Stanford University

Efficient storage of and in DNA: genomic data compression & DNA based storage

Shubham Chandak

PhD ‘21, Electrical Engineering, Stanford University

Currently Applied Scientist, S3, Amazon Web Services

-

Biochemical Engineering and Biotechnology Department Seminar

IIT Delhi

Apr 28, 2022

Outline

- Introduction to genomic sequencing technologies
- Genomic data compression: SPRING
- Using DNA as a storage medium

Introduction to genomic sequencing technologies

What is the genome?

What is genome sequencing?

Why compression?

Raw data and downstream analysis

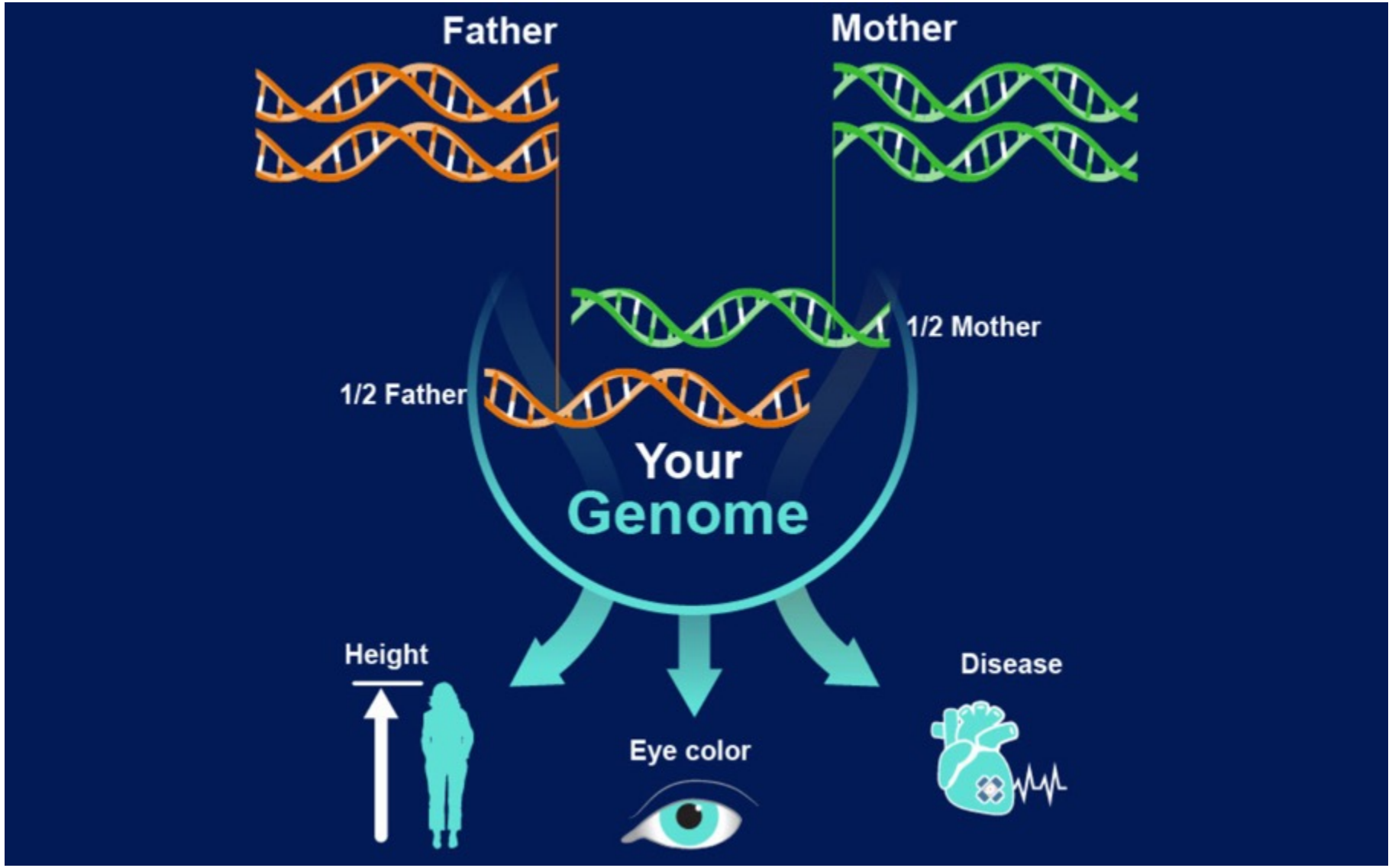


