SPRING: a next-generation compressor for FASTQ data

Shubham Chandak

Stanford University

Stanford Compression Workshop 2019

Joint work with

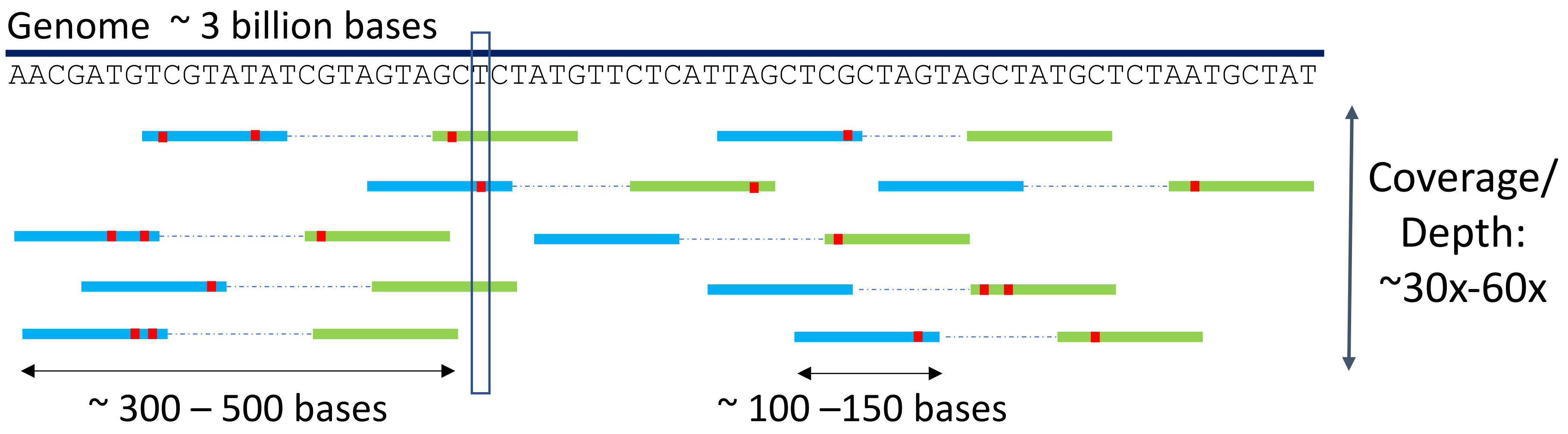
- Kedar Tatwawadi, Stanford University
- Idoia Ochoa, UIUC
- Mikel Hernaez, UIUC
- Tsachy Weissman, Stanford University

Outline

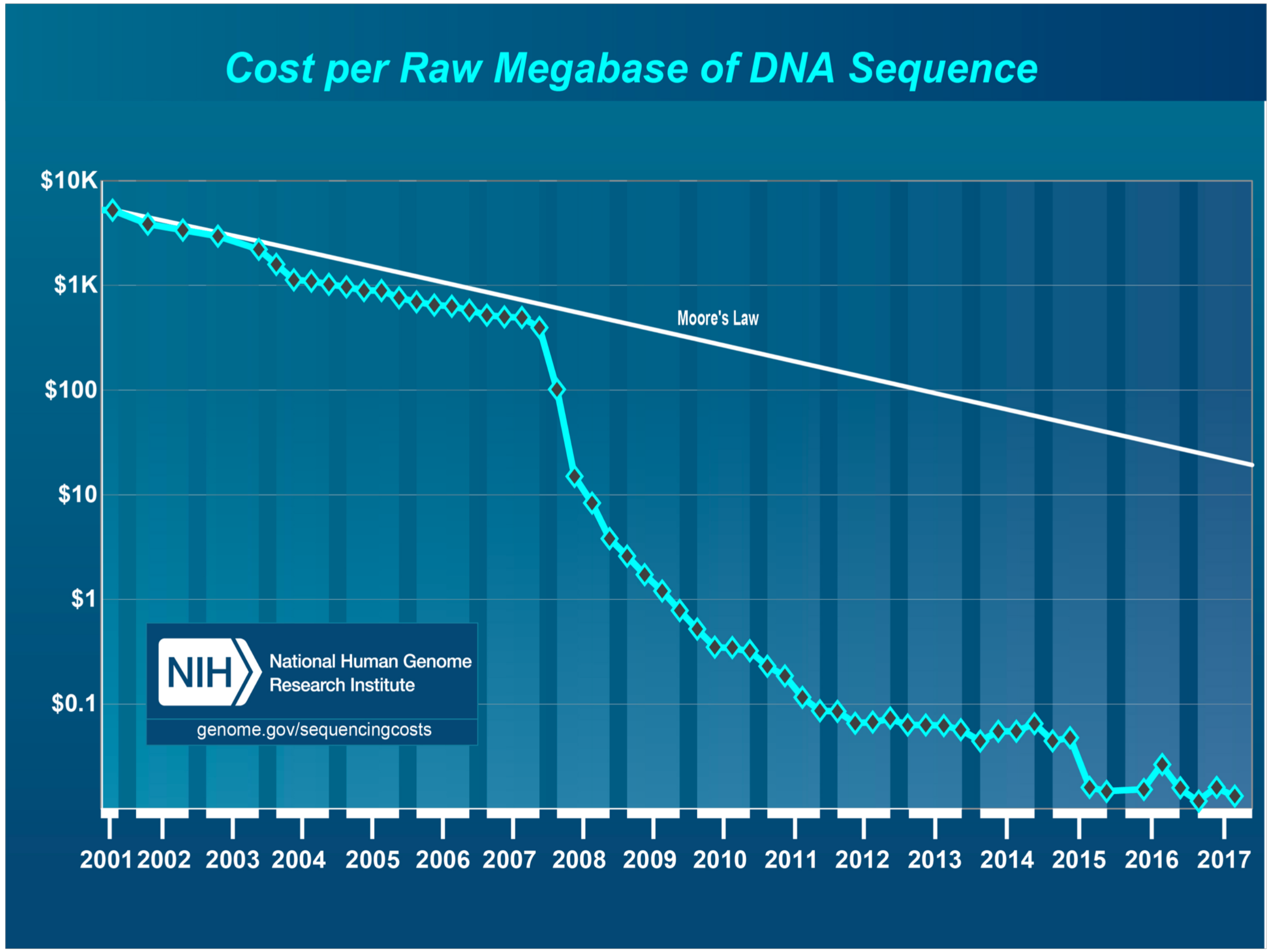
- Intro to genome sequencing
- FASTQ format and compression results
- SPRING algorithm
- SPRING as a practical tool

