



# RawHash

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

**Can Firtina**

Nika Mansouri Ghiasi

Joel Lindegger

Gagandeep Singh

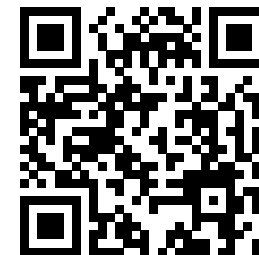
Meryem Banu Cavlak

Haiyu Mao

Onur Mutlu



[Paper](#)



[Code](#)

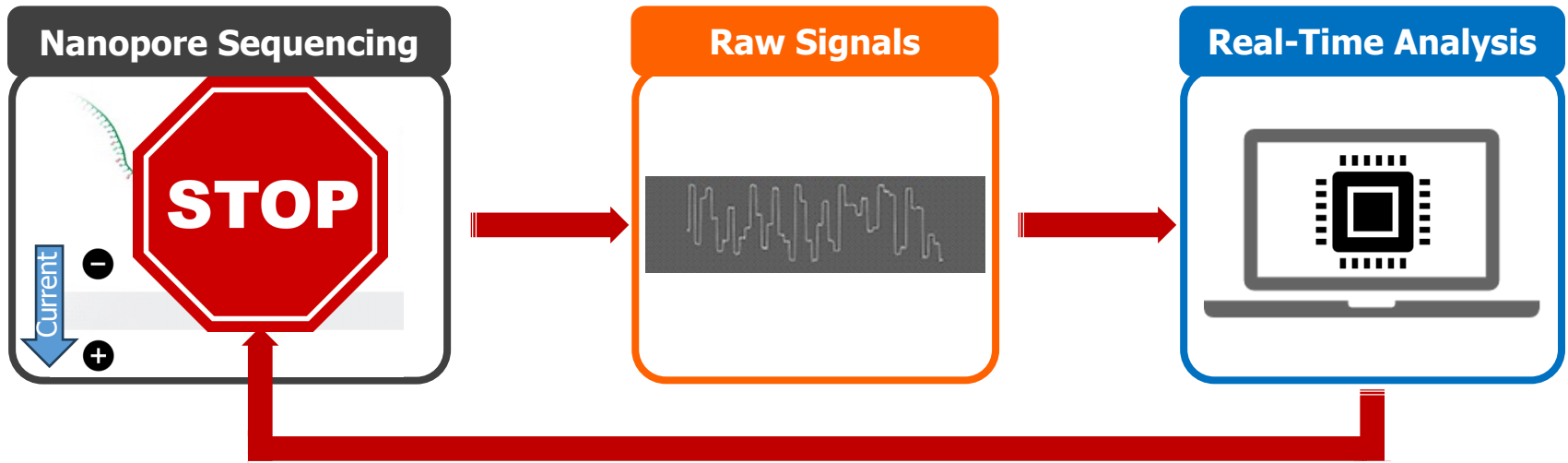
# Nanopore Sequencing

**Nanopore Sequencing:** a widely used sequencing technology

- Can sequence large fragments of nucleic acid molecules (up to >2Mbp)
- Offers high throughput
- Cost-effective
- Enables **real-time genome analysis**



# Real-Time Analysis with Nanopore Sequencing



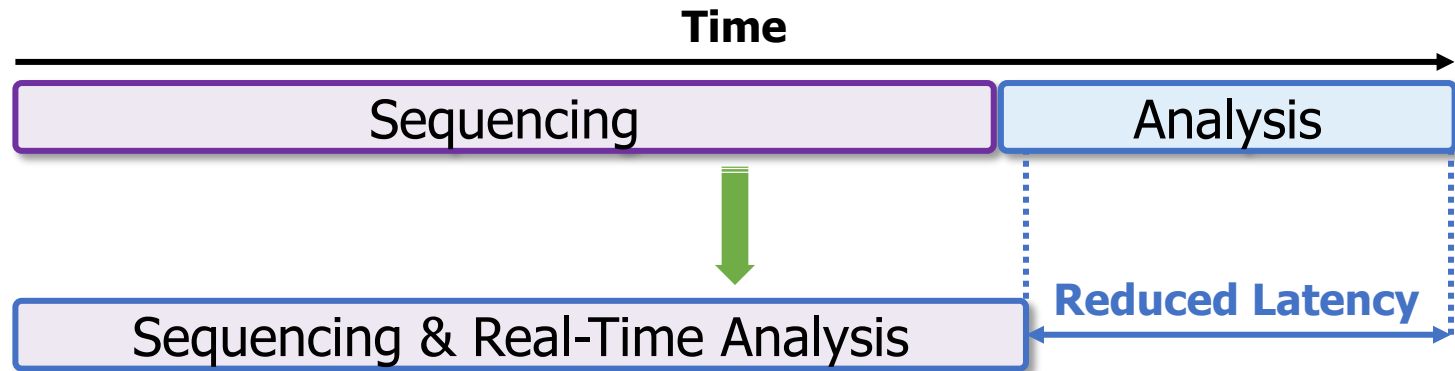
**Raw Signals:** Ionic current measurements generated at a certain **throughput**

**Real-Time Analysis:** Analyzing all raw signals by **matching the throughput**

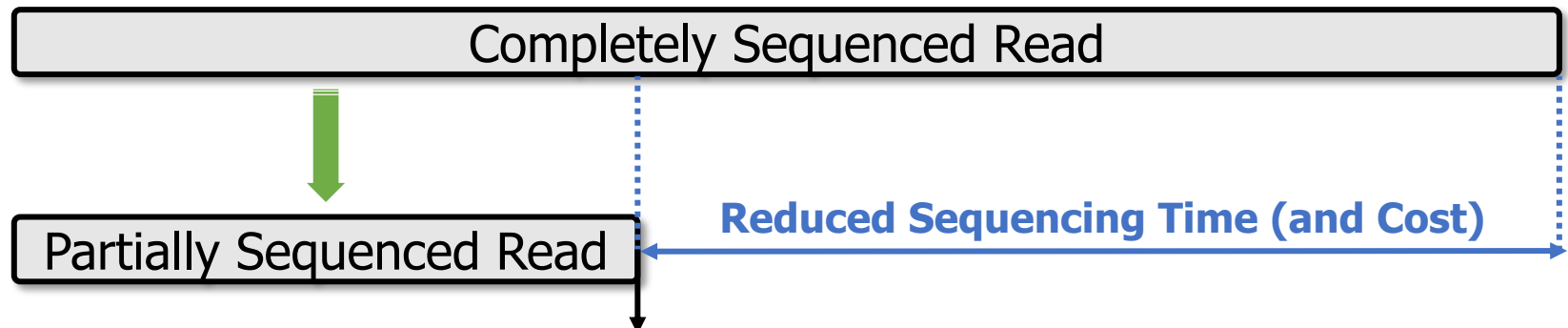
**Real-Time Decisions:** Stopping sequencing **early** based on real-time analysis

# Benefits of Real-Time Genome 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 Genome 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