Image source: https://www.genome.gov/About-Genomics/Introduction-to-Genomics

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

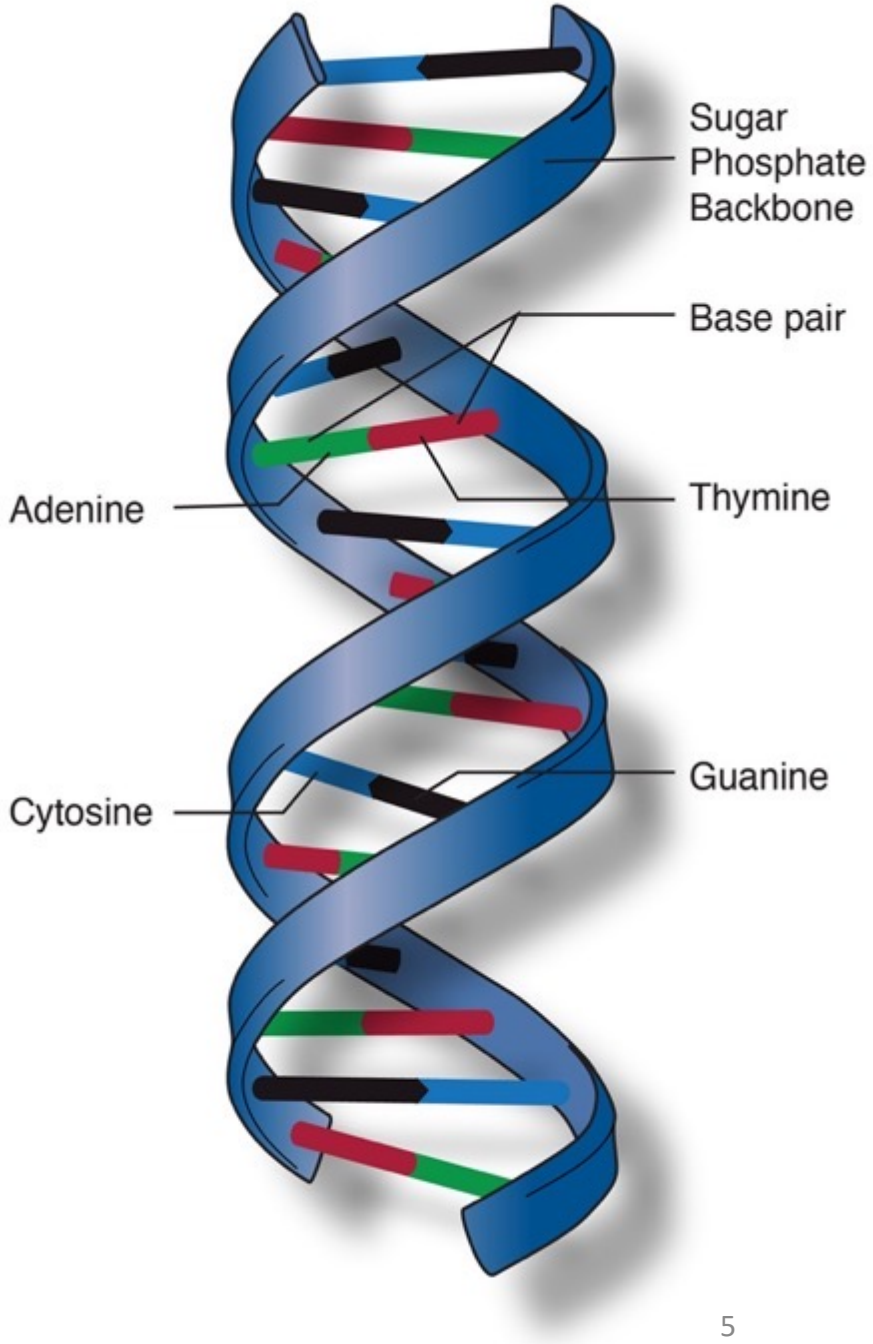


Image source: https://www.genome.gov/genetics-glossary/Double-Helix

Genome sequencing

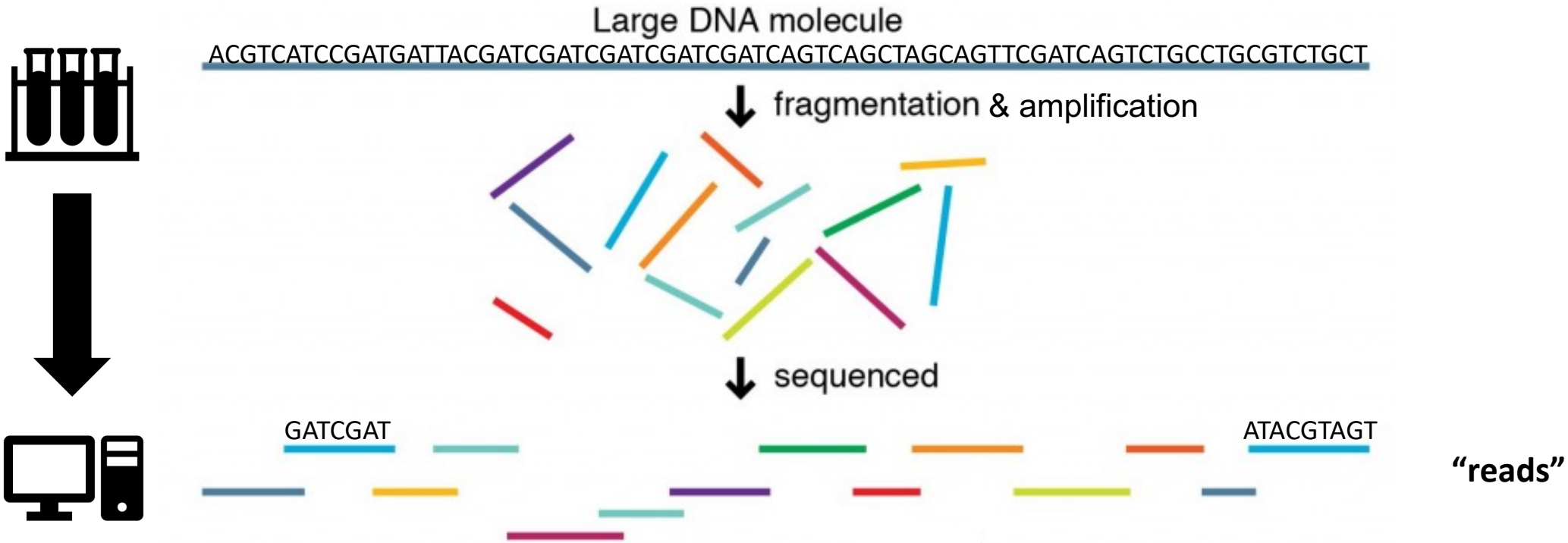
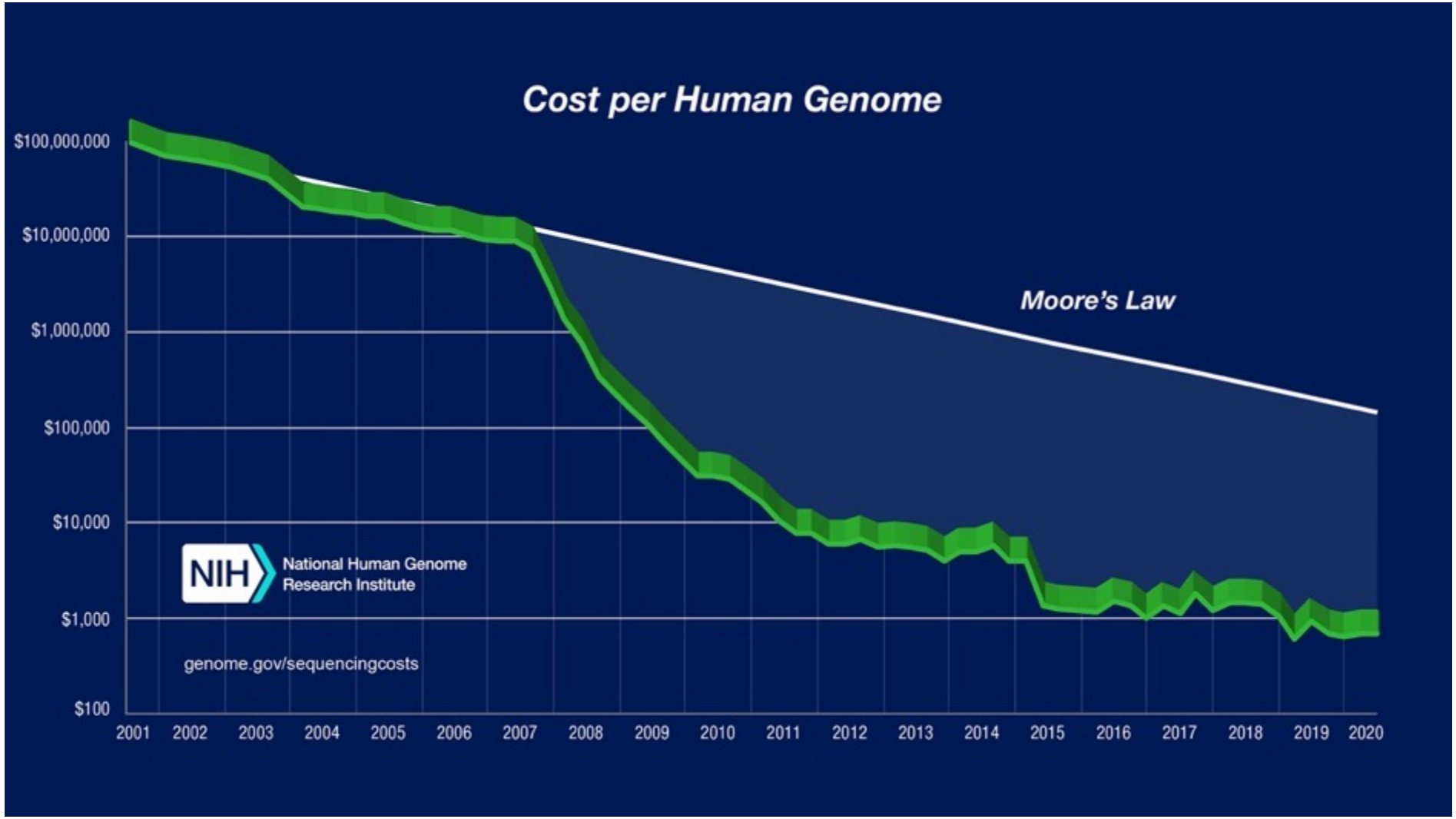


Image source: https://www.genengnews.com/uncategorized/first-nanopore-sequencing-of-human-genome/


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

500K human genomes

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

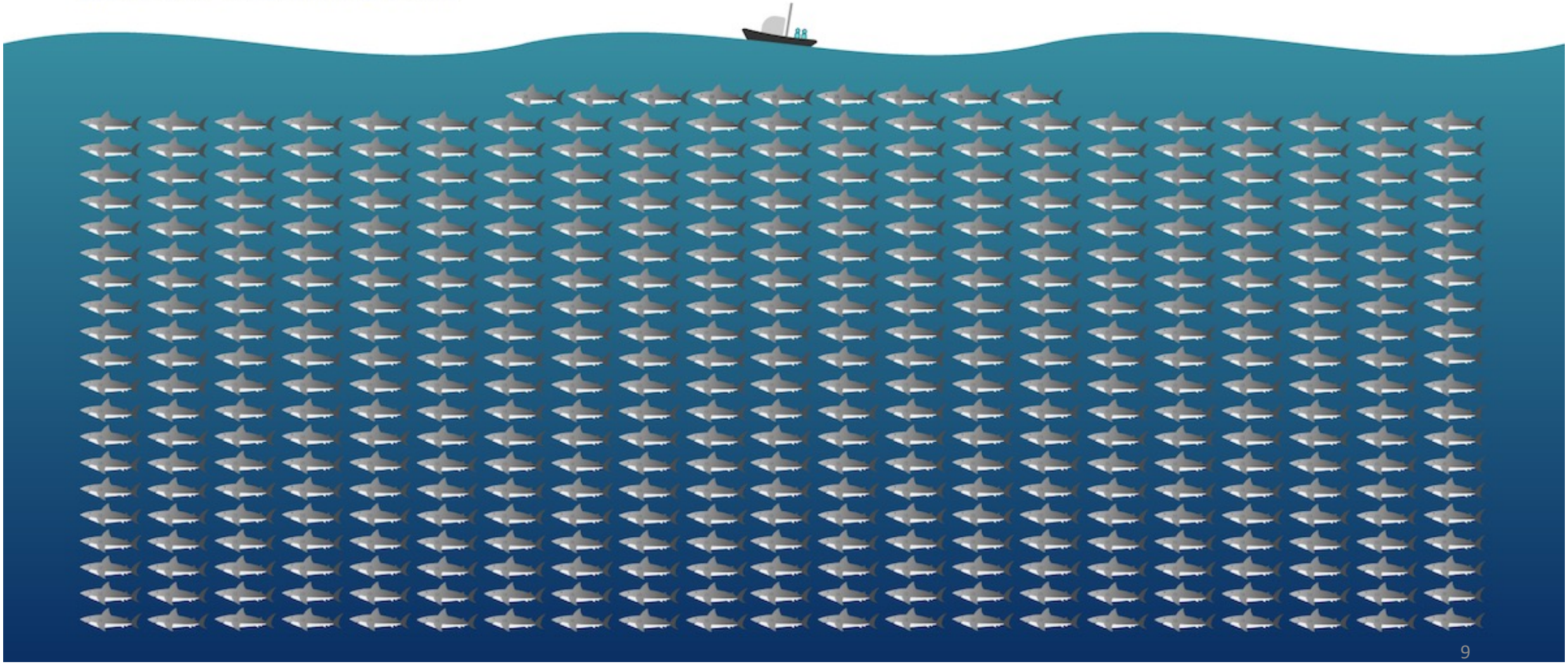


Image source: https://www.genome.gov/about-genomics/fact-sheets/Genomic-Data-Science

