAAAAAACAAACAAACAAACAAACAAAAAGTCTTA...
+
CCCF#2ADHFHHHJIJJJJJJJJJJJJJJJJIJJJJHIIJJJJJJJJIIIJJ...
⋮
```

Read

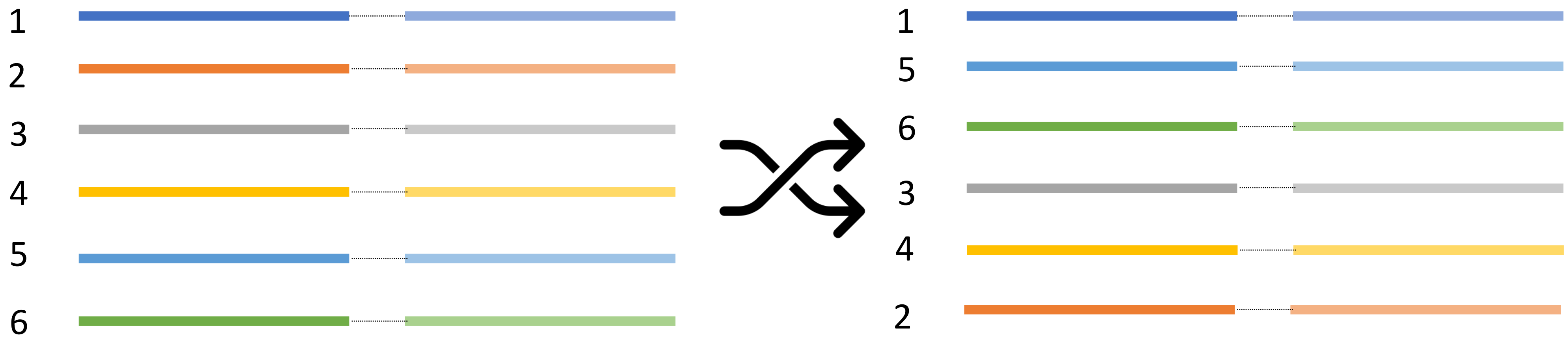
Quality scores

Read identifier

We'll mostly focus on **reads** in this talk.

Read compression

- For a typical 25x human dataset:
  - Uncompressed: 79 GB (1 byte/base)
  - Gzip: ~20 GB (2 bits/base) – still far from optimal
- Order of read pairs in FASTQ irrelevant – can this help?



Original order in FASTQ

New order (preserving read pairs)

Read compression results

Compressor	25x human
Uncompressed	79 GB
Gzip	~20 GB

Illumina NovaSeq human whole genome data, 150bp x 2

Read compression results

Compressor	25x human
Uncompressed	79 GB
Gzip	~20 GB
FaStore (allow reordering)	6 GB

Łukasz Roguski, Idoia Ochoa, Mikel Hernaez, Sebastian Deorowicz; FaStore: a space-saving solution for raw sequencing data, *Bioinformatics*, Volume 34, Issue 16, 15 August 2018, Pages 2748–2756

Read compression results

Compressor	25x human
Uncompressed	79 GB
Gzip	~20 GB
FaStore (allow reordering)	6 GB
SPRING (no reordering)	**3 GB**
SPRING (allow reordering)	**2 GB**

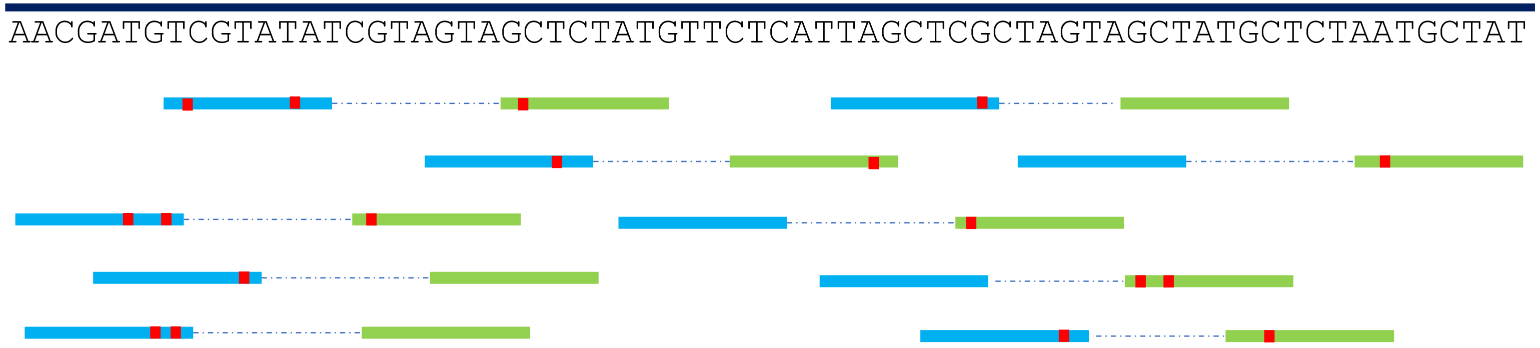
Łukasz Roguski, Idoia Ochoa, Mikel Hernaez, Sebastian Deorowicz; FaStore: a space-saving solution for raw sequencing data, *Bioinformatics*, Volume 34, Issue 16, 15 August 2018, Pages 2748–2756

Read compression results

Compressor	25x human	100x human
Uncompressed	79 GB	319 GB
Gzip	~20 GB	~80 GB
