Introduction to Real-Time Raw Nanopore Signal Analysis: RawHash and RawHash2

Can Firtina
canfirtina@gmail.com

18 March 2024
Sabanci University
BIO310 - Introduction to Bioinformatics

SAFARI

ETH zürich

Brief Self Introduction

- **Can Firtina**
  - Ph.D. Candidate in SAFARI Research Group at ETH Zurich

- **Research interests:** Bioinformatics & Computer Architecture
  - Real-time genome analysis
  - Similarity search in a large space of genomic data
  - Hardware-Algorithm co-design to accelerate genome analysis
  - Genome editing
  - Error correction

- Get to know **our group and our research**
  - **Group website:** https://safari.ethz.ch/
  - **Contact me:** canfirtina@gmail.com
  - **Website:** https://cfirtina.com
  - **Twitter (aka X):** https://twitter.com/FirtinaC

Professor Mutlu

- Onur Mutlu
  - Full Professor @ ETH Zurich ITET (INFK), since September 2015
  - Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-…
  - PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
  - https://people.inf.ethz.ch/omutlu/
  - omutlu@gmail.com (Best way to reach)
  - https://people.inf.ethz.ch/omutlu/projects.htm

- Research and Teaching in:
  - Computer architecture, computer systems, hardware security, bioinformatics
  - Memory and storage systems
  - Hardware security, safety, predictability
  - Fault tolerance
  - Hardware/software cooperation
  - Architectures for bioinformatics, health, medicine
  - …

SAFARI Research Group

Computer architecture, HW/SW, systems, bioinformatics, security, memory

SAFARI
SAFARI Research Group
safari.ethz.ch

40+ Researchers

Think BIG, Aim HIGH!

https://safari.ethz.ch

Four Key Current Directions

- Fundamentally Secure/Reliable/Safe Architectures

- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures

- Fundamentally Low-Latency and Predictable Architectures

- Algorithms & Architectures for AI/ML, Genomics, Medicine

Agenda for Today

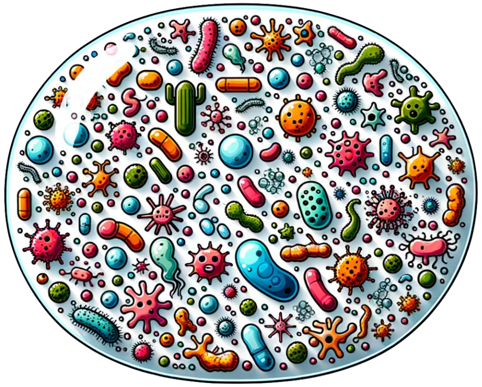
- Background
  - Sequence analysis
  - Raw nanopore signal analysis and real-time analysis

- Enabling Fast and Accurate Real-time Analysis
  - RawHash and RawHash2

- Conclusion

Sequence Analysis – Why?

Understanding **genetic variations, species, and evolution**



Predicting the **presence of pathogens** in an environment

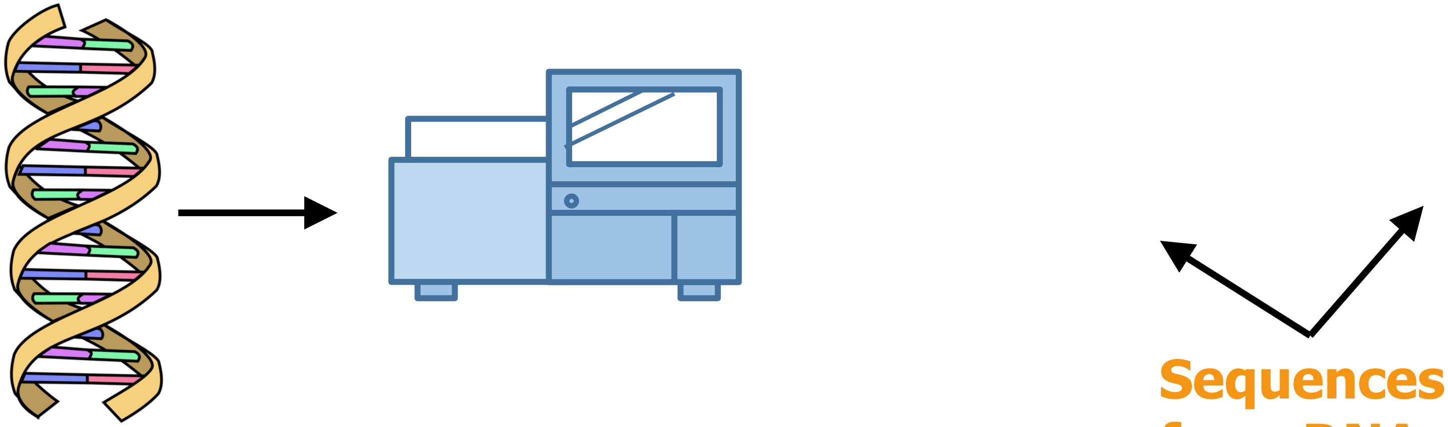
Surveillance of **disease outbreaks**

Personalized medicine

And, many, many other applications …

Sequence Analysis – How?

- **High throughput sequencing machines**
  - Quickly converts biological molecules into sequences of characters for analysis



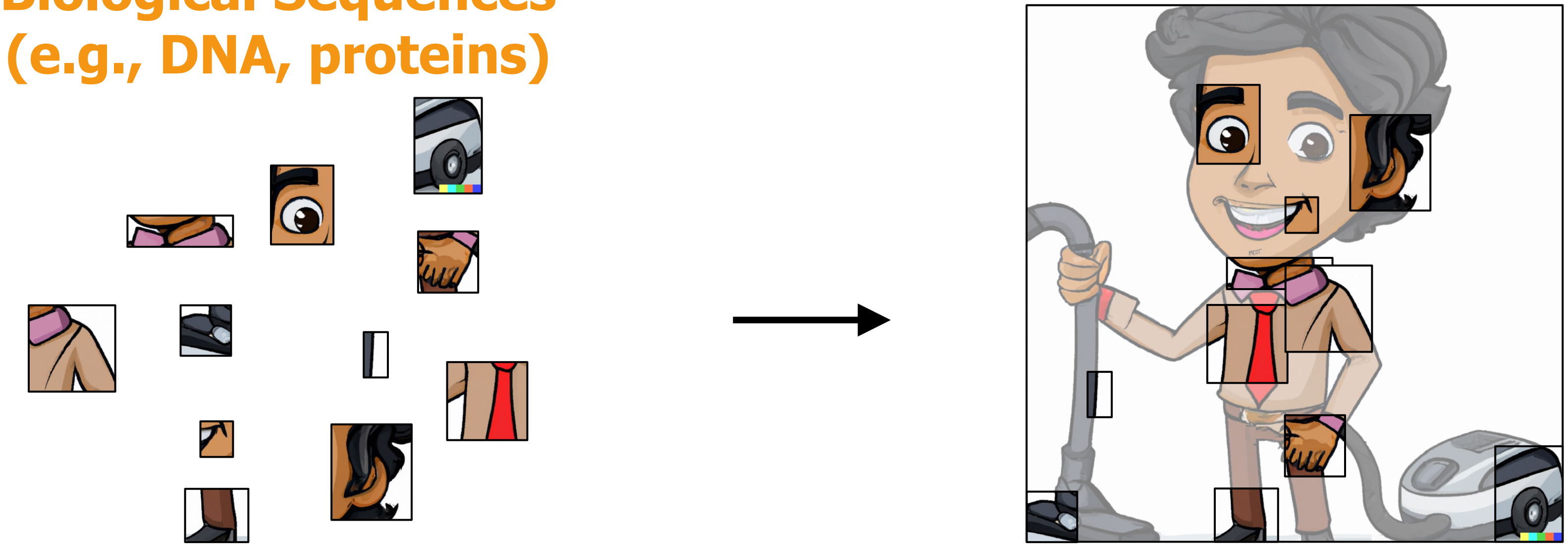
Biological Molecule (e.g., DNA)

Sequences from DNA (Reads)

Sequence Comparison is Essential

- Analyze sequences by accurately and quickly **comparing**
  - To **each other**
  - To a **template sequence** (e.g., a reference genome)

Biological Sequences (e.g., DNA, proteins)



- Essential to understand functionality of a sequence, mutations, diseases…

A Naïve Sequence Comparison Approach

- Read mapping:
  - **Mapping:** Identifies similar regions between a pair of sequences
  - **Alignment:** Identifies exact differences within similar regions (costly!)

Reference
Read

Very expensive!

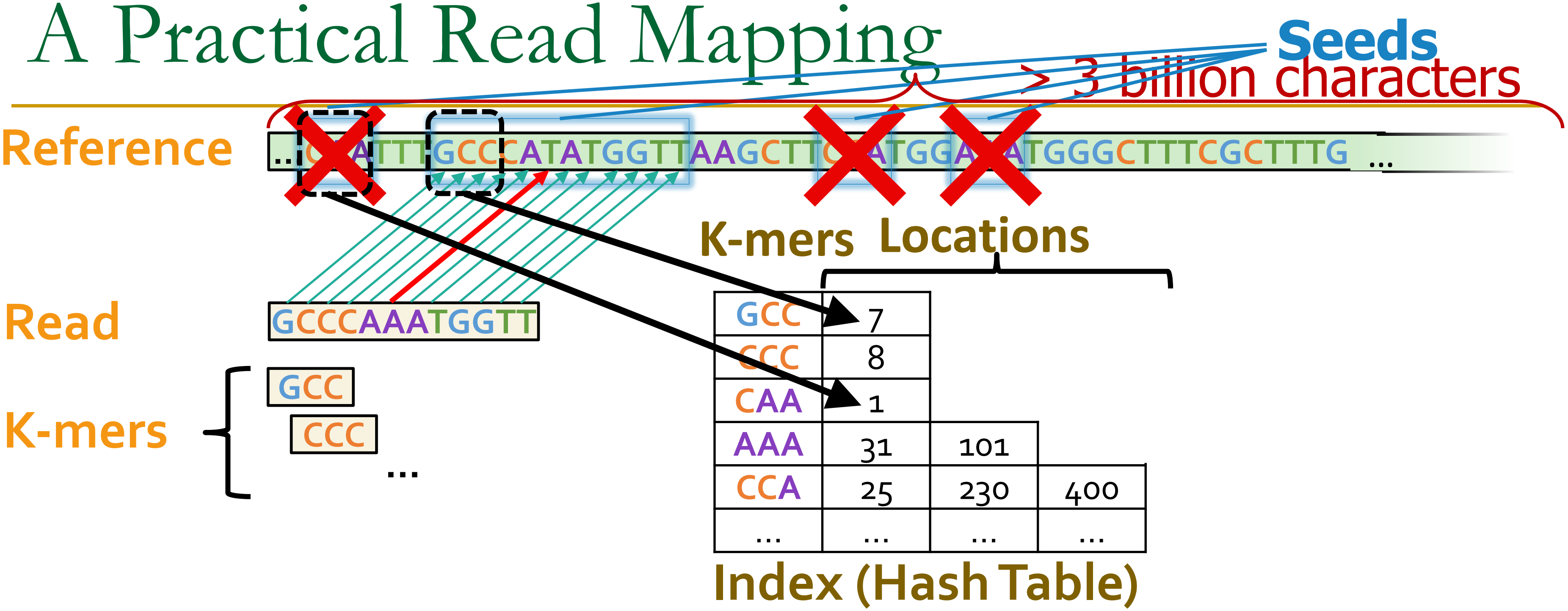
$O(m^2kn)$

m: read length

k: no. of reads

n: reference genome length

A Practical Read Mapping



GCC	7		
CCC	8		
CAA	1		
AAA	31	101	
CCA	25	230	400
…	…	…	…

Indexing	Store certain k-mers with their positions for fast query
Seeding	Determine potential matching regions (seeds)
Seed Filtering	Prune uninformative/unreliable seeds
Alignment	Determine the exact differences

Sequencing Output and Challenges

Small pieces of a puzzle
short reads (Illumina)

Large pieces of a puzzle
long reads (Nanopore & PacBio)

Which sequencing technology is the best?

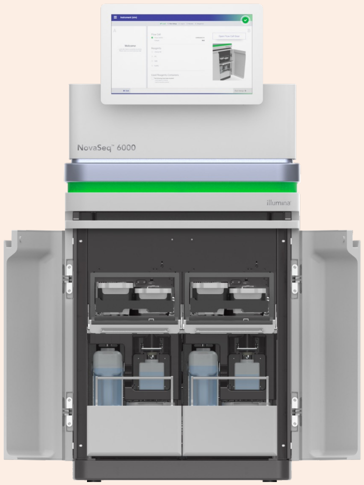
- ❑ 100-300 bp
- ❑ low error rate (~0.1%)

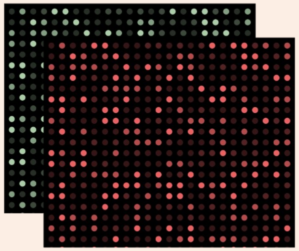
- ❑ 500-2M bp
- ❑ high error rate (~5%)

https://www.pacb.com/smrt-science/smrt-sequencing/hifi-reads-for-highly-accurate-long-read-sequencing/

Different Raw Sequencing Data

Illumina

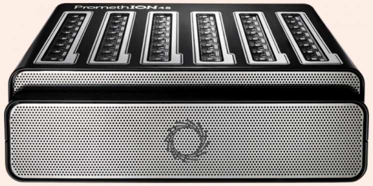


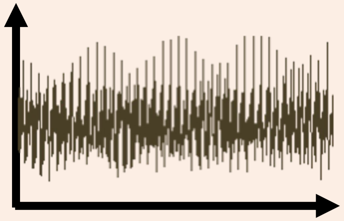


Multiple images

.BCL/.CBCL

Nanopore

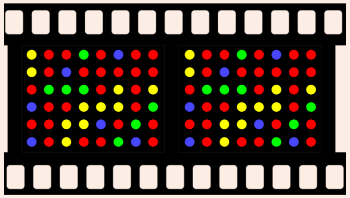




Electrical Signal

.POD5

PacBio



30-hour movie

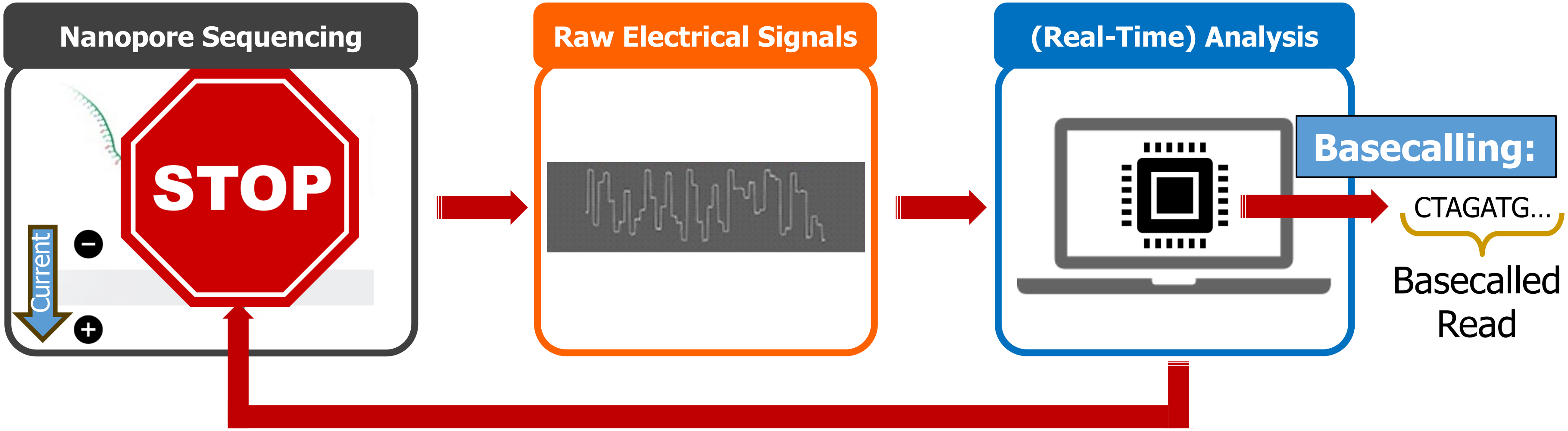
.BAM

Nanopore Sequencing

Nanopore Sequencing: a widely used sequencing technology

- **Long** reads (up to >2 million nucleotides)
- **Portable** sequencing
- Cost-effective
- More unique features: **real-time analysis**

Nanopore Sequencing & Real-time Analysis



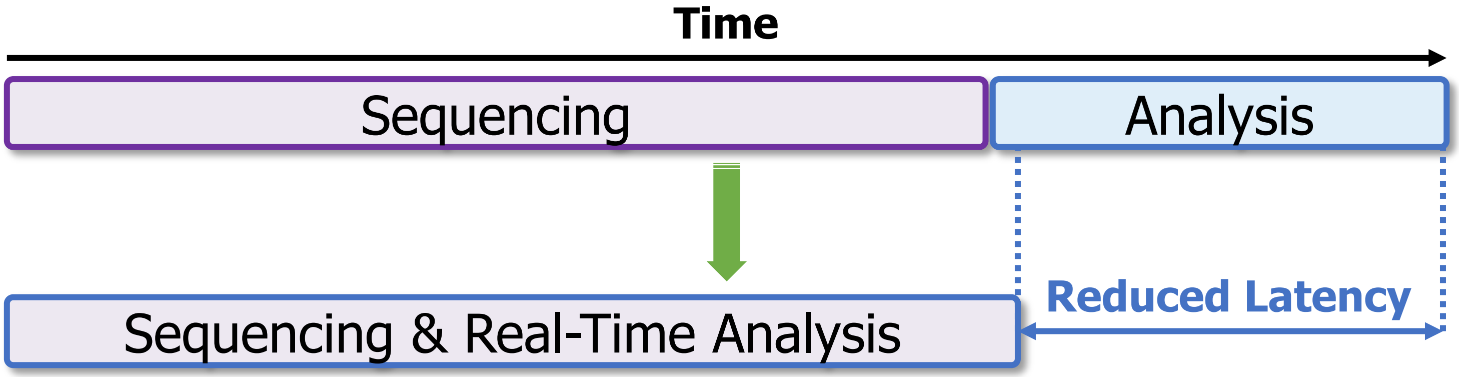
Raw Signals: Ionic current measurements generated as DNA passes through the **nanopore at a certain speed**

(Real-Time) Analysis: Translating to bases or directly analyzing raw signals

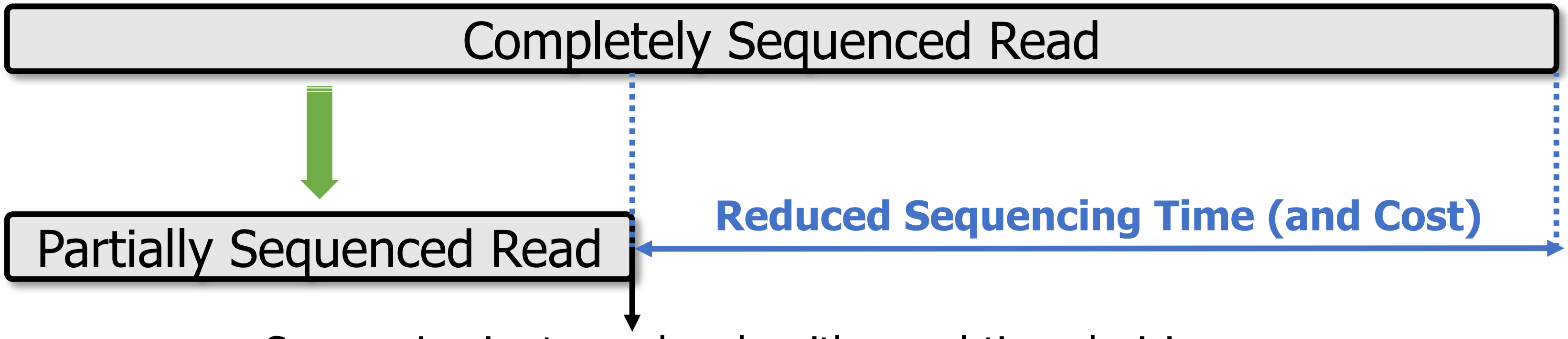
Real-Time Decisions: Stopping sequencing **early** based on real-time analysis

Benefits of Real-Time Analysis

✓ **Reducing latency** by overlapping the sequencing and analysis steps



✓ **Reducing sequencing time and cost** by stopping sequencing early



Sequencing is stopped early with a real-time decision

Challenges in Real-Time Analysis

Rapid analysis to match the nanopore sequencer throughput

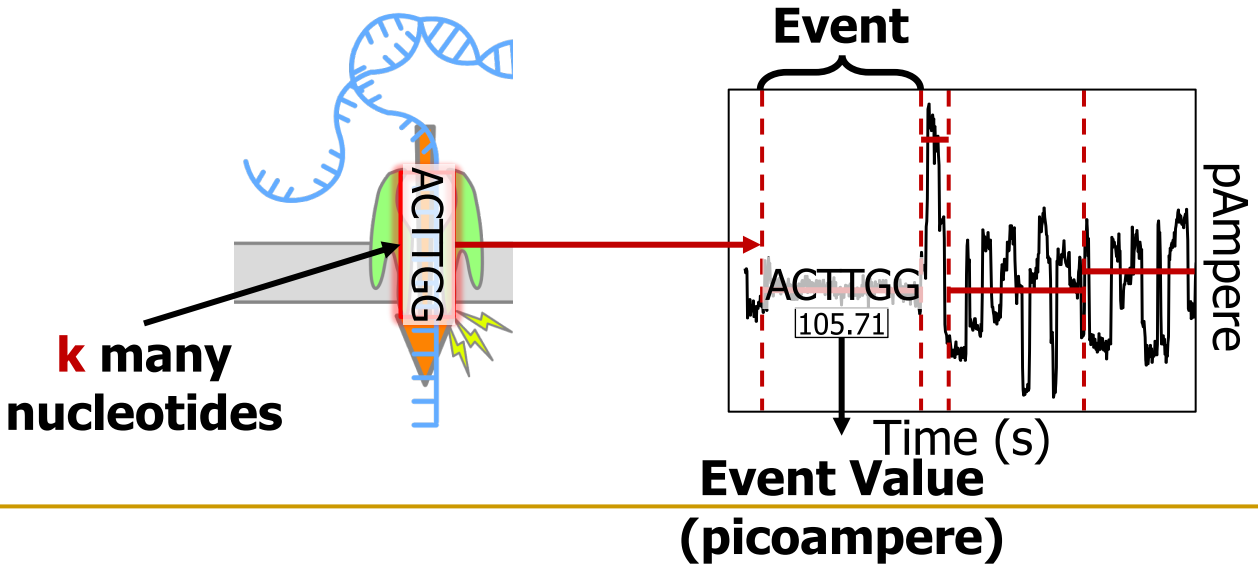
Timely decisions to stop sequencing as early as possible

Accurate analysis from noisy raw signal data

Power-efficient computation for scalability and portability

Enabling Analysis From Electrical Signals

- **K** many nucleotides (**k**-mers) sequenced at a time
- **Event:** A **segment** of the raw signal
  - Corresponds to a **particular k**-mer
  - Abrupt signal changes show sequencing of a new k-mer
  - **Statistical methods** can find these abrupt changes
  - **Event value:** average of signals **within an event**
- **Observation: Identical** k-mers generate **similar** event values during sequencing



Agenda for Today

- Background
  - Sequence analysis
  - Raw nanopore signal analysis and real-time analysis

- Enabling Fast and Accurate Real-time Analysis
  - RawHash and RawHash2

- Conclusion

ISMB/ECCB 2023

RawHash

Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes

Can Firtina

Nika Mansouri Ghiasi Joel Lindegger Gagandeep Singh

Meryem Banu Cavlak Haiyu Mao Onur Mutlu

Paper

Code

SAFARI

ETH zürich

Executive Summary

Problem: Real-time analysis of nanopore raw signals is **inaccurate** and **inefficient for large genomes**

Goal: Enable **fast** and **accurate** real-time analysis of raw signals for **large genomes**

Key Contributions:
1) The **first hash-based mechanism** that can quickly and accurately analyze raw nanopore signals for **large genomes**

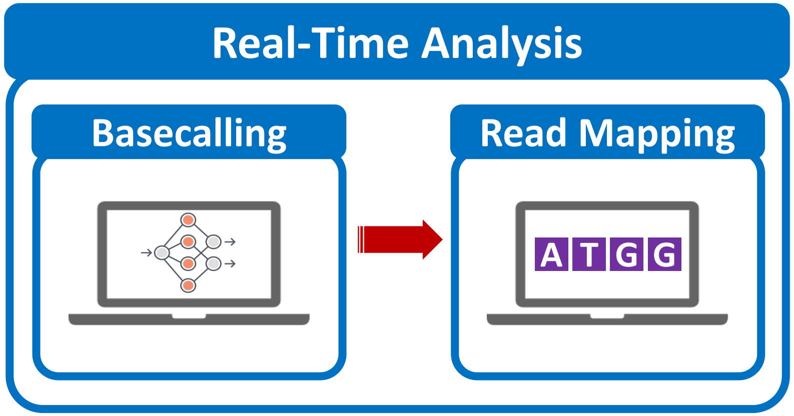
2) The novel **Sequence Until** technique can accurately and **dynamically stop the entire sequencing of all reads at once** if further sequencing is not necessary