Sequencing & downstream analysis

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

Genome assembly

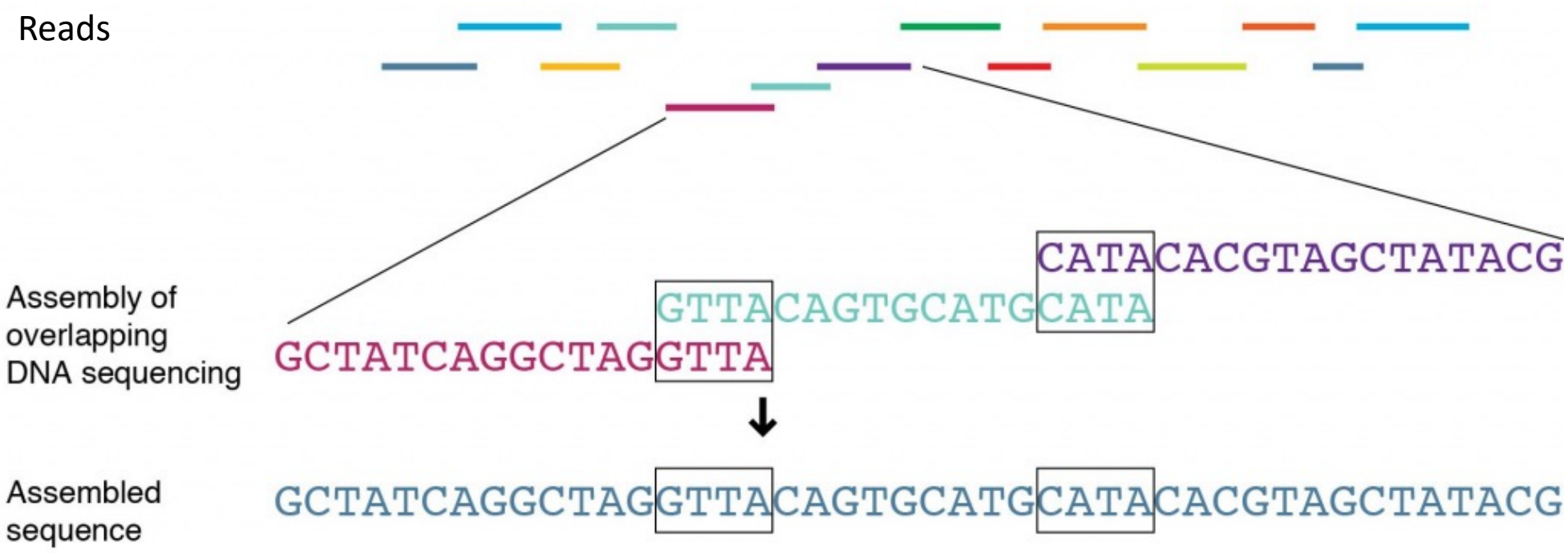


Image source: https://knowgenetics.org/whole-genome-sequencing/

Alignment and Variant Calling

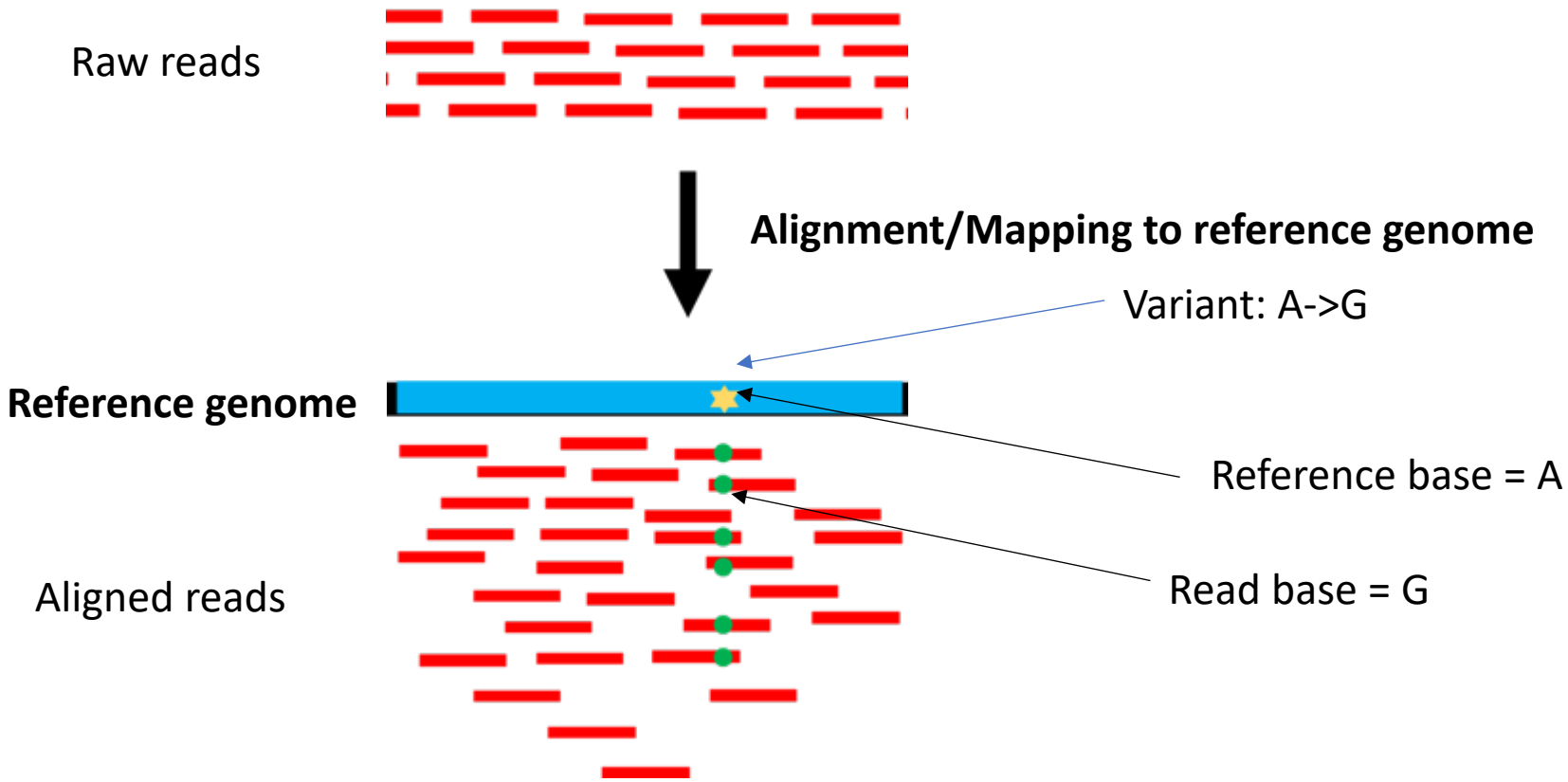
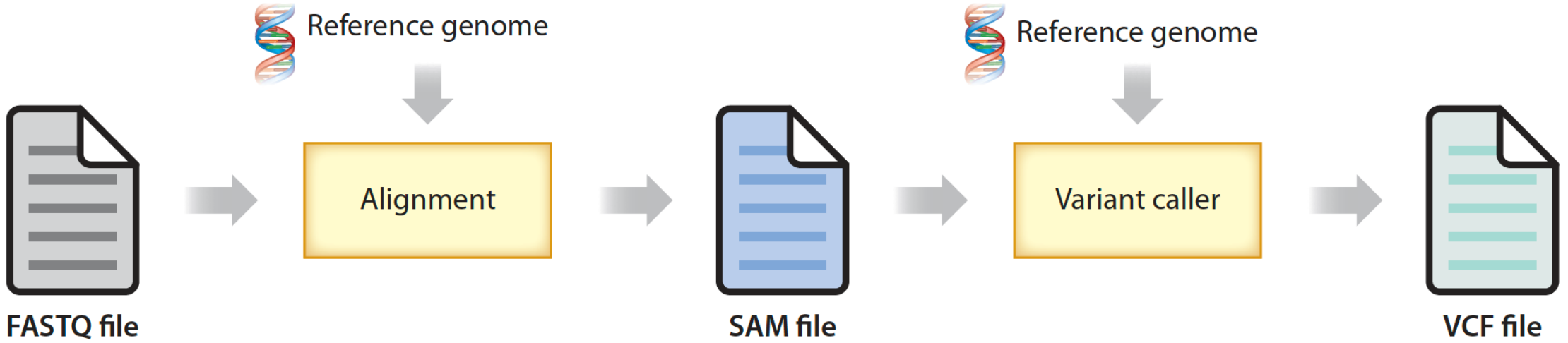


Image source: https://training.galaxyproject.org/archive/2019-02-07/topics/proteomics/tutorials/proteogenomics-dbcreation/tutorial.html/

File formats in the pipeline



This talk

Use CRAM!

Tabular data – requires random access

Image source: https://www.annualreviews.org/doi/pdf/10.1146/annurev-biodatasci-072018-021229

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

- High throughput
- Short reads
- Low error rate

Oxford Nanopore MinION

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

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

Outline

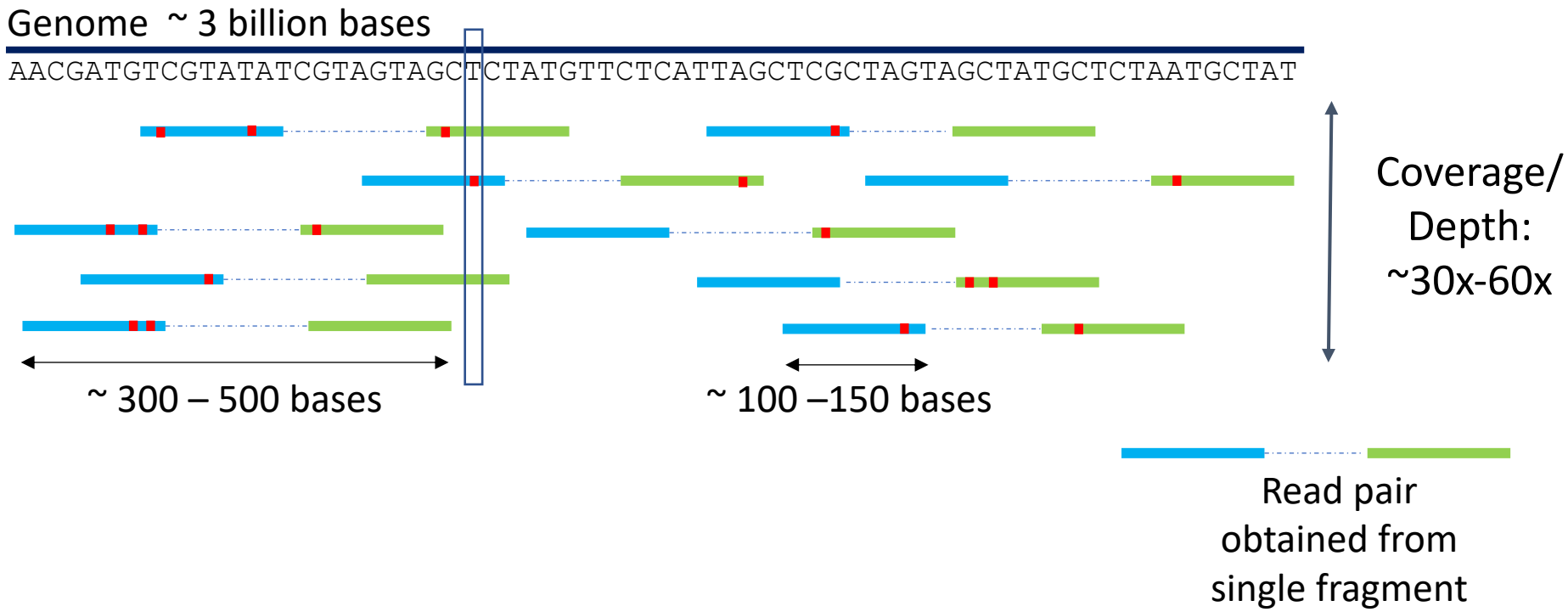
- Introduction to genomic sequencing technologies
- Genomic data compression: SPRING
- Using DNA as a storage medium

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

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

Paired-end genome sequencing

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



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

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

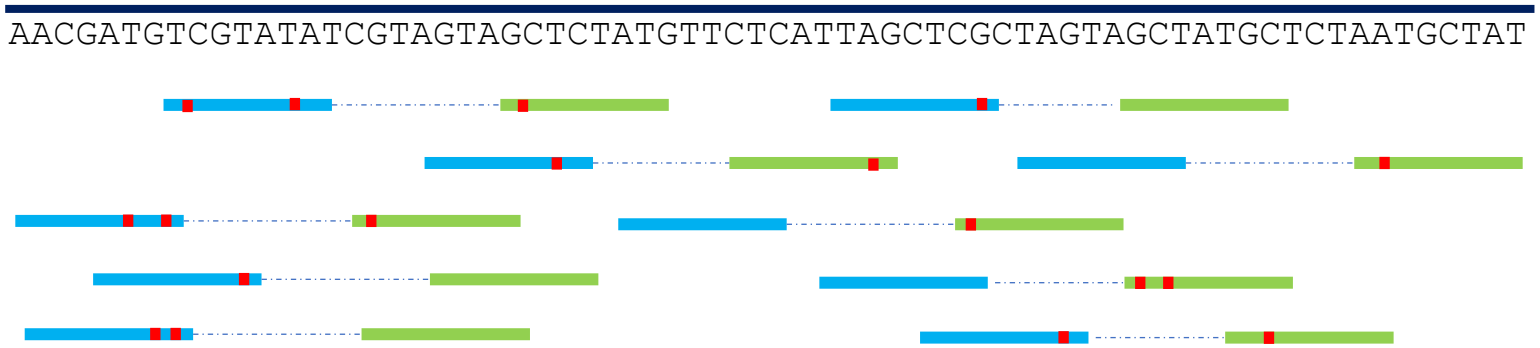
Read compression results

Compressor	25x human
Uncompressed	79 GB
Gzip	~20 GB
SPRING	**3 GB**

Read compression results

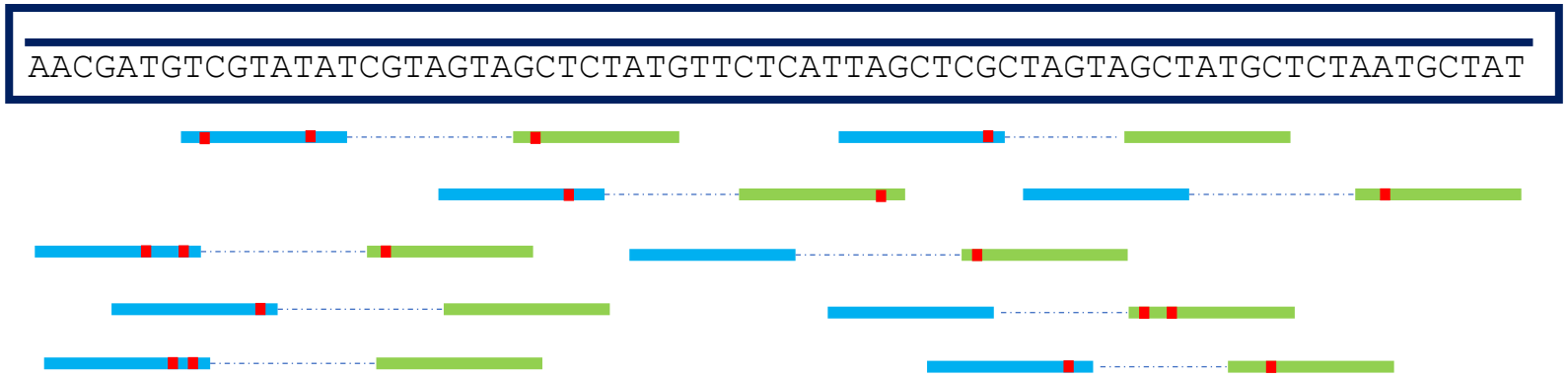
Compressor	25x human	100x human
Uncompressed	79 GB	319 GB
Gzip	~20 GB	~80 GB
SPRING	**3 GB**	**10 GB**