FaStore (allow reordering)	6 GB	13.7 GB
SPRING (no reordering)	**3 GB**	**10 GB**
SPRING (allow reordering)	**2 GB**	**5.7 GB**

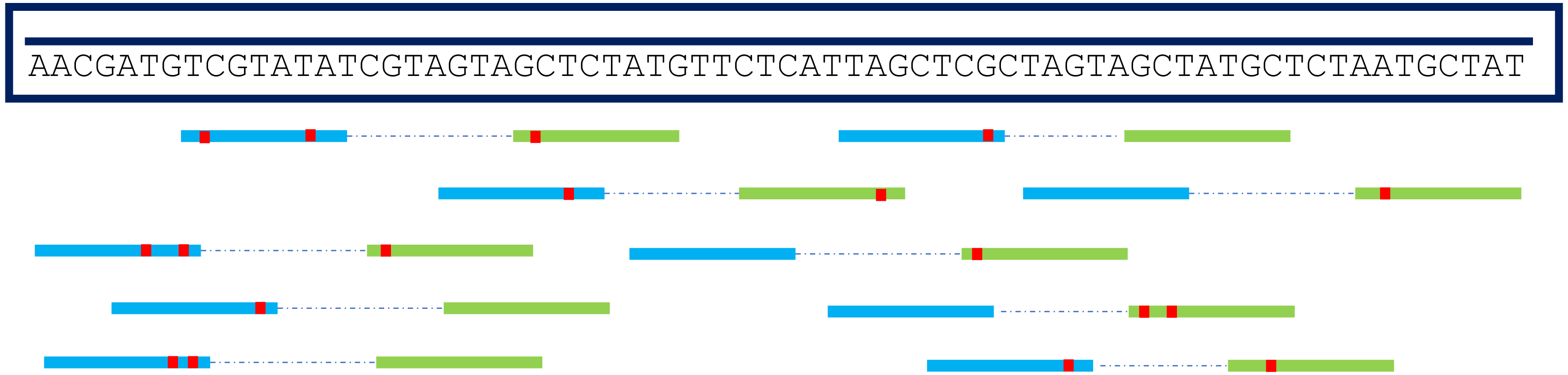
Łukasz Roguski, Idoia Ochoa, Mikel Hernaez, Sebastian Deorowicz; FaStore: a space-saving solution for raw sequencing data, *Bioinformatics*, Volume 34, Issue 16, 15 August 2018, Pages 2748–2756

Key idea


AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

- Storing reads equivalent to

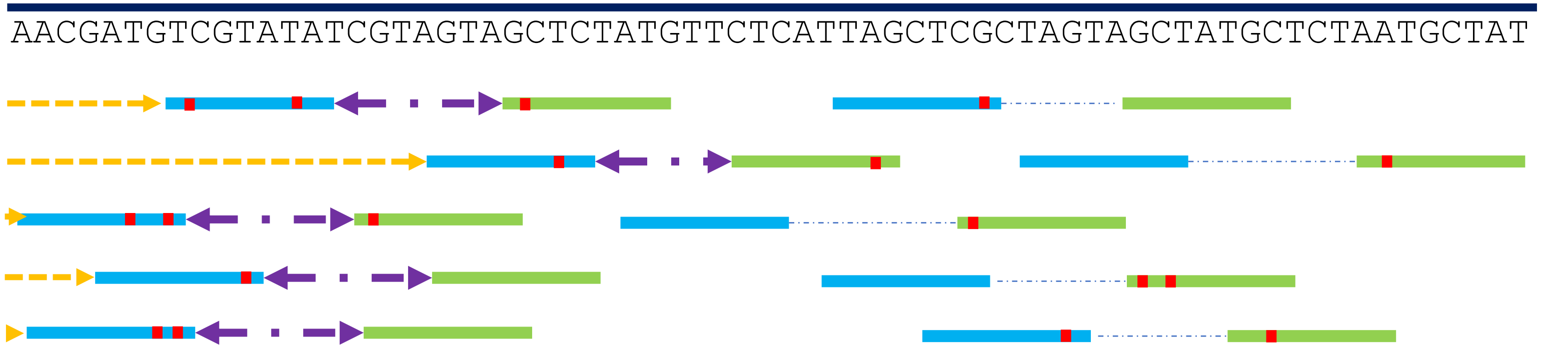
Key idea



AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

- Storing reads equivalent to
  - Store genome

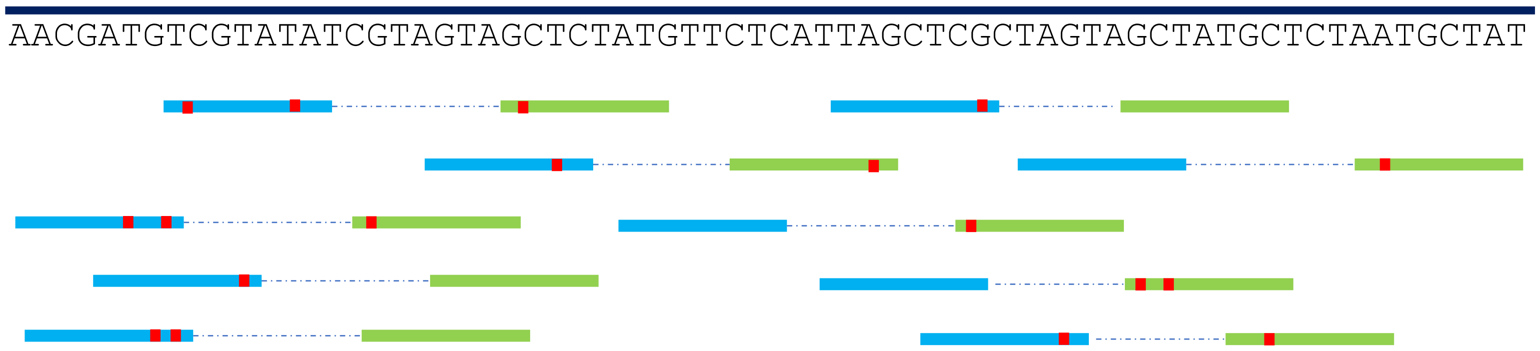
Key idea



AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

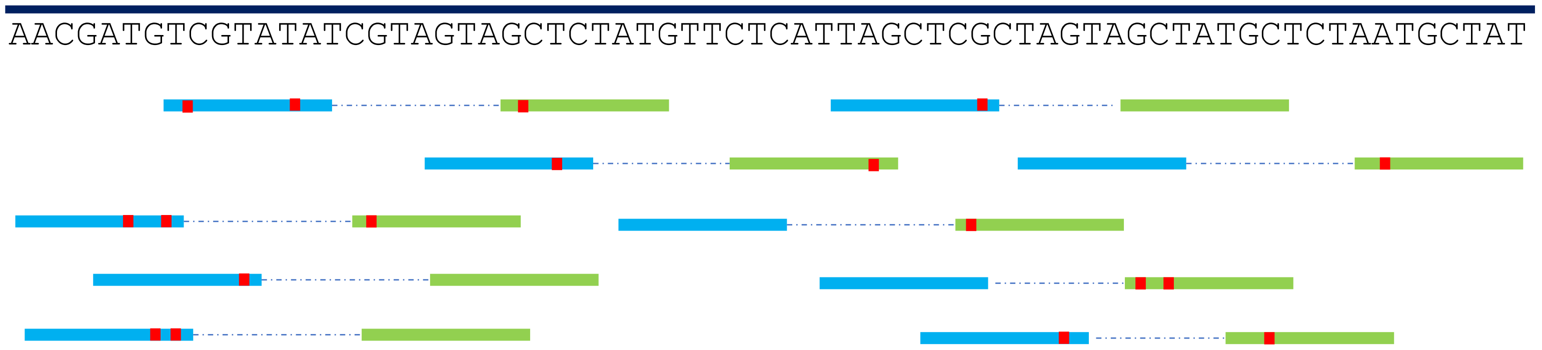
- Storing reads equivalent to
  - Store genome
  - Store read positions in genome (+ gap between paired reads)

Key idea


AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

- Storing reads equivalent to
  - Store genome
  - Store read positions in genome (+ gap between paired reads)
  - Store noise in reads

Key idea

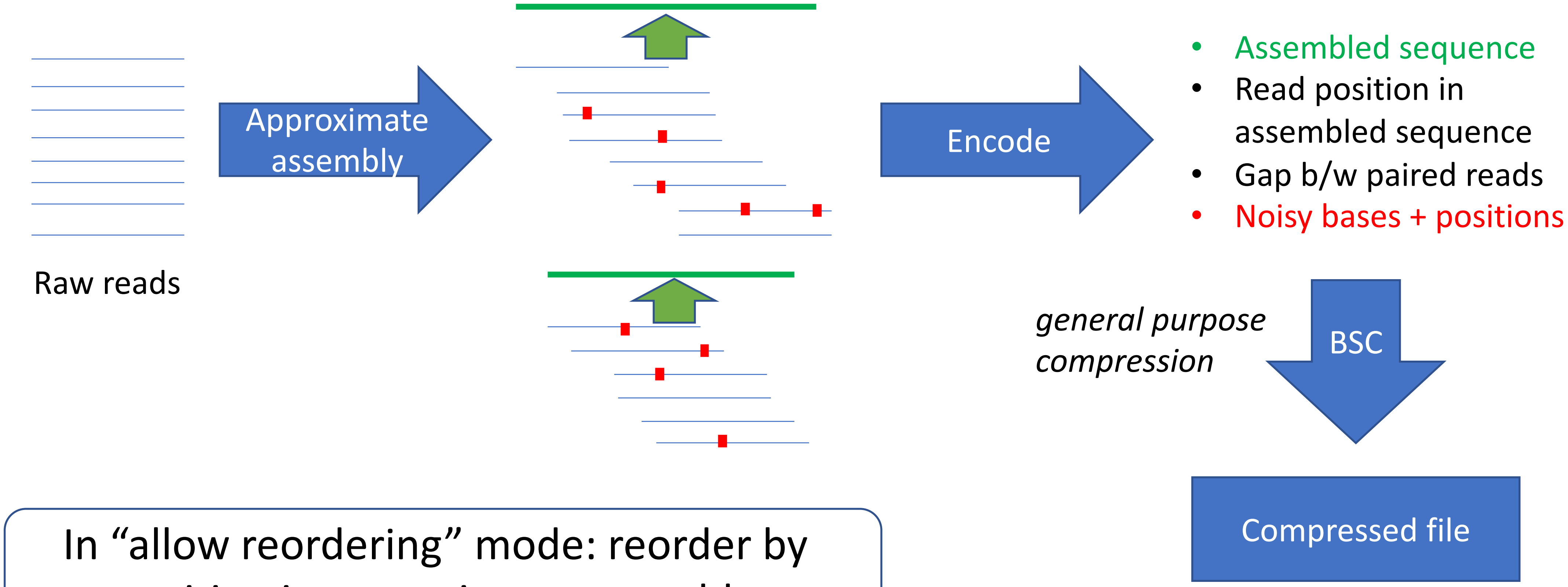

AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

- Storing reads equivalent to
  - Store genome
  - Store read positions in genome (+ gap between paired reads)
  - Store noise in reads
- Theoretical calculations show this outperforms previous compressors

Key idea

- But... How to get the genome from the reads?