Key Results: Across 3 use cases and 5 genomes of varying sizes, RawHash provides

- **25.8× and 3.4× better average throughput** compared to two state-of-the-art works
- **1.14× – 2.13× more accurate mapping results** for **large genomes**
- Sequence Until **reduces the sequencing time and cost by 15×**

Existing Solutions

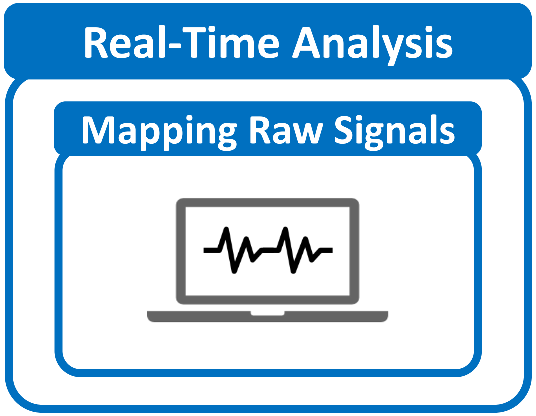
1. Deep neural networks (**DNNs**) for translating signals to **bases**



Less noisy analysis from basecalled sequences

Costly and power-hungry computational requirements

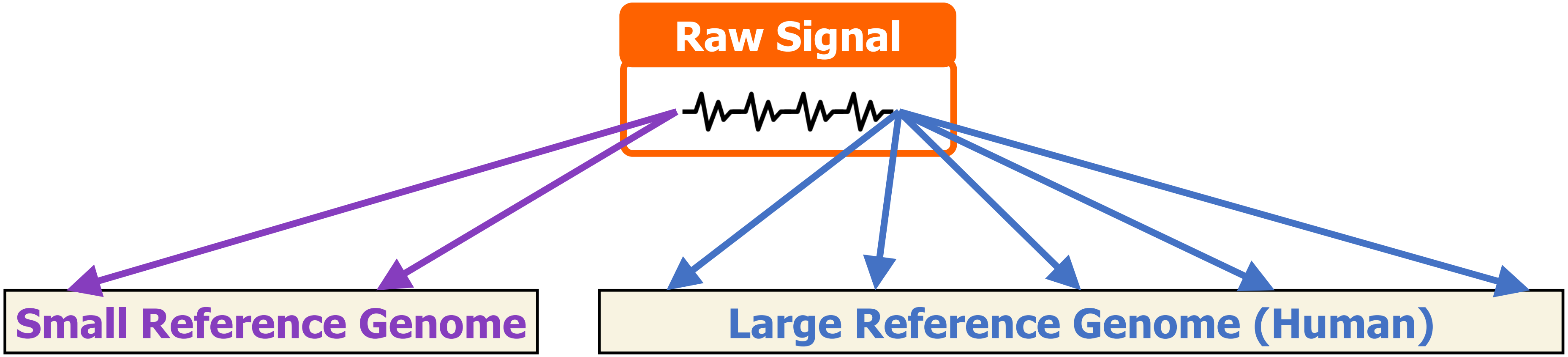
2. Mapping **signals** to reference genomes **without** basecalling



Raw signals contain richer information than bases

Efficient analysis with better scalability and portability

The Problem – Mapping Raw Signals



Fewer candidate regions in **small genomes**

Substantially **larger number of regions** to check **per read** as the genome size increases

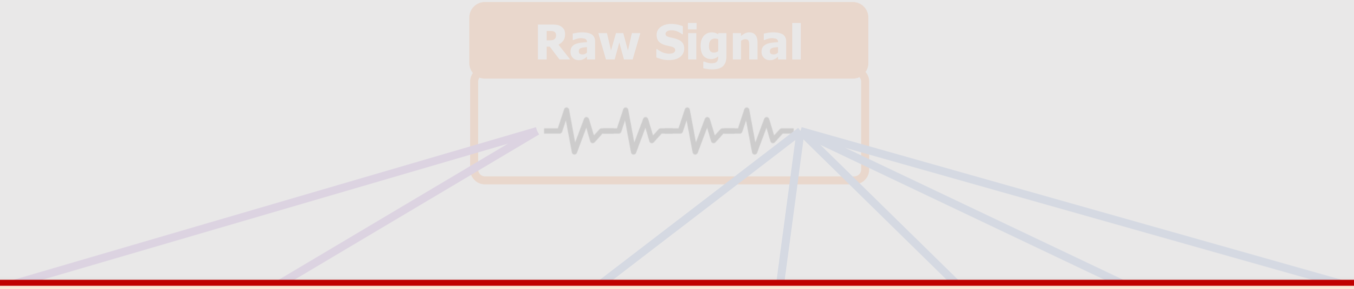
Accurate mapping

Problem: Probabilistic mechanisms on **many regions → inaccurate mapping**

High throughput

Problem: Distance calculation on **many regions → reduced throughput**

The Problem – Mapping Raw Signals



Existing solutions are
inaccurate or inefficient
for large genomes

Accurate mapping

on **many regions → inaccurate mapping**

High throughput

Problem: Distance calculation
on **many regions → reduced throughput**

Outline

Background

RawHash

Evaluation

Conclusion

Goal

Enable **fast and accurate real-time analysis** of raw nanopore signals **for large genomes**

RawHash

The **first hash-based search mechanism** to quickly and accurately map raw nanopore signals to reference genomes

Sequence Until can accurately and **dynamically stop the entire sequencing run at once** if further sequencing is unnecessary

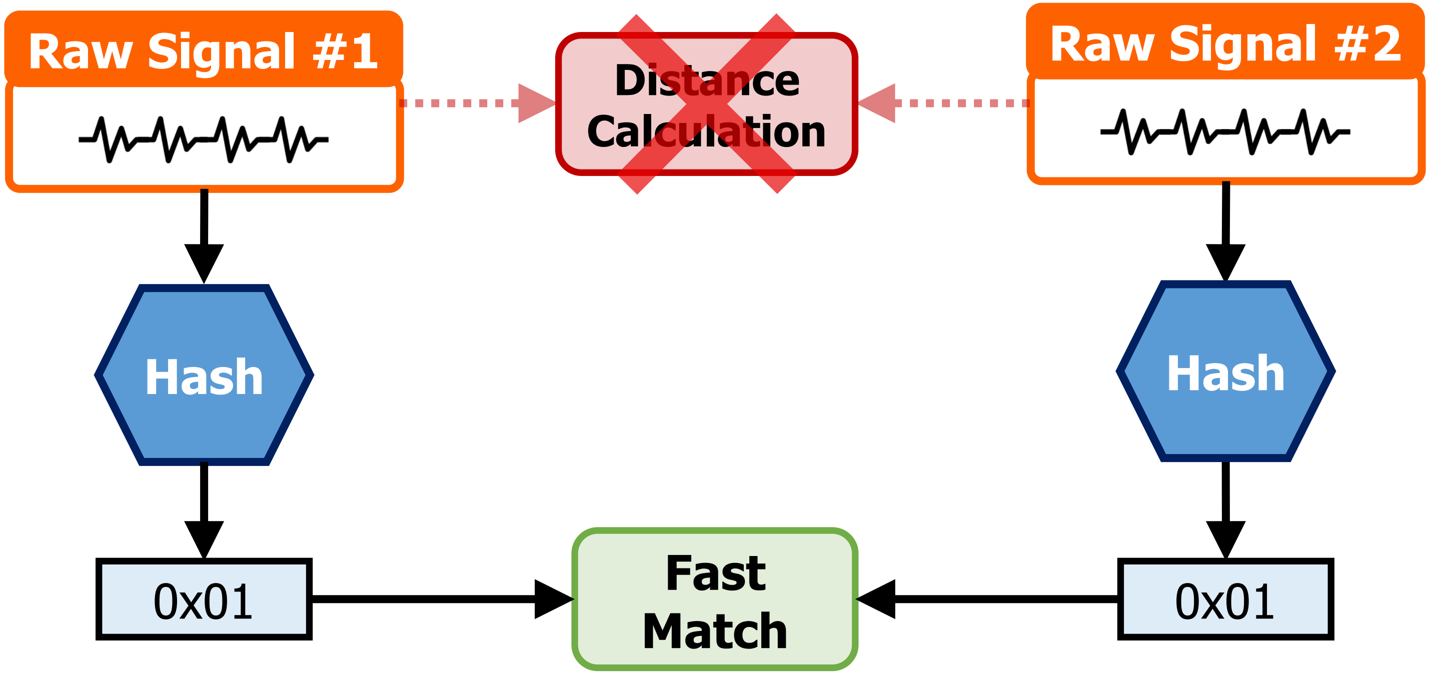
RawHash

The **first hash-based search mechanism** to quickly and accurately map raw nanopore signals to reference genomes

Sequence Until can accurately and **dynamically stop the entire sequencing run at once** if further sequencing is unnecessary

RawHash – Key Idea

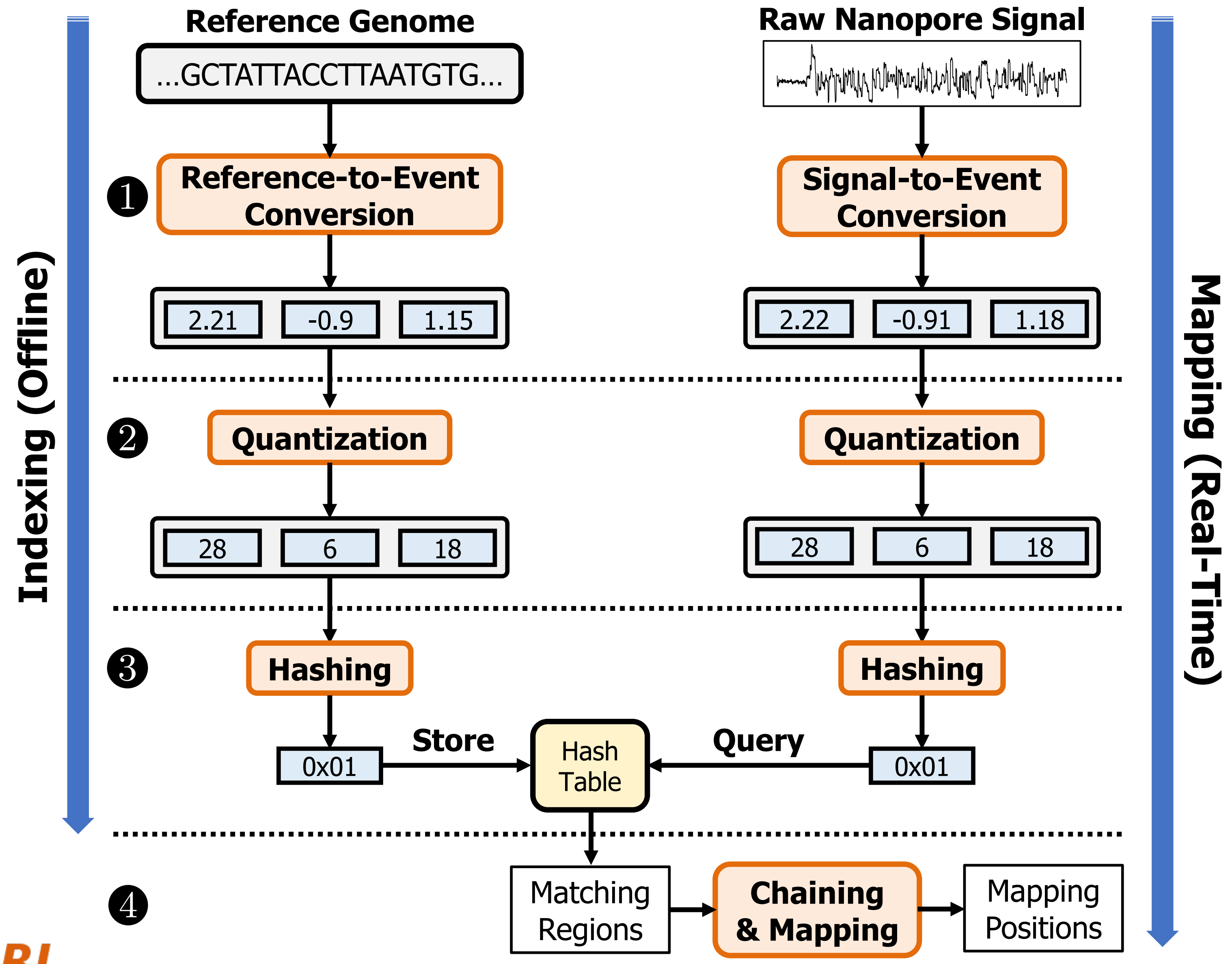
Key Observation: Identical nucleotides generate **similar** raw signals



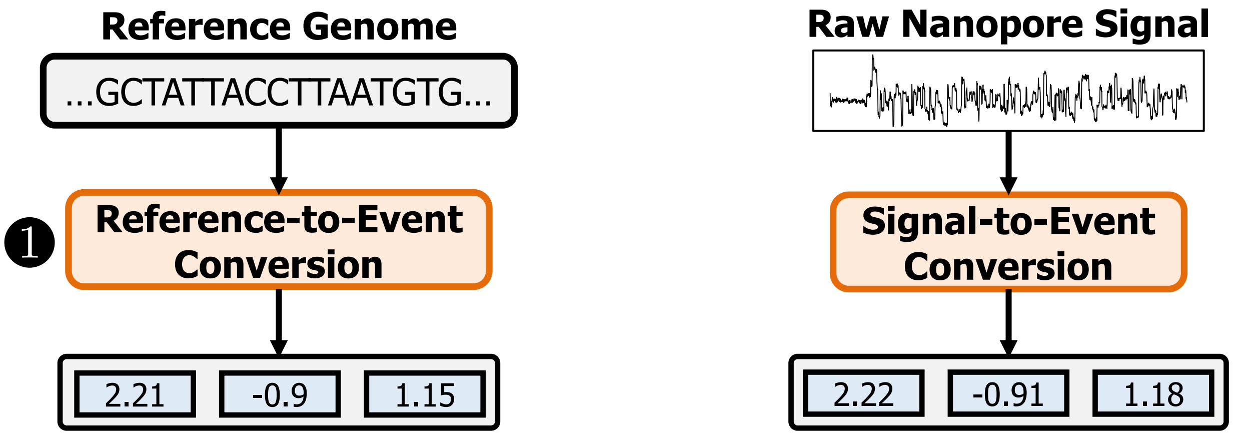
Challenge #1: Generating the **same** hash value for **similar enough** signals

Challenge #2: Accurately finding as **few** similar regions as possible

RawHash Overview



RawHash Overview



Reference Genome

...GCTATTACCTTAATGTG...

❶ Reference-to-Event Conversion

2.21 -0.9 1.15

Raw Nanopore Signal

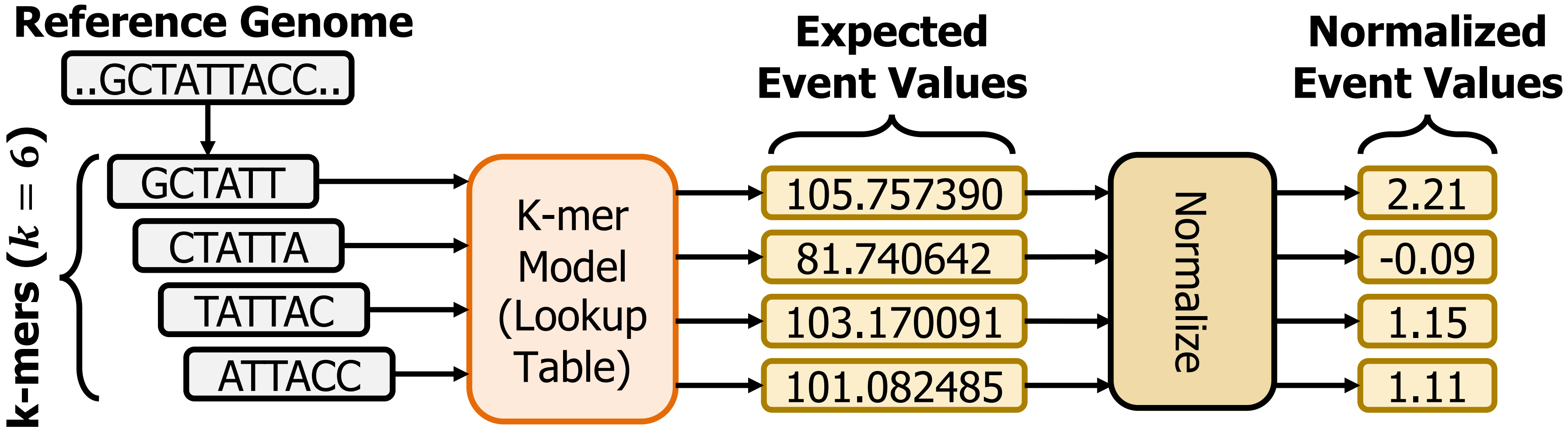
Signal-to-Event Conversion

2.22 -0.91 1.18

Reference-to-Event Conversion

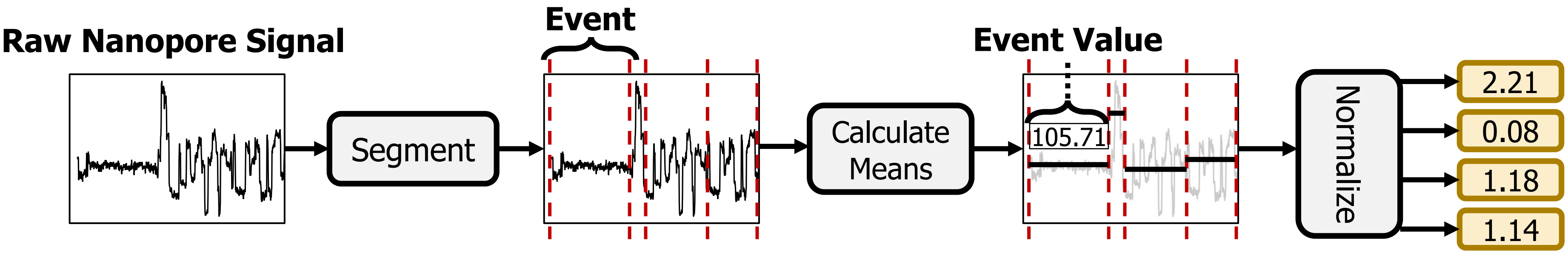
- **K-mer model:** Provides **expected** event values **for each k-mer**
  - Preconstructed based on nanopore sequencer characteristics

- Use the **k-mer model** to convert **all k-mers** of a reference genome to their **expected** event values



Signal-to-Event Conversion

- **Event detection:** Identifies signal regions corresponding to specific k-mers
  - Uses statistical test (**segmentation**) to spot abrupt signal changes



- Consecutive events ➔ consecutive k-mers

Signal-to-Event Conversion

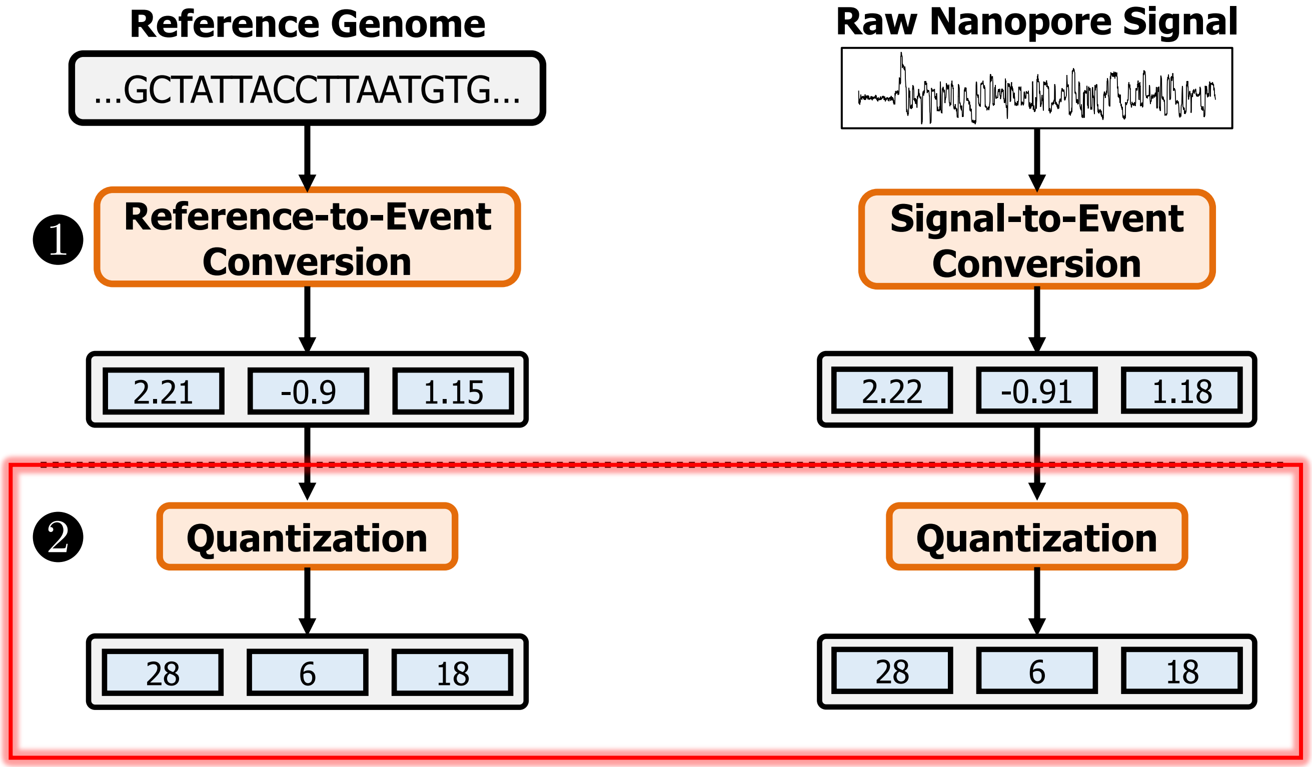
- **Event detection:** Identifies signal regions corresponding to specific k-mers
  - Uses statistical test (**segmentation**) to spot abrupt signal changes

Can we directly match signals to each other?