# 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:

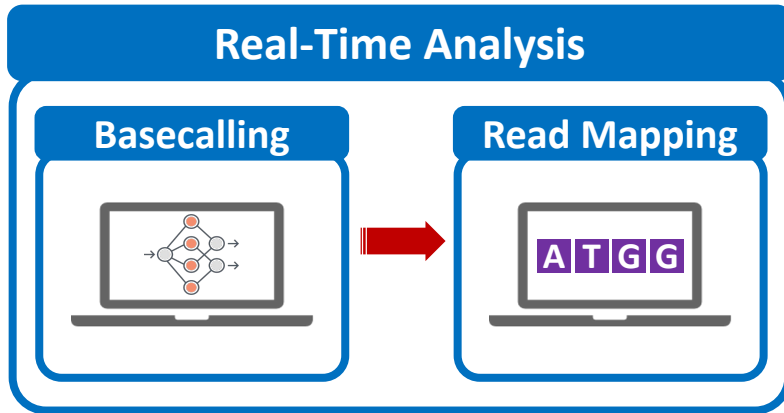
- 1) The **first hash-based mechanism** that can quickly and accurately analyze raw nanopore signals for **large genomes**
- 2) The novel **Sequence Until** technique can accurately and **dynamically stop the entire sequencing of all reads at once** if further sequencing is not necessary

**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

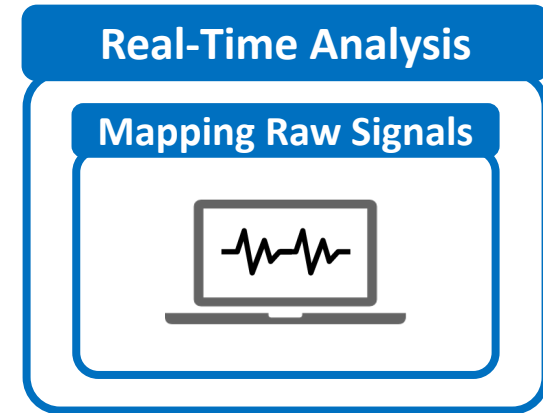
1. Deep neural networks (**DNNs**) for translating **signals** to **bases**



Less noisy analysis from basecalled sequences

**Costly and power-hungry** computational requirements

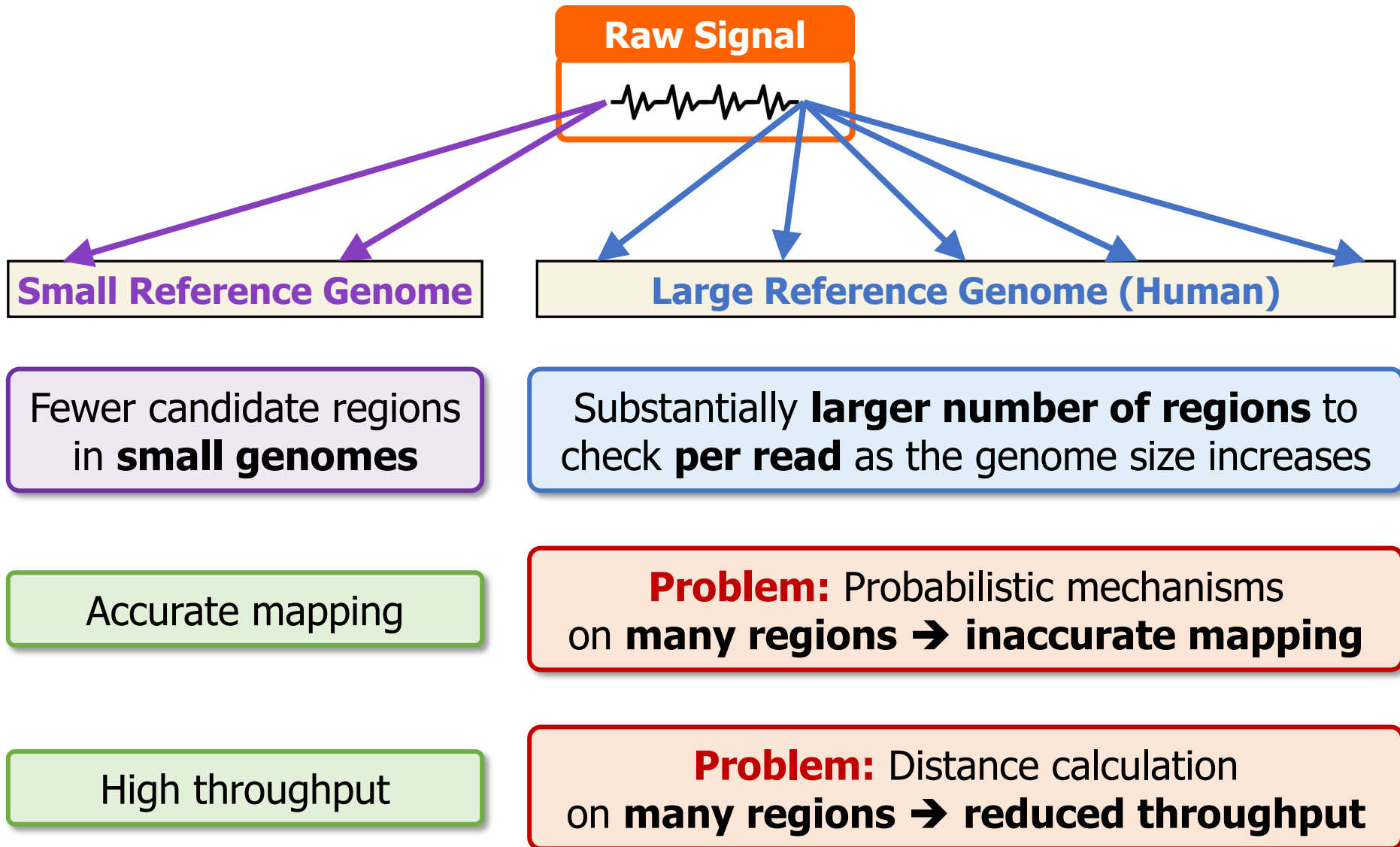
2. Mapping **signals** to reference genomes **without** basecalling