- Genome assembly too expensive - big challenges:
  - resolve repeats
  - get very long pieces of genome from shorter assemblies
- Solution: Don’t need perfect assembly for compression!

SPRING workflow



Raw reads

Approximate assembly

Encode

- Assembled sequence
- Read position in assembled sequence
- Gap b/w paired reads
- Noisy bases + positions

general purpose compression

BSC

Compressed file

https://github.com/IlyaGrebnov/libbsc

In “allow reordering” mode: reorder by position in approximate assembly

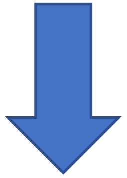
Quality and read identifier compression

- Quality – use general purpose compressor BSC (optionally apply quantization)

```
CCCF#2ADHHHHHJJJI -> BSC -> compressed bitstream
```

- Read identifier – split into tokens and use arithmetic coding[1]

```
@ERR174324.1 HSQ1009_86:1:1101:1192:2116/1
```

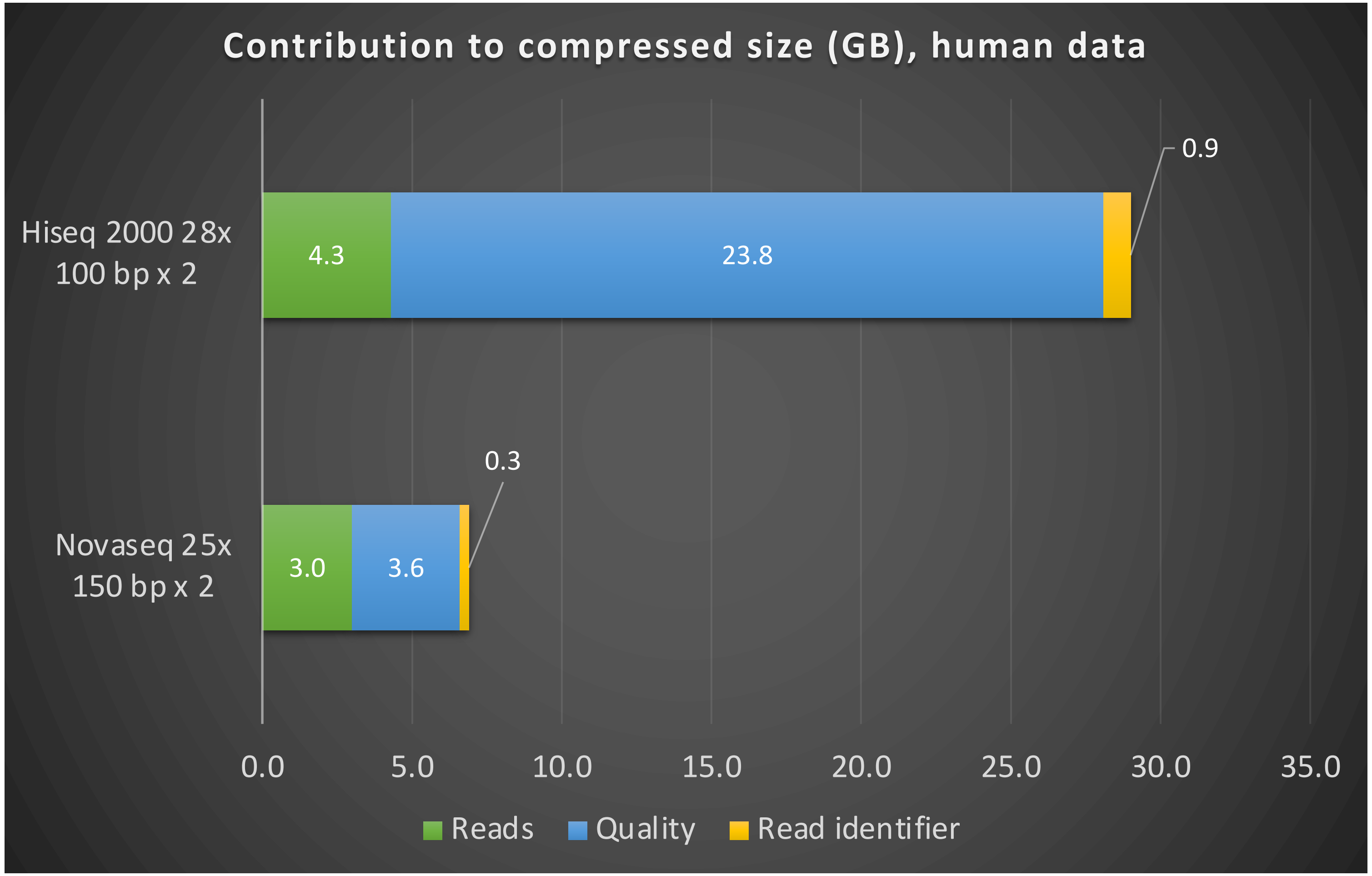


Tokenization

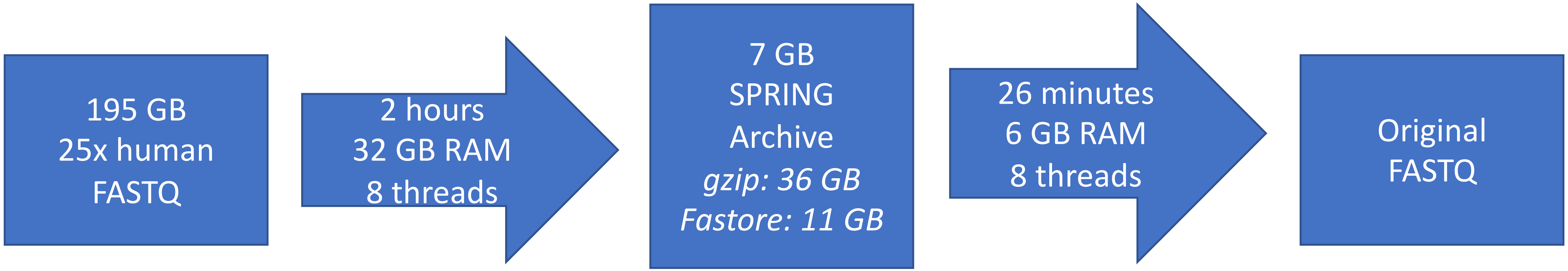
```
@ERR174324.1 HSQ1009_86:1:1101:1192:2116/1
```

[1] Bonfield, James K., and Matthew V. Mahoney. "Compression of FASTQ and SAM format sequencing data." *PloS one* 8.3 (2013): e59190.