- Consecutive events ➔ consecutive k-mers

SAFARI

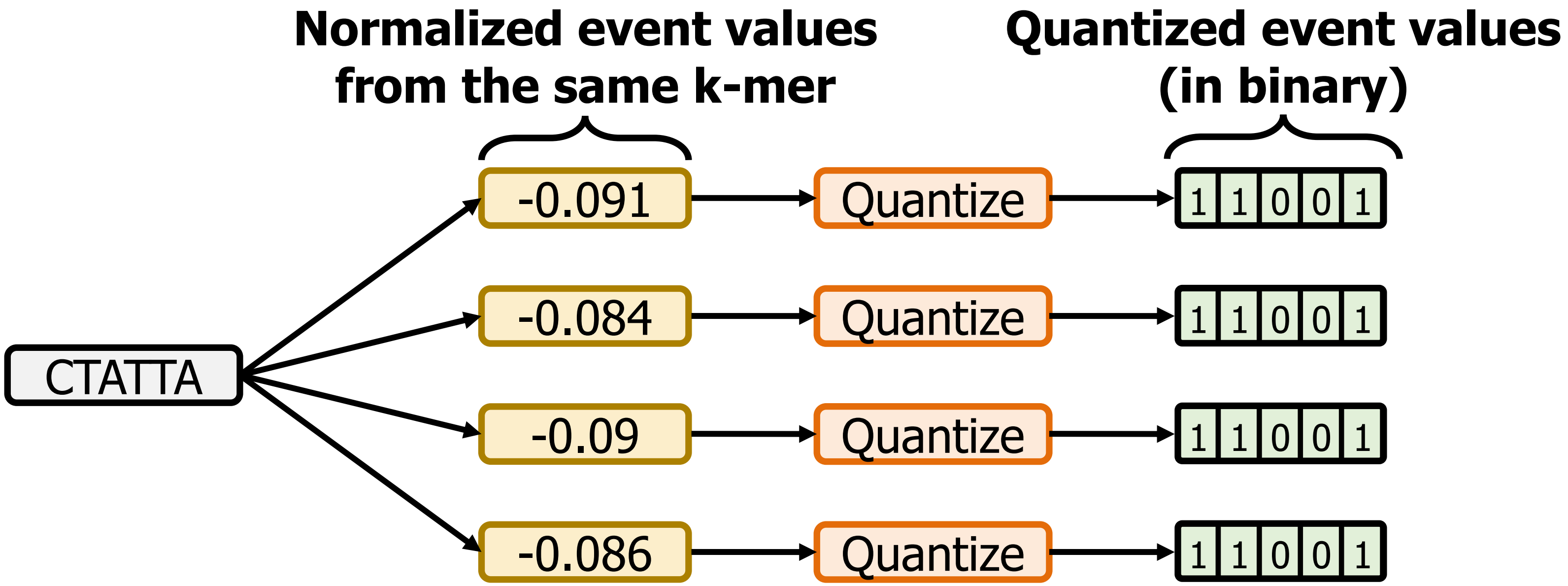
RawHash Overview



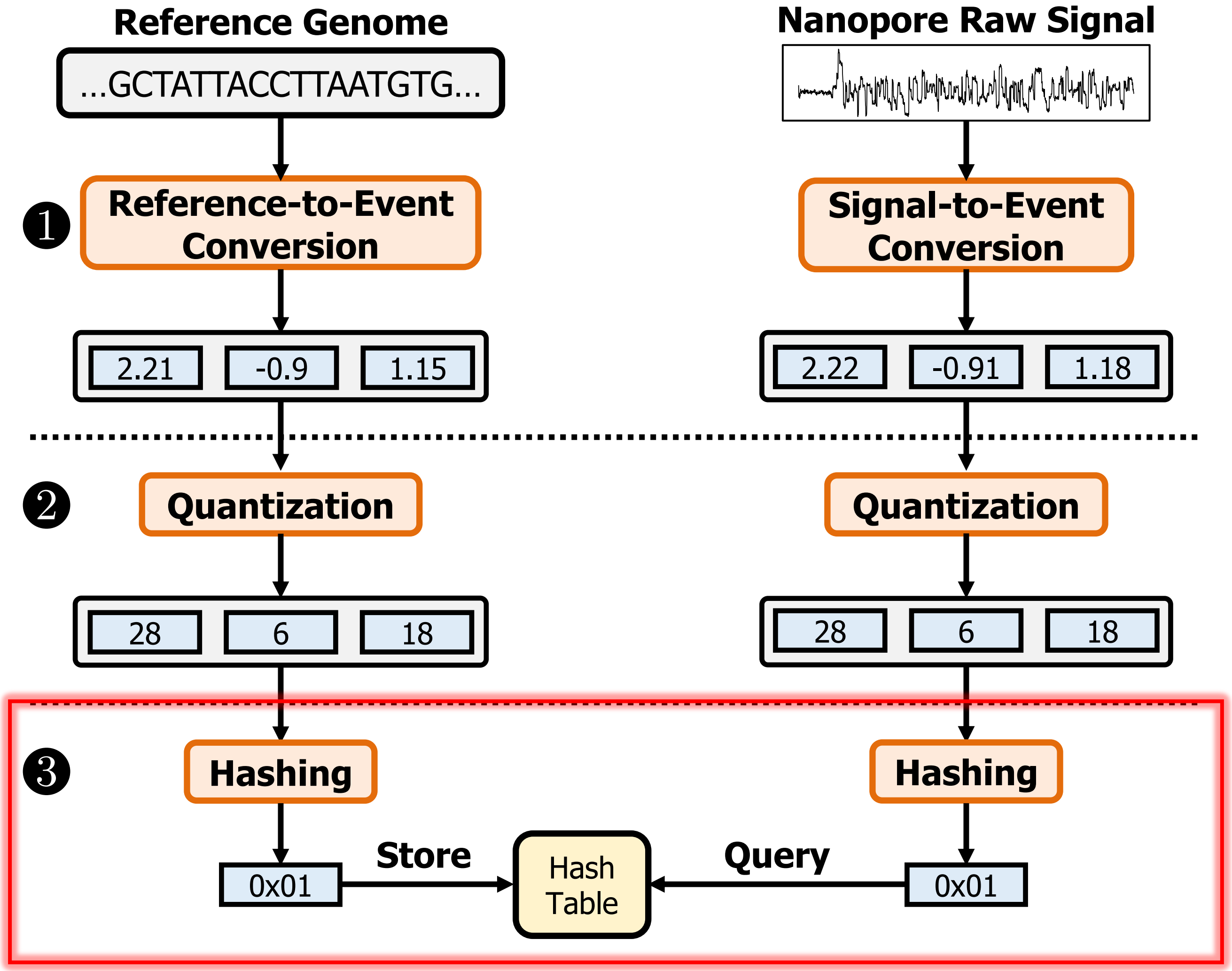
Quantizing the Event Values

- **Observation:** Slight differences in raw signals from identical k-mers
  - **Challenge:** Direct event value matching is not feasible and accurate

- **Key Idea:** Quantize the event values
  - Enables assigning **identical quantized values** to **similar event values**



RawHash Overview



Reference Genome

...GCTATTACCTTAATGTG...

Nanopore Raw Signal

❶ Reference-to-Event Conversion

Signal-to-Event Conversion

2.21	-0.9	1.15

2.22	-0.91	1.18

❷ Quantization

Quantization

28	6	18

28	6	18

❸ Hashing

Hashing

0x01

Store

Hash Table

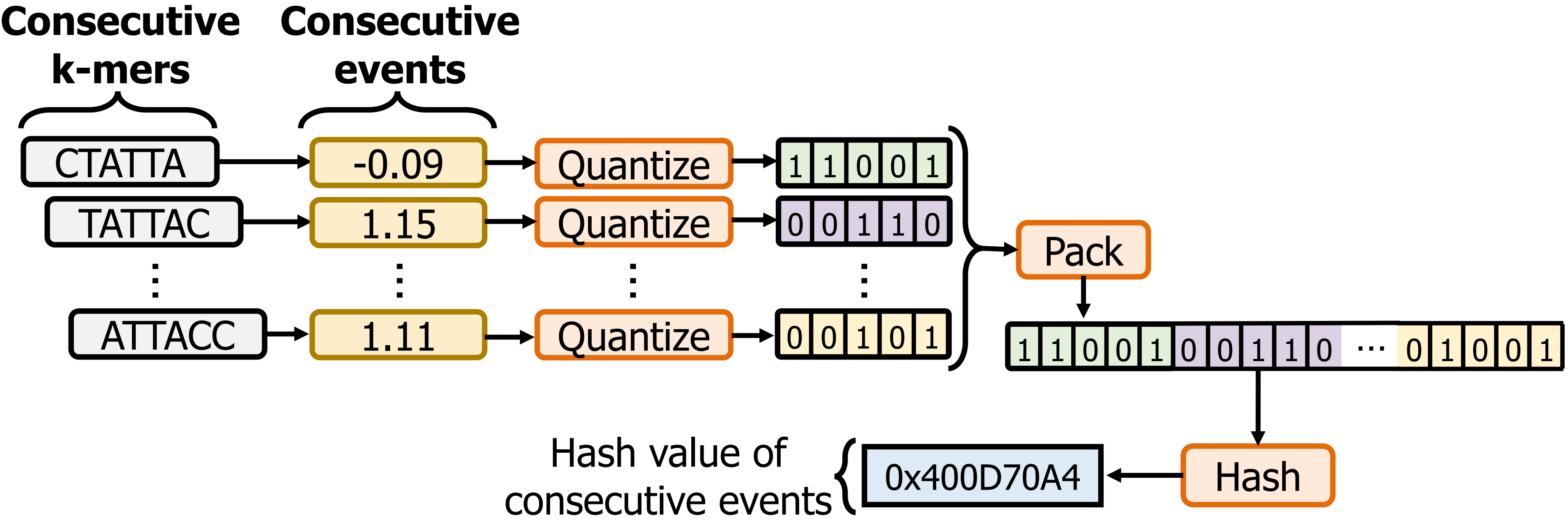
Query

0x01

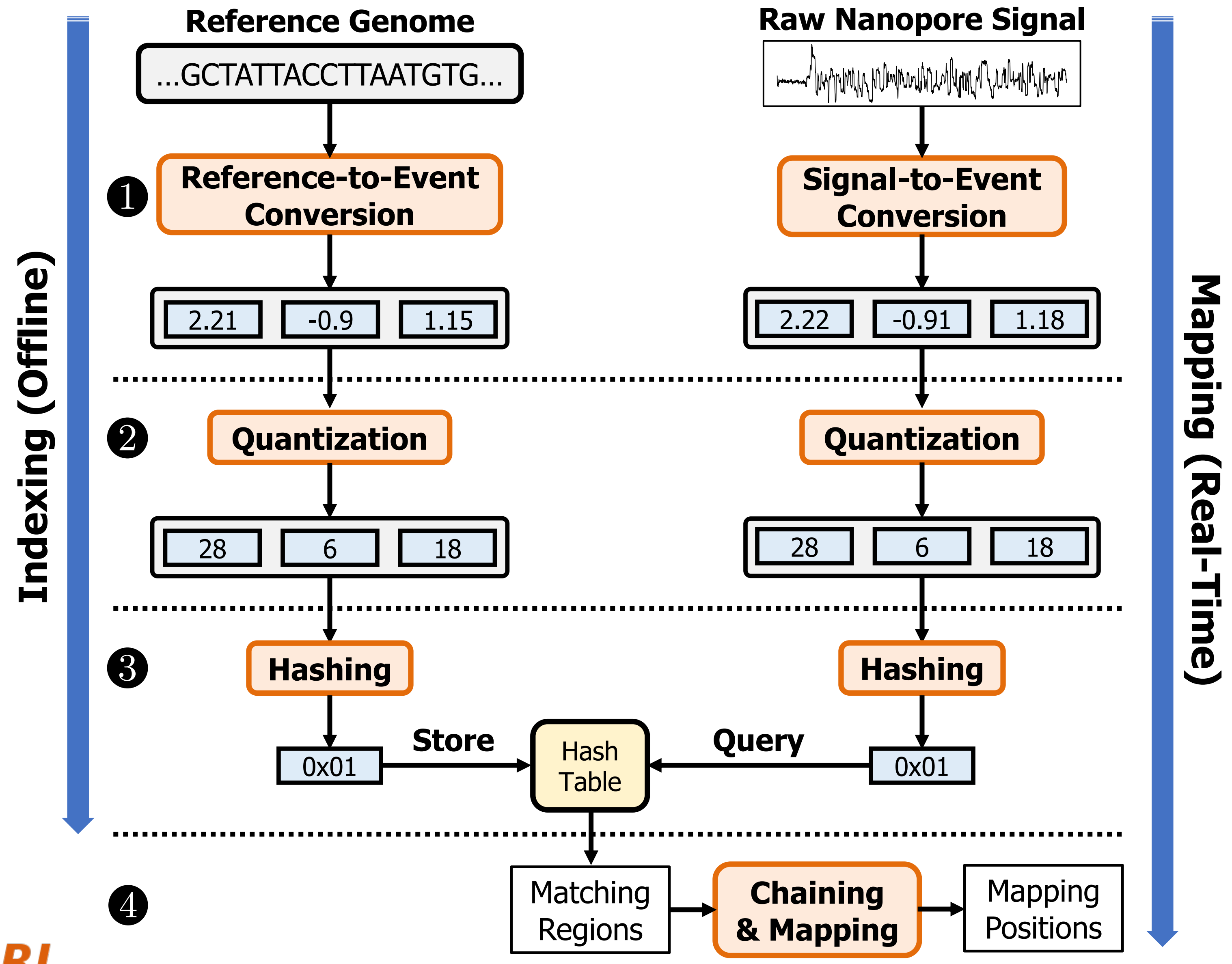
Hashing for Fast Similarity Search

- Each event usually represents a very small k-mer (6 to 9 characters)
  - **Challenge:** Short k-mers are likely to appear in many locations

- **Key Idea:** Create longer k-mers from many **consecutive events**
- **Key Benefit:** Directly match hash values to quickly identify similarities



RawHash Overview



Indexing (Offline)

Reference Genome

...GCTATTACCTTAATGTG...

❶ Reference-to-Event Conversion

2.21 | -0.9 | 1.15

❷ Quantization

28 | 6 | 18

❸ Hashing

0x01

Store

Hash Table

Query

Raw Nanopore Signal

Signal-to-Event Conversion

2.22 | -0.91 | 1.18

Quantization

28 | 6 | 18

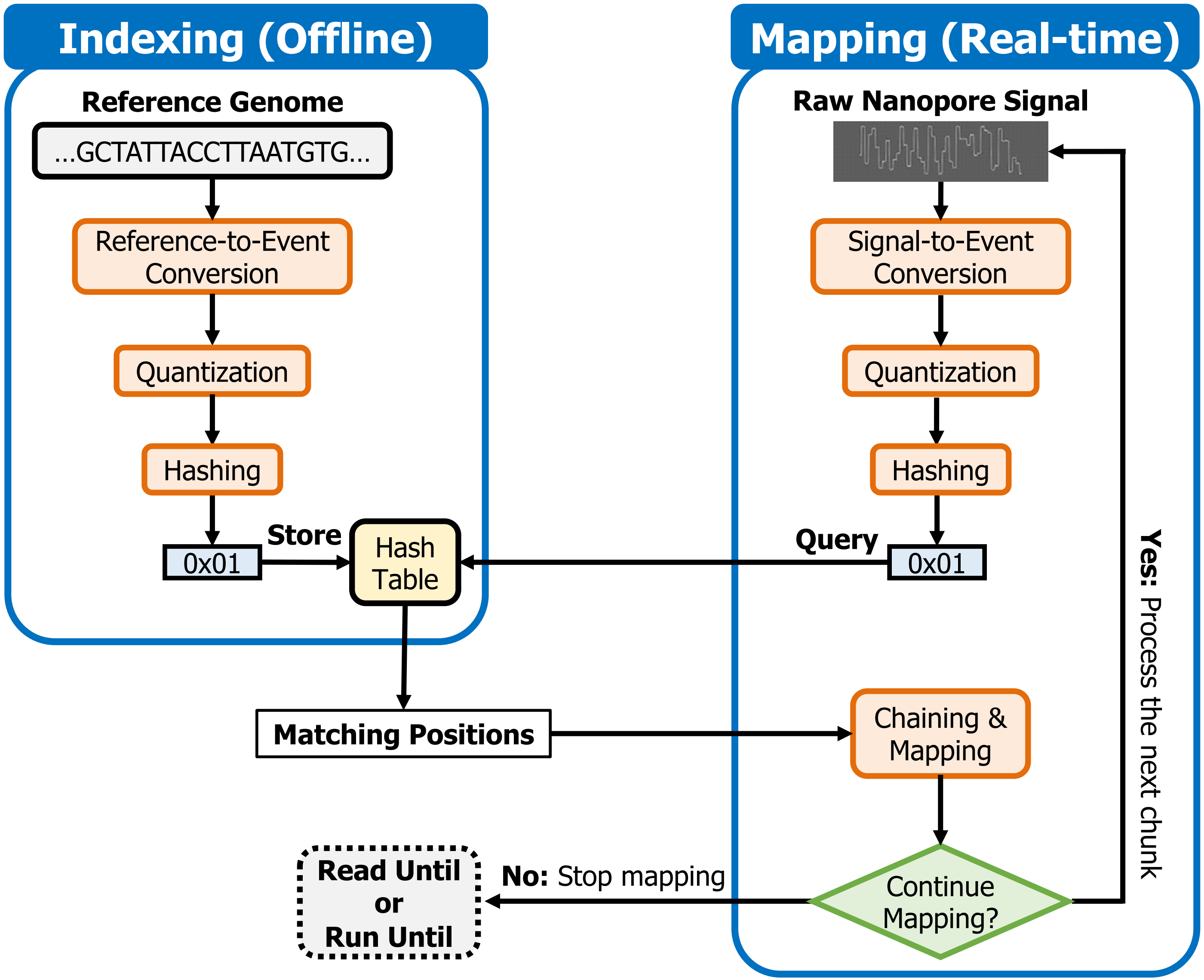
Hashing

0x01

Mapping (Real-Time)

❹ Matching Regions → Chaining & Mapping → Mapping Positions

Real-Time Mapping using Hash-based Indexing



RawHash

The **first hash-based search mechanism** to quickly and accurately map raw nanopore signals to reference genomes

Sequence Until can accurately and **dynamically stop the entire sequencing run at once** if further sequencing is unnecessary

RawHash

The **first hash-based search mechanism** to quickly and accurately map raw nanopore signals to reference genomes

Sequence Until can accurately and **dynamically stop the entire sequencing run at once** if further sequencing is unnecessary

The Sequence Until Mechanism

- **Problem:**
  - Unnecessary sequencing waste time, power and money

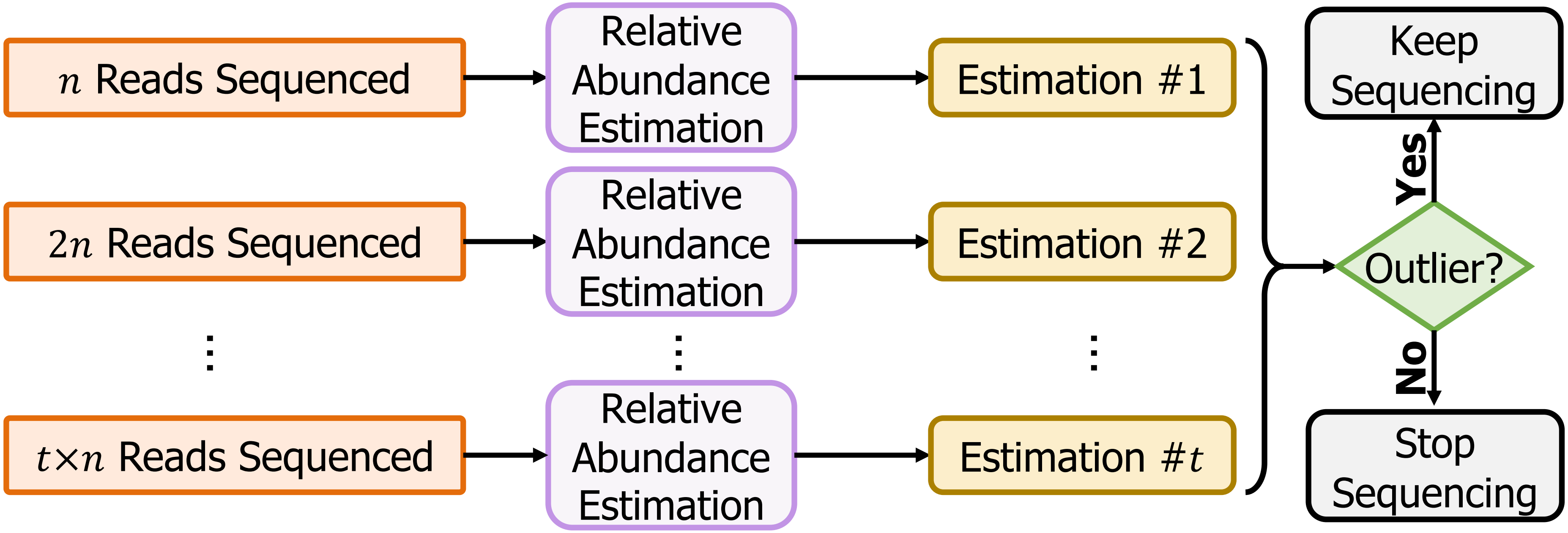
- **Key Idea:**
  - **Dynamically** decide if further sequencing of the entire sample is necessary to achieve high accuracy
  - Stop sequencing early without sacrificing accuracy

- **Potential Benefits:**
  - Significant **reduction in sequencing time and cost**

- Example real-time genome analysis use case:
  - **Relative abundance estimation**

The Sequence Until Mechanism

- **Key Steps:**
  1. Continuously generate relative abundance estimation after every n reads
  2. Keep the last t estimation results
  3. **Detect outliers** in the results via **cross-correlation** of the recent t results
  4. Absence of outliers indicates **consistent results**
     - Further sequencing **is likely** to generate consistent results ➔ Stop the sequencing



Outline

Background

RawHash

Evaluation

Conclusion

Evaluation Methodology

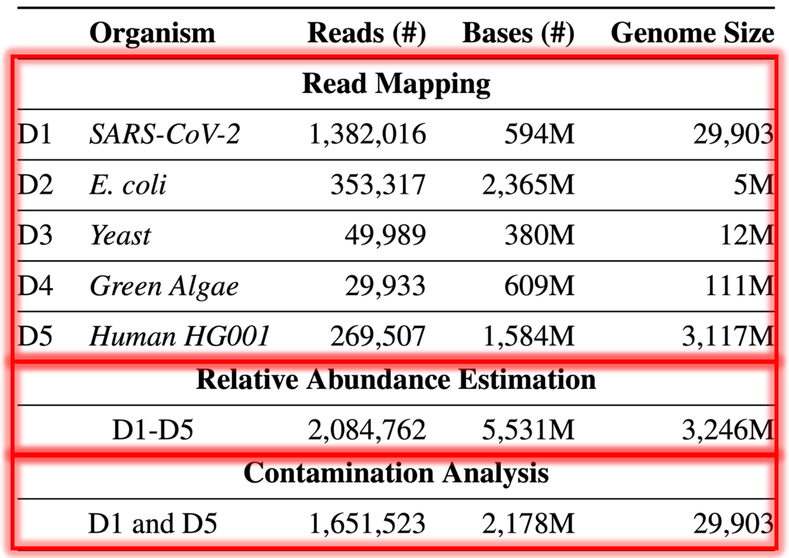
- Compared to **UNCALLED** [Kovaka+, Nat. Biotech. 2021] and **Sigmap** [Zhang+, ISMB/ECCB 2021]
  - **CPU baseline:** AMD EPYC 7742 @2.26GHz
  - **32 threads** for each tool

- **Use cases** for real-time genome analysis:
  1. Read mapping
  2. Relative abundance estimation
     - **Benefits of Sequence Until**
  3. Contamination analysis

Evaluation Methodology

- Evaluation metrics:
  - **Throughput** (bases processed per second)
  - Potential reduction in **sequencing time and cost**
  - **Accuracy**
    - **Baseline:** Mapping basecalled reads using minimap2
    - Precision, recall, and F1 scores
    - Relative abundance estimation distance to ground truth

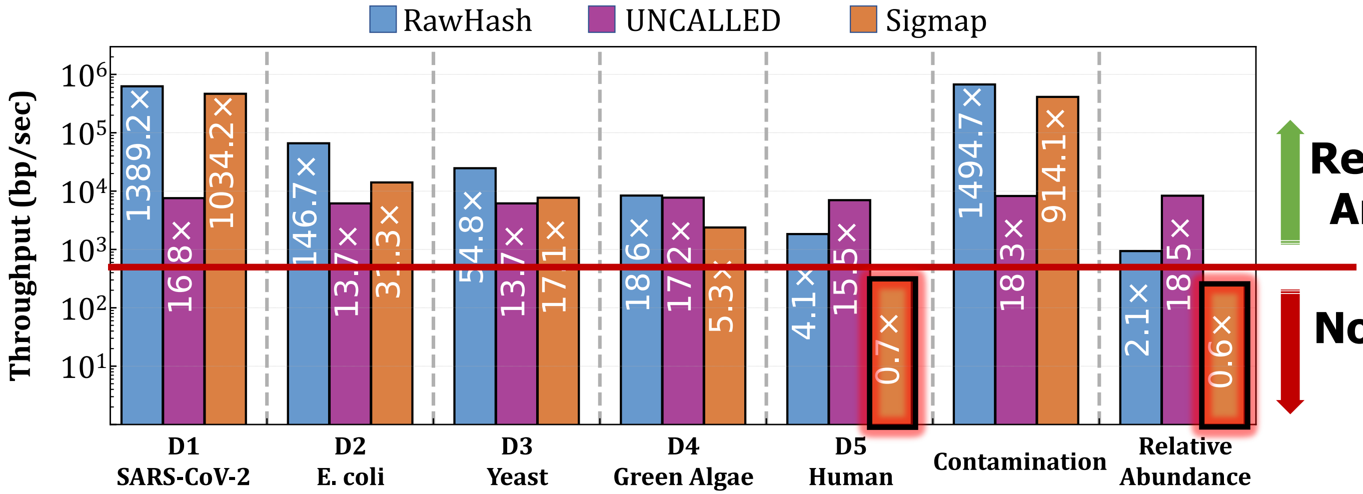
- **Datasets:**



	Organism	Reads (#)	Bases (#)	Genome Size
	Read Mapping			
D1	*SARS-CoV-2*	1,382,016	594M	29,903
D2	*E. coli*	353,317	2,365M	5M
D3	*Yeast*	49,989	380M	12M
D4	*Green Algae*	29,933	609M	111M
D5	*Human HG001*	269,507	1,584M	3,117M
	Relative Abundance Estimation			
	D1-D5	2,084,762	5,531M	3,246M
	Contamination Analysis			
	D1 and D5	1,651,523	2,178M	29,903

Throughput

- **Real-time analysis requires** faster throughput than sequencer
  - Throughput of a nanopore sequencer: ~**450 bp/sec (data generation speed)**