Key idea



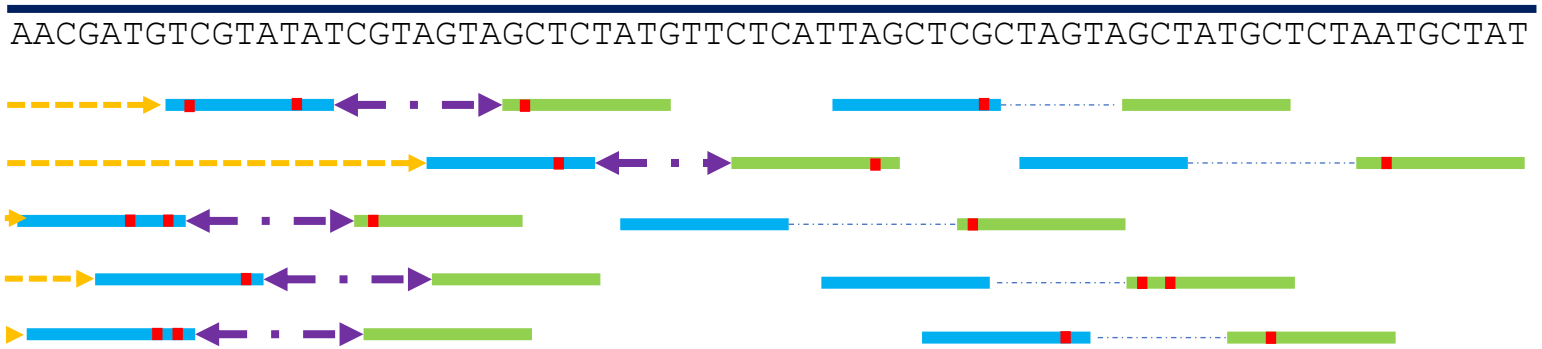
- Storing reads equivalent to

Key idea



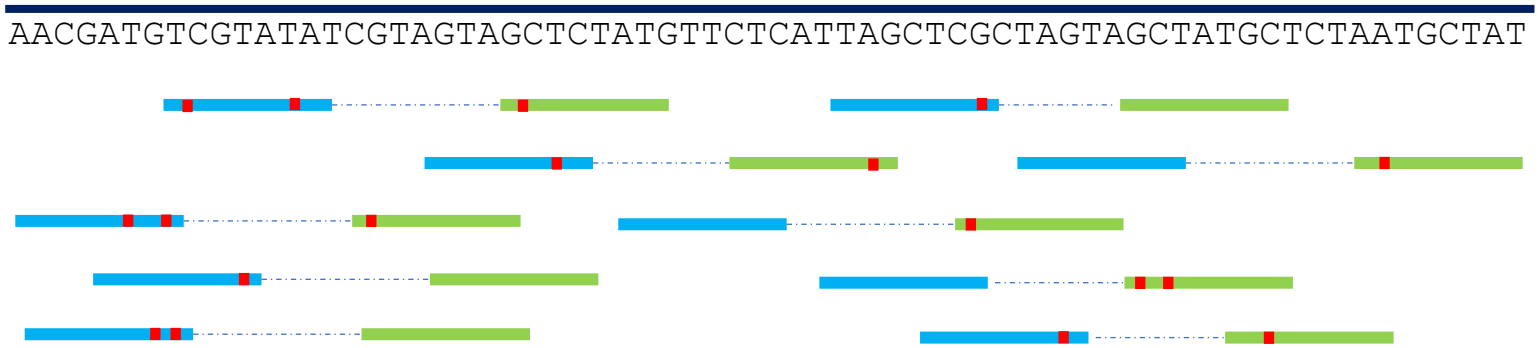
- Storing reads equivalent to
  - Store genome

Key idea



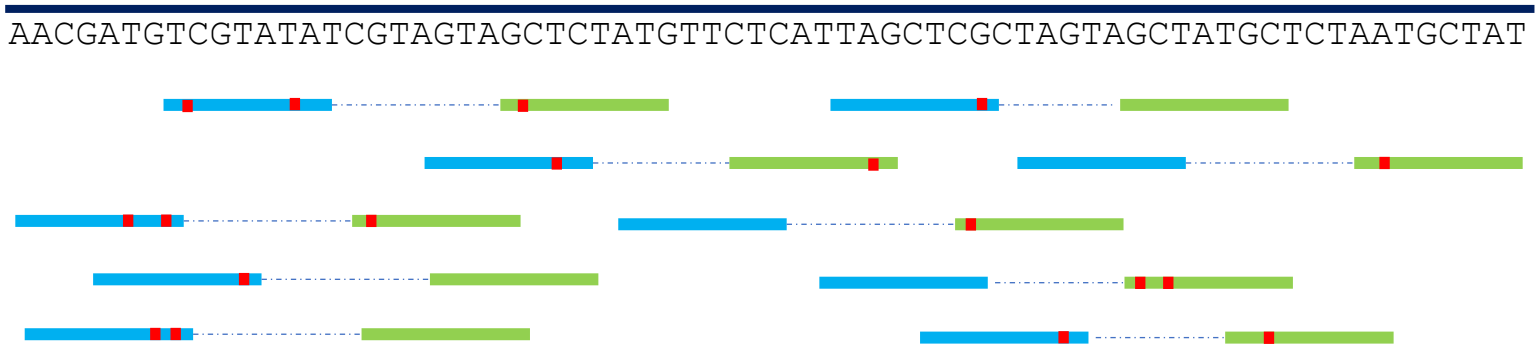
- Storing reads equivalent to
  - Store genome
  - Store read positions in genome (+ gap between paired reads)

Key idea



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

Key idea

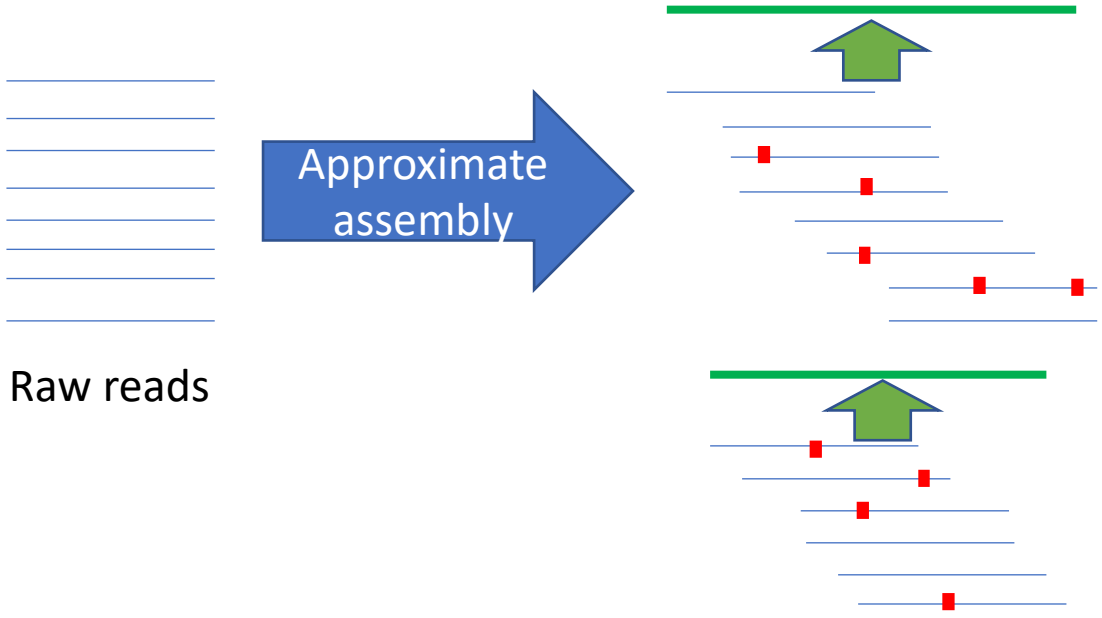


- Storing reads equivalent to
  - Store genome
  - Store read positions in genome (+ gap between paired reads)
  - Store noise in reads
- Theoretical calculations show this outperforms previous compressors

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

Encode

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

general purpose compression

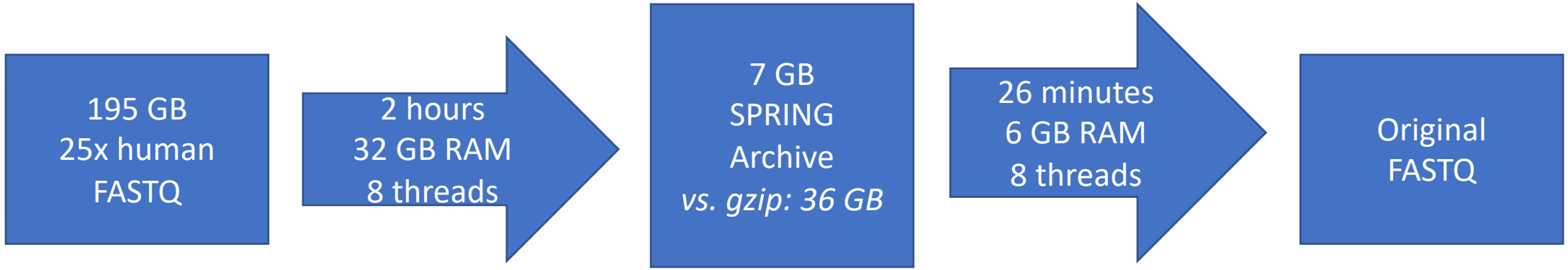
BSC

Compressed file

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

https://github.com/IlyaGrebnov/libbsc

SPRING as a practical tool



- 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/

Future directions

- Another paradigm: reference-based FASTQ compression
  - Illumina ORA/Enancio, Petagene
- More recent work on compression for long read data
  - Meng, Q., Chandak, S., Zhu, Y., & Weissman, T. (2021). NanoSpring: reference-free lossless compression of nanopore sequencing reads using an approximate assembly approach. *bioRxiv*.
  - Shubham Chandak, Kedar Tatwawadi, Srivatsan Sridhar, Tsachy Weissman, 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.