Quality and read identifier compression



SPRING as a practical tool



195 GB
25x human
FASTQ

2 hours
32 GB RAM
8 threads

7 GB
SPRING
Archive
gzip: 36 GB
Fastore: 11 GB

26 minutes
6 GB RAM
8 threads

Original
FASTQ

- Easy to use with support for:
  - Lossless and lossy modes
  - Variable length reads, long reads, etc.
  - Compressed in blocks to allow partial/streaming decompression
  - Scalable to large datasets
  - Gzipped I/O
- Github: https://github.com/shubhamchandak94/SPRING/

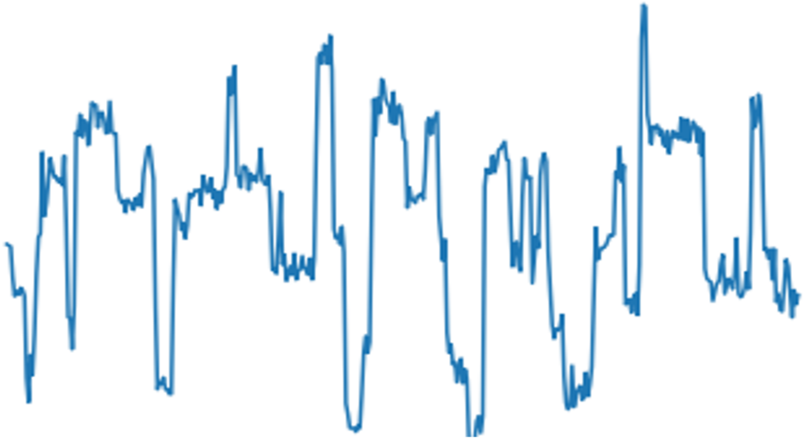
Impact and future directions

- SPRING downloaded more than 1,500 times from Conda
- Interest from industry and medical institutions in improving and adopting SPRING
- Recent compressors like PGRC[1] use similar paradigm and lossless/lossy modes, focusing on improving the approximate assembly
- SPRING is part of genie (open-source MPEG-G codec – *under development*): https://github.com/mitogen/genie
- Ongoing work on building specialized read compressor for long reads with insertion and deletion errors

[1] Kowalski, Tomasz, and Szymon Piotr Grabowski. "Engineering the Compression of Sequencing Reads." bioRxiv (2020).

Outline

- Introduction and Motivation
- SPRING: a compressor for FASTQ data
- Lossy compression of nanopore raw signal data

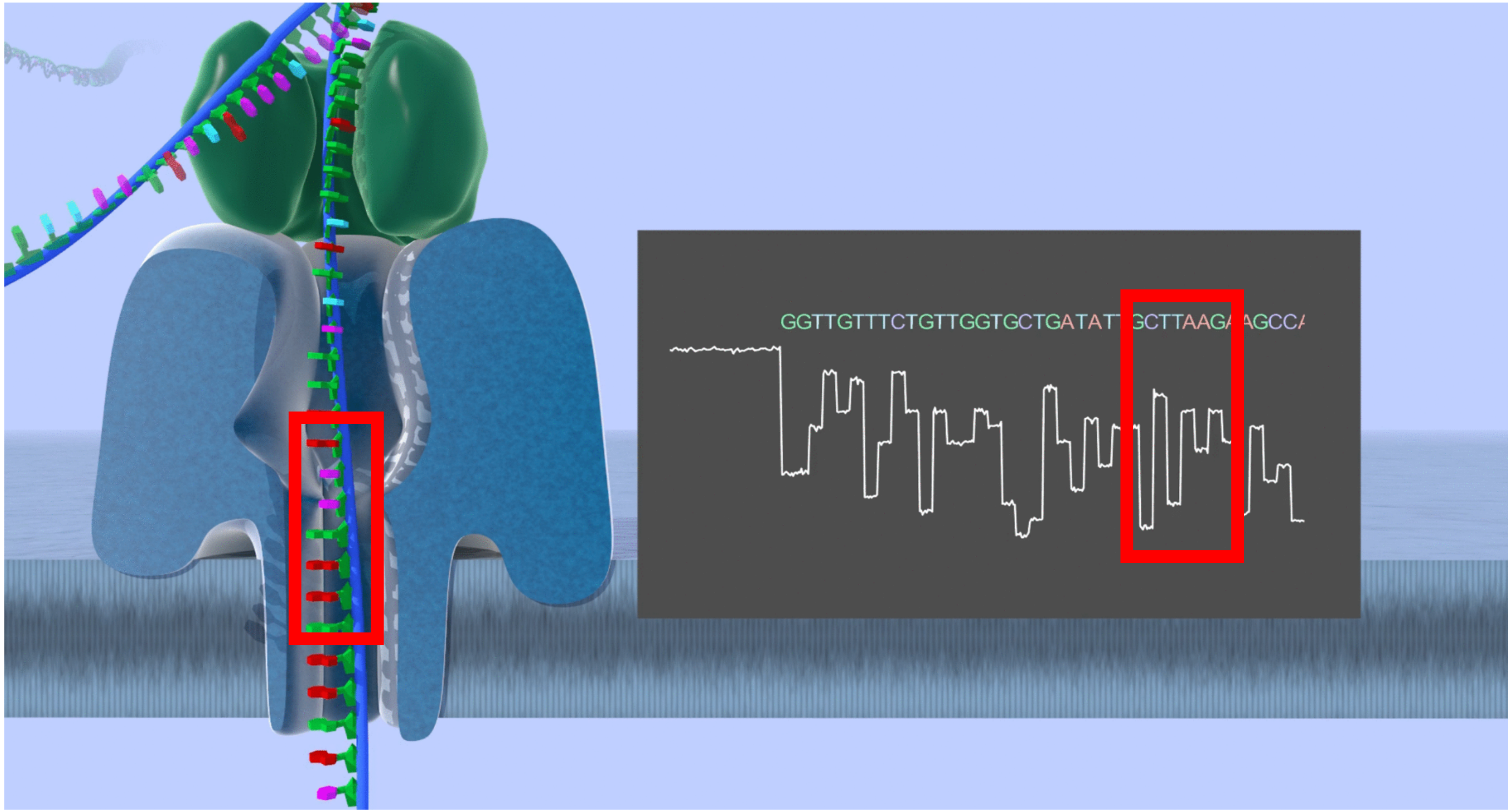


Lossy compression of nanopore raw signal data

with Kedar Tatwawadi, Srivatsan Sridhar, Tsachy Weissman

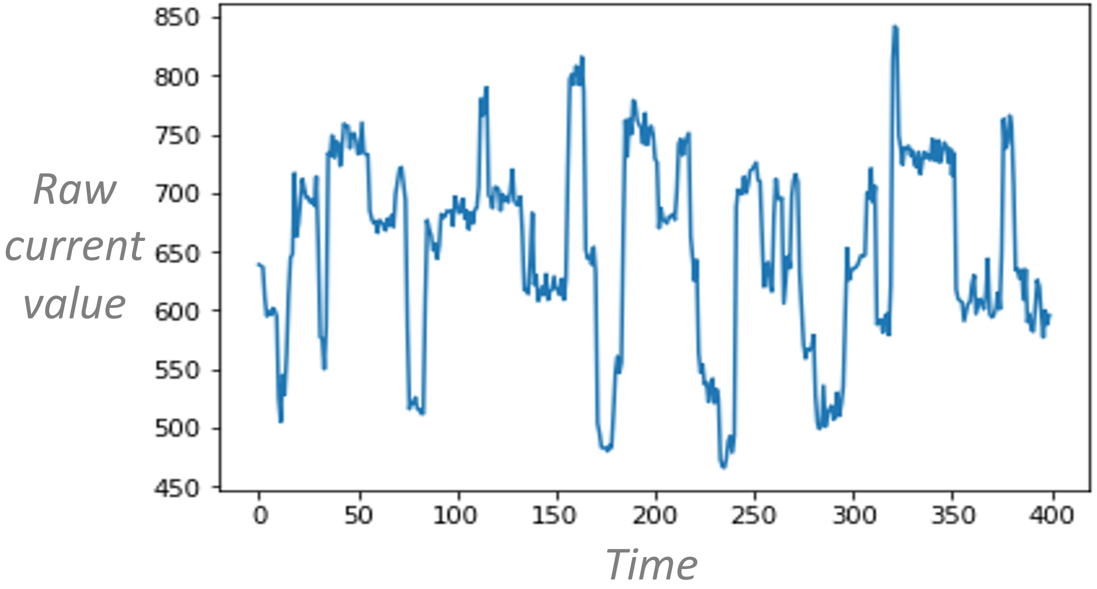
Chandak, Shubham, et al. "Impact of lossy compression of nanopore raw signal data on basecalling and consensus accuracy." *Bioinformatics,* Volume 36, Issue 22-23, 1 December 2020, Pages 5313–5321.