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 compression?



Why compression?

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

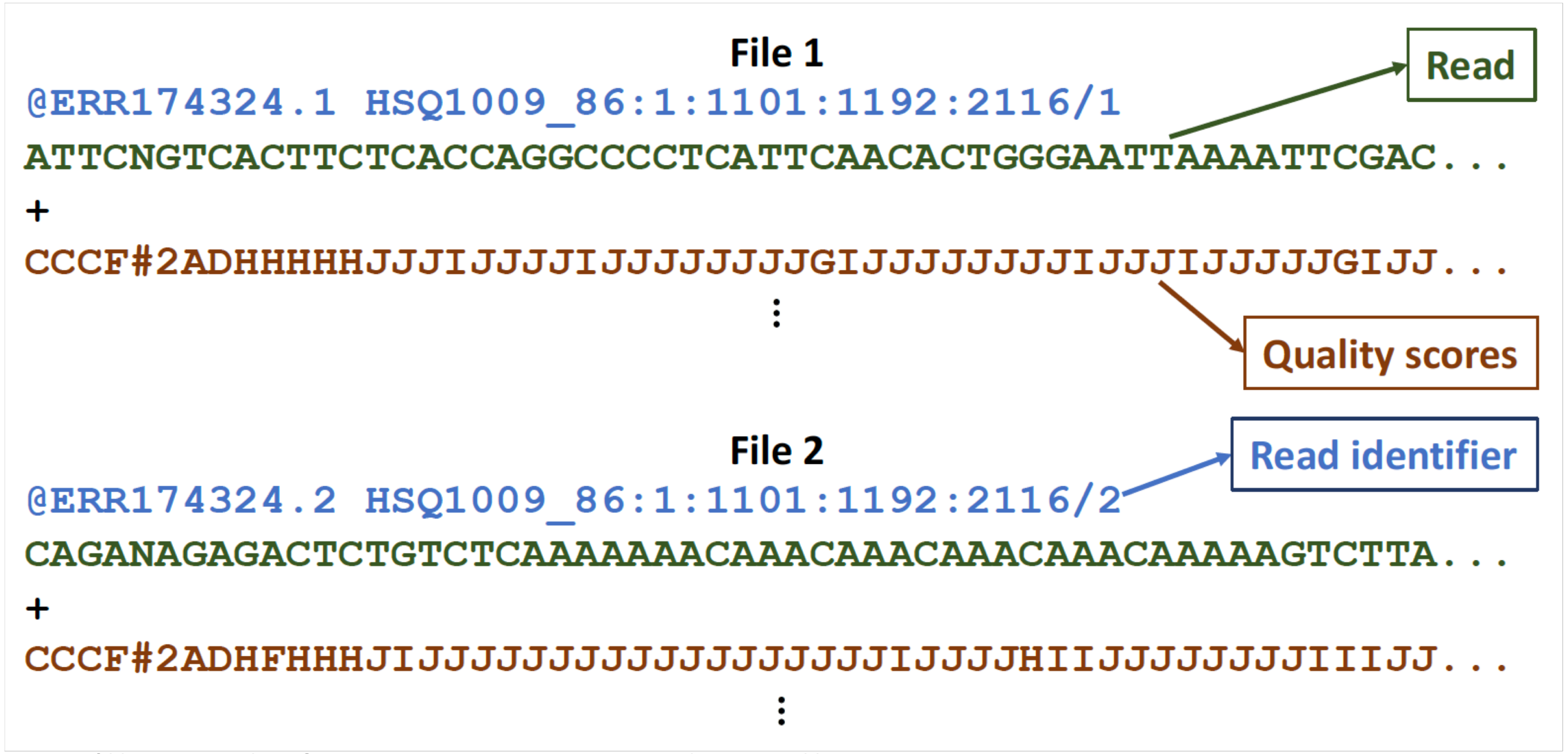
EARTH BIOGENOME PROJECT

sequencing life for the future of life

~1.5M eukaryote species

FASTQ format

FASTQ format



File 1

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

File 2

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

Read
Quality scores
Read identifier

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

Read compression

Read compression

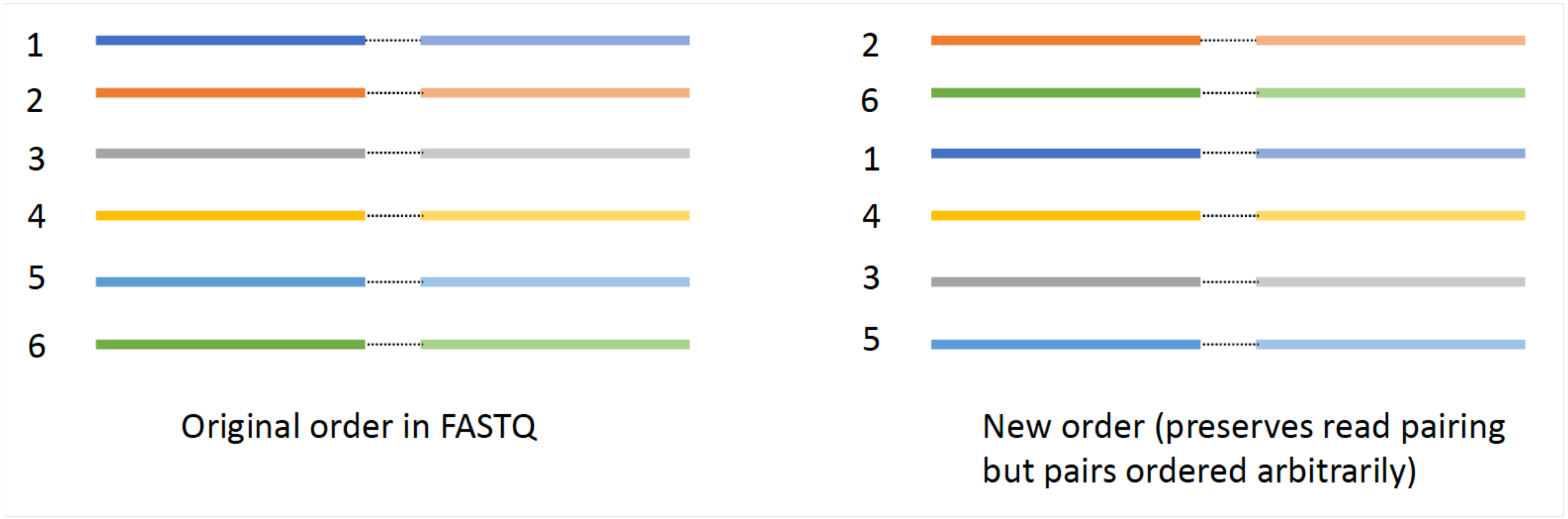
- For a typical 25x human dataset:
  - Uncompressed: 79 GB (1 byte/base)

Read compression

- For a typical 25x human dataset:
  - Uncompressed: 79 GB (1 byte/base)
  - Gzip: ~20 GB (2 bits/base) – still far from optimal

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?


1
2
3
4
5
6
Original order in FASTQ
2
6
1
4
3
5
New order (preserves read pairing but pairs ordered arbitrarily)