Raw signals contain richer information than bases

Efficient analysis with better scalability and portability

# The Problem – Mapping Raw Signals



# 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





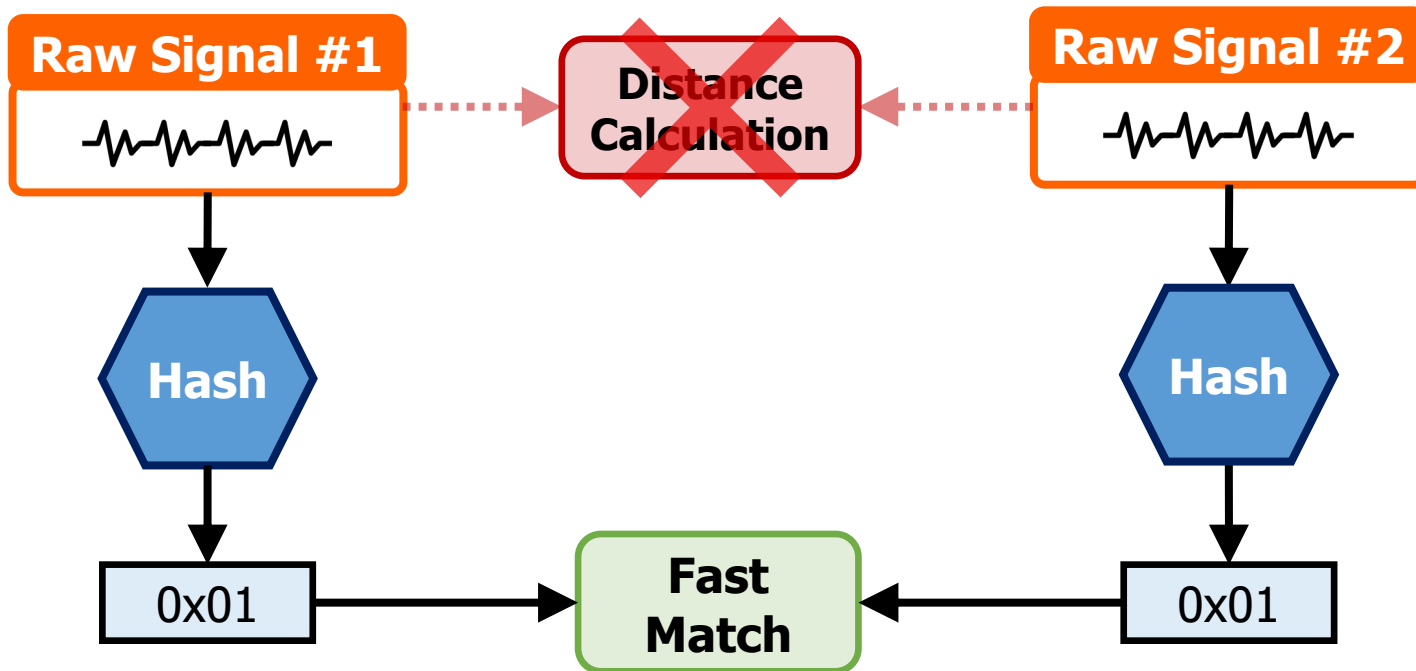
# RawHash

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

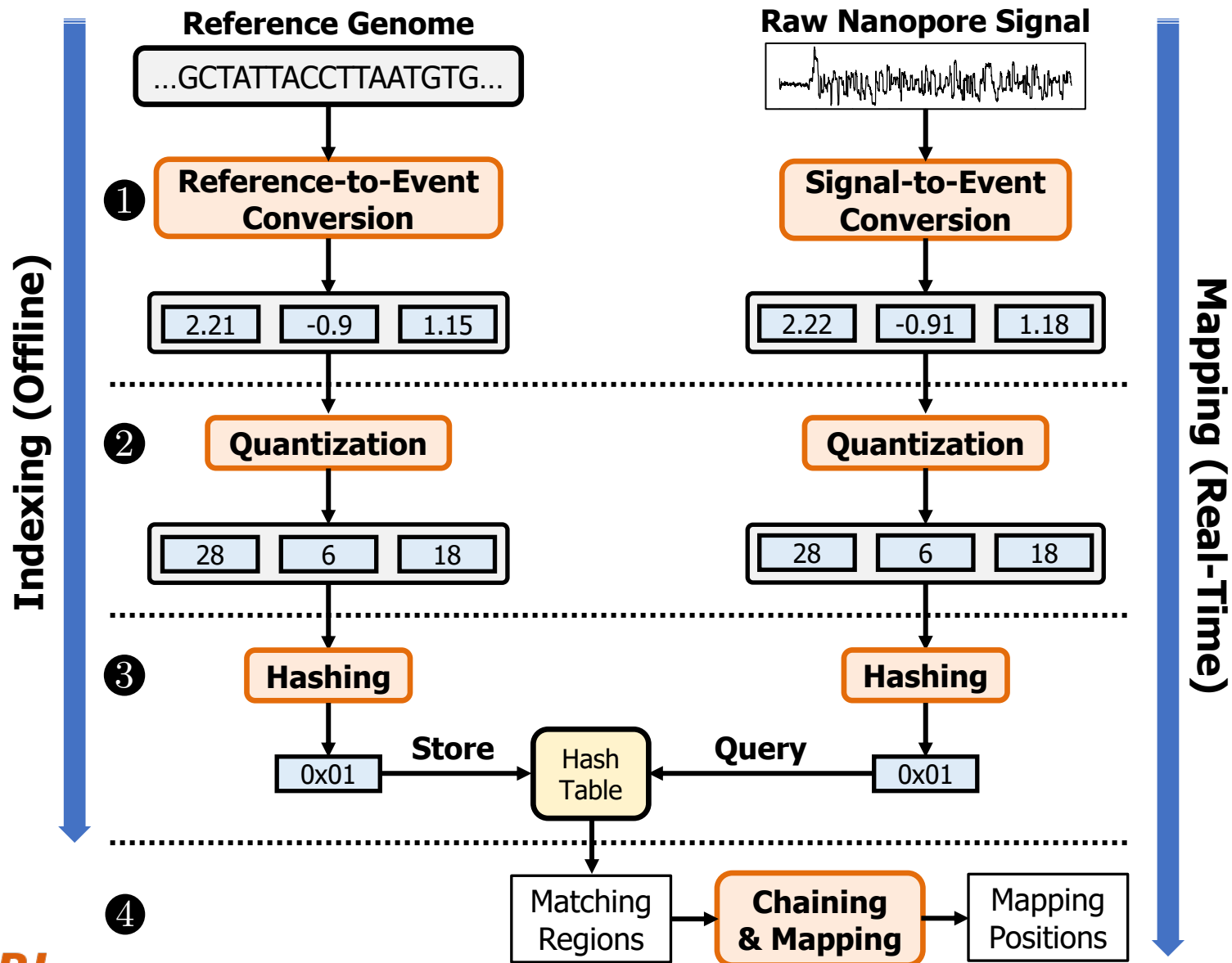
**Key Observation:** **Identical** nucleotides generate **similar** raw signals



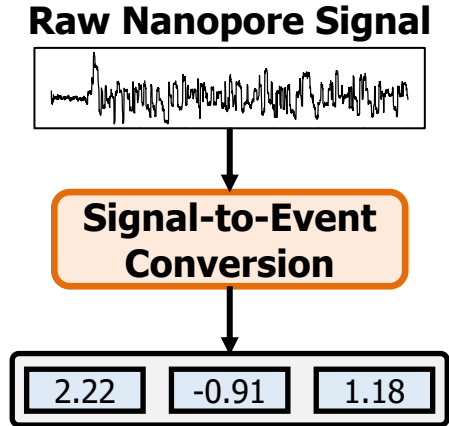
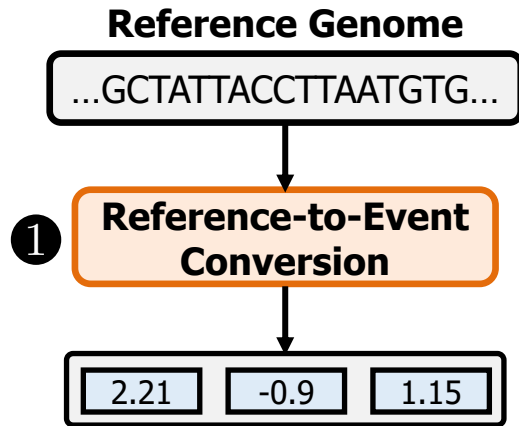
**Challenge #1:** Generating the **same** hash value for **similar enough** signals