Nanopore Sequencing



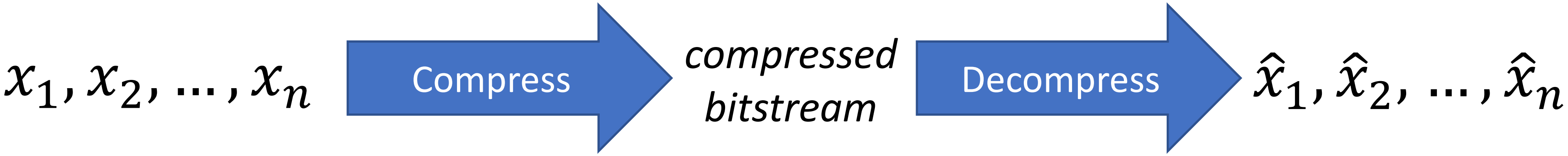
Source: https://youtu.be/E9-Rm5AoZGw

Raw signal compression



- HDF5 file (".fast5")
- ~18 bytes/base uncompressed
- VBZ: state-of-the-art lossless compressor
  - Variable byte integer encoding + zstd
  - 60% size reduction (30% over Gzip)
  - Still require **1 TB for 30x human whole genome** data
- Often need to retain raw intermediate data for (re)analysis
- Lossy compression?

Lossy time-series compression (LFZip[1]/SZ[2])



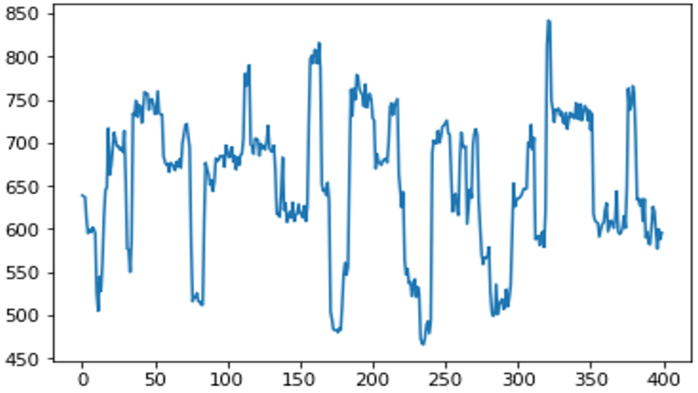
$x_1, x_2, \ldots, x_n$

Compress

compressed bitstream

Decompress

$\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$



Error constraint: $\max\limits_{i=1,\ldots,n} |x_i - \hat{x}_i| \le \epsilon$

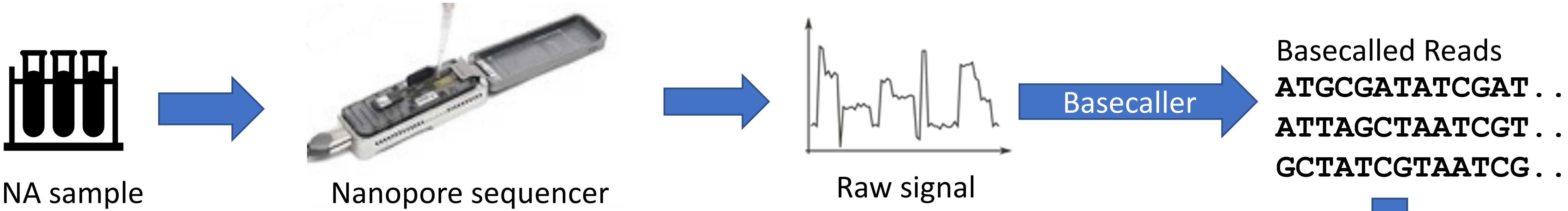
Maximum absolute error

But the actual loss metric is the downstream accuracy

[1] Chandak, S., Tatwawadi, K., Wen, C., Wang, L., Ojea, J. A., & Weissman, T. (2020, March). LFZip: Lossy compression of multivariate floating-point time series data via improved prediction. In *2020 Data Compression Conference (DCC)* (pp. 342-351). IEEE.

[2] Liang, Xin, et al. "An efficient transformation scheme for lossy data compression with point-wise relative error bound." *2018 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2018.

Basecalling and consensus



```
ATGCGATATCGAT..
ATTAGCTAATCGT..
GCTATCGTAATCG..