Read compression results

Compressor	25x human
Uncompressed	79 GB
Gzip	~20 GB

Read compression results

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

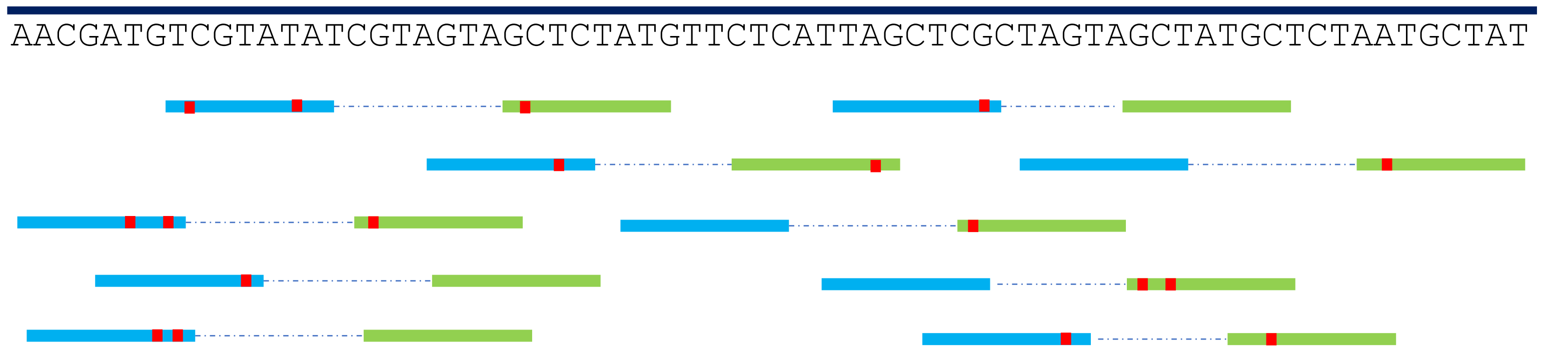
Read compression results

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

Read compression results

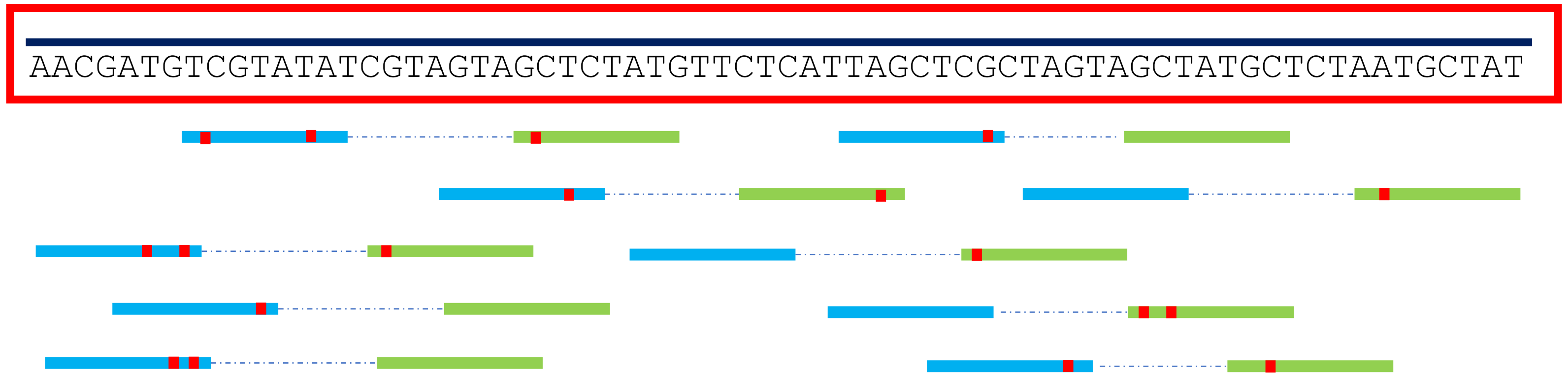
Compressor	25x human	100x human
Uncompressed	79 GB	319 GB
Gzip	~20 GB	~80 GB