Real-Time Analysis

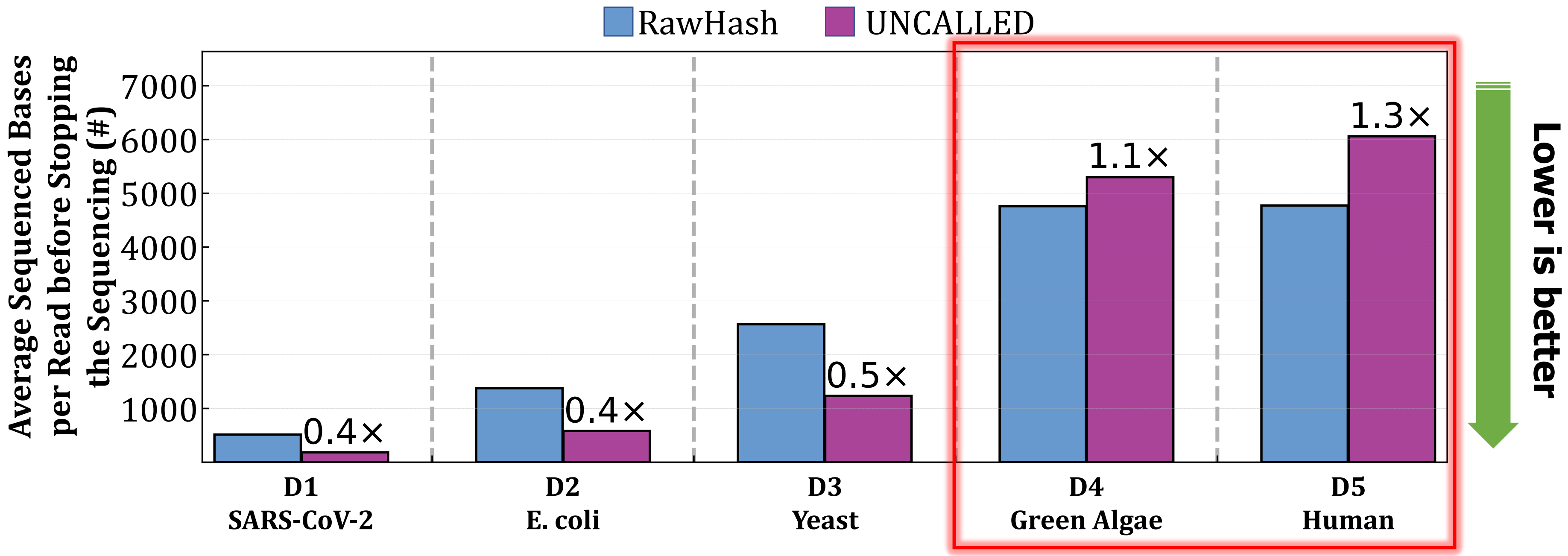
No Real-Time Analysis

25.8× **and 3.4**× better average throughput compared to **UNCALLED and Sigmap**, respectively

Sigmap **cannot** perform real-time analysis **for large genomes**

Sequencing Time

- Fewer bases to sequence ➔
  - Reduction in sequencing time and cost

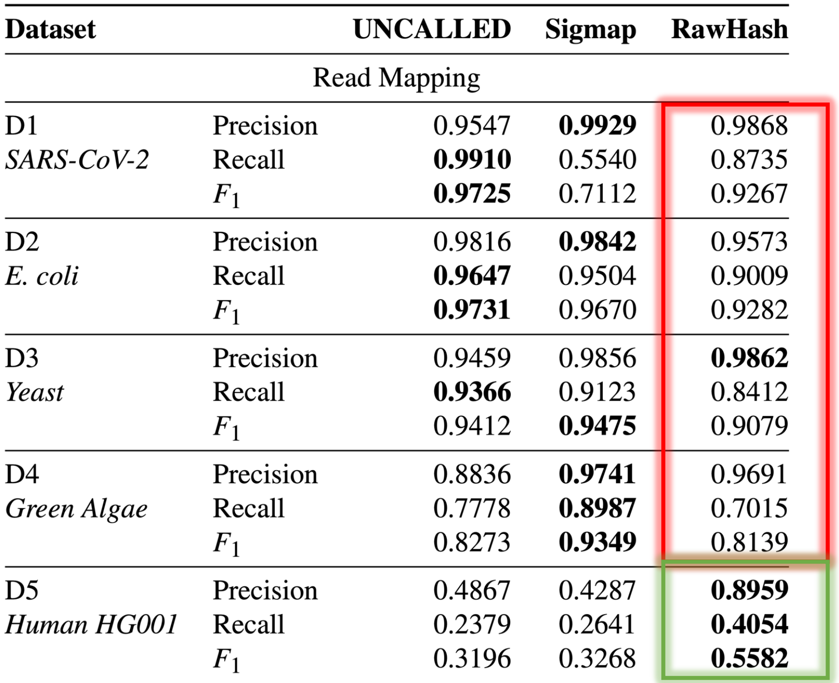


RawHash **reduces sequencing time and cost**

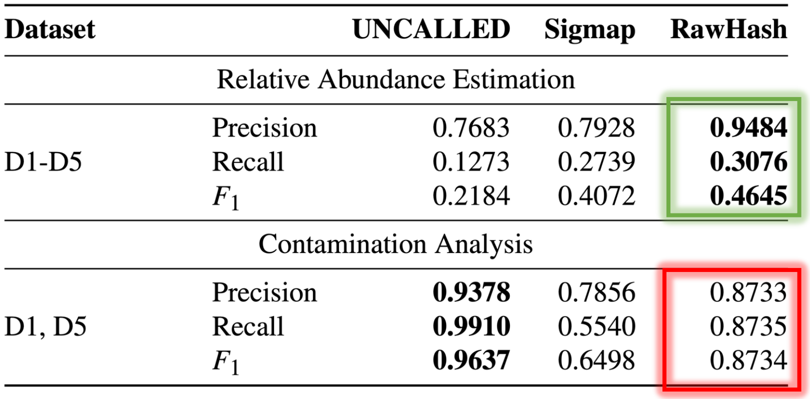
for large genomes up to **1.3**× compared to UNCALLED

Mapping Accuracy

- Read mapping accuracy of each tool and each use case



Dataset		UNCALLED	Sigmap	RawHash
Read Mapping				
D1 *SARS-CoV-2*	Precision	0.9547	**0.9929**	0.9868
	Recall	**0.9910**	0.5540	0.8735
	F_1	**0.9725**	0.7112	0.9267
D2 *E. coli*	Precision	0.9816	**0.9842**	0.9573
	Recall	**0.9647**	0.9504	0.9009
	F_1	**0.9731**	0.9670	0.9282
D3 *Yeast*	Precision	0.9459	0.9856	**0.9862**
	Recall	**0.9366**	0.9123	0.8412
	F_1	0.9412	**0.9475**	0.9079
D4 *Green Algae*	Precision	0.8836	**0.9741**	0.9691
	Recall	0.7778	**0.8987**	0.7015
	F_1	0.8273	**0.9349**	0.8139
D5 *Human HG001*	Precision	0.4867	0.4287	**0.8959**
	Recall	0.2379	0.2641	**0.4054**
	F_1	0.3196	0.3268	**0.5582**



Dataset		UNCALLED	Sigmap	RawHash
Relative Abundance Estimation				
D1-D5	Precision	0.7683	0.7928	**0.9484**
	Recall	0.1273	0.2739	**0.3076**
	F_1	0.2184	0.4072	**0.4645**
Contamination Analysis				
D1, D5	Precision	**0.9378**	0.7856	0.8733
	Recall	**0.9910**	0.5540	0.8735
	F_1	**0.9637**	0.6498	0.8734

For Large Genomes: RawHash provides the **best accuracy** in all metrics, resulting in **1.14**× - **2.13**× improvement in F_1 score

Relative Abundance Estimation Accuracy

- Estimating the ratio of genomes in a sample in real-time
  - **Distance:** Euclidean distance compared to the ground truth distance
  - The dataset includes a large reference genome

	Estimated Relative Abundance Ratios					
Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	**Distance**
Ground Truth	0.0929	0.4365	0.0698	0.1179	0.2828	N/A
UNCALLED	0.0026	0.5884	0.0615	0.1313	0.2161	0.1895
Sigmap	0.0419	0.4191	0.1038	0.0962	0.3390	0.0877
RawHash	0.1249	0.4701	0.0957	0.0629	0.2464	**0.0847**

RawHash provides the **best relative abundance estimation**

closest to the ground truth estimation

Real Implementation of Sequence Until

- Running RawHash by using
  - **RawHash (100%):** The entire sample **without Sequence Until**
  - **RawHash (7%):** RawHash **with Sequence Until** where Sequence Until dynamically stops the entire sequencing after sequencing **7% of the sample**

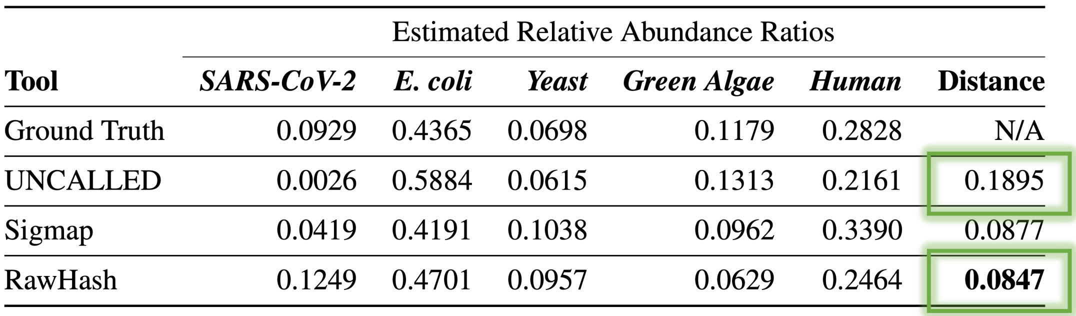
	Estimated Relative Abundance Ratios in 50,000 Random Reads					
Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	**Distance**
RawHash (100%)	0.0270	0.3636	0.3062	0.1951	0.1081	N/A
RawHash + Sequence Until (7%)	0.0283	0.3539	0.3100	0.1946	0.1133	0.0118

Sequence Until enables sequencing **only 7% (~1/15)** of the entire sample **with high accuracy**

SAFARI

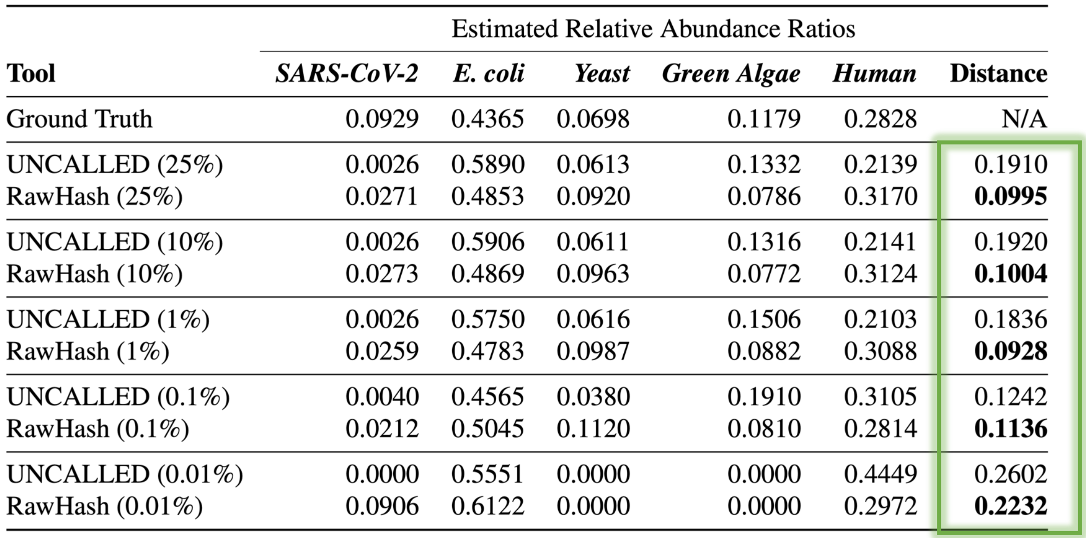
Simulating Sequence Until

- Real relative abundance results using the entire set of reads



	Estimated Relative Abundance Ratios					
Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	**Distance**
Ground Truth	0.0929	0.4365	0.0698	0.1179	0.2828	N/A
UNCALLED	0.0026	0.5884	0.0615	0.1313	0.2161	0.1895
Sigmap	0.0419	0.4191	0.1038	0.0962	0.3390	0.0877
RawHash	0.1249	0.4701	0.0957	0.0629	0.2464	**0.0847**

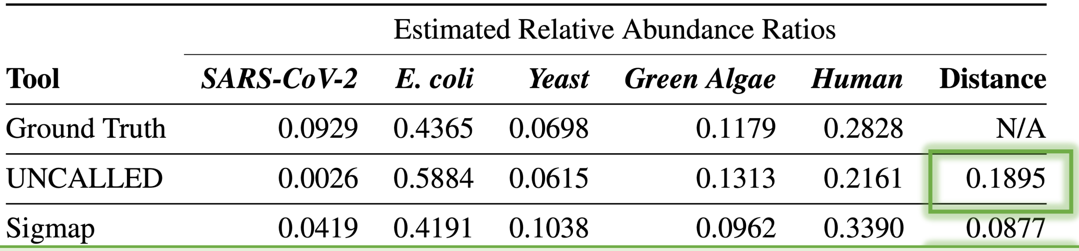
- Simulating the benefits of Sequence Until by
  - Using **a random portion** (25%, 10%, 1%, …) of the sample



	Estimated Relative Abundance Ratios					
Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	**Distance**
Ground Truth	0.0929	0.4365	0.0698	0.1179	0.2828	N/A
UNCALLED (25%)	0.0026	0.5890	0.0613	0.1332	0.2139	0.1910
RawHash (25%)	0.0271	0.4853	0.0920	0.0786	0.3170	**0.0995**
UNCALLED (10%)	0.0026	0.5906	0.0611	0.1316	0.2141	0.1920
RawHash (10%)	0.0273	0.4869	0.0963	0.0772	0.3124	**0.1004**
UNCALLED (1%)	0.0026	0.5750	0.0616	0.1506	0.2103	0.1836
RawHash (1%)	0.0259	0.4783	0.0987	0.0882	0.3088	**0.0928**
UNCALLED (0.1%)	0.0040	0.4565	0.0380	0.1910	0.3105	0.1242
RawHash (0.1%)	0.0212	0.5045	0.1120	0.0810	0.2814	**0.1136**
UNCALLED (0.01%)	0.0000	0.5551	0.0000	0.0000	0.4449	0.2602
RawHash (0.01%)	0.0906	0.6122	0.0000	0.0000	0.2972	**0.2232**

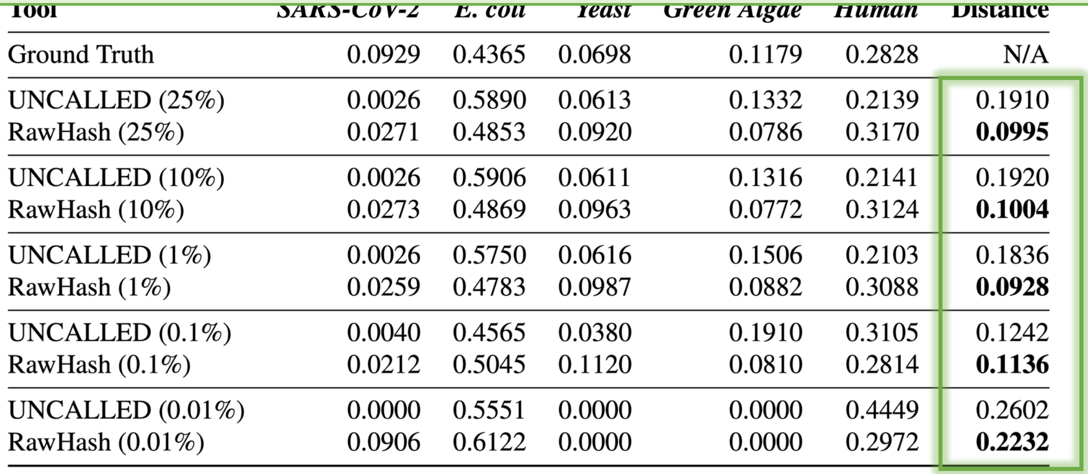
Simulating Sequence Until

- Real relative abundance results using the entire set of reads



	Estimated Relative Abundance Ratios					
Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	**Distance**
Ground Truth	0.0929	0.4365	0.0698	0.1179	0.2828	N/A
UNCALLED	0.0026	0.5884	0.0615	0.1313	0.2161	0.1895
Sigmap	0.0419	0.4191	0.1038	0.0962	0.3390	0.0877

UNCALLED and RawHash benefit from **Sequence Until** significantly **by up to 100**× reductions in sequencing time and costs



Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	**Distance**
Ground Truth	0.0929	0.4365	0.0698	0.1179	0.2828	N/A
UNCALLED (25%)	0.0026	0.5890	0.0613	0.1332	0.2139	0.1910
RawHash (25%)	0.0271	0.4853	0.0920	0.0786	0.3170	**0.0995**
UNCALLED (10%)	0.0026	0.5906	0.0611	0.1316	0.2141	0.1920
RawHash (10%)	0.0273	0.4869	0.0963	0.0772	0.3124	**0.1004**
UNCALLED (1%)	0.0026	0.5750	0.0616	0.1506	0.2103	0.1836
RawHash (1%)	0.0259	0.4783	0.0987	0.0882	0.3088	**0.0928**
UNCALLED (0.1%)	0.0040	0.4565	0.0380	0.1910	0.3105	0.1242
RawHash (0.1%)	0.0212	0.5045	0.1120	0.0810	0.2814	**0.1136**
UNCALLED (0.01%)	0.0000	0.5551	0.0000	0.0000	0.4449	0.2602
RawHash (0.01%)	0.0906	0.6122	0.0000	0.0000	0.2972	**0.2232**

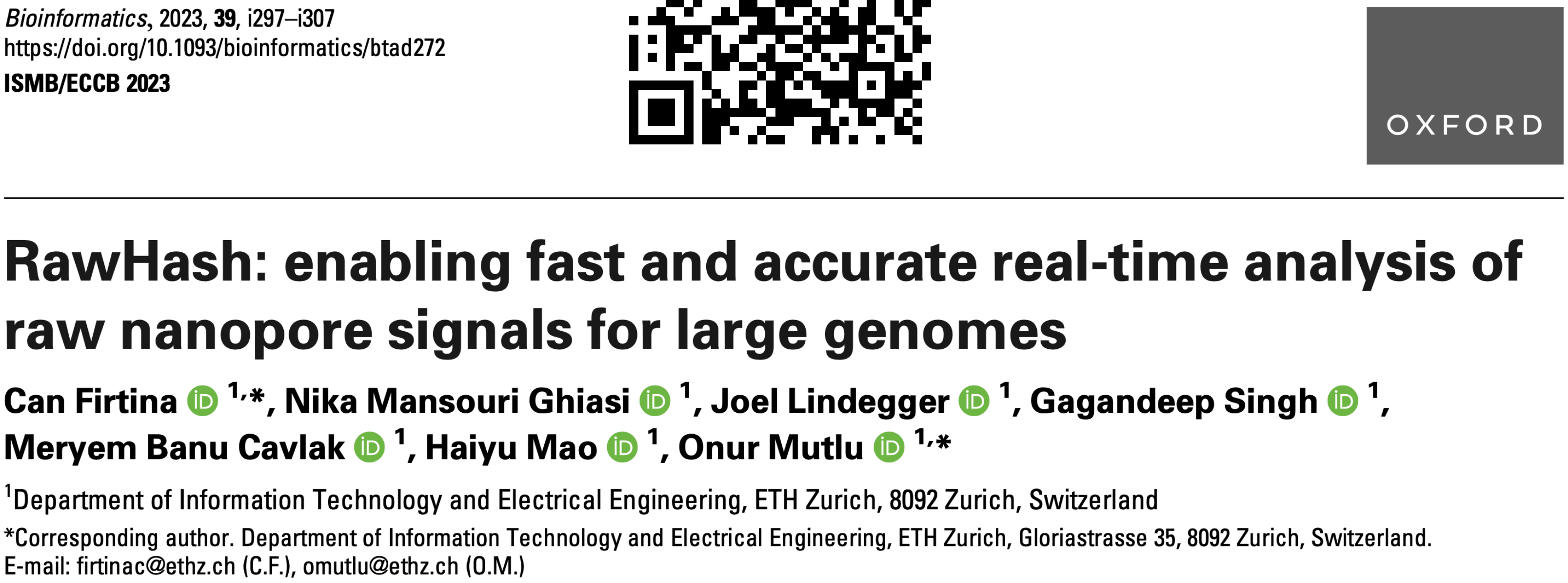
More in the Paper

- **More Results**
  - **Mapping time** per read
  - Overall **computational resources** required by each tool
    - Peak memory usage, CPU time and real time in the indexing and mapping steps
  - **Performance breakdown** of the steps in RawHash

- **Details of all mechanisms and configurations**
  - Details of the **quantization** and **hashing** mechanism
  - Details of the **parameter configurations**
  - Trade-offs between the **DNN-based approaches** and raw signal mapping approaches

RawHash

- Can Firtina, Nika Mansouri Ghiasi, Joel Lindegger, Gagandeep Singh, Meryem Banu Cavlak, Haiyu Mao, and Onur Mutlu,
"RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes"
*Proceedings of the 31st Annual Conference on Intelligent Systems for Molecular Biology (**ISMB**) and the 22nd European Conference on Computational Biology (**ECCB**)*, Jul 2023
[arXiv preprint]
[Source Code]



Bioinformatics, 2023, **39**, i297–i307
https://doi.org/10.1093/bioinformatics/btad272
ISMB/ECCB 2023

OXFORD

RawHash: enabling fast and accurate real-time analysis of raw nanopore signals for large genomes

Can Firtina [1,*], Nika Mansouri Ghiasi [1], Joel Lindegger [1], Gagandeep Singh [1], Meryem Banu Cavlak [1], Haiyu Mao [1], Onur Mutlu [1,*]

[1] Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland

*Corresponding author. Department of Information Technology and Electrical Engineering, ETH Zurich, Gloriastrasse 35, 8092 Zurich, Switzerland.
E-mail: firtinac@ethz.ch (C.F.), omutlu@ethz.ch (O.M.)