```

Consensus

```
ATGCGATAT-CGTT
ATGC-ATAT-CGAT
ATCCGATATACGAT
```

Consensus sequence: ATGCGATAT-CGAT

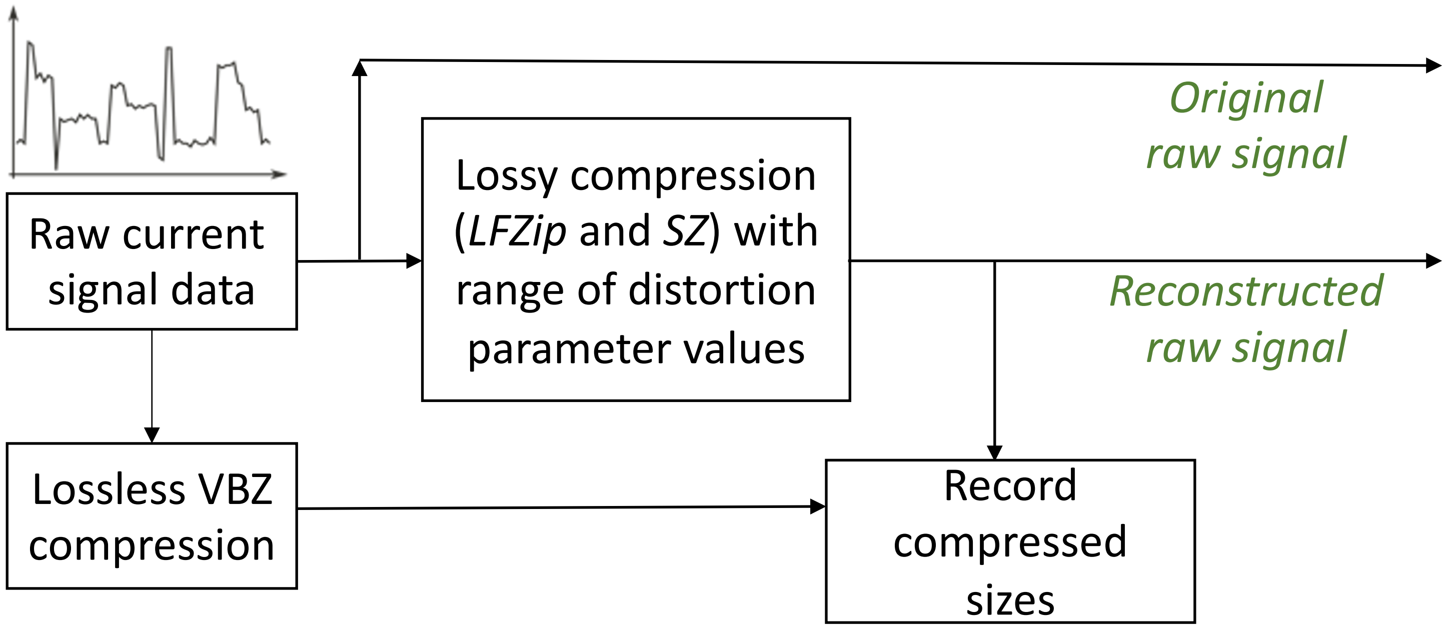
Basecalling error

Ground truth	TTGCGTATGCG--TTATCTGCTGA
Basecall	ATGC-TATGCGGCTTAGCTGC--A

Consensus error

Ground truth	TTGCGTATGCGTTATCTGCTGA
Consensus	TTGCGTATACGTTATCT-CTGA
Read 1	ATGC-TATACGGCATCG-CTGA
Read 2	TTGCGTATACGTTAACT-CTGA
Read 3	T-GCGTATACTTTATCTGCTCA

Evaluation pipeline: compression



Raw current signal data

Lossy compression (*LFZip* and *SZ*) with range of distortion parameter values

Original raw signal

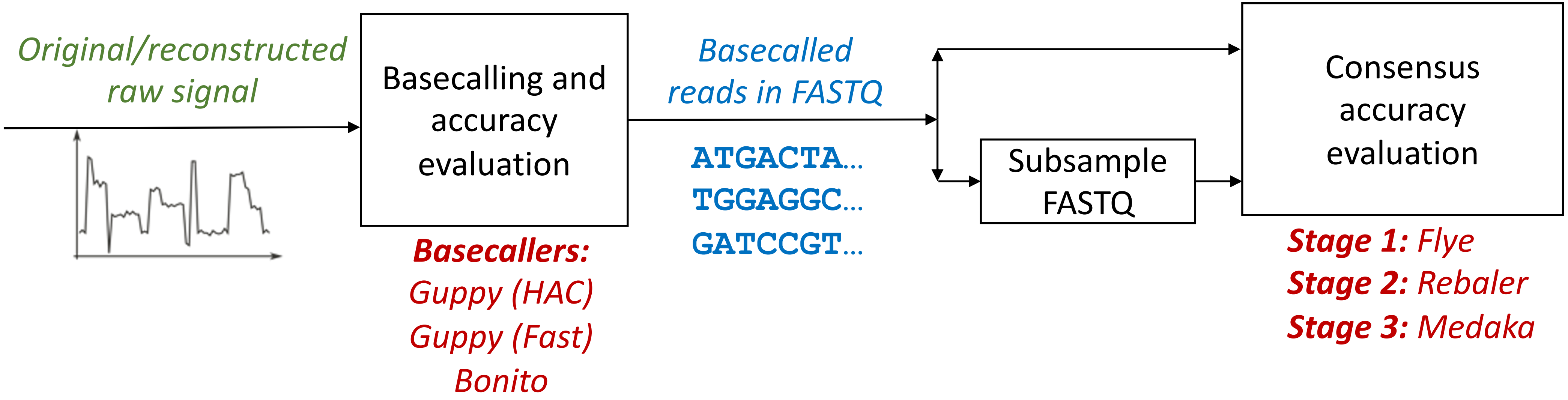
Reconstructed raw signal

Lossless VBZ compression

Record compressed sizes

Lossless and lossy compression of raw signal data

Evaluation pipeline: downstream accuracy



Original/reconstructed raw signal

Basecalling and accuracy evaluation

Basecallers:
Guppy (HAC)
Guppy (Fast)
Bonito

Basecalled reads in FASTQ

ATGACTA…
TGGAGGC…
GATCCGT…

Subsample FASTQ

Consensus accuracy evaluation

Stage 1: *Flye*
Stage 2: *Rebaler*
Stage 3: *Medaka*

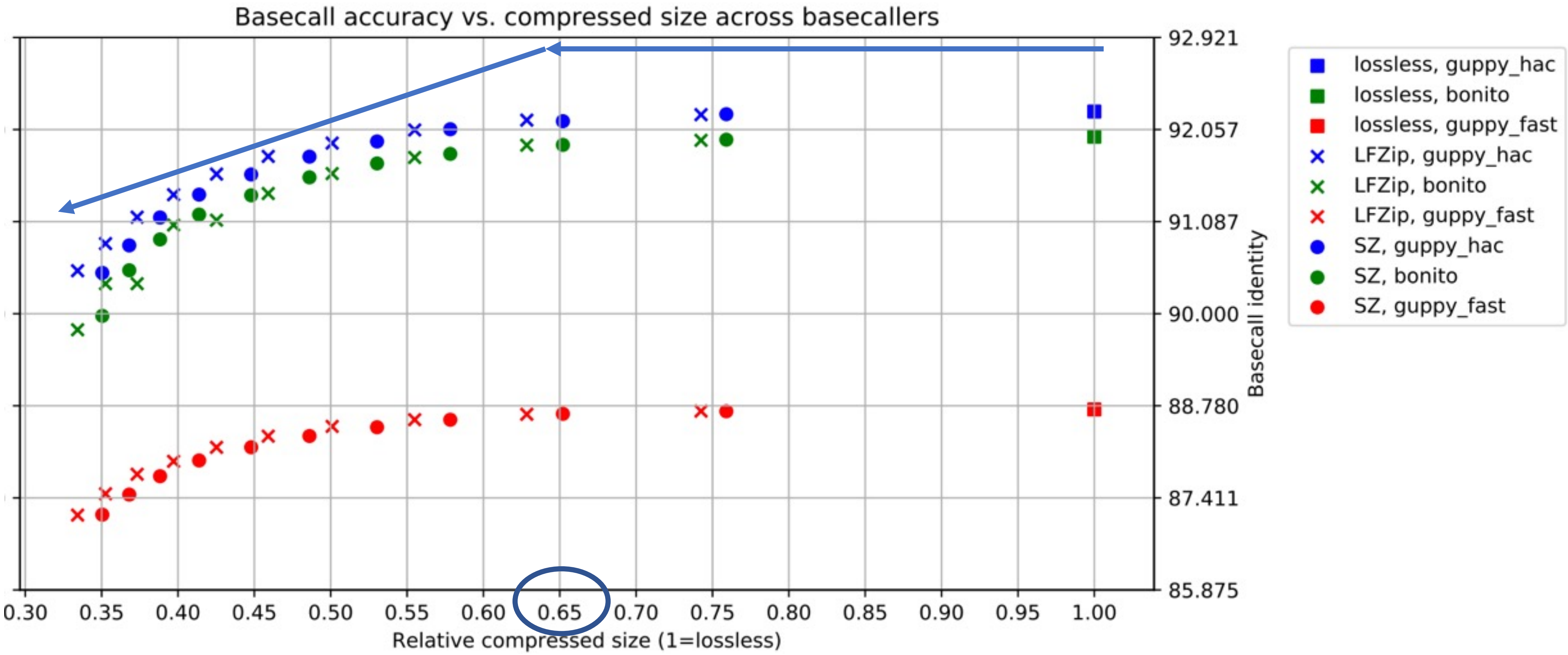
Basecalling and consensus accuracy analysis

Note: Attempt to “future-proof” by testing various tools/use cases

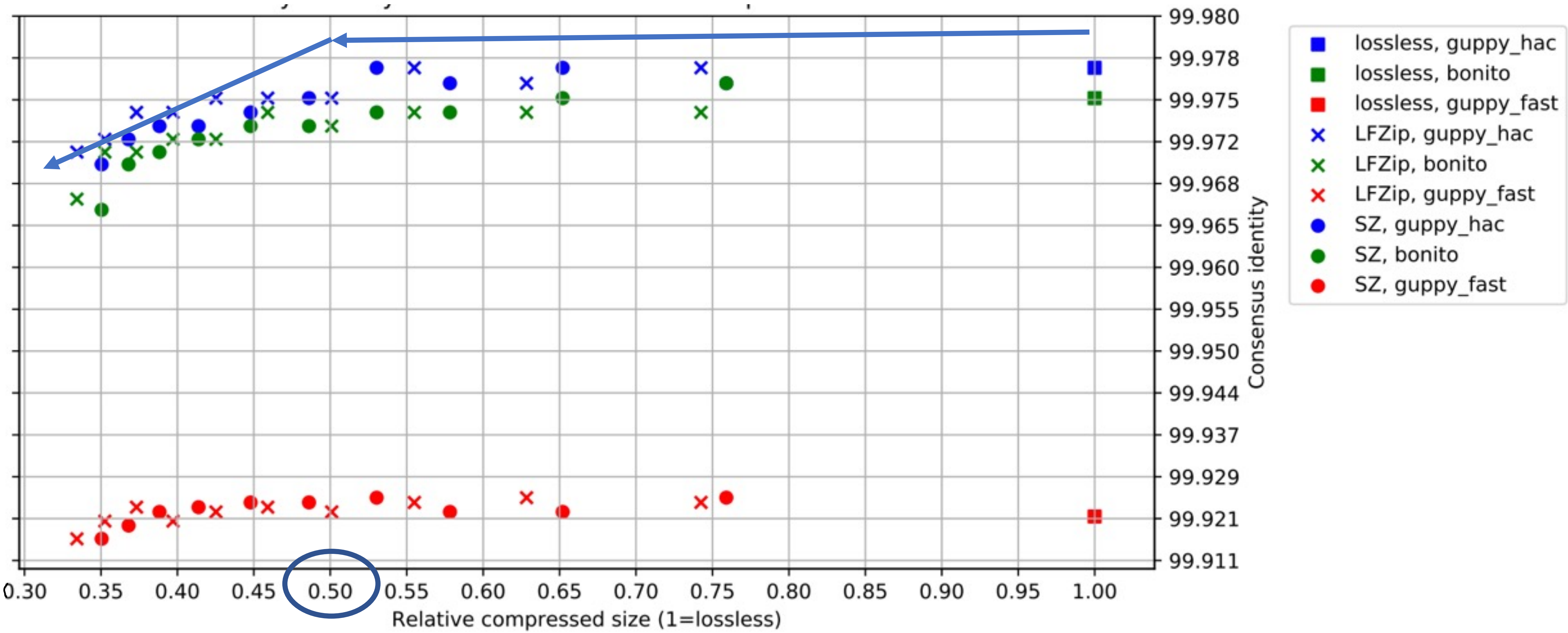
Evaluation pipeline

- Human and 3 bacterial datasets for basecalling accuracy
  - Use benchmark datasets with known ground-truth genome
- Bacterial datasets for consensus accuracy
  - Tested at multiple subsampling levels
- Tested all combinations of
  - Dataset