FaStore (allow reordering)	6 GB	13.7 GB
SPRING (no reordering)	**3 GB**	**10 GB**
SPRING (allow reordering)	**2 GB**	**5.7 GB**

Key idea


AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

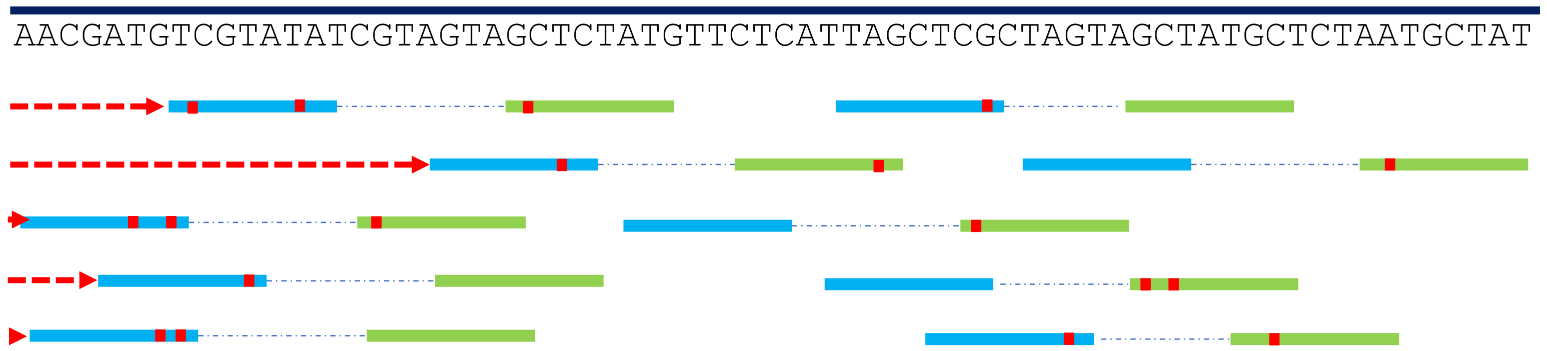
- Storing reads equivalent to

Key idea


AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

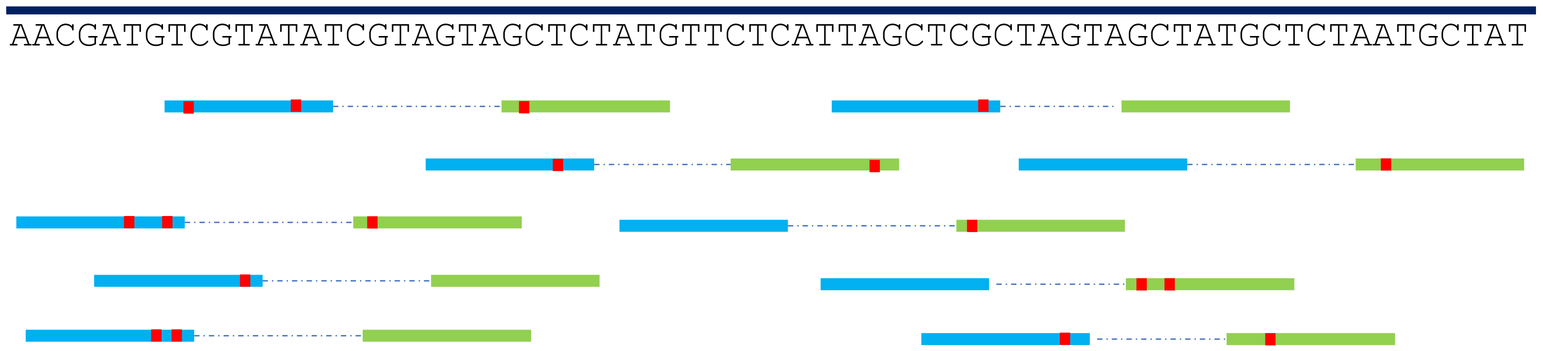
- Storing reads equivalent to
    - Store genome