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

# RawHash Overview

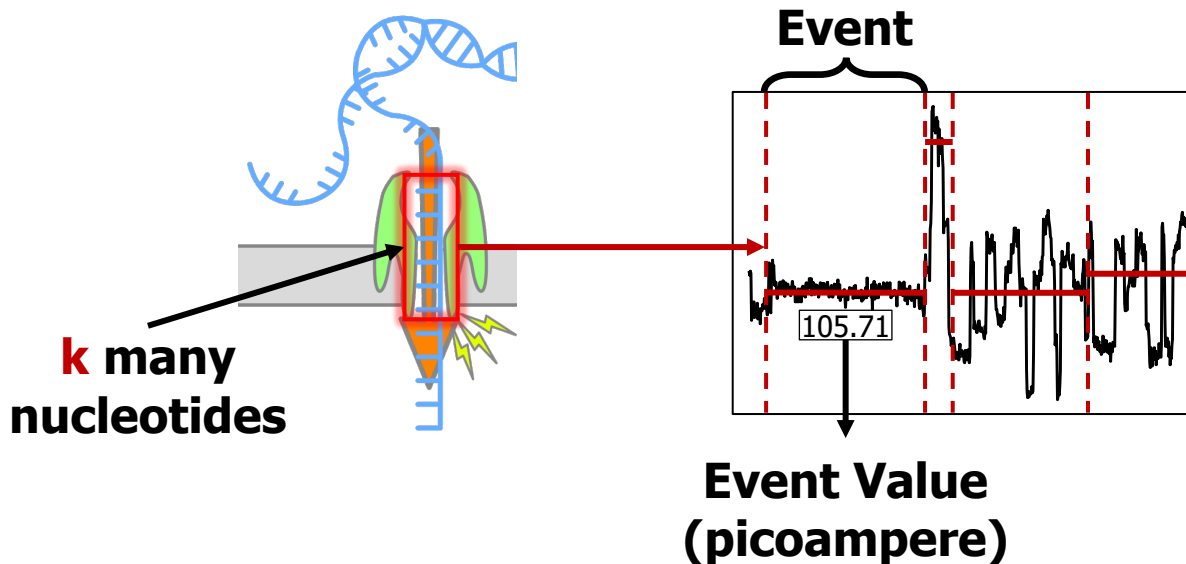


# RawHash Overview



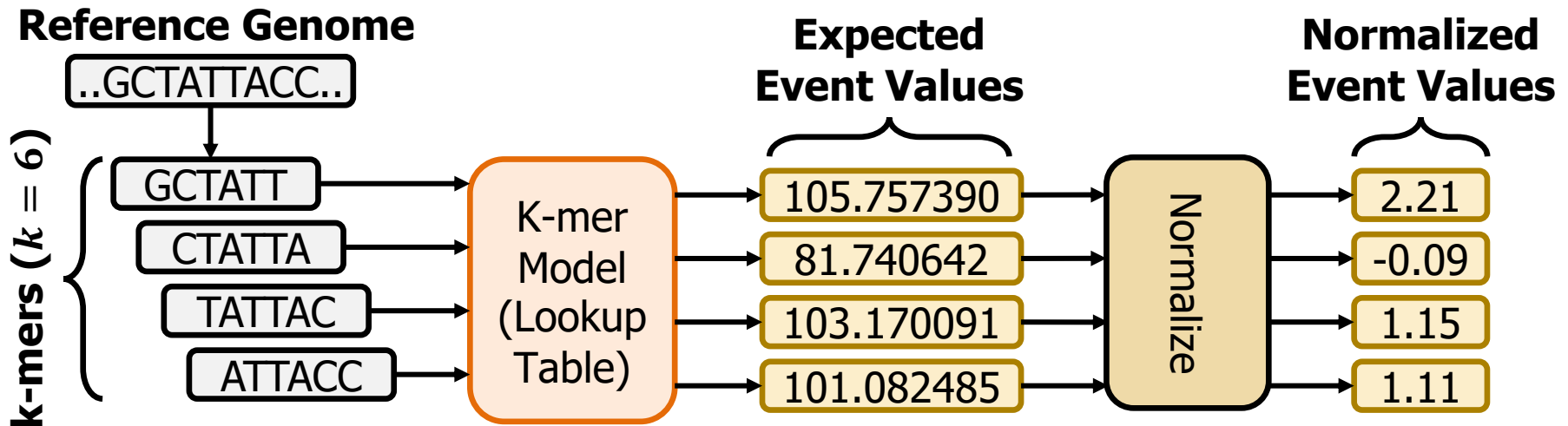
# Events in Raw Nanopore Signals

- **Event:** A **segment** of the raw signal
  - Corresponds to a **particular k-mer**
- **Event detection** finds these segments to identify **k-mers**
  - Start and end positions are marked by abrupt signal changes
  - Statistical methods identify these abrupt changes
  - **Event value:** average of signals **within an event**