Outline

- Introduction to genomic sequencing technologies
- Genomic data compression: SPRING
- Using DNA as a storage medium

Lau, Billy T., **Chandak S.**, et al. "Magnetic DNA random access memory with nanopore readouts and exponentially-scaled combinatorial addressing." *bioRxiv* (2021).

S. Chandak et al.; "Overcoming high nanopore basecaller error rates for DNA storage via basecaller-decoder integration and convolutional codes," ***ICASSP 2020.***

S. Chandak et al.; "Improved read/write cost tradeoff in DNA-based data storage using LDPC codes," ***Allerton 2019.***

Team and funding


Shubham Chandak


Joachim Neu


Jay Mardia


Billy Lau


Matt Kubit

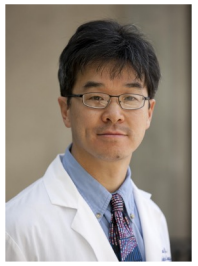

Reyna Hulett


Peter Griffin


Sharmili Roy


Tsachy Weissman


Mary Wootters

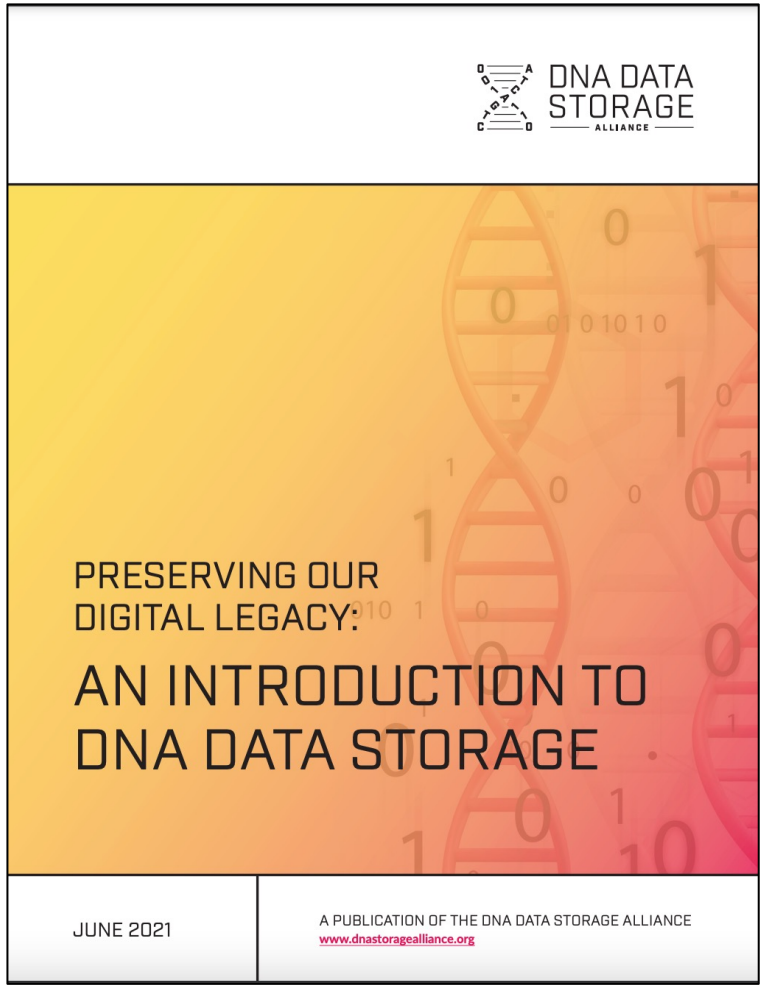

Hanlee Ji

SemiSynBio: Highly scalable random access DNA data storage with nanopore-based reading

Beckman Center Innovative Technology Seed Grant

Scalable Long-Term DNA Storage with Error Correction and Random-Access Retrieval

National Institutes of Health



BBC | Sign in | Home | News | Sport | Reel | Worklife | Trav

NEWS

Home | War in Ukraine | Coronavirus | Climate | Video | World | Asia | UK | Business | Tech | Science

Science

Scientists claim big advance in using DNA to store data

By Paul Rincon
Science editor, BBC News website

1 December 2021

FOUNDERS

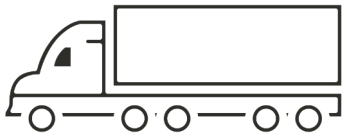
Illumina

Microsoft

Twist Bioscience

Western Digital

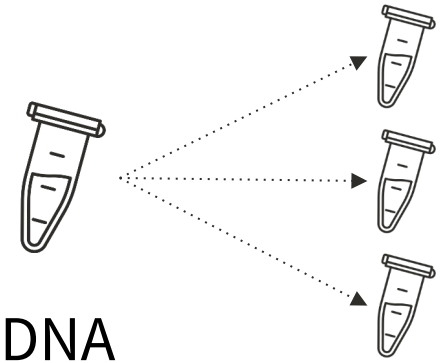
Why DNA-based Storage?



40,000 x 5 TByte
HDDs
40 tons

10’s of years

200 Petabyte



DNA
1 gram

1000’s of years

Easy duplication

Building Blocks

- Ability to "**read/sequence**" the DNA from the solution.

Oxford NANOPORE Technologies

- Ability to "**write/synthesize**" artificial DNA (sequence of {A,C,G,T})

TWIST BIOSCIENCE

CustomArray | GenScript Make Research Easy

Agilent Technologies

Current ability: short DNA oligo sequences (~**150** length) at scale (Array Synthesis)

Typical DNA Storage System

File

Can only synthesize short DNA oligo sequences ~150 bases

Typical DNA Storage System

File

0010101010101
0000101001010
0100011010110
0100100010100
1010010100101
0100110101010
1010000

Binary file

Segment

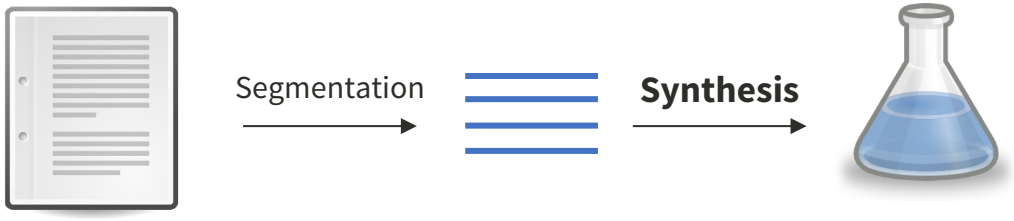
00101010101010000101001010010
00010010101100100010110101000
10100101010011010111010100000

Convert to DNA

00 – A
01 – C …

AGGGGGGACCAGGC
ACAGGCATCAGTTCA
GGCTATTAGCTCGAA

Typical DNA Storage System



File

Order of sequences lost in the solution!

0010101010101
0000101001010
0100011010110
0100100010100
1010010100101
0100110101010
1010000

Binary file

Segment

0010101010101000010100101001
0001001010110010001011010100
1010010101001101011101010000

Convert to DNA

00 – A
01 – C …

AGGGGGGACCAGGC
ACAGGCATCAGTTCA
GGCTATTAGCTCGAA

Typical DNA Storage System

Order of sequences lost in the solution! – Add Index

```
000010101010101000010100101001
000101001010010010001010010100
001010010101001001010101010000
```

Length of index in binary segment at least $\log_2$(number of segments)

Typical DNA Storage System

File

Segmentation

Add Index

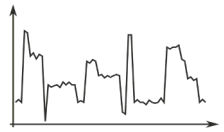
Synthesis

Nanopore sequencing

Sequencing/Basecalling is not perfect!