Key idea


AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

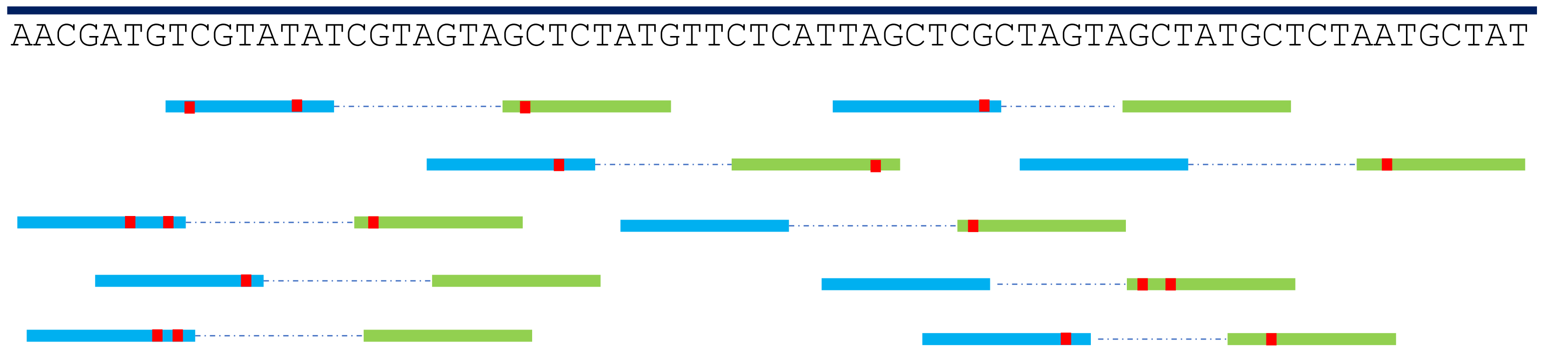
- Storing reads equivalent to
  - Store genome
  - Store read positions in genome

Key idea


AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

- Storing reads equivalent to
  - Store genome
  - Store read positions in genome
  - Store noise in reads

Key idea


AACGATGTCGTATATCGTAGTAGCTCTATGTTCTCATTAGCTCGCTAGTAGCTATGCTCTAATGCTAT

- Storing reads equivalent to
  - Store genome
  - Store read positions in genome
  - Store noise in reads
- Entropy calculations show this outperforms previous compressors

Key idea

- But... How to get the genome from the reads?

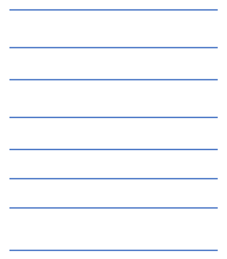
Key idea

- 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

Key idea

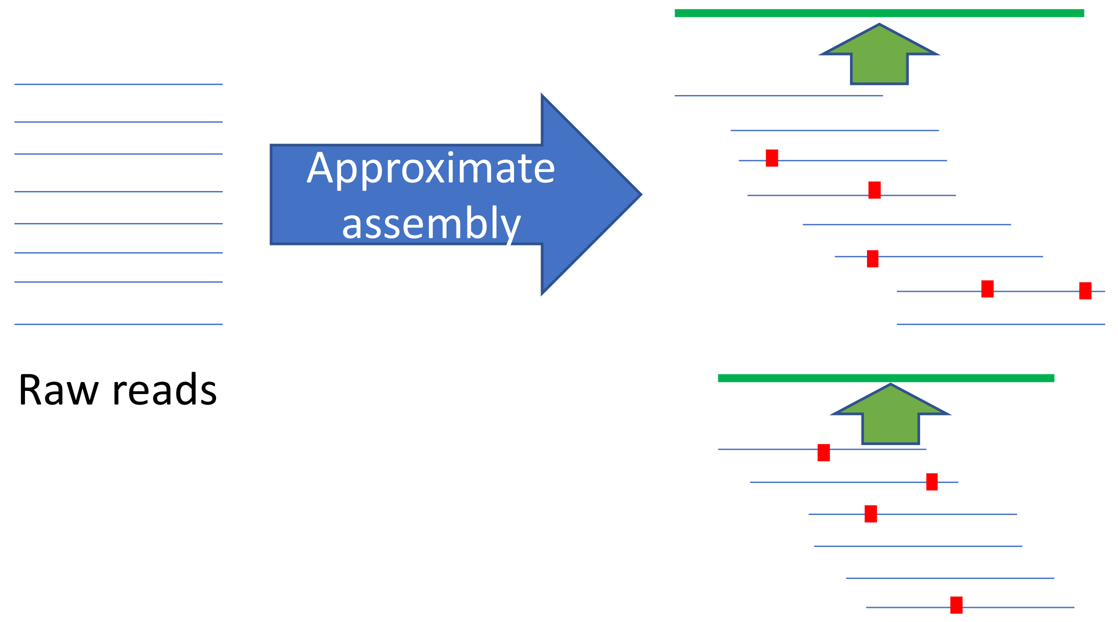
- But... How to get the genome from the reads?
- Genome assembly too expensive - big challenges:
    - resolve repeats
    - get very long pieces of genome from shorter assemblies
- Solution: Don’t need perfect assembly for compression!