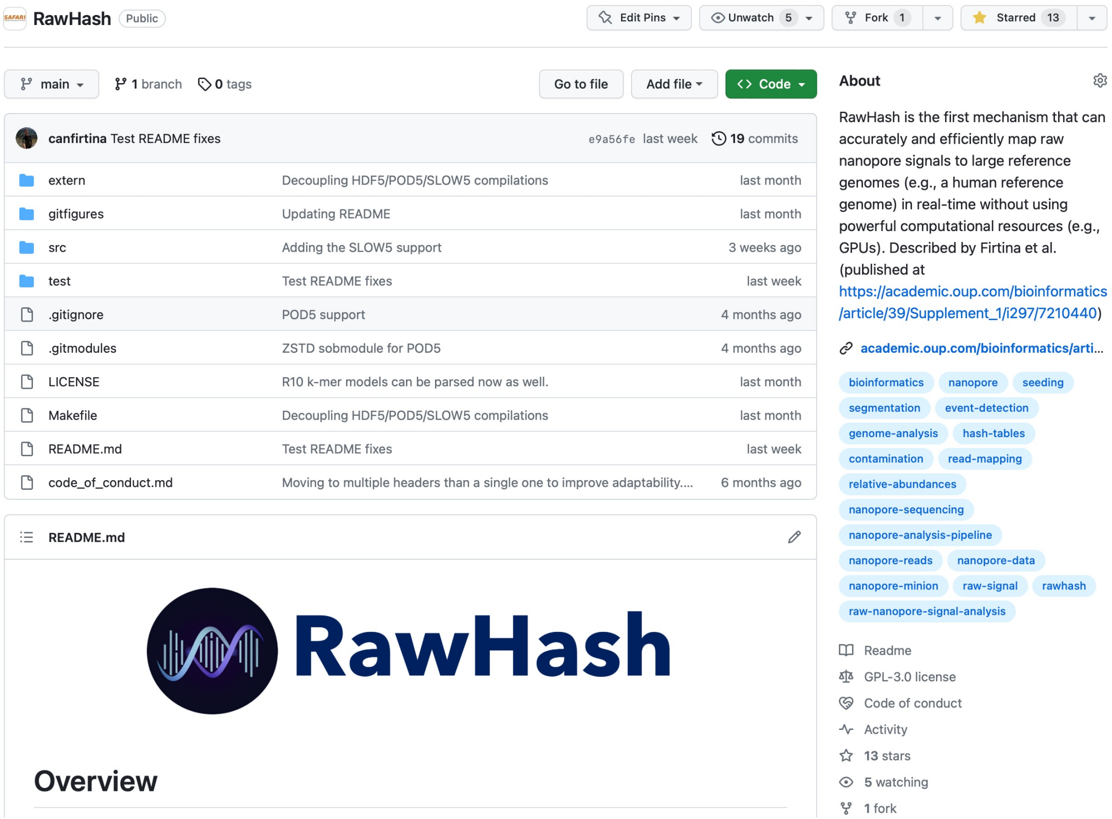
RawHash Source Code

- Supports **all major raw signal file formats and flow cell versions**
  - FAST5, POD5, S/BLOW5 file formats

- Easy-to-use scripts
  - To download all the datasets
  - To reproduce all of our results

- You can write your outlier function for Sequence Until
  - Easily integrate Sequence Until

- Upcoming Feature:
  - Integrating the MinKNOW API



canfirtina Test README fixes e9a56fe last week 19 commits

extern	Decoupling HDF5/POD5/SLOW5 compilations	last month
gitfigures	Updating README	last month
src	Adding the SLOW5 support	3 weeks ago
test	Test README fixes	last week
.gitignore	POD5 support	4 months ago
.gitmodules	ZSTD sobmodule for POD5	4 months ago
LICENSE	R10 k-mer models can be parsed now as well.	last month
Makefile	Decoupling HDF5/POD5/SLOW5 compilations	last month
README.md	Test README fixes	last week
code_of_conduct.md	Moving to multiple headers than a single one to improve adaptability....	6 months ago

README.md

RawHash

Overview

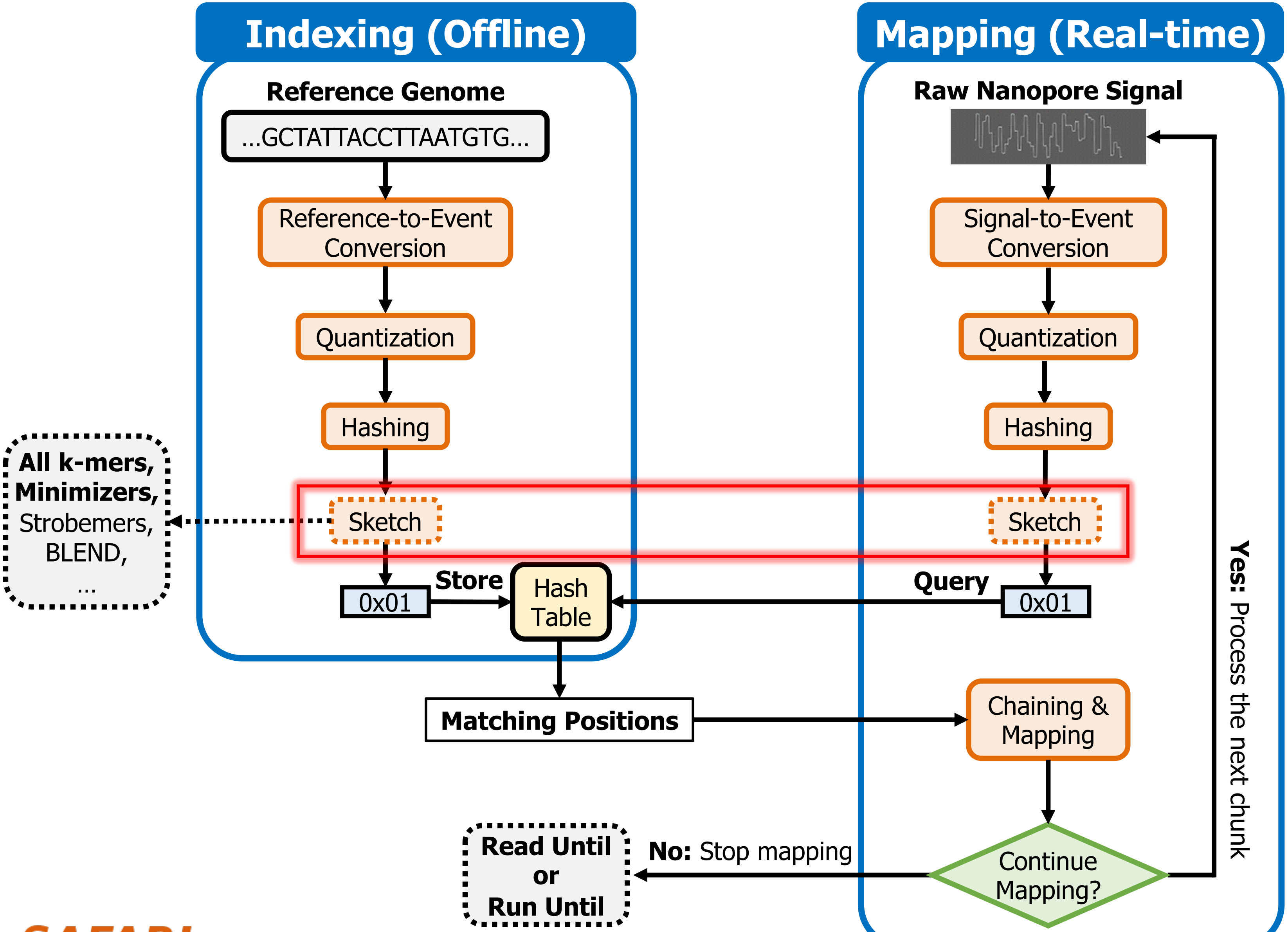
About

RawHash is the first mechanism that can accurately and efficiently map raw nanopore signals to large reference genomes (e.g., a human reference genome) in real-time without using powerful computational resources (e.g., GPUs). Described by Firtina et al. (published at https://academic.oup.com/bioinformatics/article/39/Supplement_1/i297/7210440)

academic.oup.com/bioinformatics/arti...

https://github.com/CMU-SAFARI/RawHash

Sketching with Hash-based Indexing



SAFARI

Outline

Background

RawHash

Evaluation

Conclusion

Conclusion

Key Contributions:
1) The **first hash-based mechanism** that can quickly and accurately analyze raw nanopore signals for **large genomes**

2) The novel **Sequence Until** technique can accurately and **dynamically stop the entire sequencing of all reads at once** if further sequencing is not necessary

Key Results: Across 3 use cases and 5 genomes of varying sizes, RawHash provides

- **25.8× and 3.4× better average throughput** compared to two state-of-the-art works
- **1.14× – 2.13× more accurate mapping results** for **large genomes**
- Sequence Until **reduces the sequencing time and cost by 15×**

Many opportunities for analyzing raw nanopore signals in real-time:

- Many hash-based **sketching techniques** can now be used for raw signals
- **Indexing is very cheap:** Many future use cases with the on-the-fly index construction
- We should rethink the algorithms to perform downstream analysis fully using raw signals

ISMB/ECCB 2023

RawHash

Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes

Can Firtina

Nika Mansouri Ghiasi Joel Lindegger Gagandeep Singh

Meryem Banu Cavlak Haiyu Mao Onur Mutlu

Paper

Code

SAFARI

ETH zürich

Fast and Accurate Real-Time Genome Analysis

- Can Firtina, Melina Soysal, Joel Lindegger, and Onur Mutlu,
  "RawHash2: Accurate and Fast Mapping of Raw Nanopore Signals using a Hash-based Seeding Mechanism"
  *Preprint on **arxiv***, September 2023.
  [arXiv version]
  [RawHash2 Source Code]

RawHash2: Accurate and Fast Mapping of Raw Nanopore Signals using a Hash-based Seeding Mechanism

Can Firtina Melina Soysal Joel Lindegger Onur Mutlu
ETH Zürich

Optimizations in RawHash2 (1)

- **More sensitive** chaining implementation with penalty scores
  - Benefits: Enables filtering dissimilar regions quickly
  - Downside: Additional computations with costly log operations

- **Weighted mapping decisions**
  - Benefit #1: ‘Learned’ mapping decisions based on the weights chosen from empirical analysis
  - Benefit #2: Faster and more accurate decisions

- Frequency **filters**
  - Filters the seeds that frequently appear before chaining
  - Benefits: Reduced workload on chaining without significantly affecting accuracy
  - Downside: Less sensitive mapping due to removed seeds

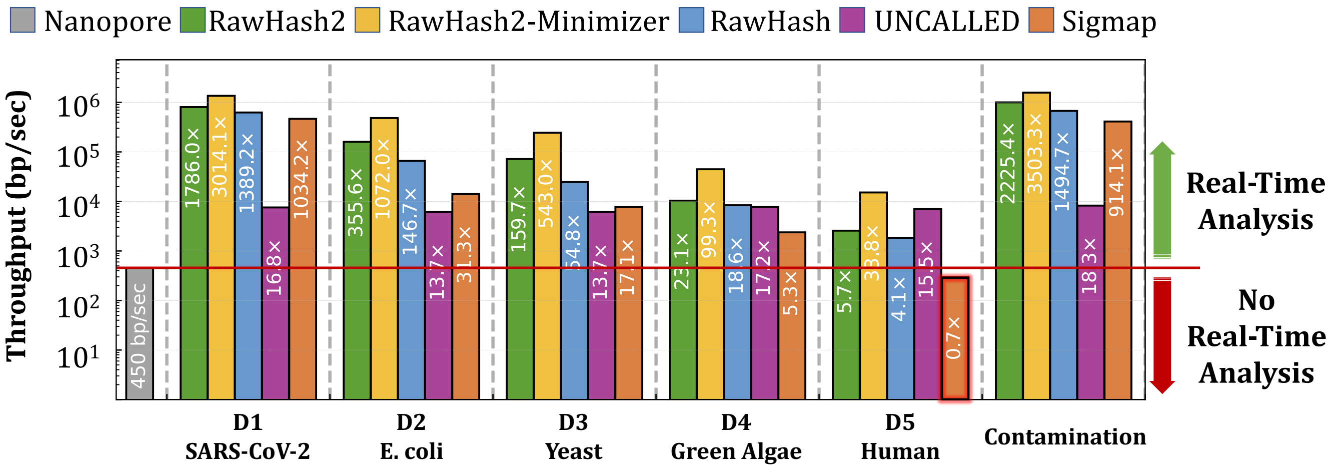
Optimizations in RawHash2 (2)

- New sketching techniques such as **minimizers** and **BLEND**
  - Enables integration of widely studied sketching techniques
  - Benefits: Can take advantage of these techniques (e.g., reduced storage requirements)

- Support for the recent improvements in the technology
  - Support for **new data formats**: POD5 and S/BLOW5
  - Support for **newer nanopore chemistry** versions: R10.4

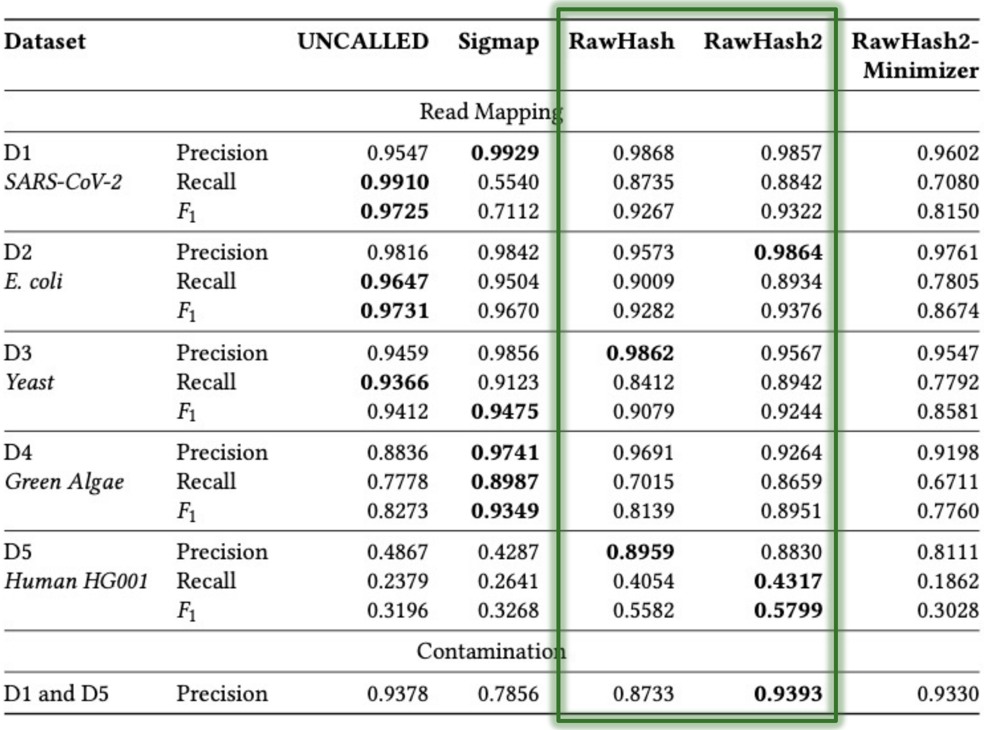
Results – Throughput

- **Real-time analysis requires** faster throughput than sequencer
  - Throughput of a nanopore sequencer: ~**450 bp/sec (data generation speed)**



2.3× better average throughput RawHash

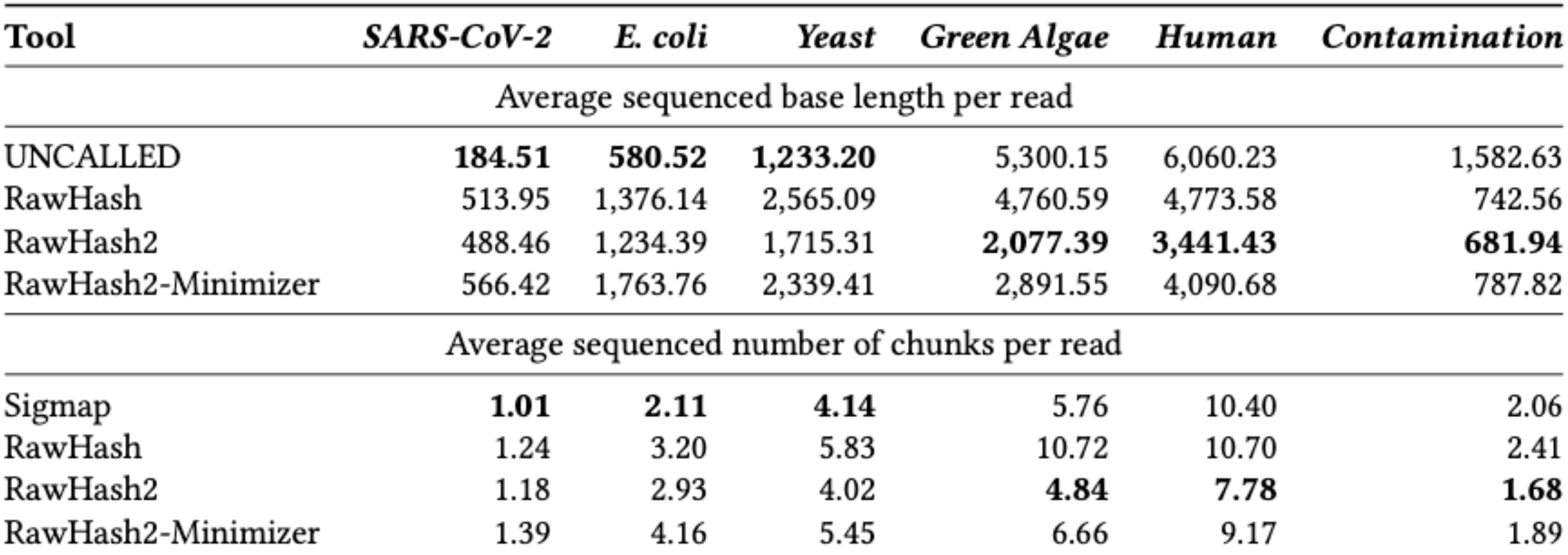
Results – Accuracy



Dataset		UNCALLED	Sigmap	RawHash	RawHash2	RawHash2-Minimizer
			Read Mapping			
D1 *SARS-CoV-2*	Precision	0.9547	**0.9929**	0.9868	0.9857	0.9602
	Recall	**0.9910**	0.5540	0.8735	0.8842	0.7080
	F_1	**0.9725**	0.7112	0.9267	0.9322	0.8150
D2 *E. coli*	Precision	0.9816	0.9842	0.9573	**0.9864**	0.9761
	Recall	**0.9647**	0.9504	0.9009	0.8934	0.7805
	F_1	**0.9731**	0.9670	0.9282	0.9376	0.8674
D3 *Yeast*	Precision	0.9459	0.9856	**0.9862**	0.9567	0.9547
	Recall	**0.9366**	0.9123	0.8412	0.8942	0.7792
	F_1	0.9412	**0.9475**	0.9079	0.9244	0.8581
D4 *Green Algae*	Precision	0.8836	**0.9741**	0.9691	0.9264	0.9198
	Recall	0.7778	**0.8987**	0.7015	0.8659	0.6711
	F_1	0.8273	**0.9349**	0.8139	0.8951	0.7760
D5 *Human HG001*	Precision	0.4867	0.4287	**0.8959**	0.8830	0.8111
	Recall	0.2379	0.2641	0.4054	**0.4317**	0.1862
	F_1	0.3196	0.3268	0.5582	**0.5799**	0.3028
			Contamination			
D1 and D5	Precision	0.9378	0.7856	0.8733	**0.9393**	0.9330

RawHash2 is more accurate than RawHash **in all cases**

Results – Average Sequencing Length



Tool	*SARS-CoV-2*	*E. coli*	*Yeast*	*Green Algae*	*Human*	*Contamination*
			Average sequenced base length per read			
UNCALLED	**184.51**	**580.52**	**1,233.20**	5,300.15	6,060.23	1,582.63
RawHash	513.95	1,376.14	2,565.09	4,760.59	4,773.58	742.56
RawHash2	488.46	1,234.39	1,715.31	**2,077.39**	**3,441.43**	**681.94**
RawHash2-Minimizer	566.42	1,763.76	2,339.41	2,891.55	4,090.68	787.82
			Average sequenced number of chunks per read			
Sigmap	**1.01**	**2.11**	**4.14**	5.76	10.40	2.06
RawHash	1.24	3.20	5.83	10.72	10.70	2.41
RawHash2	1.18	2.93	4.02	**4.84**	**7.78**	**1.68**
RawHash2-Minimizer	1.39	4.16	5.45	6.66	9.17	1.89

RawHash2 uses fewer bases to sequence than RawHash in all cases

RawHash2 uses the smallest number of bases to sequence for larger genomes

Fast and Accurate Real-Time Genome Analysis

- Can Firtina, Melina Soysal, Joel Lindegger, and Onur Mutlu,
  "RawHash2: Accurate and Fast Mapping of Raw Nanopore Signals using a Hash-based Seeding Mechanism"
  Preprint on ***arxiv***, September 2023.
  [arXiv version]
  [RawHash2 Source Code]

RawHash2: Accurate and Fast Mapping of Raw Nanopore Signals using a Hash-based Seeding Mechanism

Can Firtina Melina Soysal Joel Lindegger Onur Mutlu

ETH Zürich

Agenda for Today

- Background
  - Sequence analysis
  - Raw nanopore signal analysis and real-time analysis

- Enabling Fast and Accurate Real-time Analysis
  - RawHash and RawHash2

- Conclusion

The Future is Bright for Genome Analysis

- We covered various recent ideas to
  - Analyze genomes in ways that were not possible before

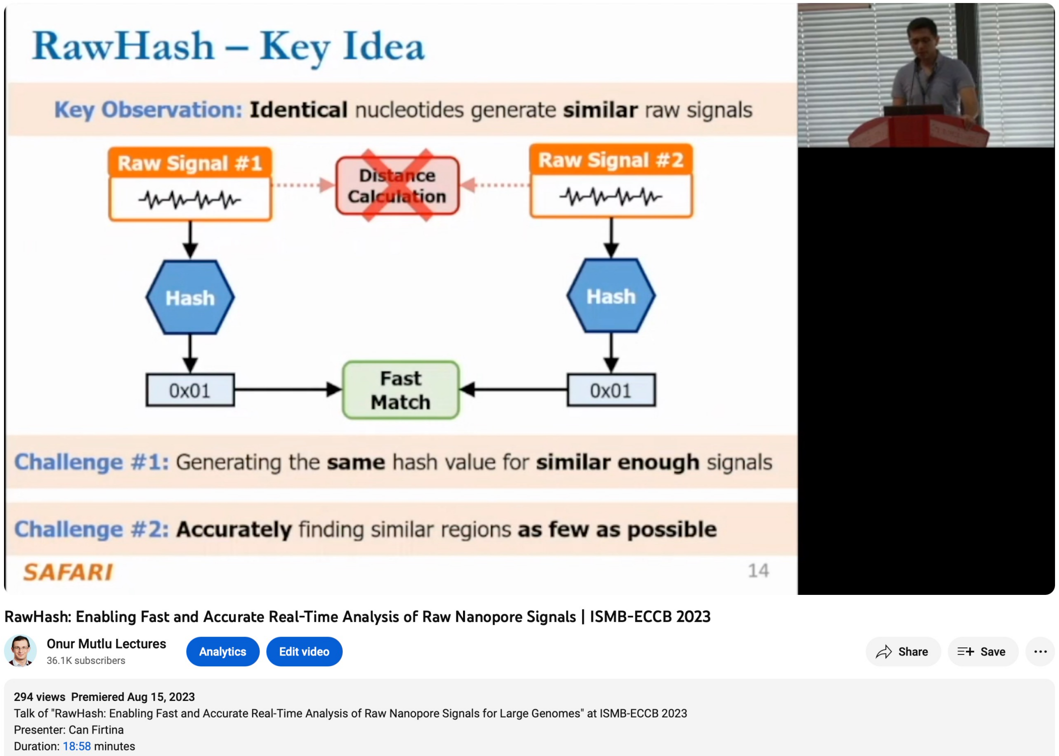
- Enabling cost-effective, portable, fast, and accurate genome analysis has many implications
  - What are the new applications to enable with these unique benefits?

- Can we do even better?
  - Understanding and modifying the sequencing process for analyzing other types of biological data

- **Many future opportunities exist**
  - **Especially with new sequencing technologies**
  - **Especially with new applications and use cases**

More on Real-Time Genome Analysis

- Can Firtina,
  "RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes"
  Proceedings Talk at ISMB-ECCB, Lyon, France, 25 July 2023.
  [Slides (pptx) (pdf)]
  [Talk Video (18 minutes)]



RawHash – Key Idea

Key Observation: **Identical** nucleotides generate **similar** raw signals

Challenge #1: Generating the **same** hash value for **similar enough** signals

Challenge #2: **Accurately** finding similar regions **as few as possible**

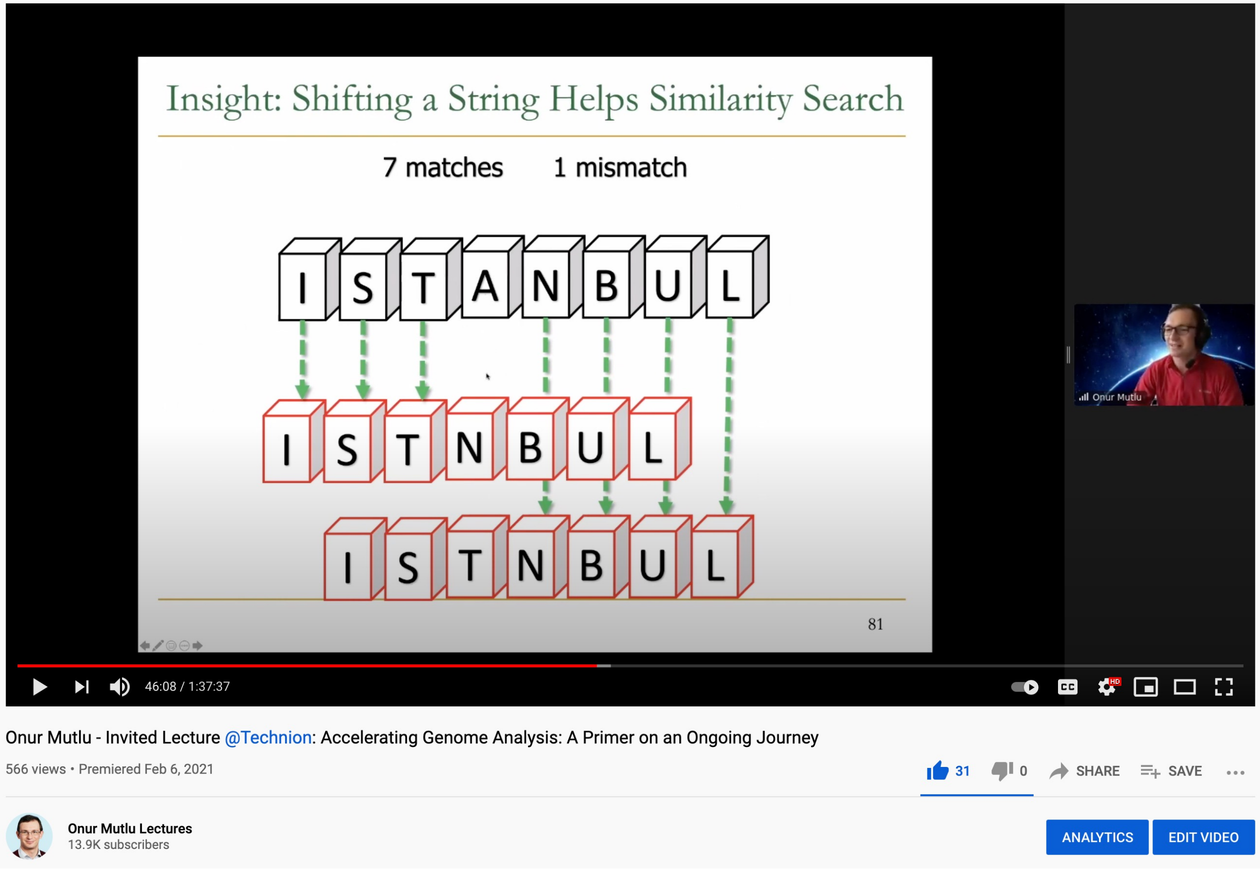
RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals | ISMB-ECCB 2023

Onur Mutlu Lectures
36.1K subscribers

294 views Premiered Aug 15, 2023
Talk of "RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes" at ISMB-ECCB 2023
Presenter: Can Firtina
Duration: 18:58 minutes

Fast Genome Analysis…

- Onur Mutlu,
  "Accelerating Genome Analysis: A Primer on an Ongoing Journey"
  Invited Lecture at Technion, Virtual, 26 January 2021.
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 37 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]



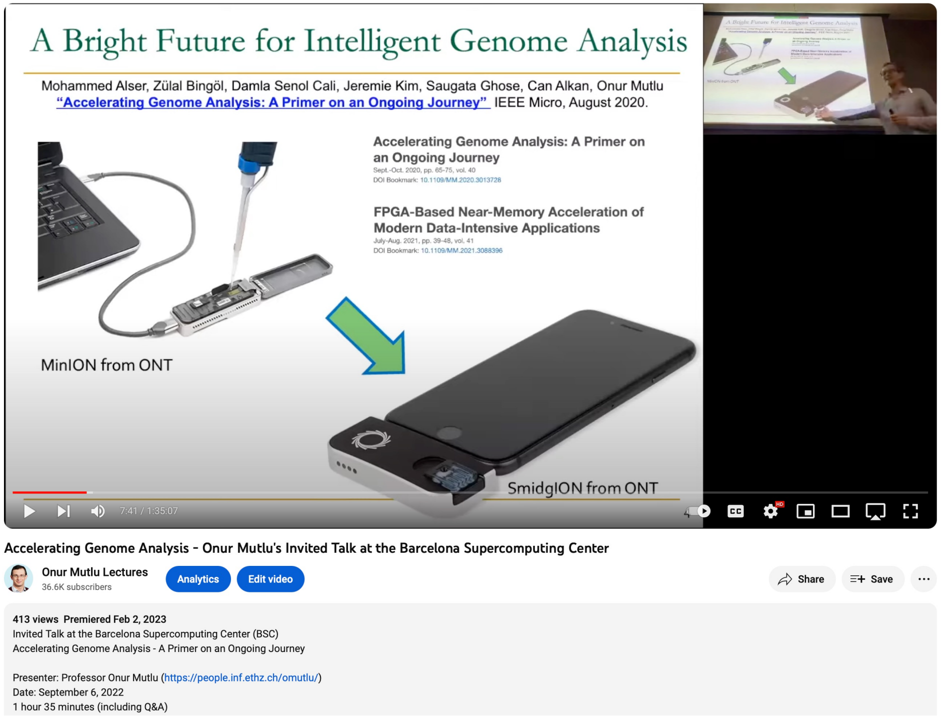
Insight: Shifting a String Helps Similarity Search

Onur Mutlu - Invited Lecture @Technion: Accelerating Genome Analysis: A Primer on an Ongoing Journey

566 views • Premiered Feb 6, 2021

More on Fast Genome Analysis…

- Onur Mutlu,
  "Accelerating Genome Analysis"
  *Invited Talk at the Barcelona Supercomputing Center (**BSC**)*, Barcelona, Spain, 6 September 2022.
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 35 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]
  [Related Invited Paper (at Computational and Structural Biology Journal, 2022)]



A Bright Future for Intelligent Genome Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu
"Accelerating Genome Analysis: A Primer on an Ongoing Journey" IEEE Micro, August 2020.