  - Compressor
  - Downstream tool
- Evaluated methylation accuracy and homopolymer accuracy
  - not discussed in this talk

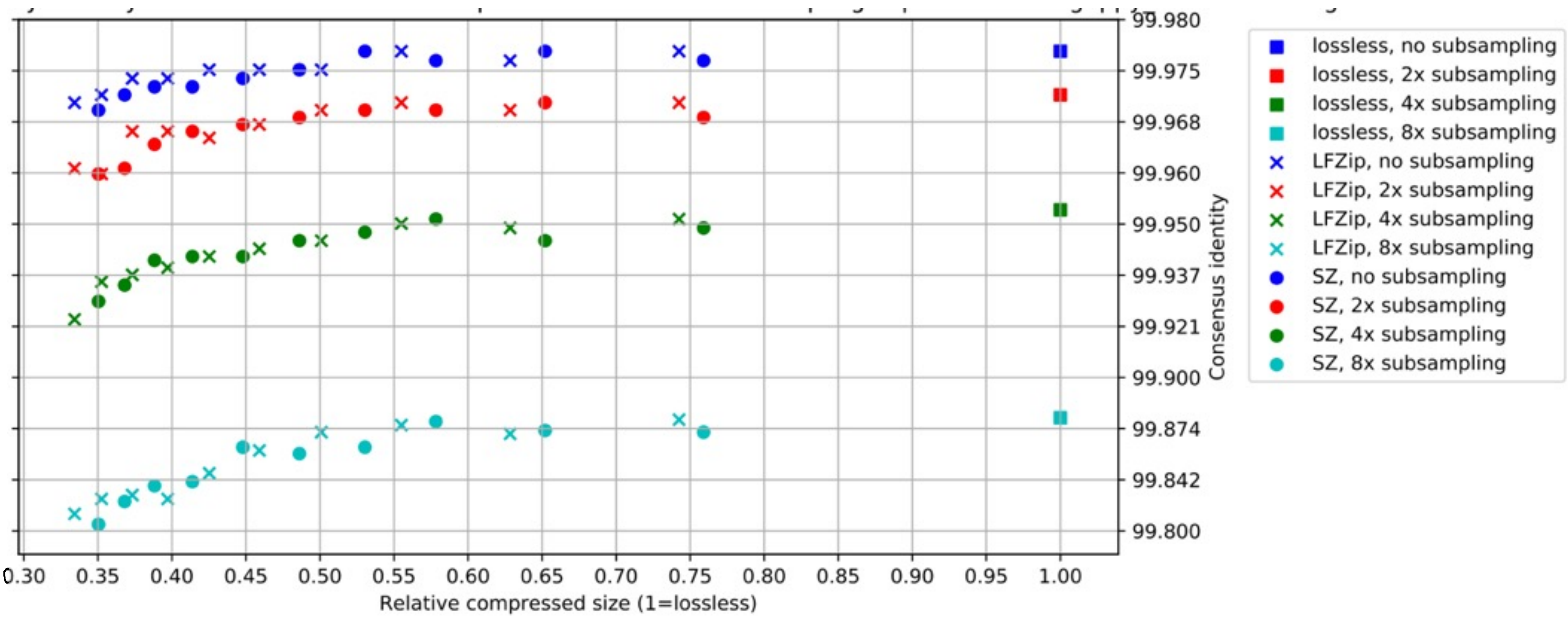
Basecalling accuracy



Consensus accuracy



Subsampling experiments



Summary

- Achieve 35-50% reduction over best lossless compression
    - Negligible loss in accuracy
    - Consistent observations across datasets, coverage, downstream tools
- Highly practical
    - LFZip simply reduces the data resolution
    - Can be adopted at the nanopore sequencer device itself
- This is the first work on the topic, and much remains to be explored:
    - Specialized lossy compressors for this data, retraining of downstream models
    - Further evaluation on human data with improved benchmark datasets
- Evaluation scripts, data, plots: https://github.com/shubhamchandak94/lossy_compression_evaluation

Publications: genomic data compression

- **S. Chandak**, K. Tatwawadi, S. Sridhar and T. Weissman; Impact of lossy compression of nanopore raw signal data on basecall and consensus accuracy, ***Bioinformatics 2020.***
- **S. Chandak**, K. Tatwawadi, I. Ochoa, M. Hernaez and T. Weissman; SPRING: A next-generation compressor for FASTQ data, ***Bioinformatics 2019.***
- **S. Chandak**, K. Tatwawadi and T. Weissman; Compression of genomic sequencing reads via hash-based reordering: algorithm and analysis, ***Bioinformatics 2018.***

Publications: Storage in DNA

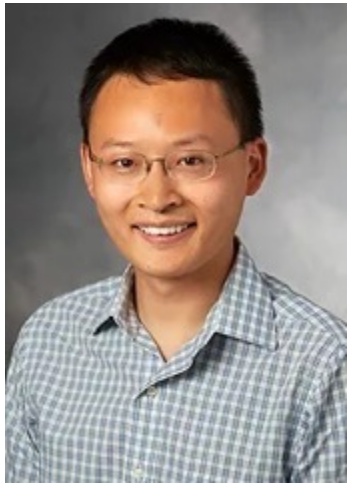
- Journal paper ***in preparation.***