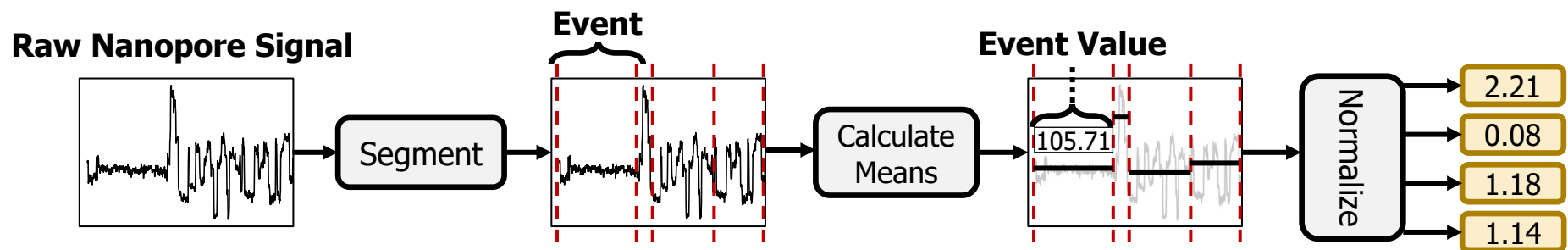
# Reference-to-Event Conversion

- **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

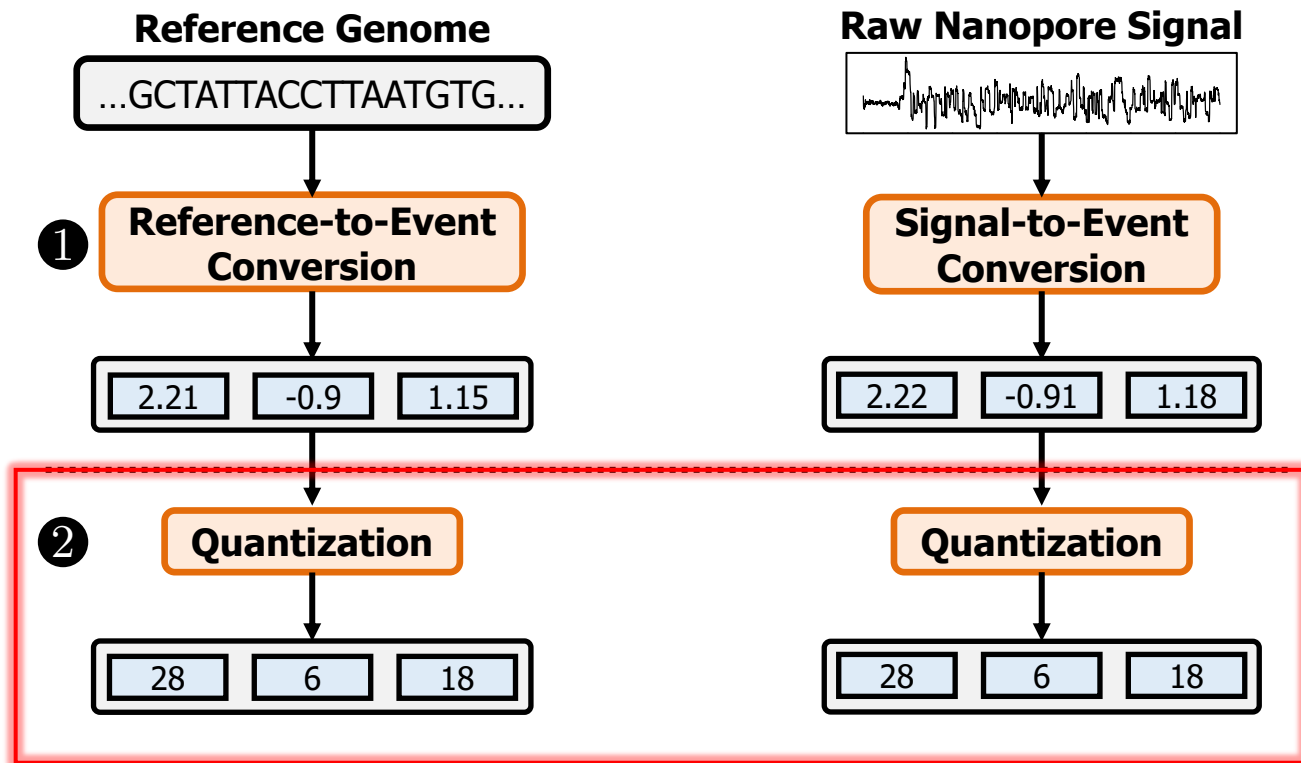
- **Event detection:** Identifies signal regions corresponding to specific k-mers
  - Uses statistical test (**segmentation**) to spot abrupt signal changes

Can we match events (k-mers) between reference genome and raw signals?