SPRING workflow

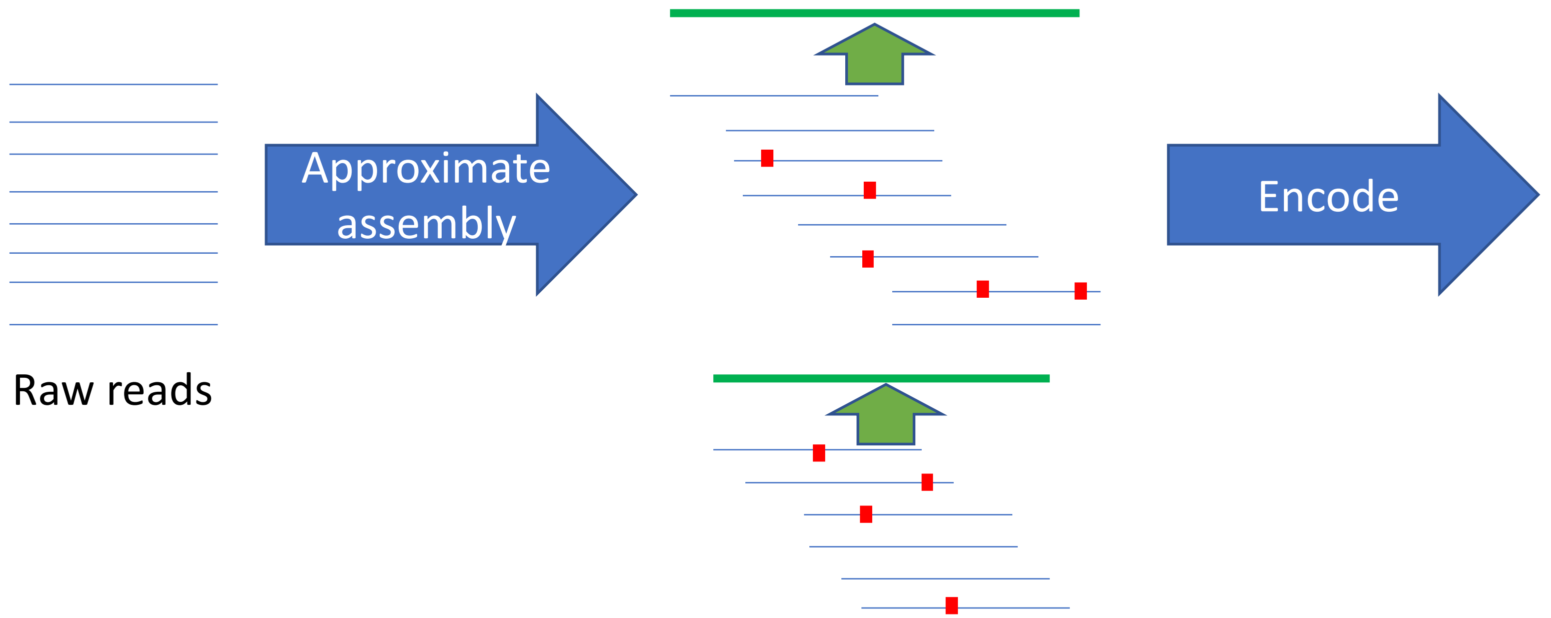


Raw reads

SPRING workflow

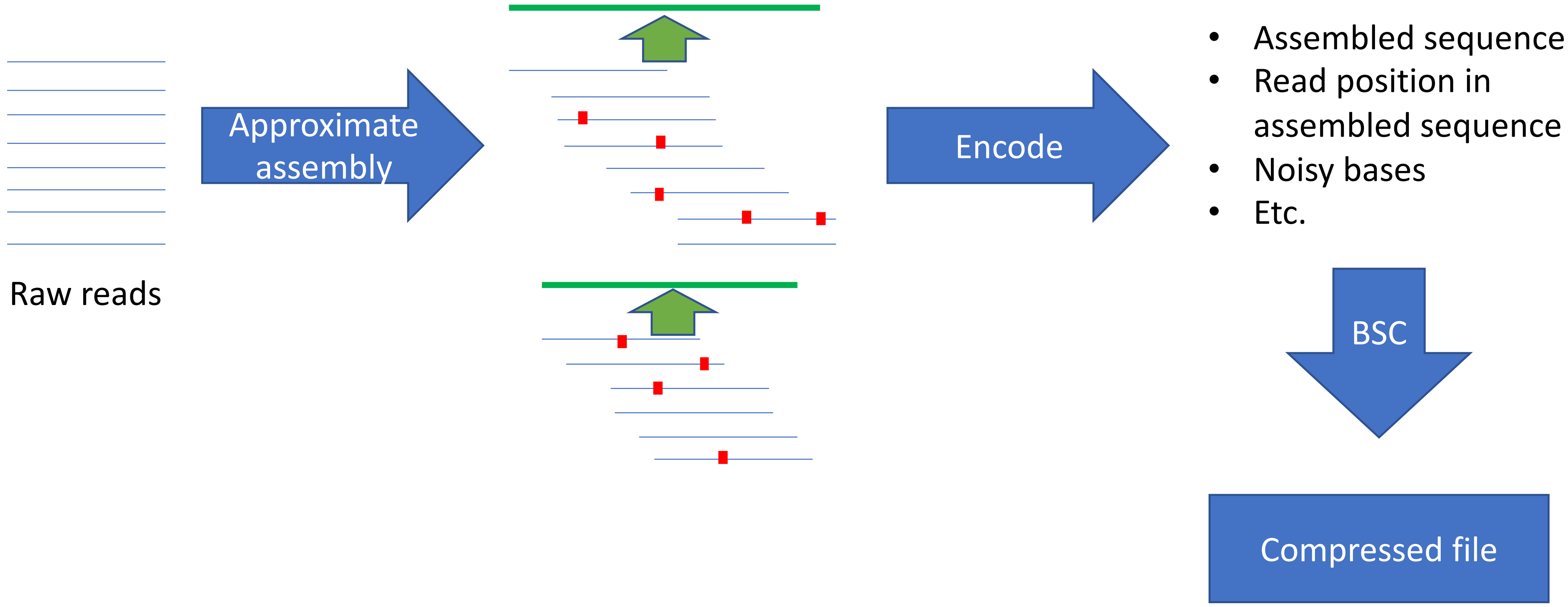


SPRING workflow



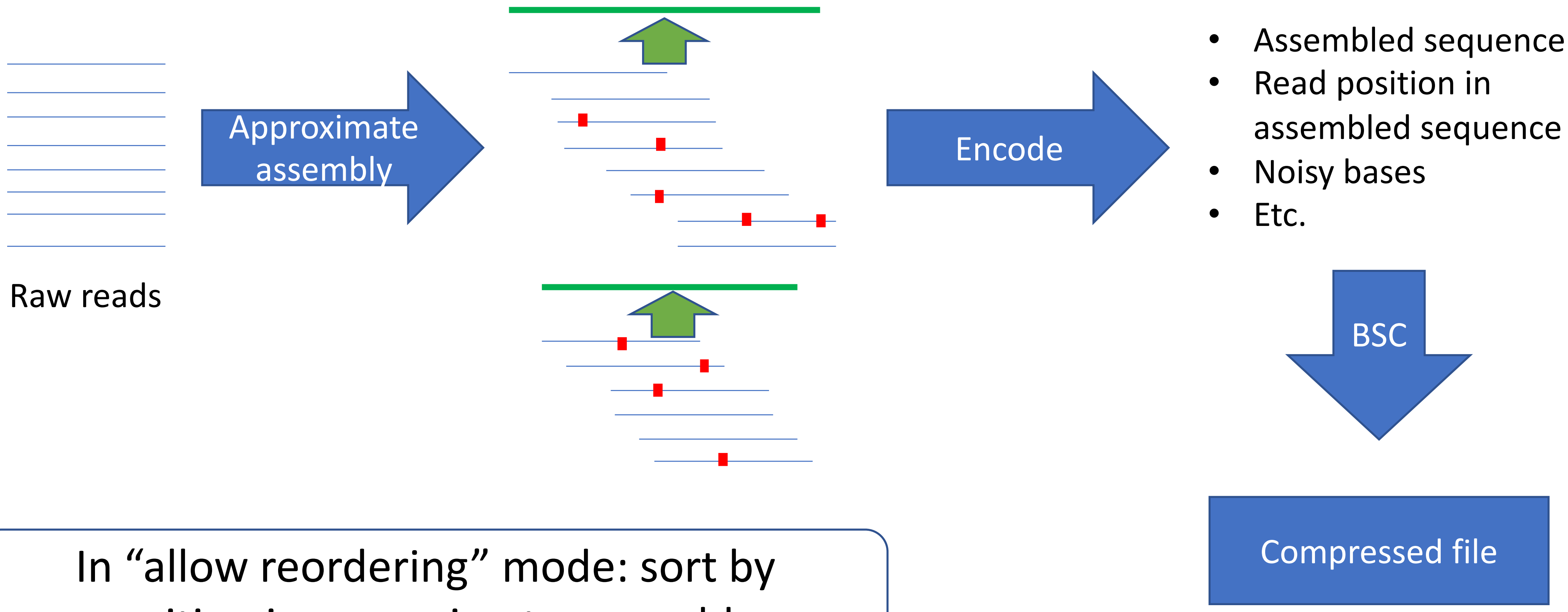
- Assembled sequence
- Read position in assembled sequence