1. Errors within an oligo sequence (substitution/insertion/deletion)

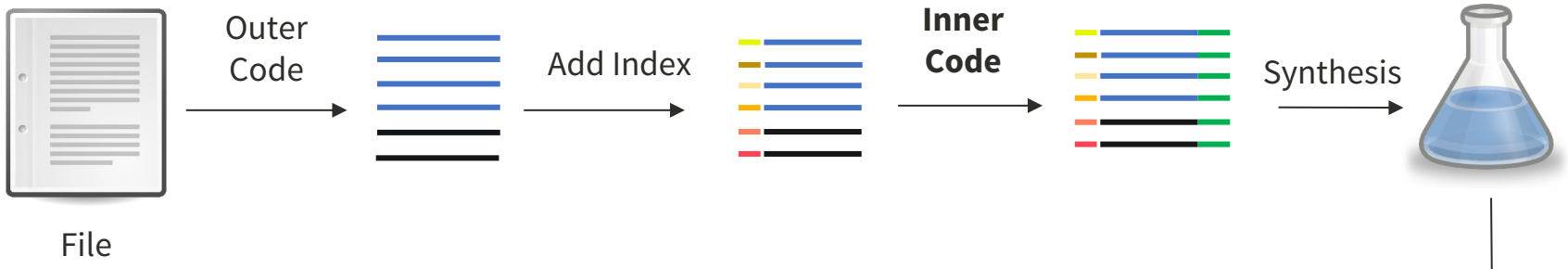
2. Missing Oligo sequences



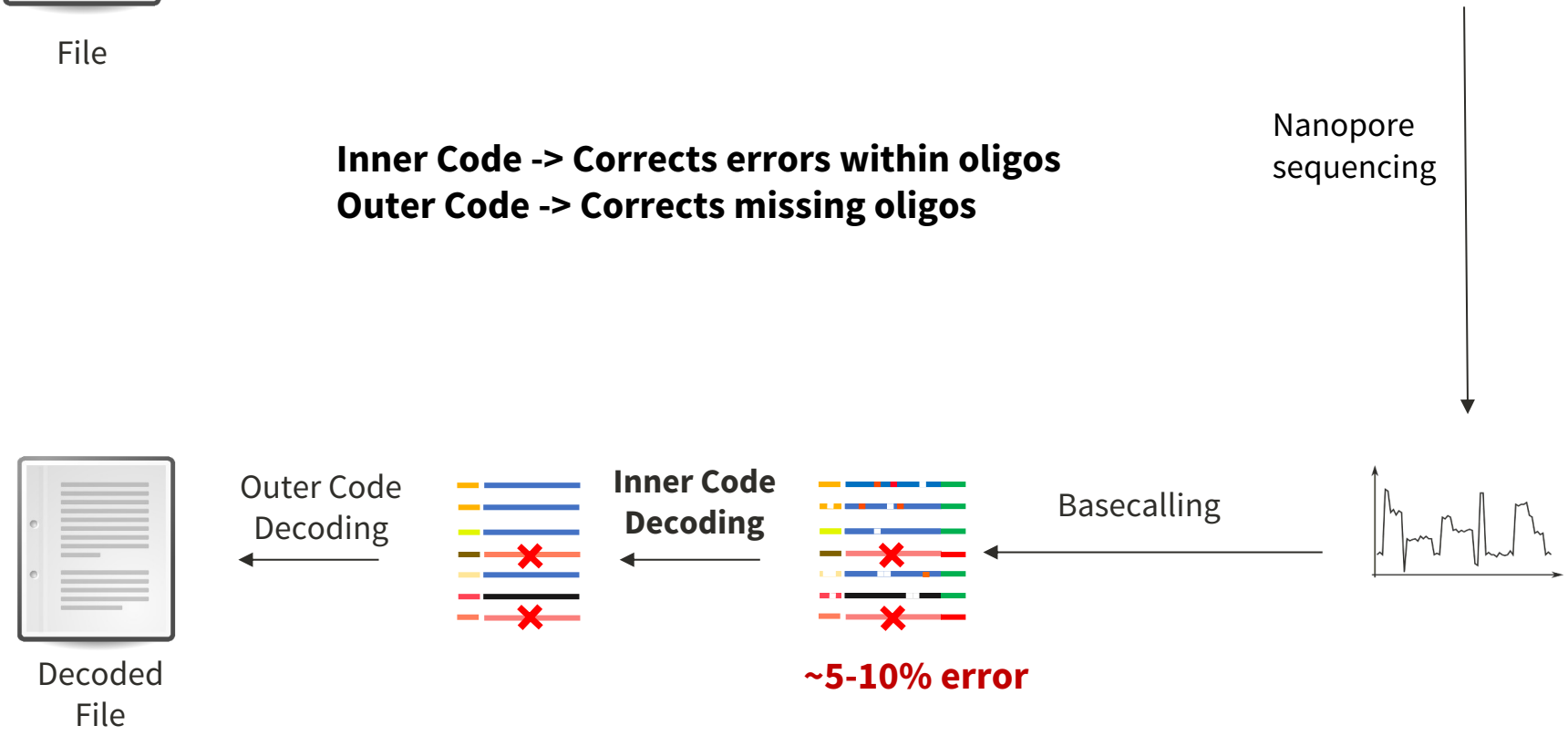
Basecalling

~5-10% error

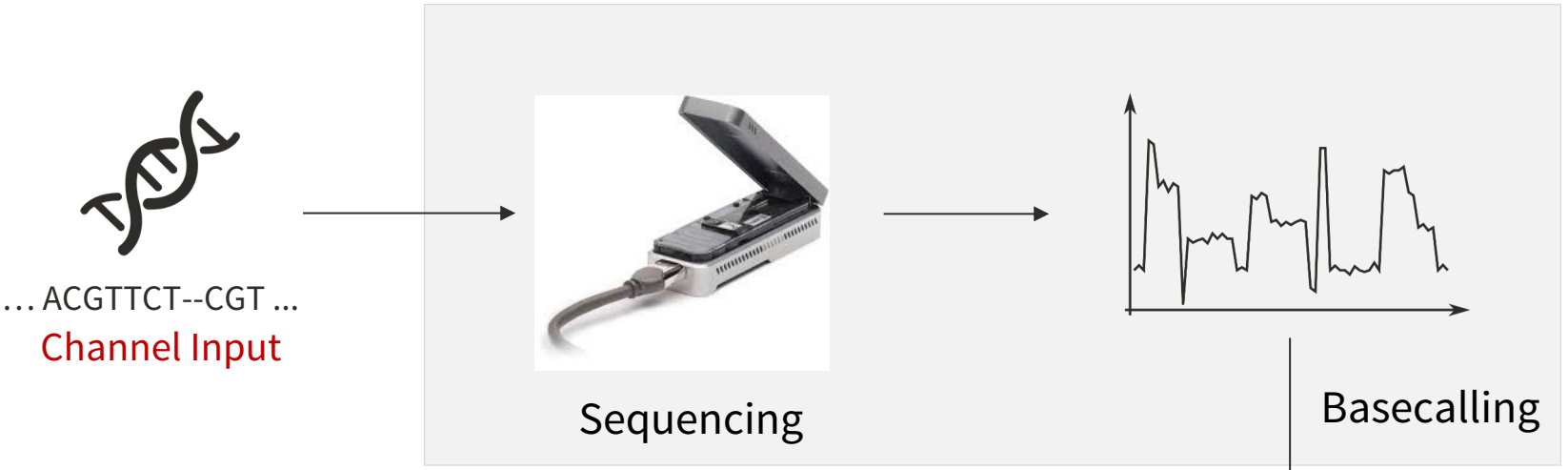
Typical DNA Storage System



Inner Code -> Corrects errors within oligos
Outer Code -> Corrects missing oligos

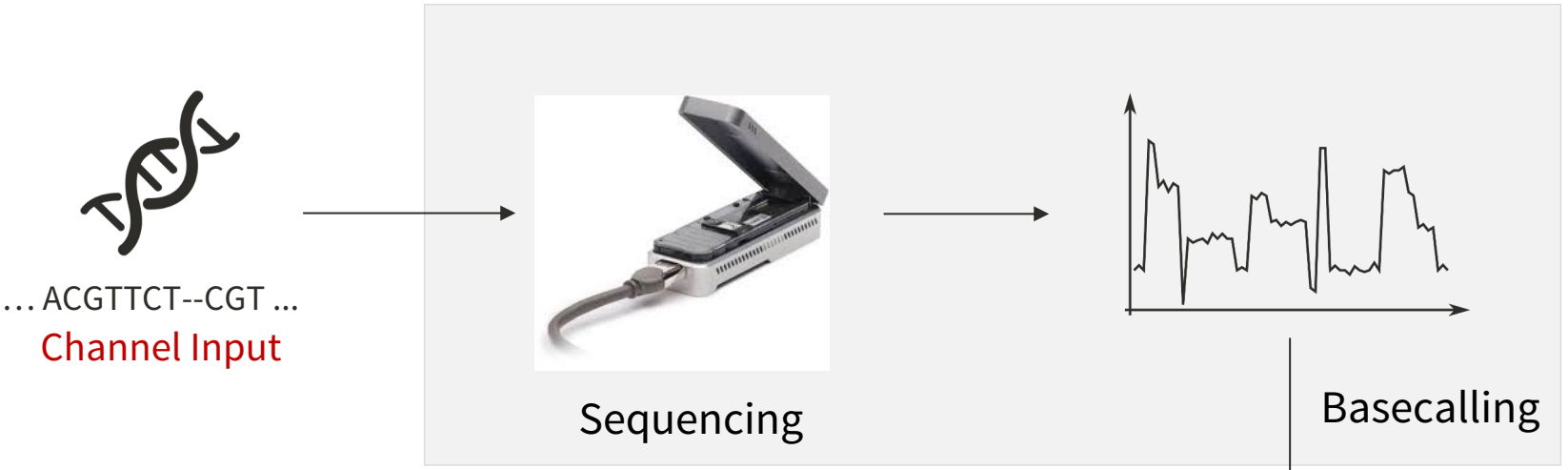


Channel Model – Insertion/Deletion Channel



Channel Model – Insertion/Deletion Channel

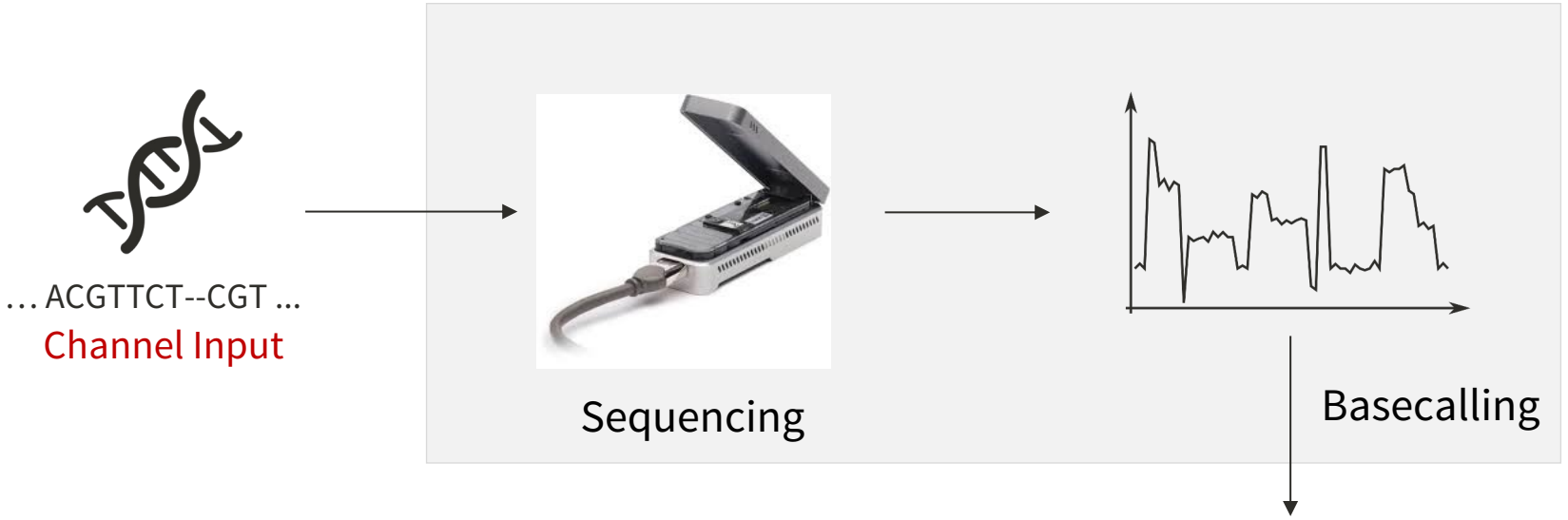
- **Basecalling Error:** No good practical error correction code for 5-10% Insertion/Deletions



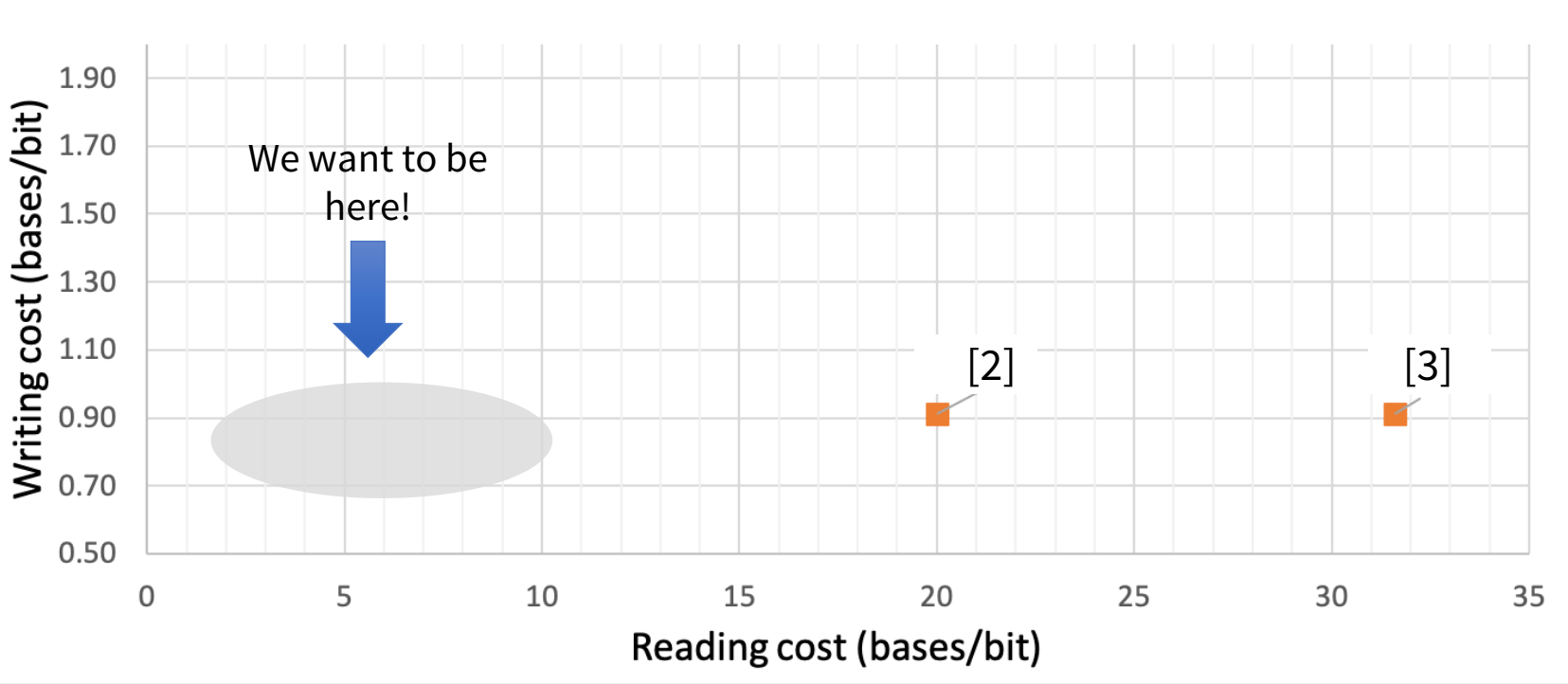
Channel Model – Insertion/Deletion Channel