Accelerating Genome Analysis: A Primer on an Ongoing Journey
Sept.-Oct. 2020, pp. 65-75, vol. 40
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
July-Aug. 2021, pp. 39-48, vol. 41
DOI Bookmark: 10.1109/MM.2021.3088396

Accelerating Genome Analysis - Onur Mutlu's Invited Talk at the Barcelona Supercomputing Center

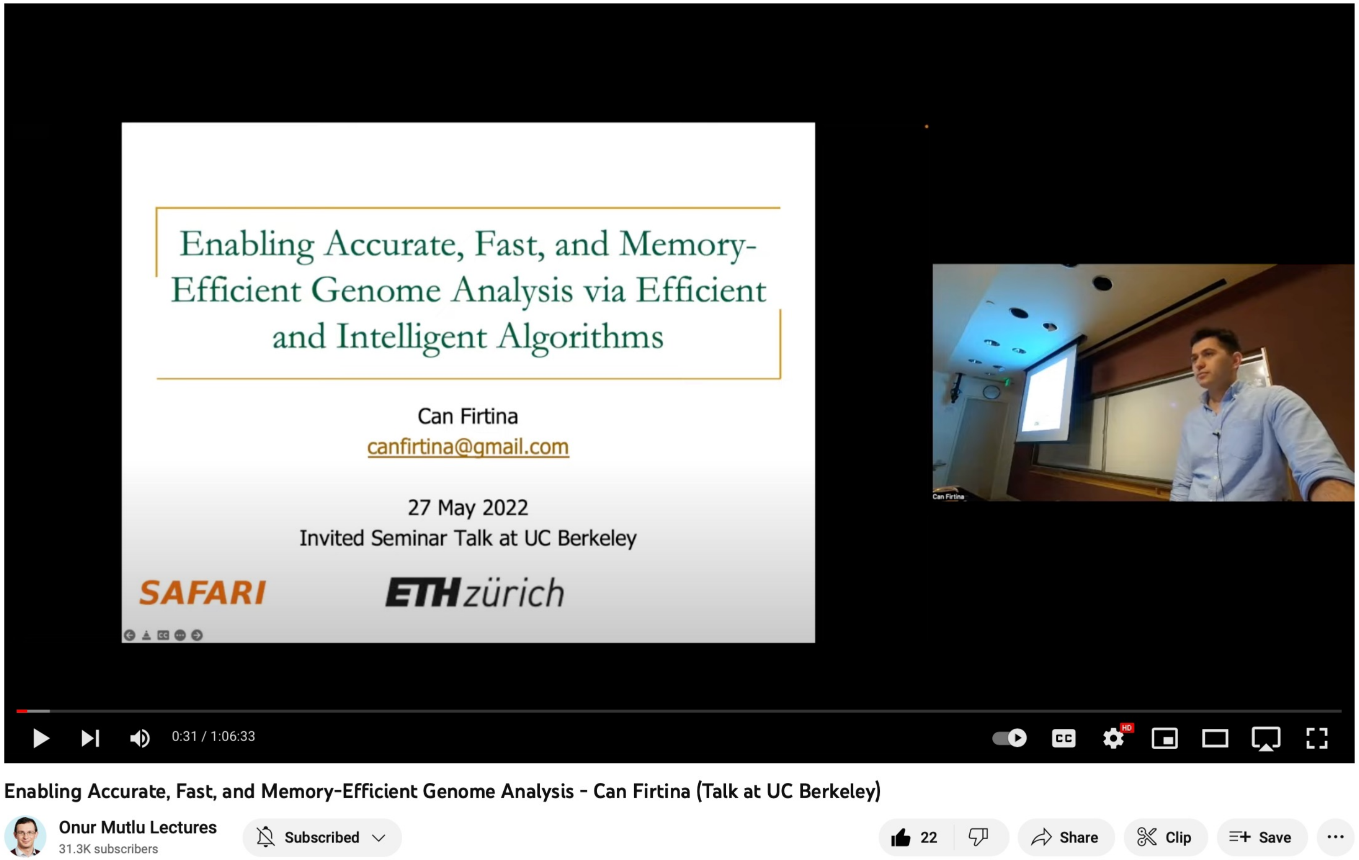
Onur Mutlu Lectures
36.6K subscribers

413 views Premiered Feb 2, 2023
Invited Talk at the Barcelona Supercomputing Center (BSC)
Accelerating Genome Analysis - A Primer on an Ongoing Journey

Presenter: Professor Onur Mutlu (https://people.inf.ethz.ch/omutlu/)
Date: September 6, 2022
1 hour 35 minutes (including Q&A)

More on Accelerating Genome Analysis

- Can Firtina,
  "Enabling Accurate, Fast, and Memory-Efficient Genome Analysis via Efficient and Intelligent Algorithms"
  Talk at UC Berkeley, Berkeley, CA, United States, May 27, 2022.
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 6 minutes)]



Enabling Accurate, Fast, and Memory-Efficient Genome Analysis via Efficient and Intelligent Algorithms

Can Firtina
canfirtina@gmail.com

27 May 2022
Invited Seminar Talk at UC Berkeley

Enabling Accurate, Fast, and Memory-Efficient Genome Analysis - Can Firtina (Talk at UC Berkeley)

Accelerating Genome Analysis [DAC 2023]

- Onur Mutlu and Can Firtina,
  "Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
  *Invited Special Session Paper in Proceedings of the 60th Design Automation Conference (**DAC**)*, San Francisco, CA, USA, July 2023.
  [Slides (pptx) (pdf)]
  [Talk Video (38 minutes, including Q&A)]
  [Related Invited Paper]
  [arXiv version]

Accelerating Genome Analysis via Algorithm-Architecture Co-Design

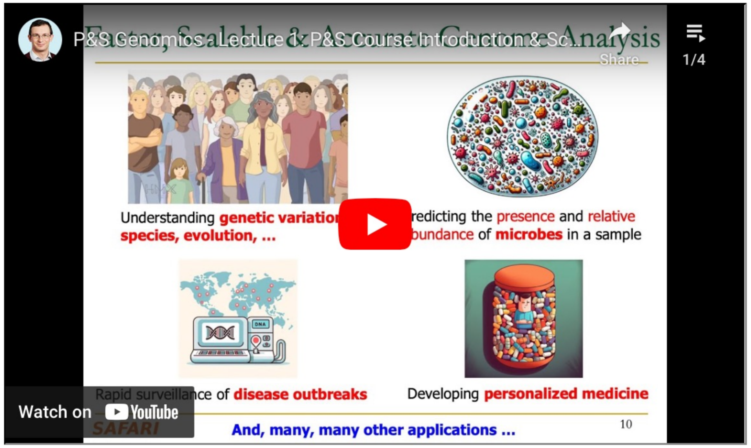
Onur Mutlu Can Firtina

ETH Zürich

https://ieeexplore.ieee.org/document/10247887

Genomics Course (Spring 2024)

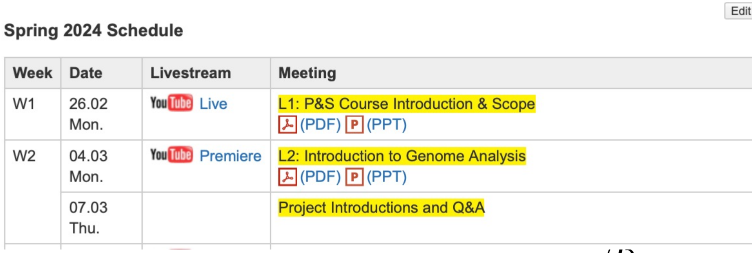
- **Spring 2024 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2024/doku.php?id=bioinformatics
- **Fall 2023 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2023/doku.php?id=bioinformatics
- **Youtube Livestream (Spring 2024):**
  - https://youtube.com/playlist?list=PL5Q2soXY2Zi_UT4zTiLxRmK_zbgz6M93Z
- Project course
  - Taken by Bachelor's/Master's students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings



P&S Genomics - Lecture 1: P&S Course Introduction & Sc...
Faster, Scalable & Accurate Genome Analysis
Share
1/4
Understanding genetic variation, species, evolution, ...
Predicting the presence and relative abundance of microbes in a sample
Rapid surveillance of disease outbreaks
Developing personalized medicine
And, many, many other applications ...
10
Watch on YouTube
SAFARI

Complete Lecture Playlist (Fall 2023):

P&S Genomics - Lecture 1: P&S Course Introduction & Sc...
Why Need Faster & Scalable Genome Analysis
Share
1/18
Understanding genetic variation, species, evolution, ...
Predicting the presence and relative abundance of microbes in a sample
Rapid surveillance of disease outbreaks
Developing personalized medicine
And, many, many other applications ...
4
Watch on YouTube
SAFARI



Spring 2024 Schedule

Week	Date	Livestream	Meeting
W1	26.02 Mon.	YouTube Live	L1: P&S Course Introduction & Scope (PDF) (PPT)
W2	04.03 Mon.	YouTube Premiere	L2: Introduction to Genome Analysis (PDF) (PPT)
	07.03 Thu.		Project Introductions and Q&A

https://www.youtube.com/onurmutlulectures

Introduction to Real-Time Raw Nanopore Signal Analysis: RawHash and RawHash2

Can Firtina

canfirtina@gmail.com

18 March 2024

Sabanci University

BIO310 - Introduction to Bioinformatics

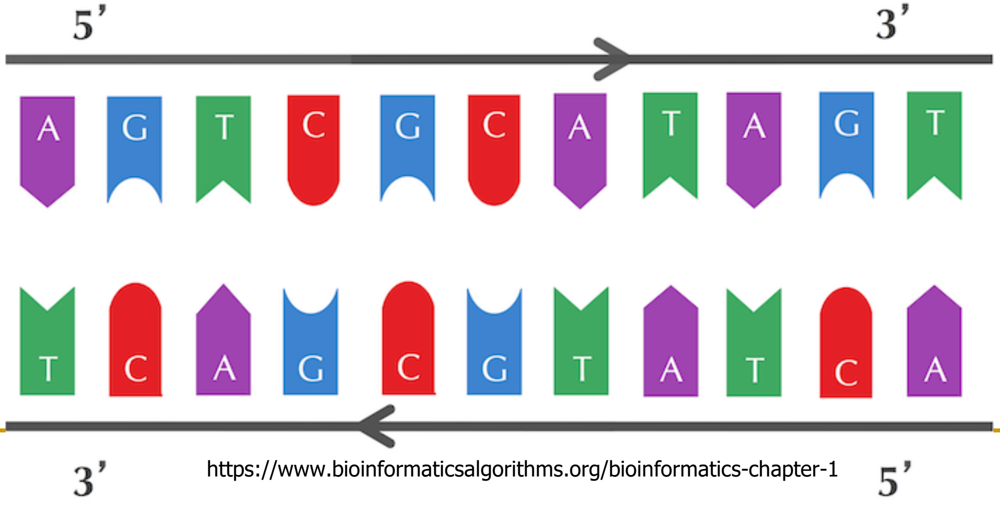
SAFARI

ETH zürich

Backup Slides

Challenges in Read Mapping

- Need to find many mappings of each read
- Need to tolerate variances/sequencing errors in each read
- Need to map each read very fast (i.e., performance is important, life critical in some cases)
- Need to map reads to both forward and reverse strands



https://www.bioinformaticsalgorithms.org/bioinformatics-chapter-1

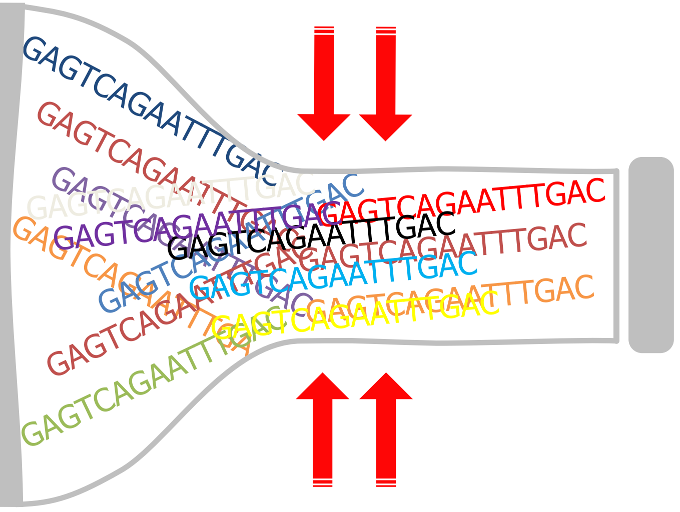
Analysis is Bottlenecked in Read Mapping!!

48 Human whole genomes

at 30× coverage

in about 2 days

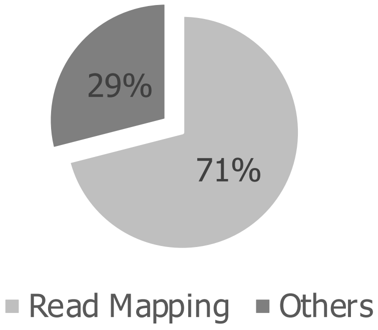
Illumina NovaSeq 6000



1 Human genome

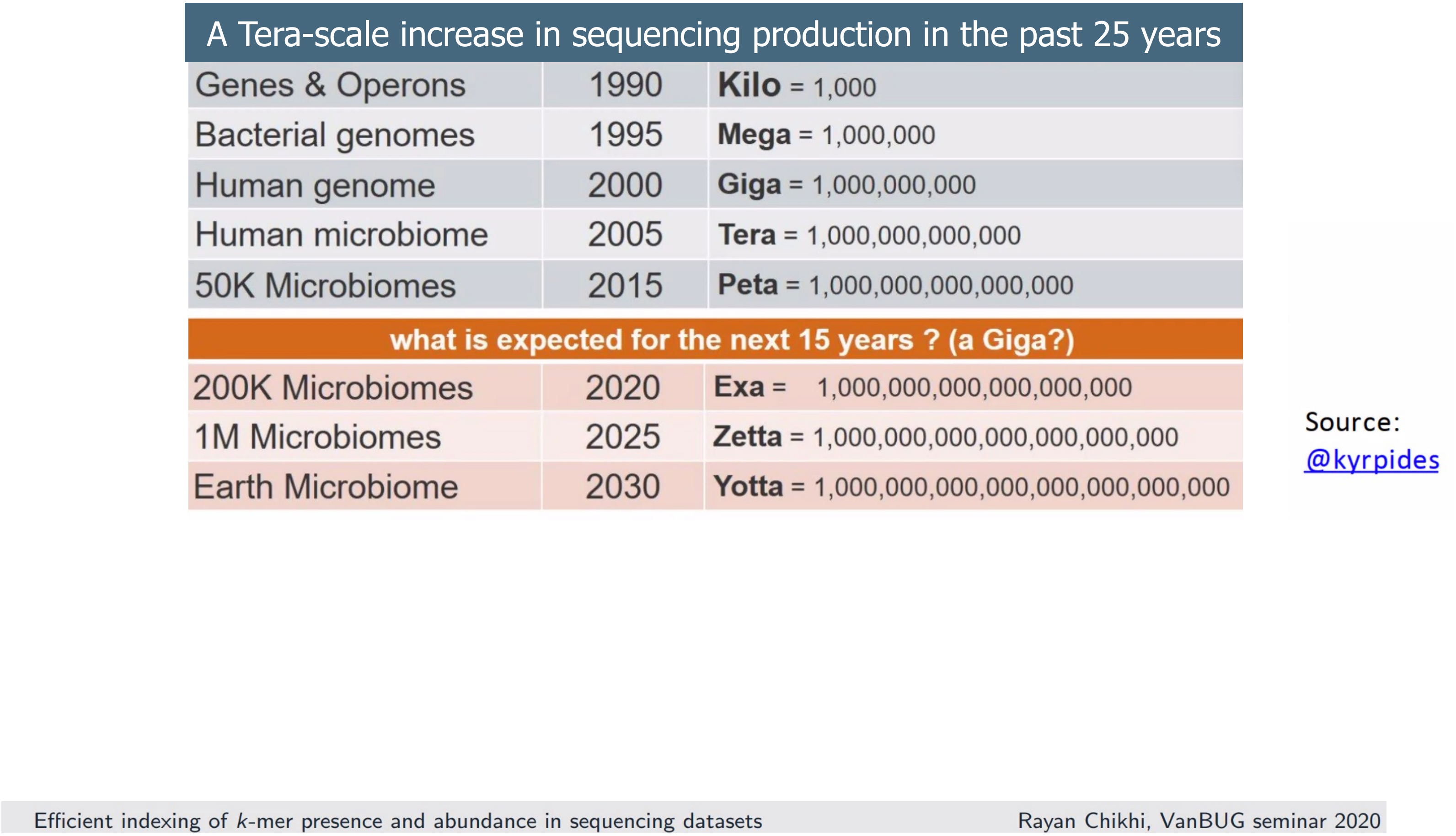
32 CPU hours

on a 48-core processor



Goyal+, "Ultra-fast next generation human genome sequencing data processing using DRAGENTM bio-IT processor for precision medicine", *Open Journal of Genetics*, 2017.

A Tsunami of Sequencing Data



A Tera-scale increase in sequencing production in the past 25 years

Genes & Operons	1990	**Kilo** = 1,000
Bacterial genomes	1995	**Mega** = 1,000,000
Human genome	2000	**Giga** = 1,000,000,000
Human microbiome	2005	**Tera** = 1,000,000,000,000
50K Microbiomes	2015	**Peta** = 1,000,000,000,000,000
what is expected for the next 15 years ? (a Giga?)		
200K Microbiomes	2020	**Exa** = 1,000,000,000,000,000,000
1M Microbiomes	2025	**Zetta** = 1,000,000,000,000,000,000,000
Earth Microbiome	2030	**Yotta** = 1,000,000,000,000,000,000,000,000

Source: @kyrpides

Efficient indexing of *k*-mer presence and abundance in sequencing datasets

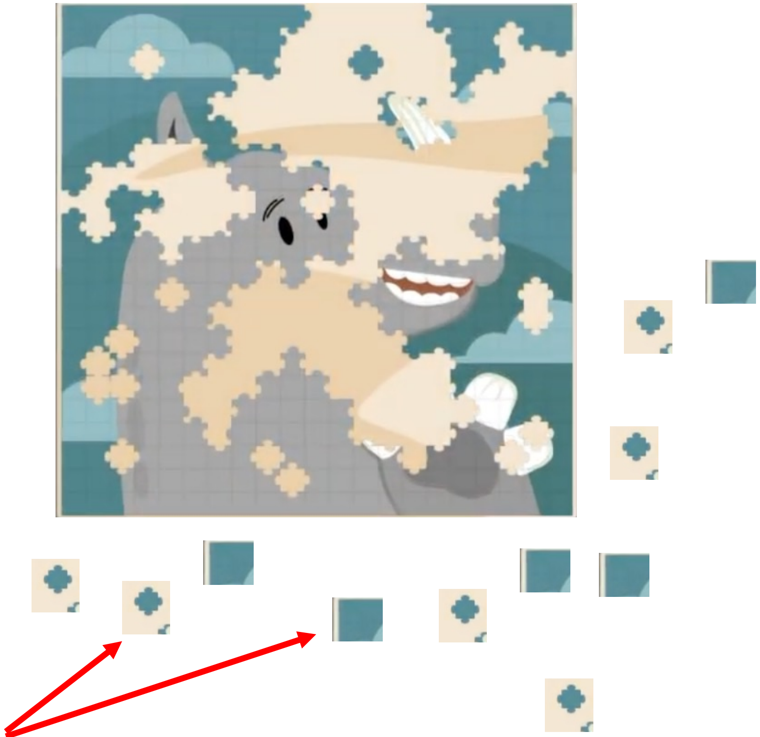
Rayan Chikhi, VanBUG seminar 2020

Solving the Puzzle

.FASTA file

Reference genome

.FASTQ file



Reads

https://www.pacb.com/smrt-science/smrt-sequencing/hifi-reads-for-highly-accurate-long-read-sequencing/

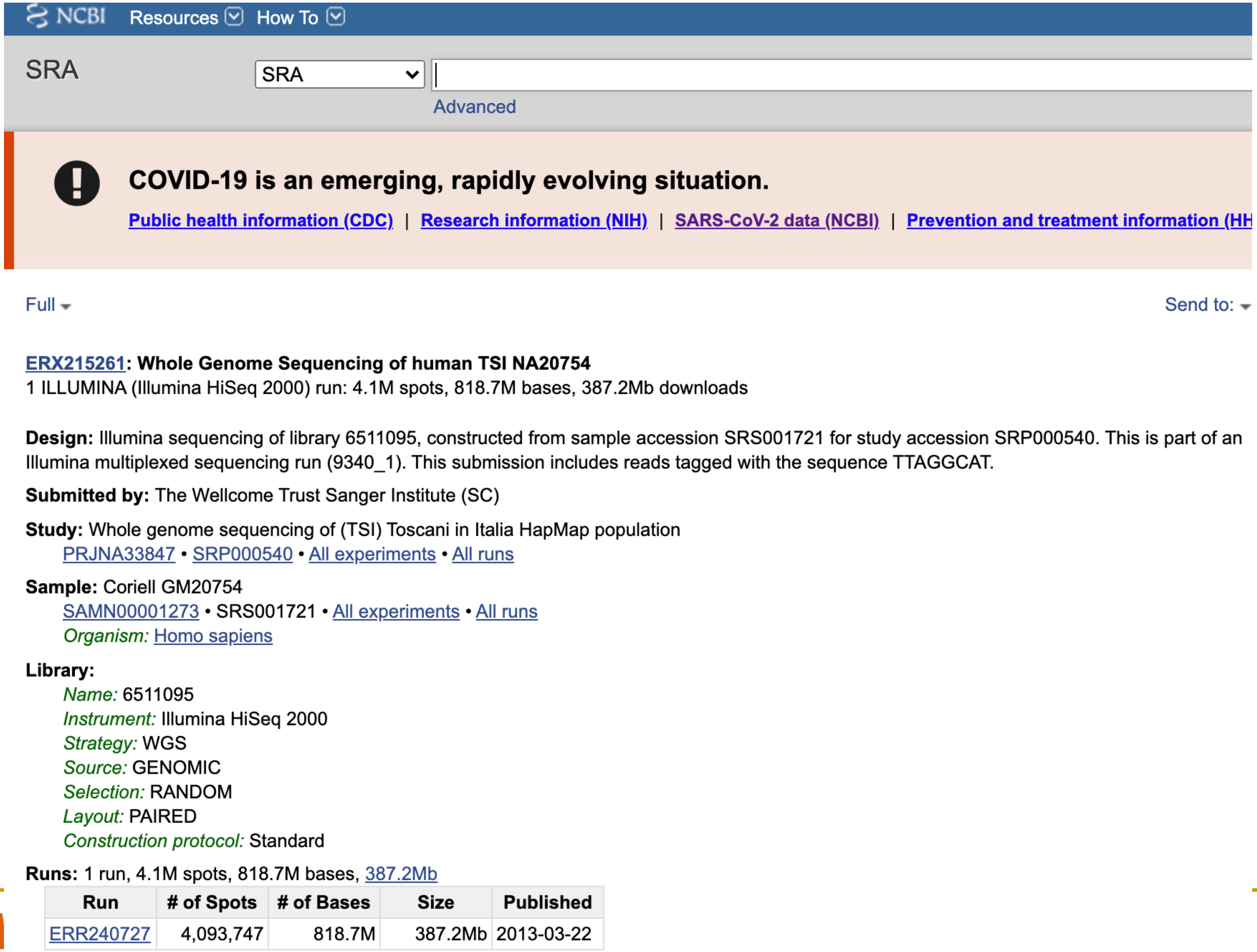
Obtaining the Human Reference Genome

- **GRCh38.p13**
- Description: Genome Reference Consortium Human Build 38 patch release 13 (GRCh38.p13)
- Organism name: Homo sapiens (human)
- Date: 2019/02/28
- 3,099,706,404 bases
- Compressed .fna file (964.9 MB)
- https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39

```
>NC_000001.11 Homo sapiens chromosome 1, GRCh38.p13 Primary Assembly
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
….
```

Obtaining .FASTQ Files

- https://www.ncbi.nlm.nih.gov/sra/ERR240727



COVID-19 is an emerging, rapidly evolving situation.