- **S. Chandak**, J. Neu, K. Tatwawadi, J. Mardia, B. Lau, M. Kubit, R. Hulett, P. Griffin, M. Wootters, T. Weissman and H. Ji; “Overcoming high nanopore basecaller error rates for DNA storage via basecaller-decoder integration and convolutional codes,” ***ICASSP 2020.***
- **S. Chandak**, K. Tatwawadi, B. Lau, J. Mardia, M. Kubit, J. Neu, P. Griffin, M. Wootters, T. Weissman and H. Ji; “Improved read/write cost tradeoff in DNA-based data storage using LDPC codes,” ***Allerton 2019.***

Publications: time series & multimedia compression

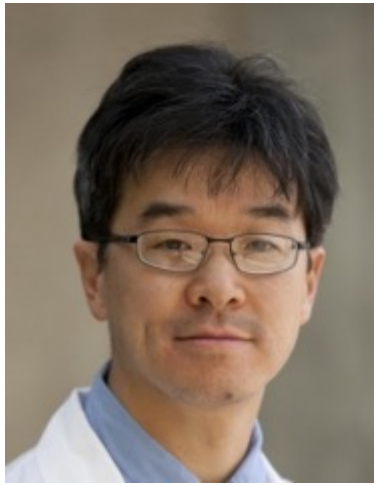
- R. Prabhakar, **S. Chandak**, C. Chiu, R. Liang, H. Nguyen, K. Tatwawadi and T. Weissman; "Reducing latency and bandwidth for video streaming using keypoint extraction and digital puppetry," ***DCC 2021.***
- **S. Chandak**, K. Tatwawadi, C. Wen, L. Wang, J.A. Ojea and T. Weissman; "LFZip: Lossy compression of multivariate floating-point time series data via improved prediction," ***DCC 2020.***
- A. Bhown, S. Mukherjee, S. Yang, **S. Chandak**, I. Fischer-Hwang, K. Tatwawadi and T. Weissman; "Humans are still the best lossy image compressors," ***DCC 2019.***

Acknowledgements

Exam committee



James Zou

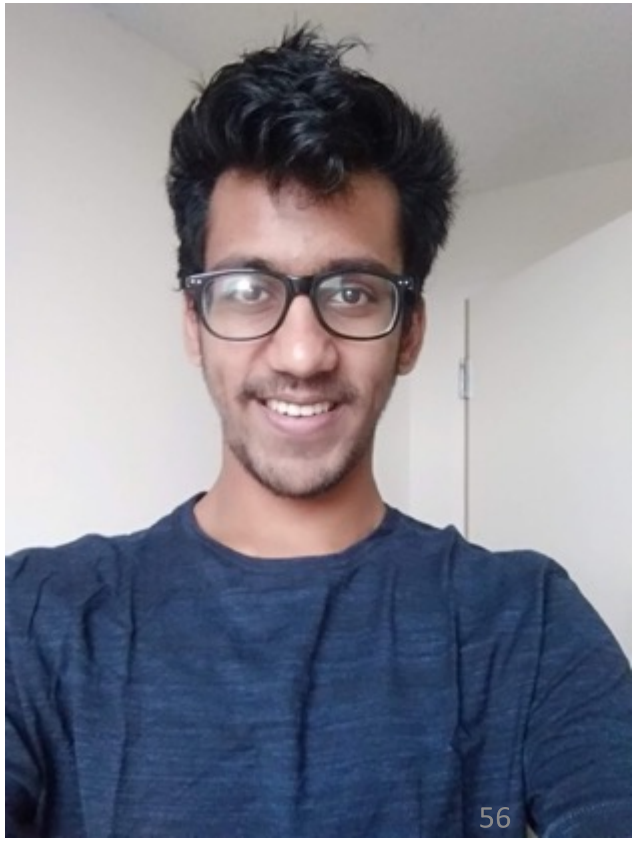


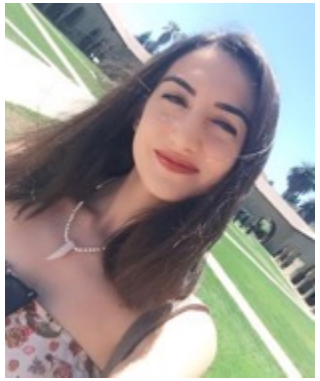
Hanlee Ji

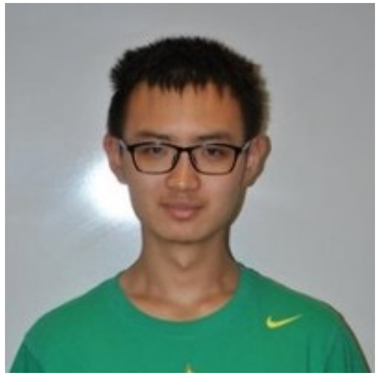
Mary Wootters

Ayfer Özgür

Tsachy Weissman







PHILIPS
Cam dge In vation

EE@Stanford

- Doug
- Suzanne
- Meo
- Rachel
- Marisa
- Denise
- and many more behind the scenes!

Thank you!