- **Basecalling Error:** No good practical error correction code for 5-10% Insertion/Deletions

- **Common Idea:** Sequence the input lot of times (~30-40x)
  - cluster *index*, and perform "averaging" to reduce the error



Previous Works

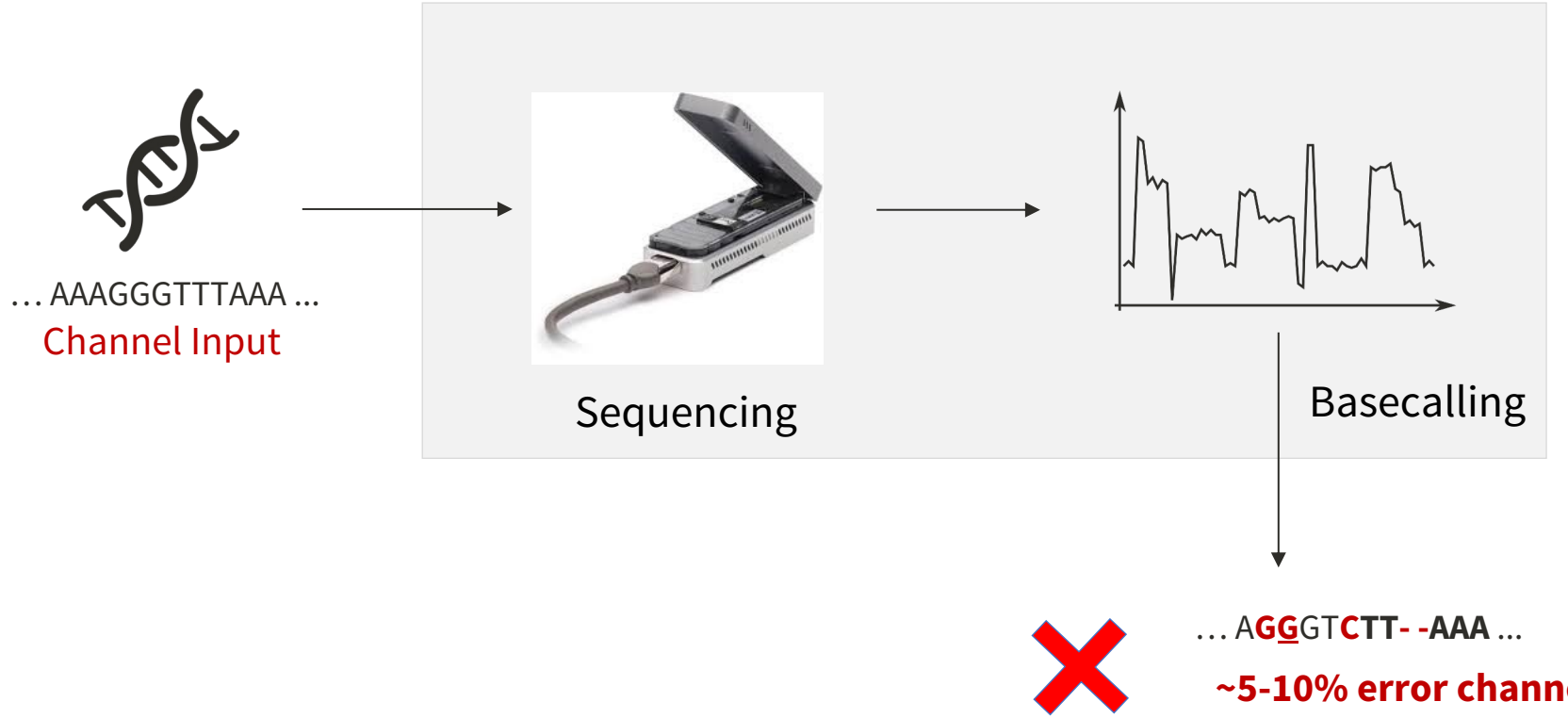


Tradeoff between reading and writing costs

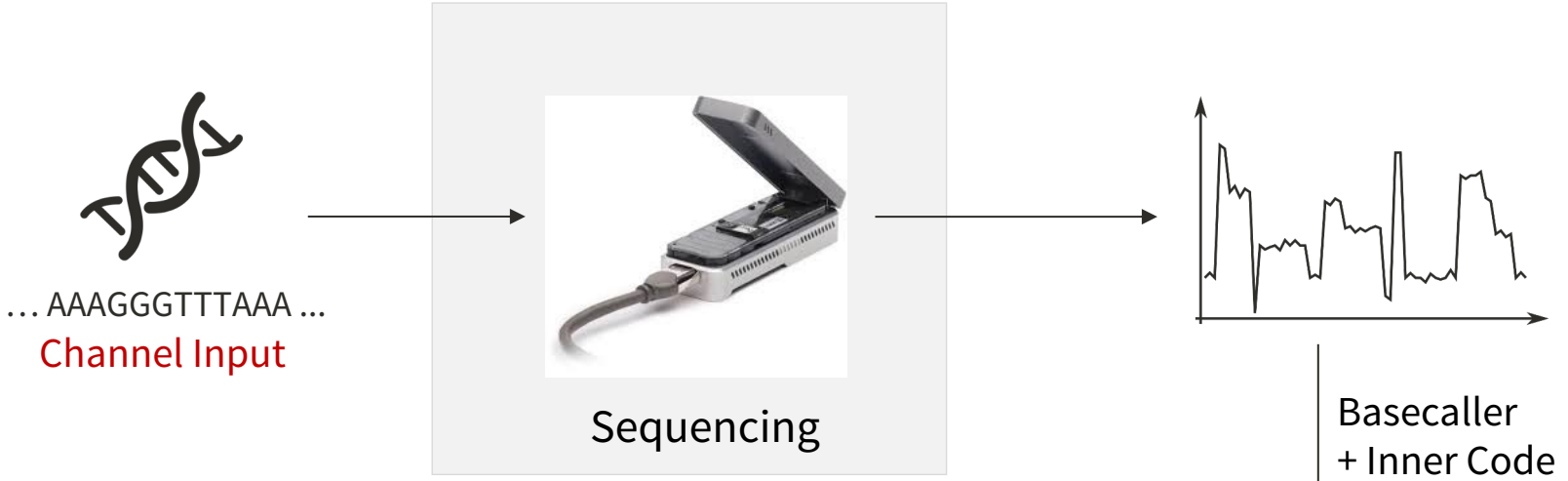
[2] L. Organick *et al.*, “Random access in large-scale DNA data storage," *Nature biotechnology*, vol. 36, no. 3, p. 242, 2018.

[3] Randolph Lopez et al., “DNA assembly for nanopore data storage readout,” Nature communications, vol. 10, no. 1, pp. 2933, 2019.

Nanopore Error Model



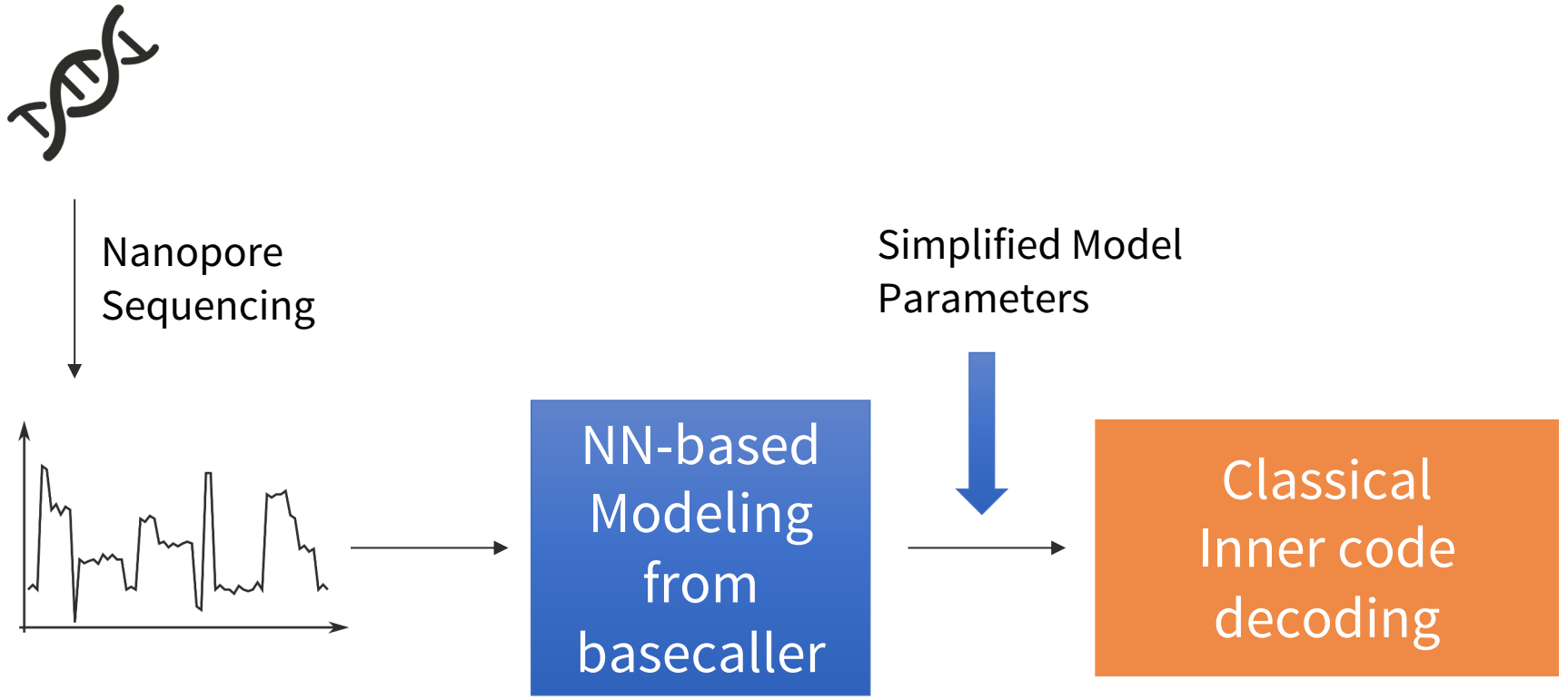
Key Insight!