Public health information (CDC) | Research information (NIH) | SARS-CoV-2 data (NCBI) | Prevention and treatment information (HH

ERX215261: Whole Genome Sequencing of human TSI NA20754
1 ILLUMINA (Illumina HiSeq 2000) run: 4.1M spots, 818.7M bases, 387.2Mb downloads

Design: Illumina sequencing of library 6511095, constructed from sample accession SRS001721 for study accession SRP000540. This is part of an Illumina multiplexed sequencing run (9340_1). This submission includes reads tagged with the sequence TTAGGCAT.

Submitted by: The Wellcome Trust Sanger Institute (SC)

Study: Whole genome sequencing of (TSI) Toscani in Italia HapMap population
PRJNA33847 • SRP000540 • All experiments • All runs

Sample: Coriell GM20754
SAMN00001273 • SRS001721 • All experiments • All runs
Organism: Homo sapiens

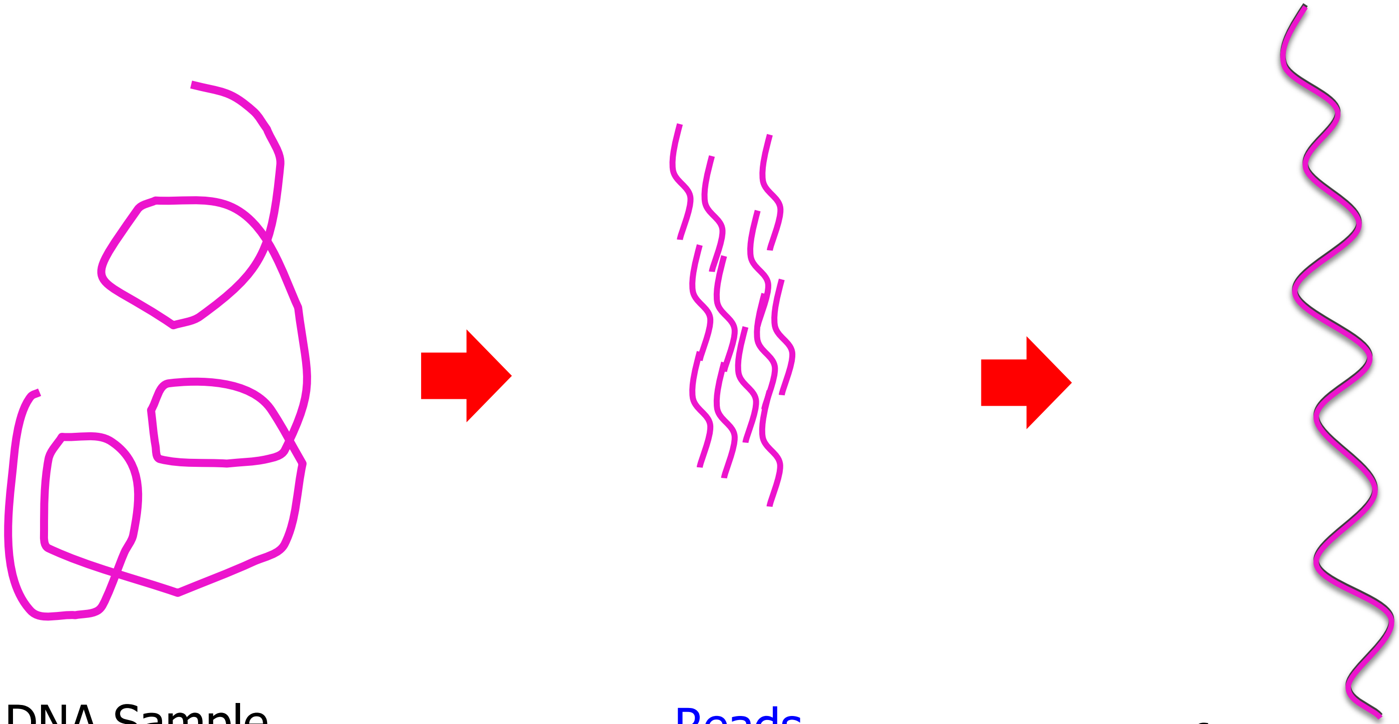
Library:
Name: 6511095
Instrument: Illumina HiSeq 2000
Strategy: WGS
Source: GENOMIC
Selection: RANDOM
Layout: PAIRED
Construction protocol: Standard

Runs: 1 run, 4.1M spots, 818.7M bases, 387.2Mb

Run	# of Spots	# of Bases	Size	Published
ERR240727	4,093,747	818.7M	387.2Mb	2013-03-22

Read Mapping

Map reads to a known reference genome with some minor differences allowed



DNA Sample
“chemical format”

Reads
“text format”

Reference genome

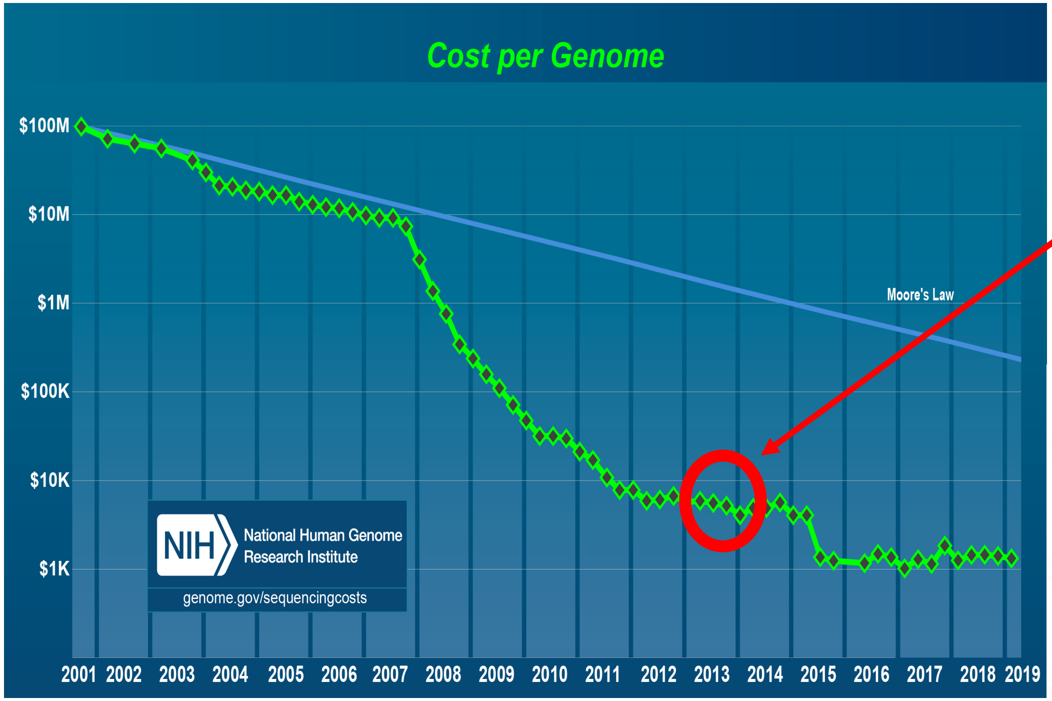
Subject genome
“text format”

Read Mapping Algorithms: Two Styles

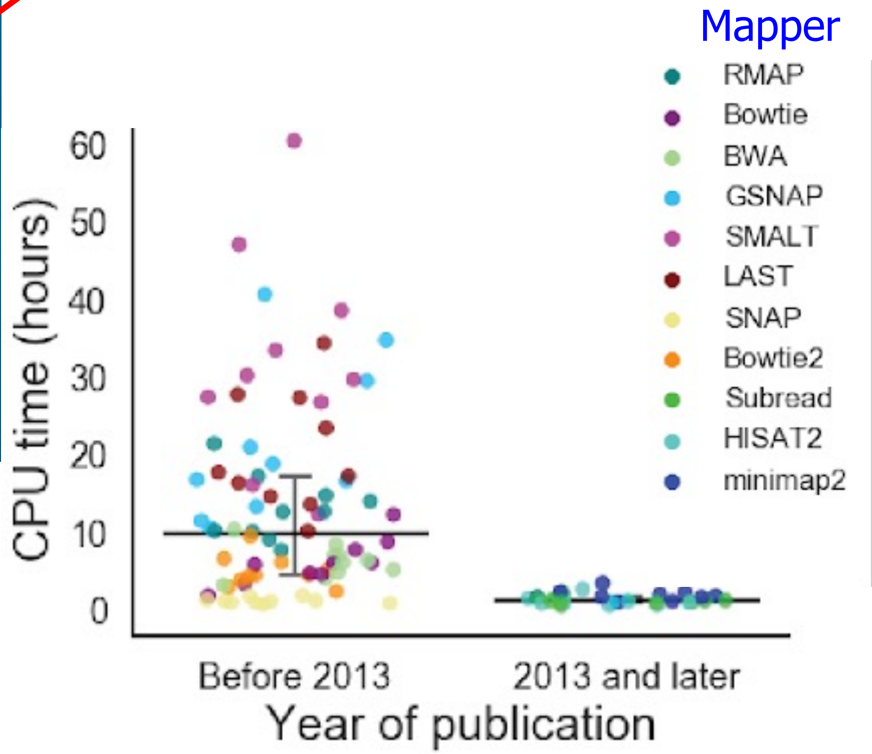
- Hash based seed-and-extend (hash table, suffix array, suffix tree)
  - Index the "k-mers" in the genome into a hash table (pre-processing)
  - When searching a read, find the location of a k-mer in the read; then extend through alignment
  - More sensitive (can find all mapping locations), but slow
  - Requires large memory; this can be reduced with cost to run time

- Burrows-Wheeler Transform & Ferragina-Manzini Index based aligners
  - BWT is a compression method used to compress the genome index
  - Perfect matches can be found very quickly, memory lookup costs increase for imperfect matches
  - Reduced sensitivity

The Need for Speed



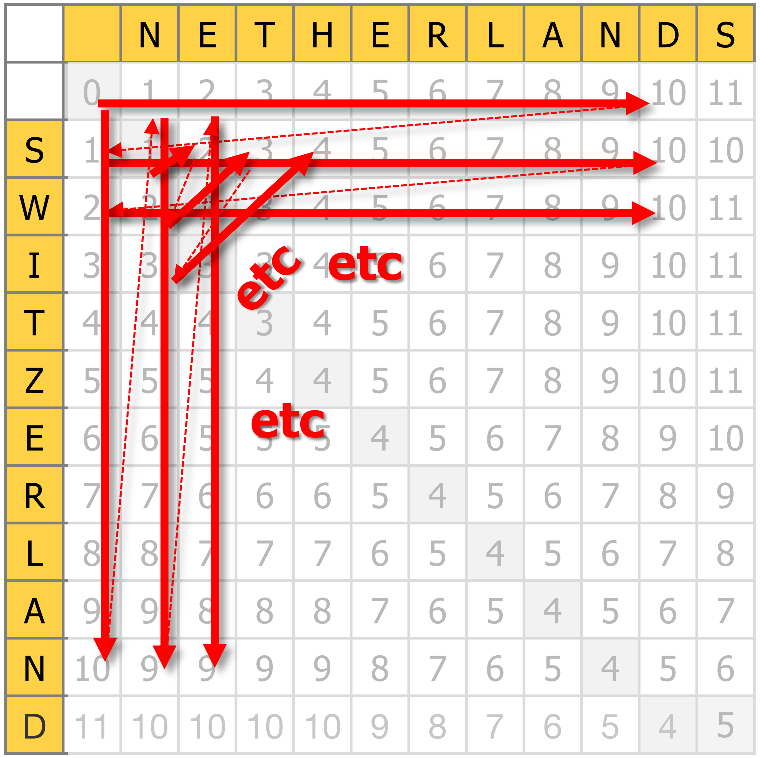
Did we realize the **need** for **faster** genome analysis?



Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021

Sequence Alignment in Unavoidable

- Quadratic-time dynamic-programming algorithm **WHY?!**
  - Enumerating all possible prefixes



		N	E	T	H	E	R	L	A	N	D	S
	0	1	2	3	4	5	6	7	8	9	10	11
S	1	1	2	3	4	5	6	7	8	9	10	10
W	2	2	2	3	4	5	6	7	8	9	10	11
I	3	3	3	3	4	[illegible]	6	7	8	9	10	11
T	4	4	4	3	4	5	6	7	8	9	10	11
Z	5	5	5	4	4	5	6	7	8	9	10	11
E	6	6	5	5	5	4	5	6	7	8	9	10
R	7	7	6	6	6	5	4	5	6	7	8	9
L	8	8	7	7	7	6	5	4	5	6	7	8
A	9	9	8	8	8	7	6	5	4	5	6	7
N	10	9	9	9	9	8	7	6	5	4	5	6
D	11	10	10	10	10	9	8	7	6	5	4	5

NETHERLANDS x SWITZERLAND

- NETHERLANDS x S
  NETHERLANDS x SW
  NETHERLANDS x SWI
  NETHERLANDS x SWIT
  NETHERLANDS x SWITZ
  NETHERLANDS x SWITZE
  NETHERLANDS x SWITZER
  NETHERLANDS x SWITZERL
  NETHERLANDS x SWITZERLA
  NETHERLANDS x SWITZERLAN
  NETHERLANDS x SWITZERLAND

Sequence Alignment in Unavoidable

- Quadratic-time dynamic-programming algorithm

  Enumerating all possible prefixes

- Data dependencies limit the computation parallelism

  Processing row (or column) after another

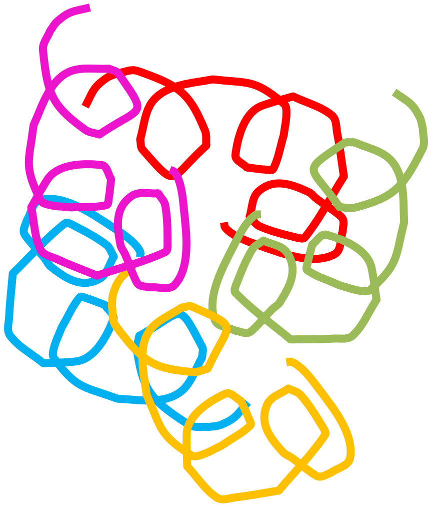
- Entire matrix is computed even though strings can be dissimilar.

  Number of differences is computed only at the backtraking step.

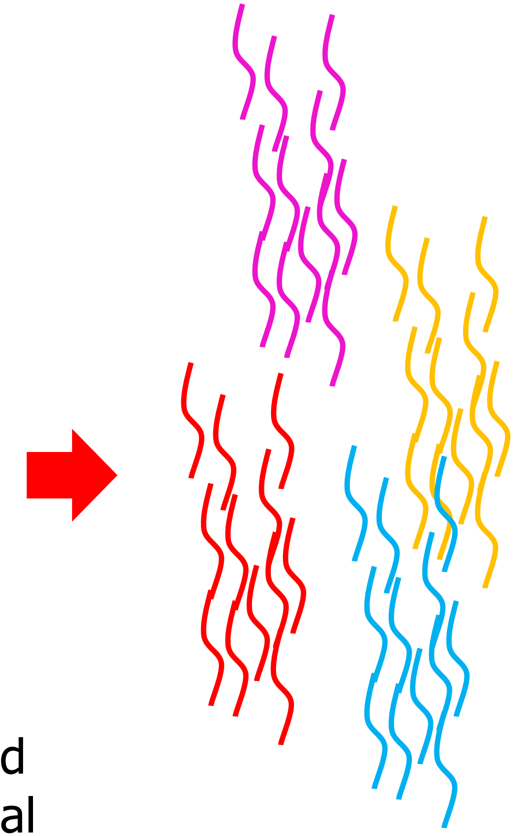
		N	E	T	H	E	R	L	A	N	D	S
	0	1	2	3	4	5	6	7	8	9	10	11
S	1	1	2	3	4	5	6	7	8	9	10	10
W	2	2	2	3	4	5	6	7	8	9	10	11
I	3	3	3	3	4	5	6	7	8	9	10	11
T	4	4	4	3	4	5	6	7	8	9	10	11
Z	5	5	5	4	4	5	6	7	8	9	10	11
E	6	6	5	5	5	4	5	6	7	8	9	10
R	7	7	6	6	6	5	4	5	6	7	8	9
L	8	8	7	7	7	6	5	4	5	6	7	8
A	9	9	8	8	8	7	6	5	4	5	6	7
N	10	9	9	9	9	8	7	6	5	4	5	6
D	11	10	10	10	10	9	8	7	6	5	4	5

Metagenomics Analysis

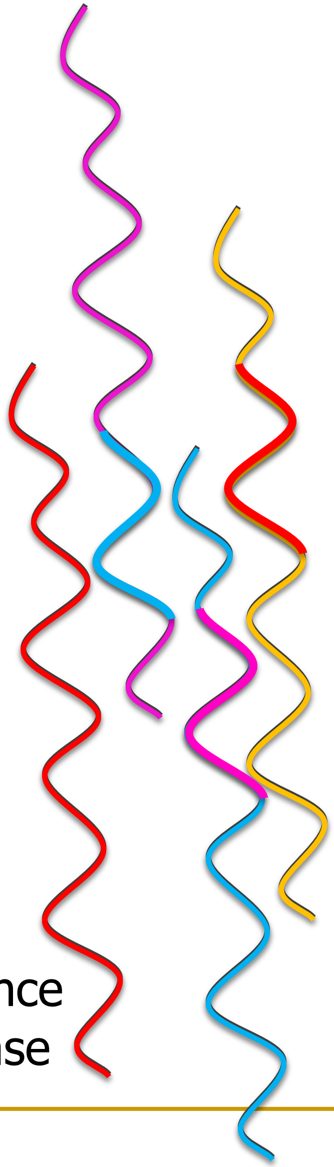
Reads from different unknown donors at sequencing time are mapped to many known reference genomes



genetic material recovered directly from environmental samples

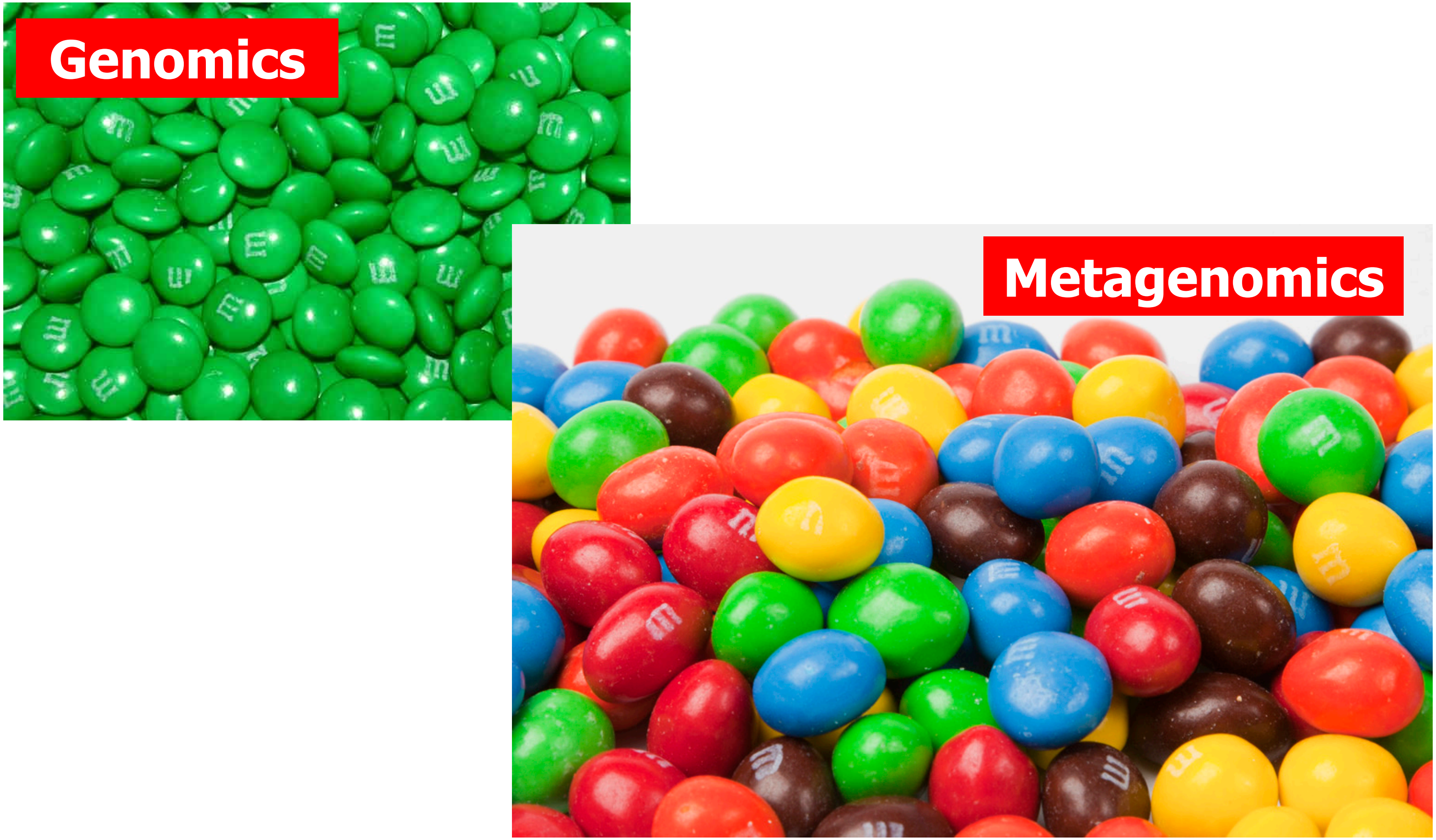


Reads
"text format"