- Consecutive events → consecutive k-mers

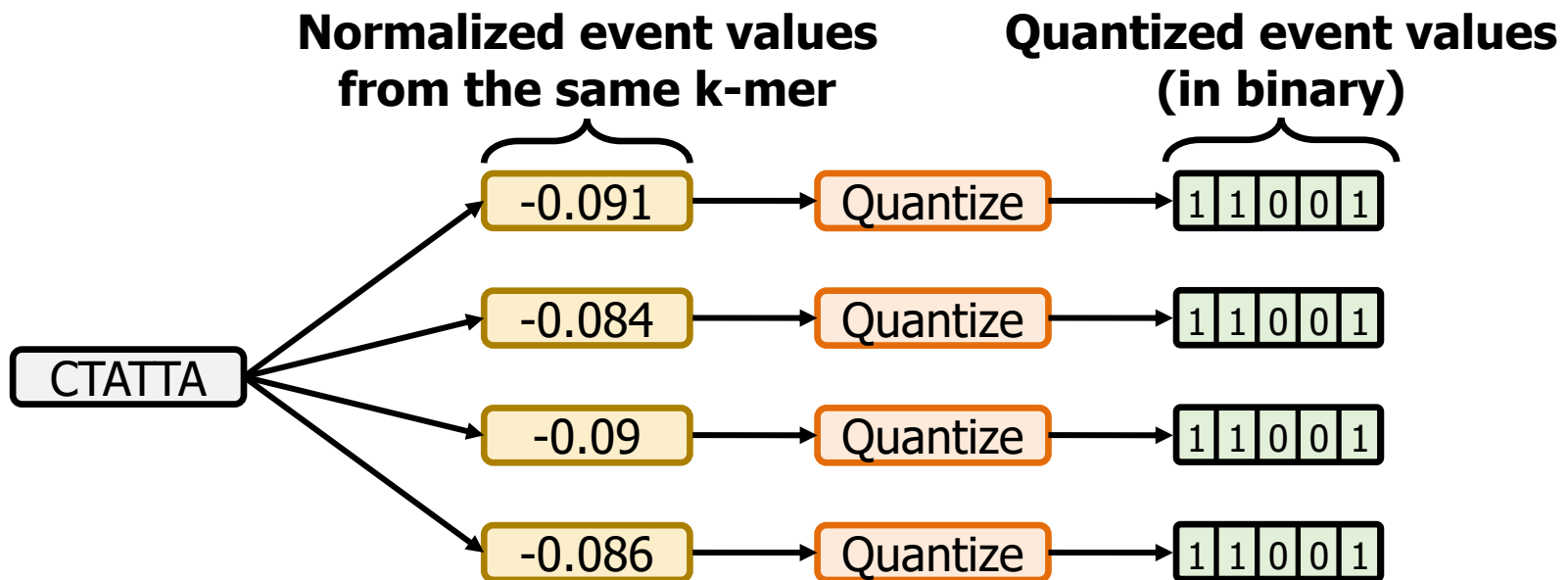


# RawHash Overview

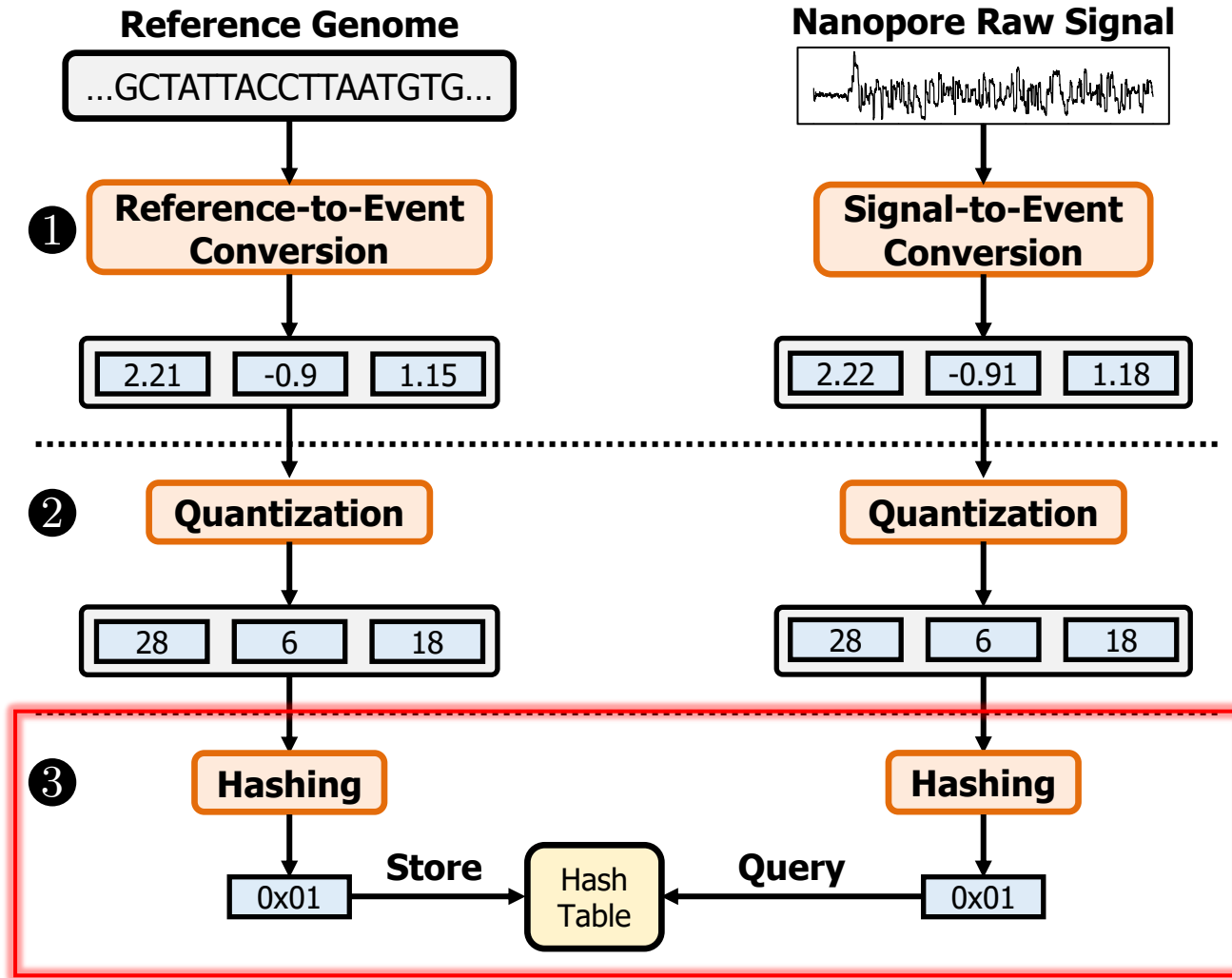


# Quantizing the Event Values

- **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**

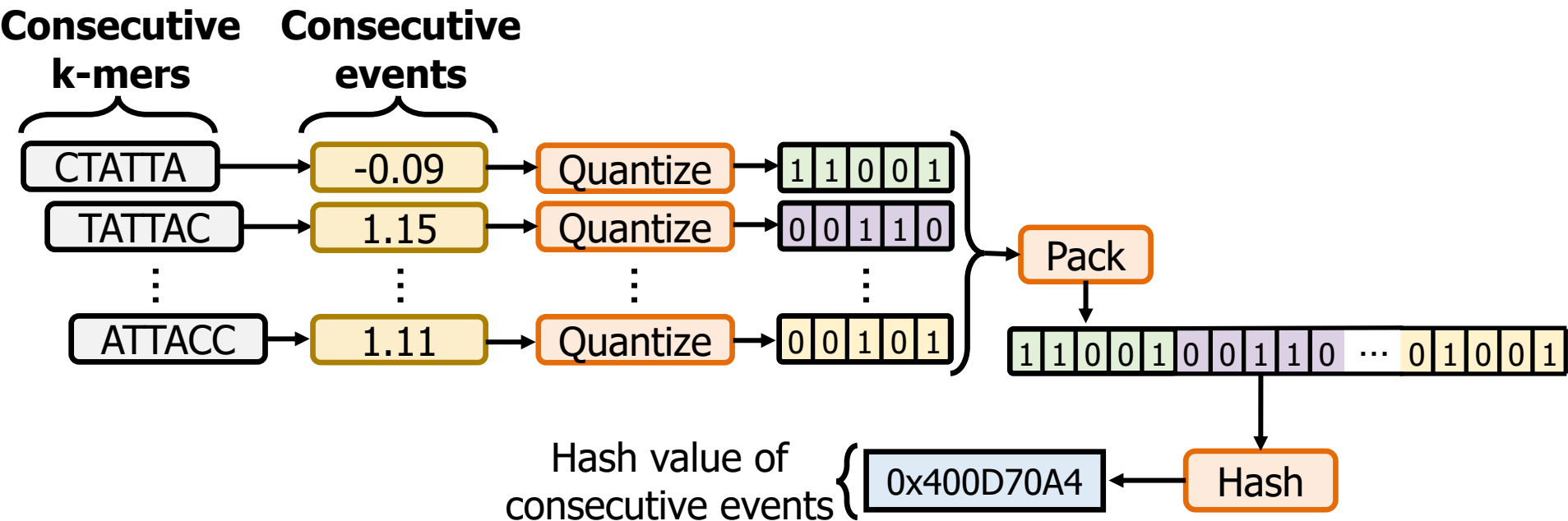


# RawHash Overview

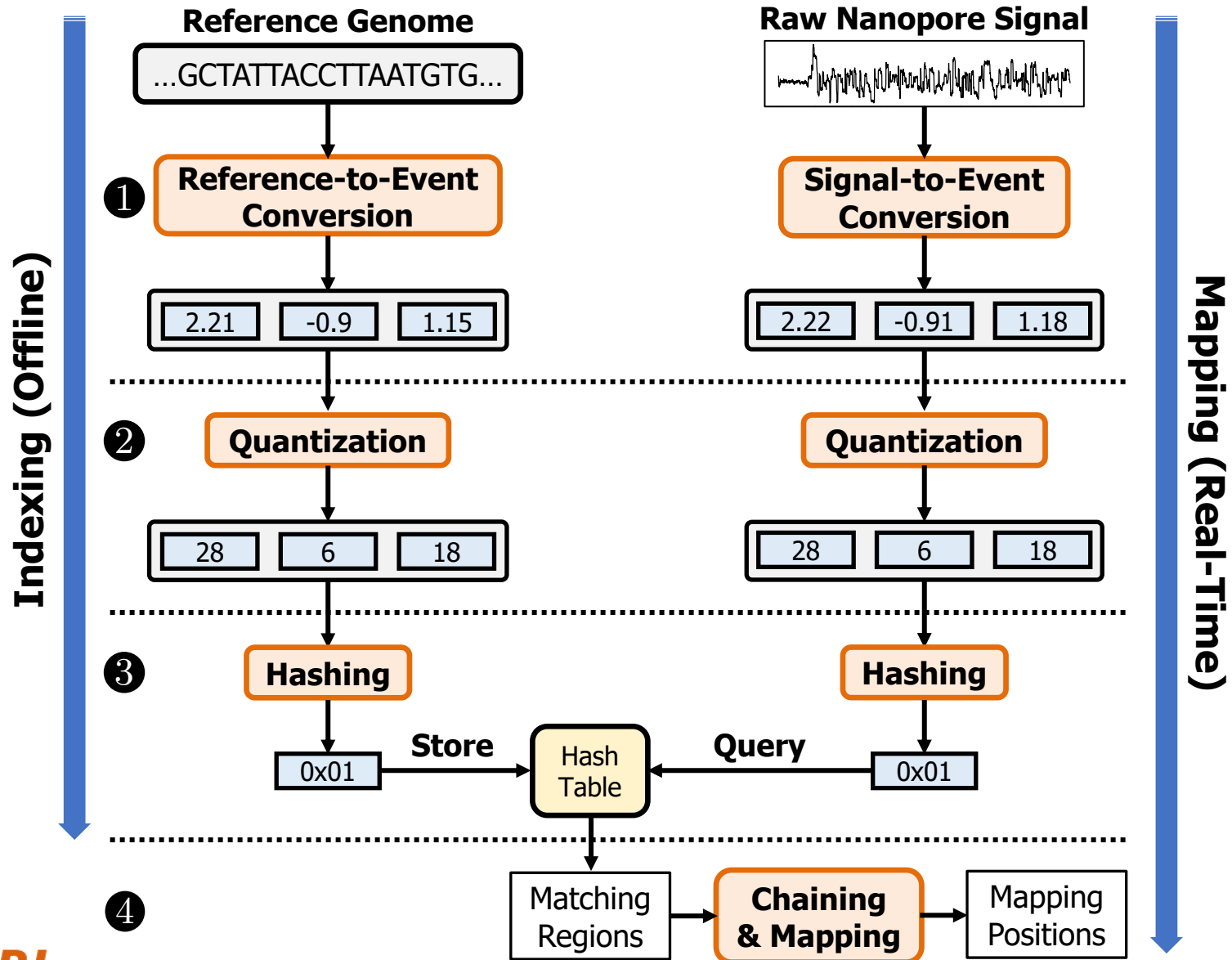


# Hashing for Fast Similarity Search

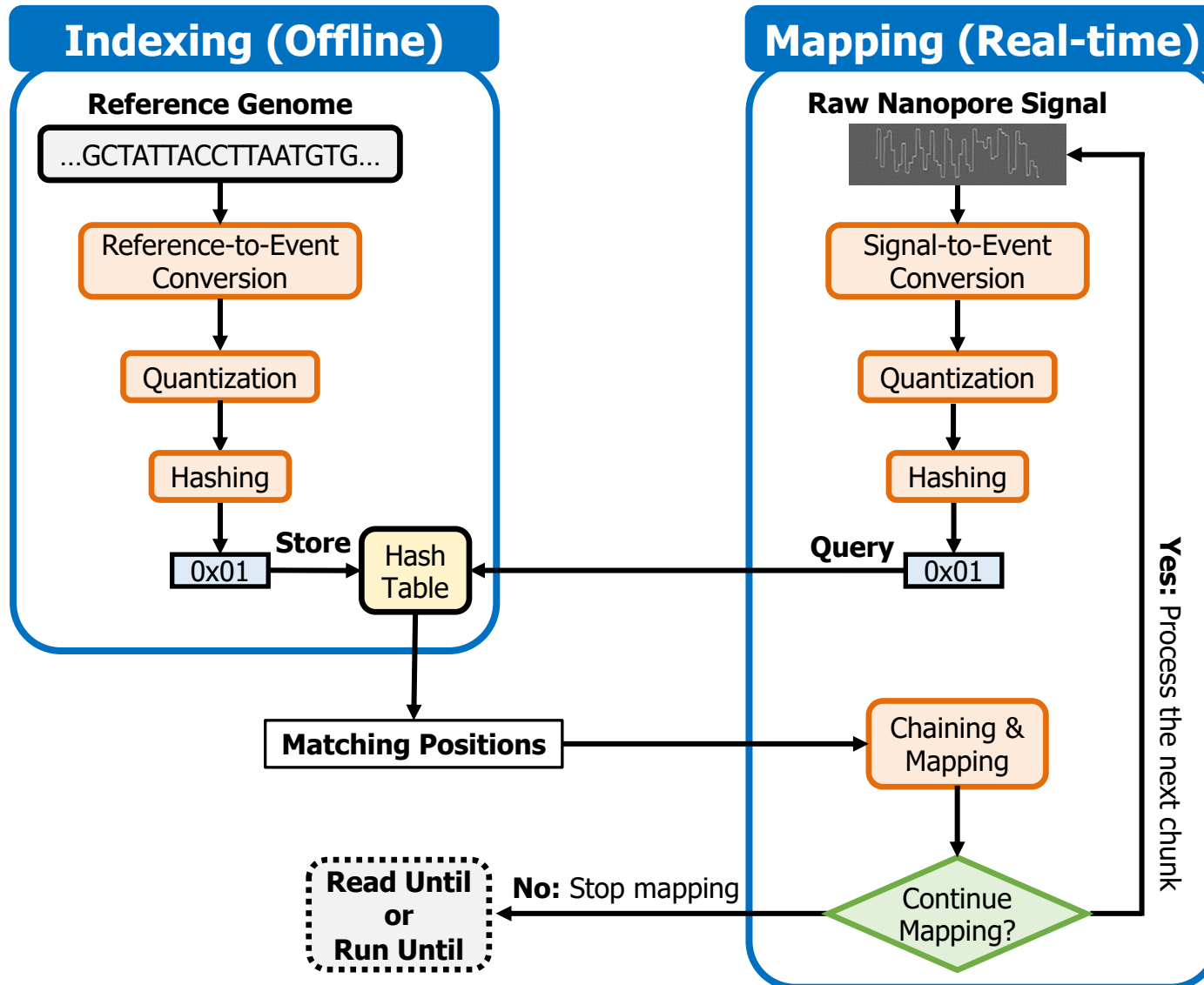