- Noisy bases
- Etc.

SPRING workflow



- Assembled sequence
- Read position in assembled sequence
- Noisy bases
- Etc.

SPRING workflow

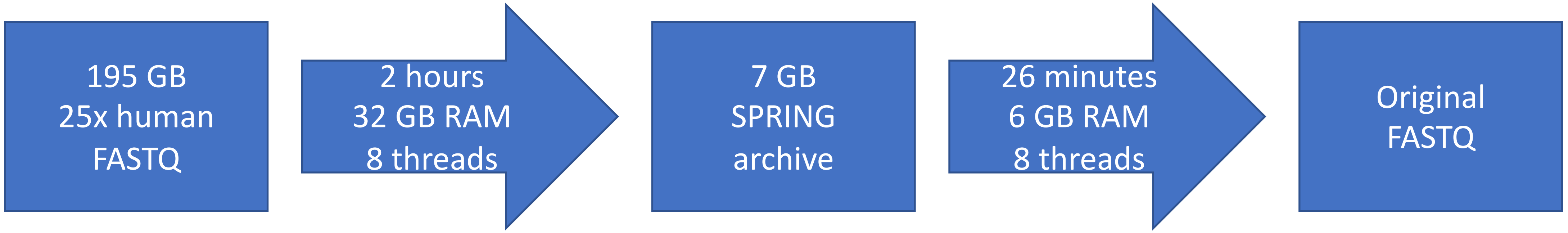


- Assembled sequence
- Read position in assembled sequence
- Noisy bases
- Etc.