Basecalling can improve if the basecaller is aware of the inner code structure

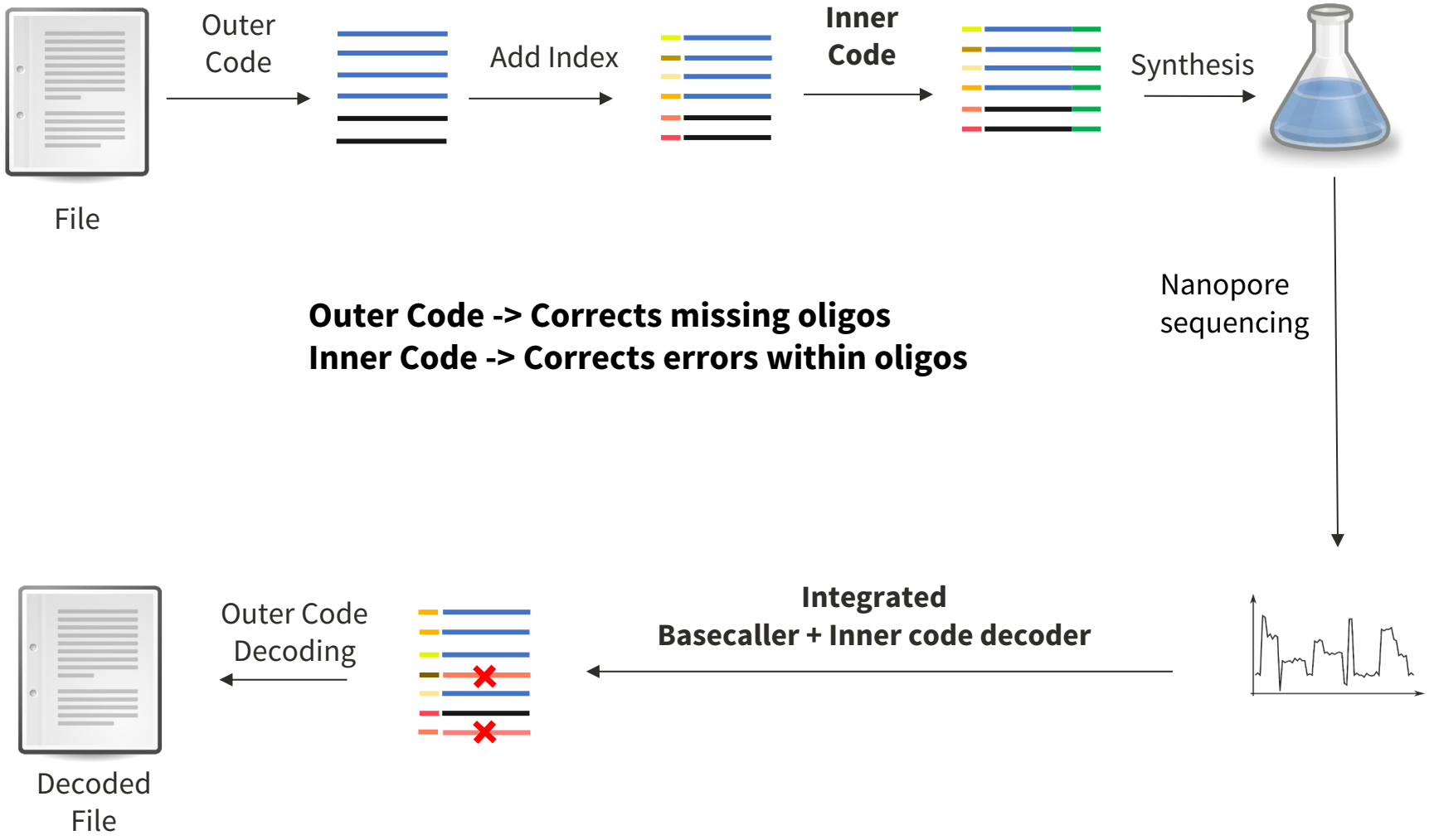
1. **Work with raw current signal rather than Basecalled DNA**
2. **Model Too Complex to work directly: Use *Machine Learning* to simplify**

Inner code decoding

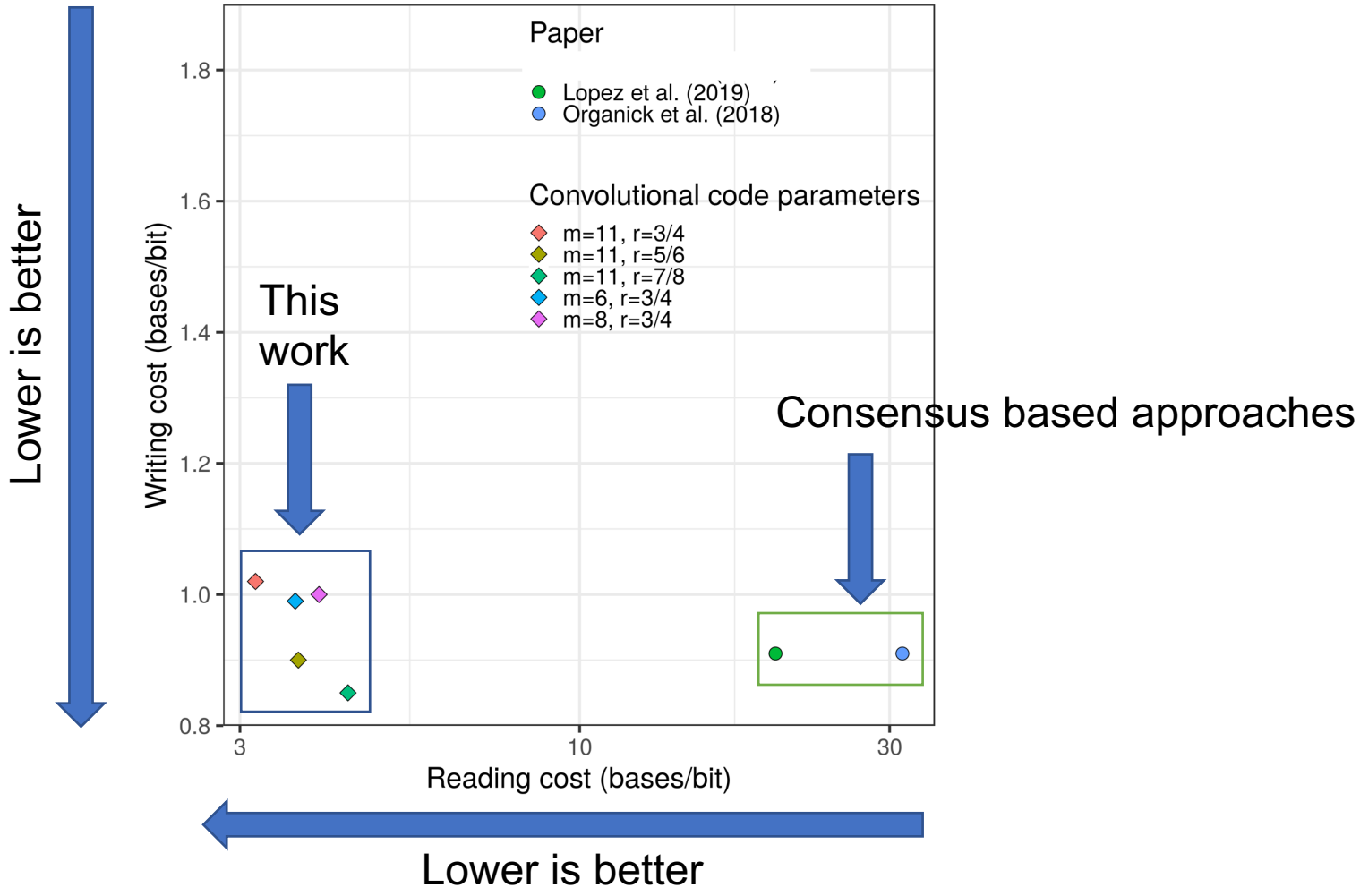


NN-based model simplifies the complex model into a simpler Markov model!
It repurposes the basecaller's NN model which is optimized based on large amounts of data.

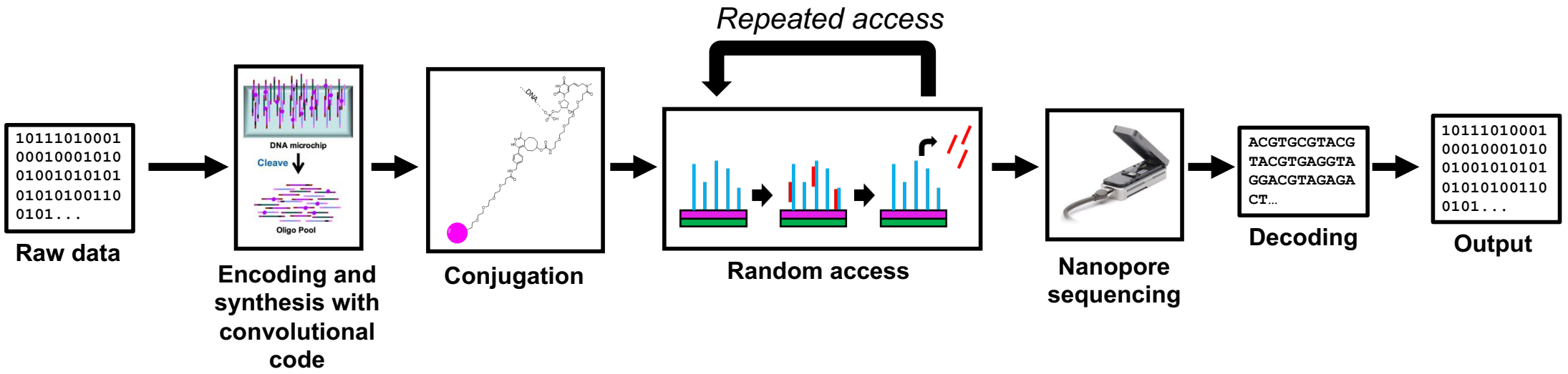
Integrated Decoding System



Results


Lower is better
Paper
Lopez et al. (2019)
Organick et al. (2018)
Convolutional code parameters
m=11, r=3/4
m=11, r=5/6
m=11, r=7/8
m=6, r=3/4
m=8, r=3/4
This work
Consensus based approaches
Writing cost (bases/bit)
1.8
1.6
1.4
1.2
1.0
0.8
3
10
30
Reading cost (bases/bit)
Lower is better

Overall approach



Future directions

- Automation of DNA data retrieval with liquid handling robots
- Possibility of real-time data decoding with nanopore sequencers
- Design cheaper, possibly more error prone synthesis platforms

Thank you!