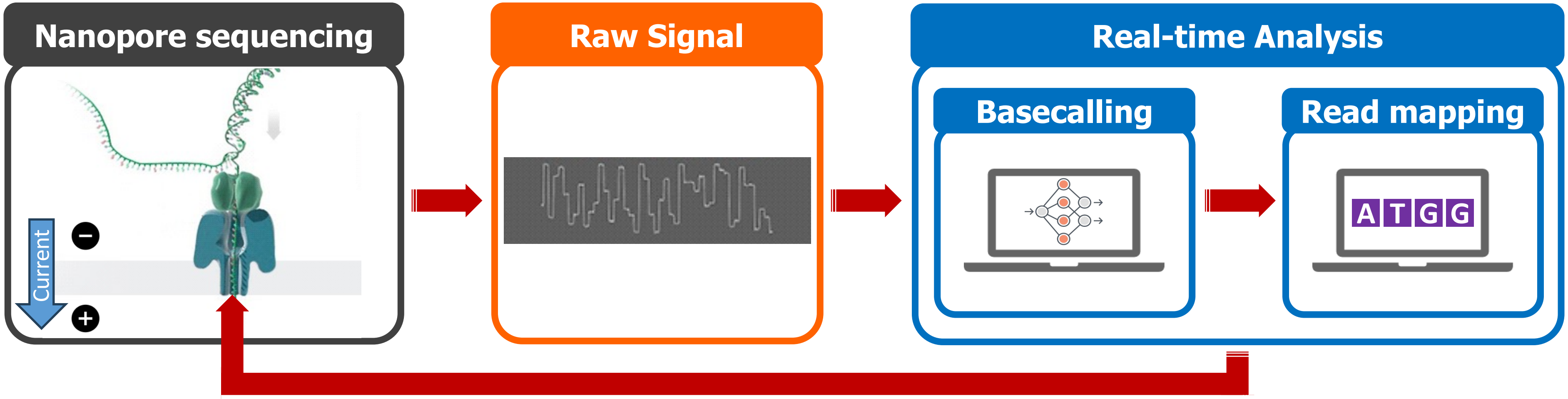
Reference Database

Genomics vs. Metagenomics



Existing Solutions – Real-time Basecalling

Deep neural networks (**DNNs**) for translating **signals** to **bases**

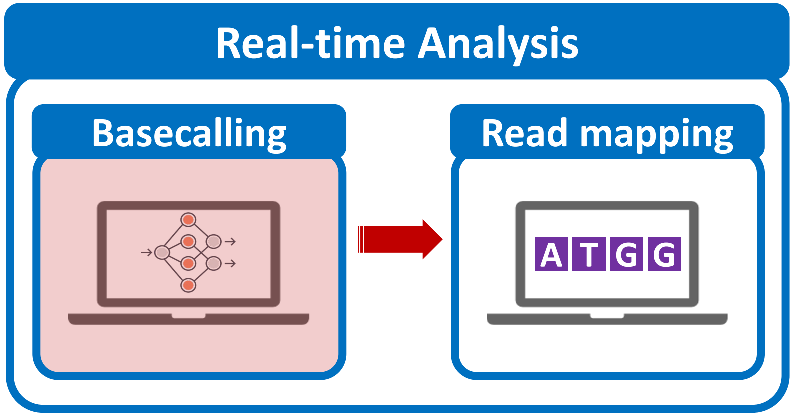


DNNs provide **less noisy analysis** from basecalled sequences

Costly and power-hungry computational requirements

The Problem

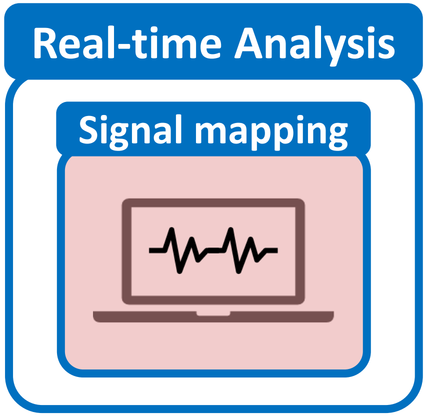
The existing solutions are **ineffective for large genomes**



Real-time Analysis

Basecalling

Read mapping



Real-time Analysis

Signal mapping

Costly and energy-hungry computations to basecall each read:
Portable sequencing becomes challenging with resource-constrained devices

Larger number of reference regions **cannot be handled accurately or quickly**, rendering existing solutions **ineffective for large genomes**

Applications of Read Until

Depletion: Reads mapping to a particular reference genome is ejected

- Removing contaminated reads from a sample
- Relative abundance estimation
- Controlling low/high-abundance genomes in a sample
- Controlling the sequencing of depth of a genome

Enrichment: Reads **not** mapping to a particular reference genome is ejected

- Purifying the sample to ensure it contains only the selected genomes
- Removing the host genome (e.g., human) in contamination analysis

Applications of Run Until and Sequence Until

Run Until: Stopping the sequencing without informative decision from analysis

- Stopping when reads reach to a particular depth of coverage
- Stopping when the abundance of all genomes reach a particular threshold

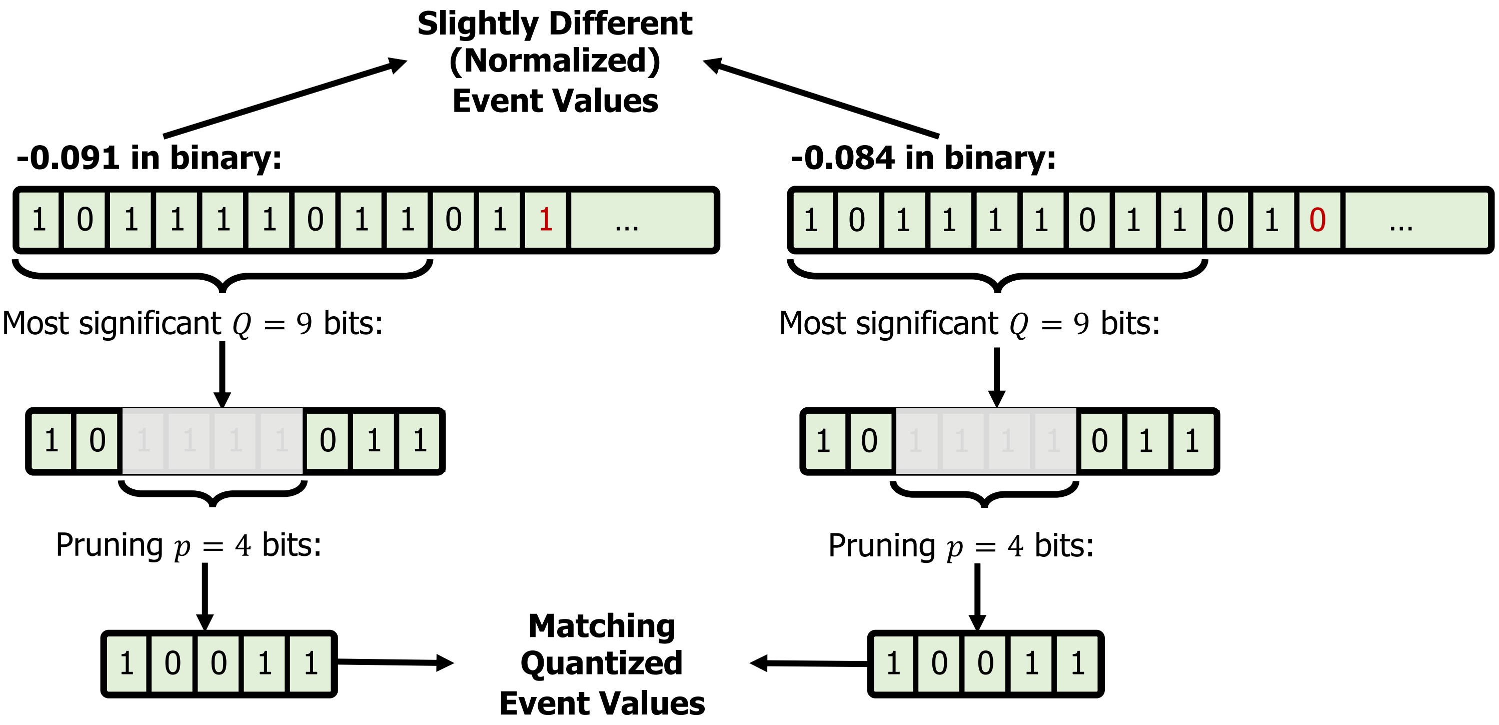
Sequence Until: Stopping the sequencing based on information decision

- Stopping when relative abundance estimations do not change substantially (for high-abundance genomes)
- Stopping when finding that the sample is contaminated with a particular set of genomes
- …

Details: Quantizing the Event Values

- **Observation:** Identical k-mers generate similar raw signals
  - **Challenge:** Their corresponding event values can be slightly different

- **Key Idea:** Quantize the event values
  - To enable assigning the **same quantized value** to the **similar event values**



SAFARI

Average Sequenced Bases and Chunks

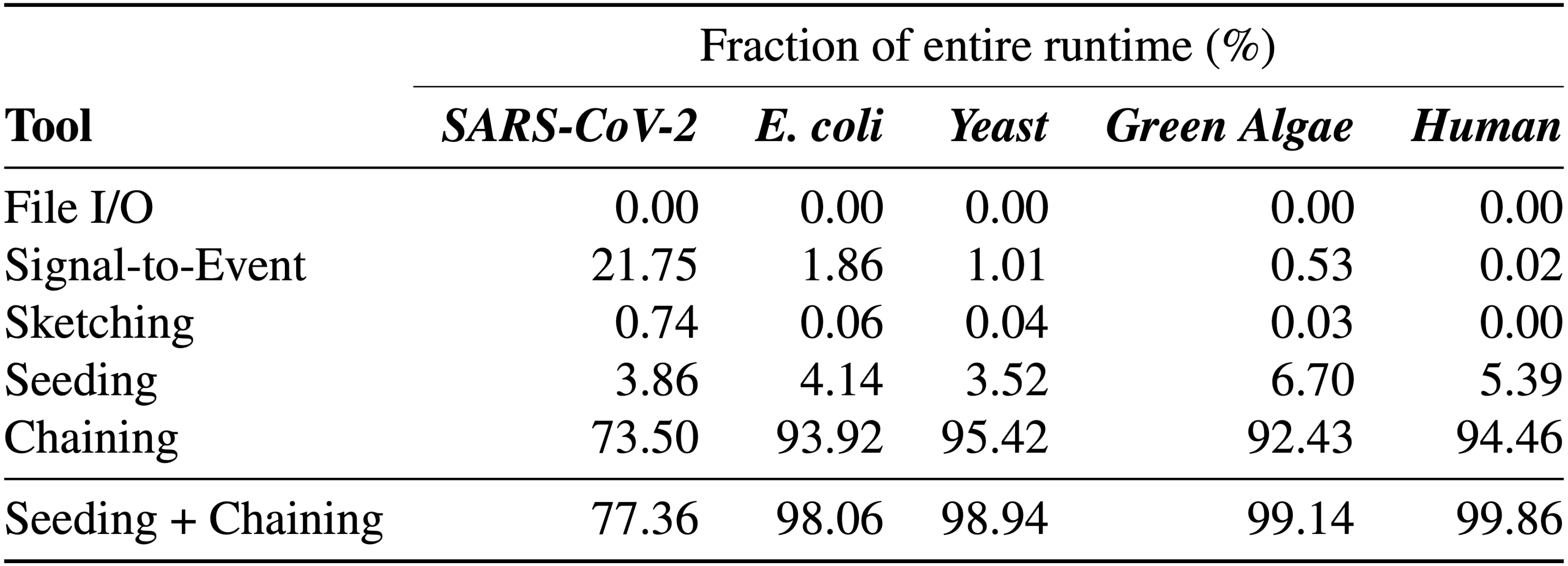
Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***
			Average sequenced base length per read		
UNCALLED	**184.51**	**580.52**	**1,233.20**	5,300.15	6,060.23
RawHash	513.95	1,376.14	2,565.09	**4,760.59**	**4,773.58**
			Average sequenced number of chunks per read		
Sigmap	**1.01**	**2.11**	**4.14**	**5.76**	**10.40**
RawHash	1.24	3.20	5.83	10.72	10.70

RawHash **reduces sequencing time and cost for large genomes**

up to **1.3**× compared to UNCALLED

Although Sigmap processes less number of chunks than RawHash, it fails to provide real-time analysis capabilities for large genomes

Breakdown Analysis of the RawHash Steps



Tool	Fraction of entire runtime (%)				
	SARS-CoV-2	***E. coli***	***Yeast***	***Green Algae***	***Human***
File I/O	0.00	0.00	0.00	0.00	0.00
Signal-to-Event	21.75	1.86	1.01	0.53	0.02
Sketching	0.74	0.06	0.04	0.03	0.00
Seeding	3.86	4.14	3.52	6.70	5.39
Chaining	73.50	93.92	95.42	92.43	94.46
Seeding + Chaining	77.36	98.06	98.94	99.14	99.86

The entire runtime is **bottlenecked by the chaining step**

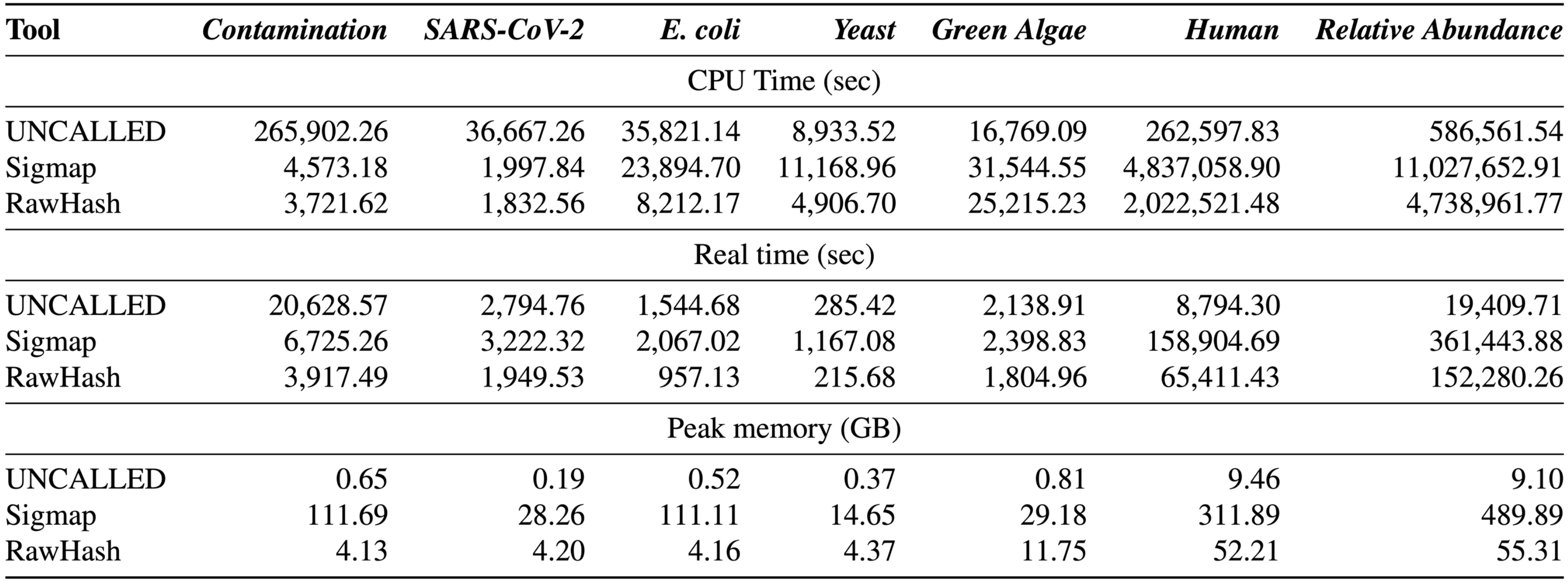
Required Computation Resources in Indexing

Tool	*Contamination*	*SARS-CoV-2*	*E. coli*	*Yeast*	*Green Algae*	*Human*	*Relative Abundance*
			CPU Time (sec)				
UNCALLED	8.72	9.00	11.08	18.62	285.88	4,148.10	4,382.38
Sigmap	0.02	0.04	8.66	24.57	449.29	36,765.24	40,926.76
RawHash	0.18	0.13	2.62	4.48	34.18	1,184.42	788.88
			Real time (sec)				
UNCALLED	1.01	1.04	2.67	7.79	280.27	4,190.00	4,471.82
Sigmap	0.13	0.25	9.31	25.86	458.46	37,136.61	41,340.16
RawHash	0.14	0.10	1.70	2.06	15.82	278.69	154.68
			Peak memory (GB)				
UNCALLED	0.07	0.07	0.13	0.31	11.96	48.44	47.81
Sigmap	0.01	0.01	0.40	1.04	8.63	227.77	238.32
RawHash	0.01	0.01	0.35	0.76	5.33	83.09	152.80

The indexing step of RawHash is **orders of magnitude faster** than the indexing steps of UNCALLED and Sigmap, especially **for large genomes**

RawHash requires **larger memory space** than UNCALLED

Required Computation Resources in Mapping

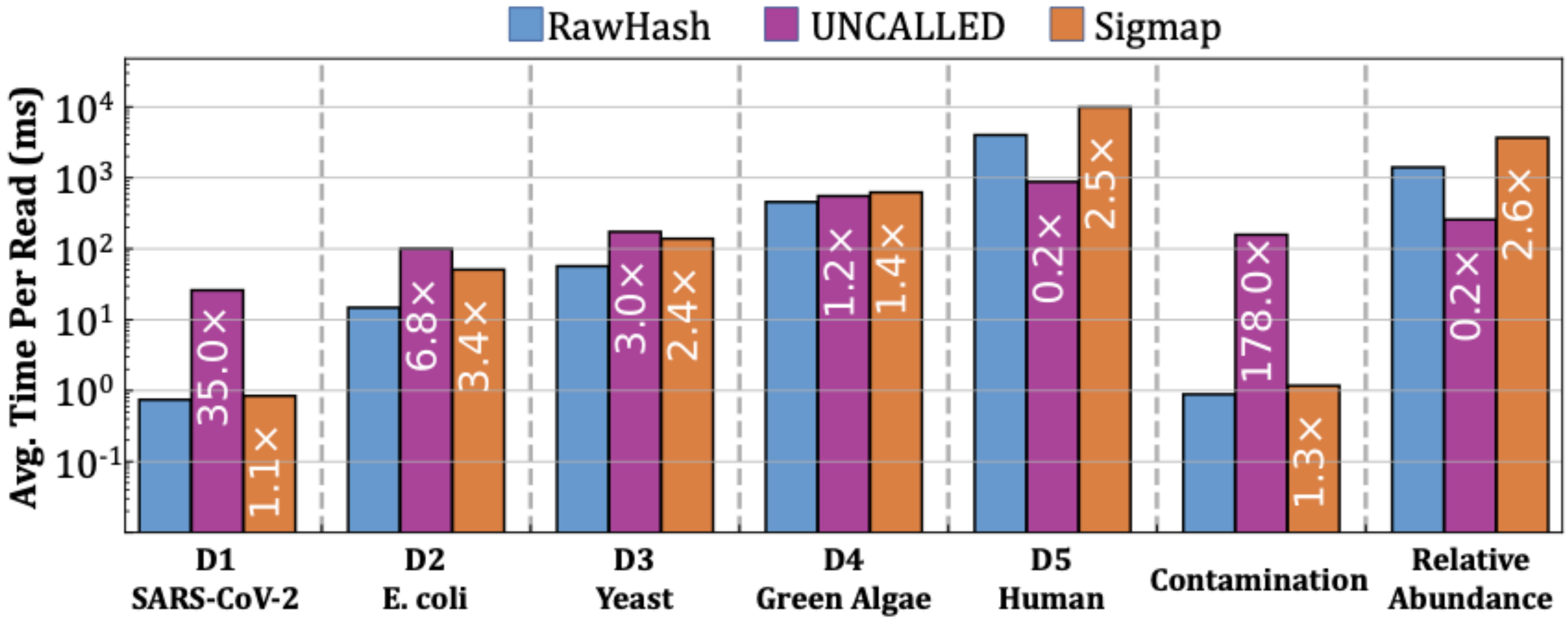


Tool	***Contamination***	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	***Relative Abundance***
				CPU Time (sec)			
UNCALLED	265,902.26	36,667.26	35,821.14	8,933.52	16,769.09	262,597.83	586,561.54
Sigmap	4,573.18	1,997.84	23,894.70	11,168.96	31,544.55	4,837,058.90	11,027,652.91
RawHash	3,721.62	1,832.56	8,212.17	4,906.70	25,215.23	2,022,521.48	4,738,961.77
				Real time (sec)			
UNCALLED	20,628.57	2,794.76	1,544.68	285.42	2,138.91	8,794.30	19,409.71
Sigmap	6,725.26	3,222.32	2,067.02	1,167.08	2,398.83	158,904.69	361,443.88
RawHash	3,917.49	1,949.53	957.13	215.68	1,804.96	65,411.43	152,280.26
				Peak memory (GB)			
UNCALLED	0.65	0.19	0.52	0.37	0.81	9.46	9.10
Sigmap	111.69	28.26	111.11	14.65	29.18	311.89	489.89
RawHash	4.13	4.20	4.16	4.37	11.75	52.21	55.31

The mapping step of RawHash is **significantly faster than Sigmap** for all genomes, and **faster than UNCALLED for small genomes**

RawHash requires **larger memory space** than UNCALLED

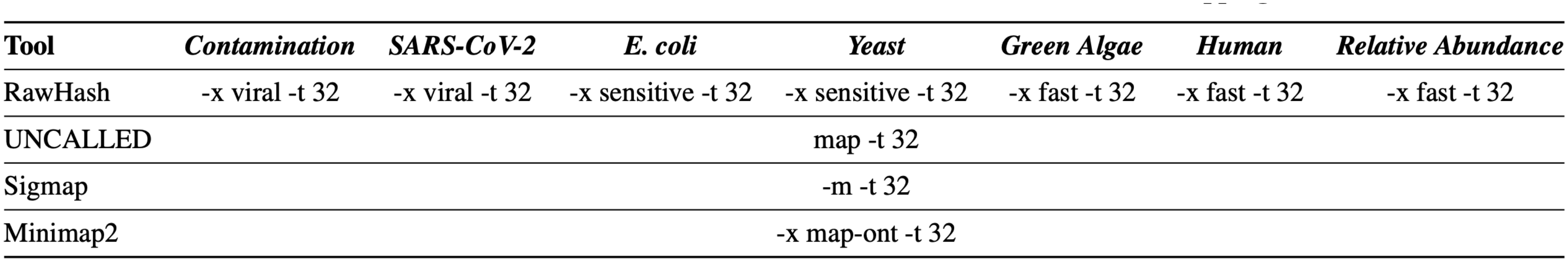
Average Mapping Time per Read



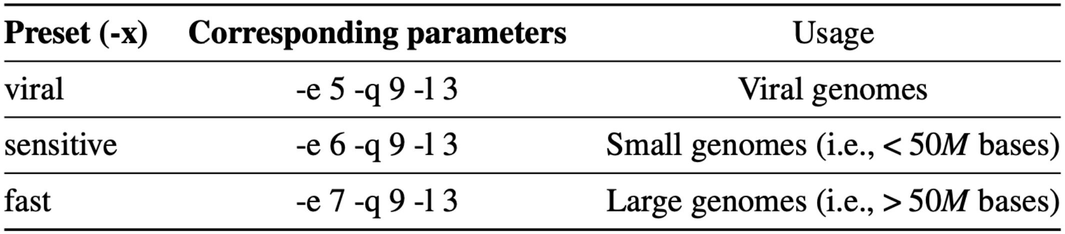
The mapping step of RawHash is **significantly faster than Sigmap** for all genomes, and **faster than UNCALLED for small genomes**

SAFARI

Parameter Configurations

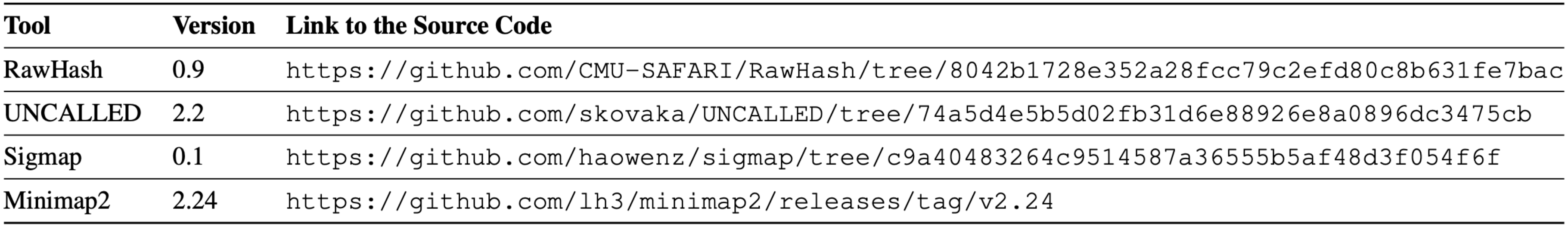


Tool	*Contamination*	*SARS-CoV-2*	*E. coli*	*Yeast*	*Green Algae*	*Human*	*Relative Abundance*
RawHash	-x viral -t 32	-x viral -t 32	-x sensitive -t 32	-x sensitive -t 32	-x fast -t 32	-x fast -t 32	-x fast -t 32
UNCALLED	map -t 32						
Sigmap	-m -t 32						
Minimap2	-x map-ont -t 32						



Preset (-x)	**Corresponding parameters**	Usage
viral	-e 5 -q 9 -l 3	Viral genomes
sensitive	-e 6 -q 9 -l 3	Small genomes (i.e., $< 50M$ bases)
fast	-e 7 -q 9 -l 3	Large genomes (i.e., $> 50M$ bases)

Versions



Tool	Version	Link to the Source Code
RawHash	0.9	https://github.com/CMU-SAFARI/RawHash/tree/8042b1728e352a28fcc79c2efd80c8b631fe7bac
UNCALLED	2.2	https://github.com/skovaka/UNCALLED/tree/74a5d4e5b5d02fb31d6e88926e8a0896dc3475cb
Sigmap	0.1	https://github.com/haowenz/sigmap/tree/c9a40483264c9514587a36555b5af48d3f054f6f
Minimap2	2.24	https://github.com/lh3/minimap2/releases/tag/v2.24