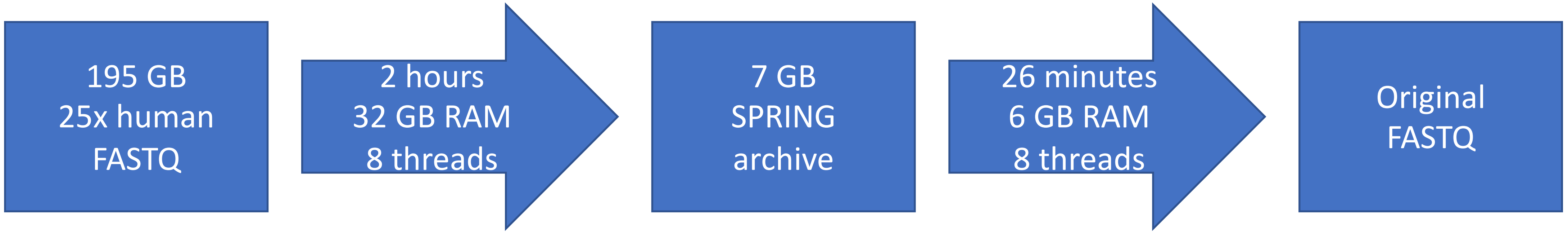
In “allow reordering” mode: sort by position in approximate assembly

SPRING as a practical tool

SPRING as a practical tool

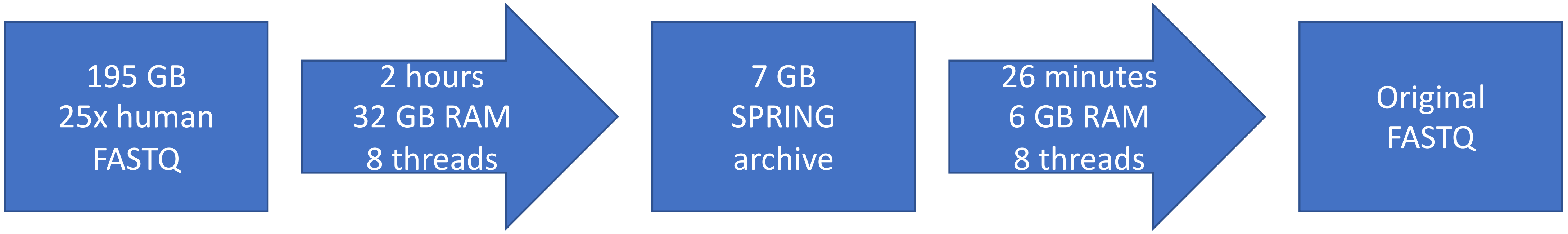


SPRING as a practical tool



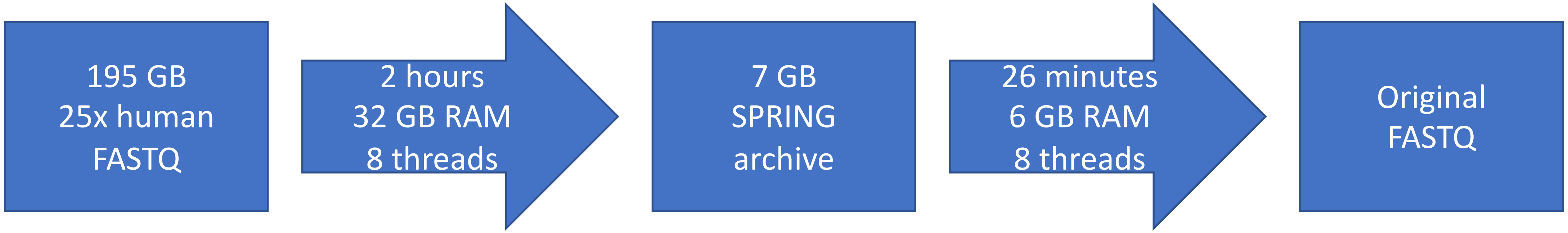
- Support for:
  - Lossless and lossy modes
  - Variable length reads, long reads, etc.
  - Random access

SPRING as a practical tool



- Support for:
  - Lossless and lossy modes
  - Variable length reads, long reads, etc.
  - Random access
- Github: https://github.com/shubhamchandak94/SPRING/

SPRING as a practical tool



- Support for:
  - Lossless and lossy modes
  - Variable length reads, long reads, etc.
  - Random access
- Github: https://github.com/shubhamchandak94/SPRING/
- Currently integrating with genie, an upcoming open source MPEG-G codec

Thank you!

References

- Shubham Chandak, Kedar Tatwawadi, Tsachy Weissman; Compression of genomic sequencing reads via hash-based reordering: algorithm and analysis, *Bioinformatics*, Volume 34, Issue 4, 15 February 2018, Pages 558–567
- Shubham Chandak, Kedar Tatwawadi, Idoia Ochoa, Mikel Hernaez, Tsachy Weissman; SPRING: a next-generation compressor for FASTQ data, *Bioinformatics*, bty1015
- 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
- Alberti C. et al. (2018) An introduction to MPEG-G, the new ISO standard for genomic information representation. https://www.biorxiv.org/content/early/2018/10/08/426353.
- BSC: https://github.com/IlyaGrebnov/libbsc
- genie (open source MPEG-G codec): https://mitogen.github.io/
- Image credits:
  - https://www.genome.gov/27541954/dna-sequencing-costs-data/
  - https://twitter.com/nature/status/1050115893957730305
  - http://www.earlham.ac.uk/newsroom/decoding-life-earth