- Each event usually represents a very small k-mer (6 to 9 characters)
  - **Challenge:** Short k-mers are likely to appear in many locations
- **Key Idea:** Create longer k-mers from many **consecutive events**
- **Key Benefit:** Directly match hash values to quickly identify similarities



# RawHash Overview



# 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**

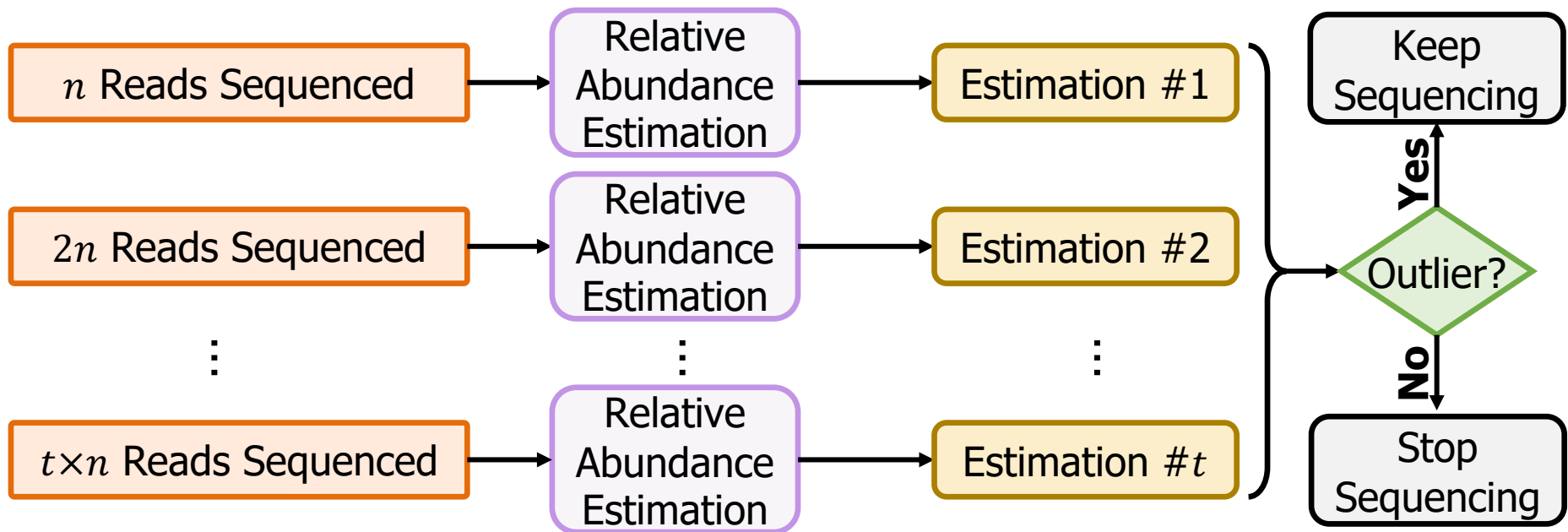
- Example real-time genome analysis use case:

- **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

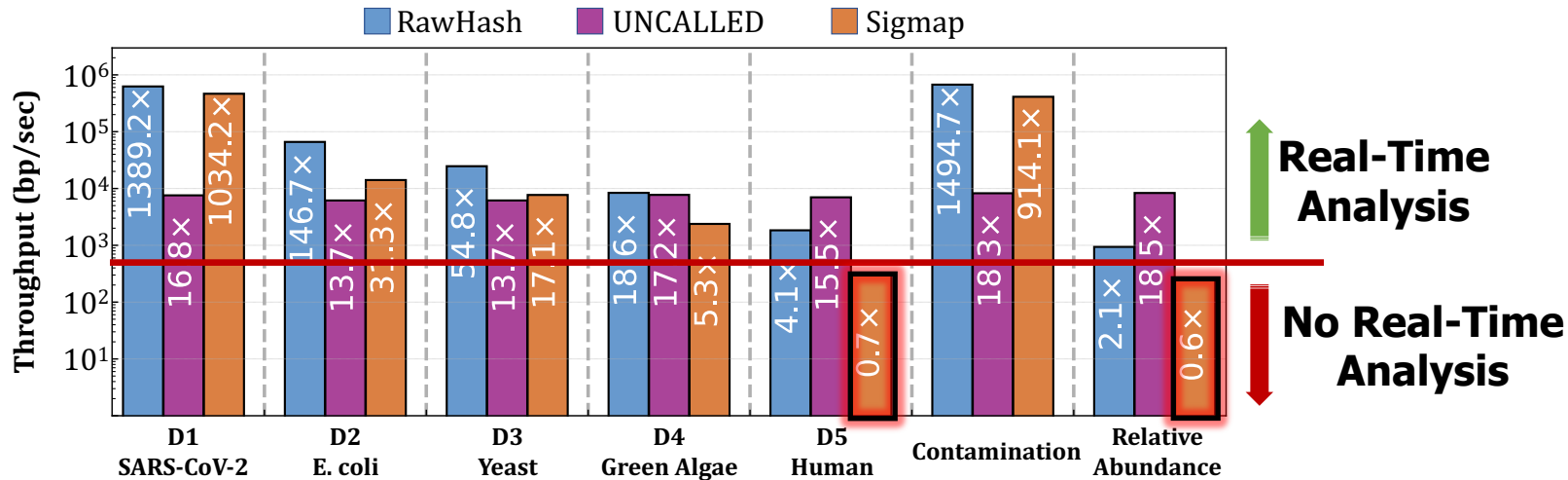
- Evaluation metrics:
  - **Throughput** (bases processed per second)
  - Potential reduction in **sequencing time and cost**
  - **Accuracy**
    - **Baseline:** Mapping basecalled reads using minimap2
    - Precision, recall, and F1 scores
    - Relative abundance estimation distance to ground truth

- **Datasets:**

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

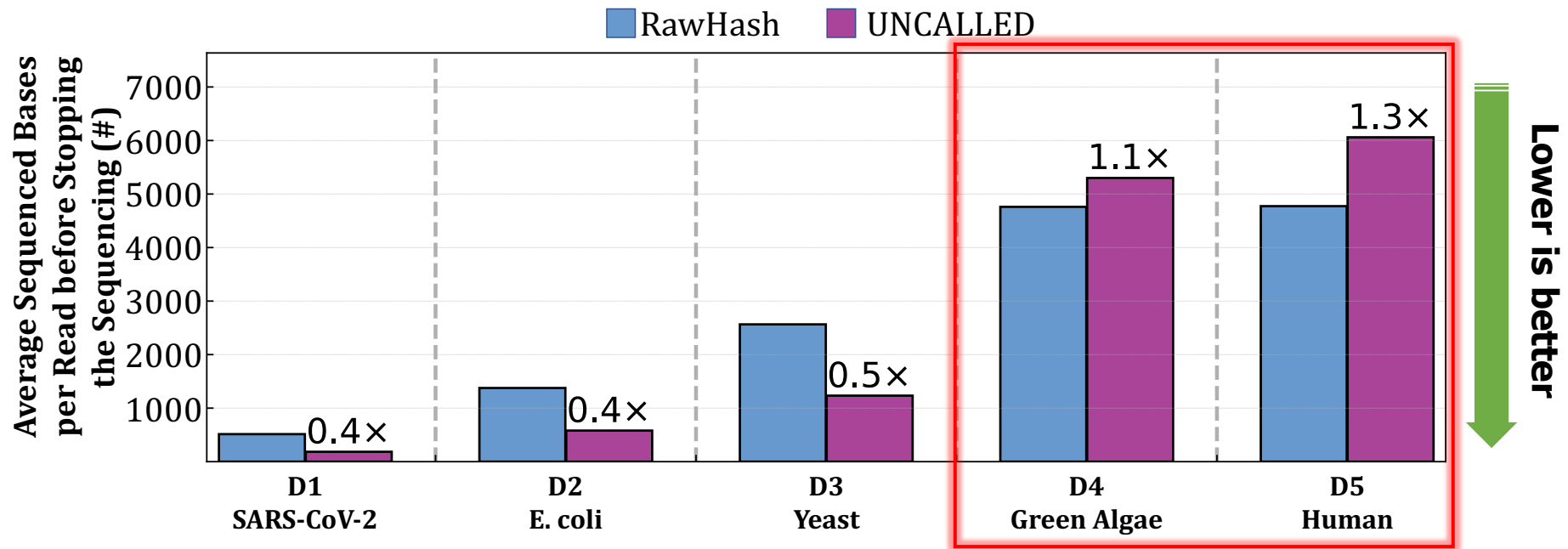


**25.8x and 3.4x** better average throughput compared to **UNCALLED and Sigmap**, respectively

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 1.3x** compared to UNCALLED

# Mapping Accuracy

- Read mapping accuracy of each tool and each use case

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

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

Tool	Estimated Relative Abundance Ratios					Distance
	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>	
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	<b>0.0847</b>

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**

Tool	Estimated Relative Abundance Ratios in 50,000 Random Reads					
	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>	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**

# Simulating Sequence Until

- Real relative abundance results using the entire set of reads

Tool	Estimated Relative Abundance Ratios					
	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>	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	<b>0.0847</b>

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

Tool	Estimated Relative Abundance Ratios					
	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>	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	<b>0.0995</b>
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	<b>0.1004</b>
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	<b>0.0928</b>
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	<b>0.1136</b>
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	<b>0.2232</b>

# Simulating Sequence Until

- Real relative abundance results using the entire set of reads

Tool	Estimated Relative Abundance Ratios					Distance
	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>	
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	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>	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	<b>0.0995</b>
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	<b>0.1004</b>
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	<b>0.0928</b>
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	<b>0.1136</b>
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	<b>0.2232</b>

# 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**

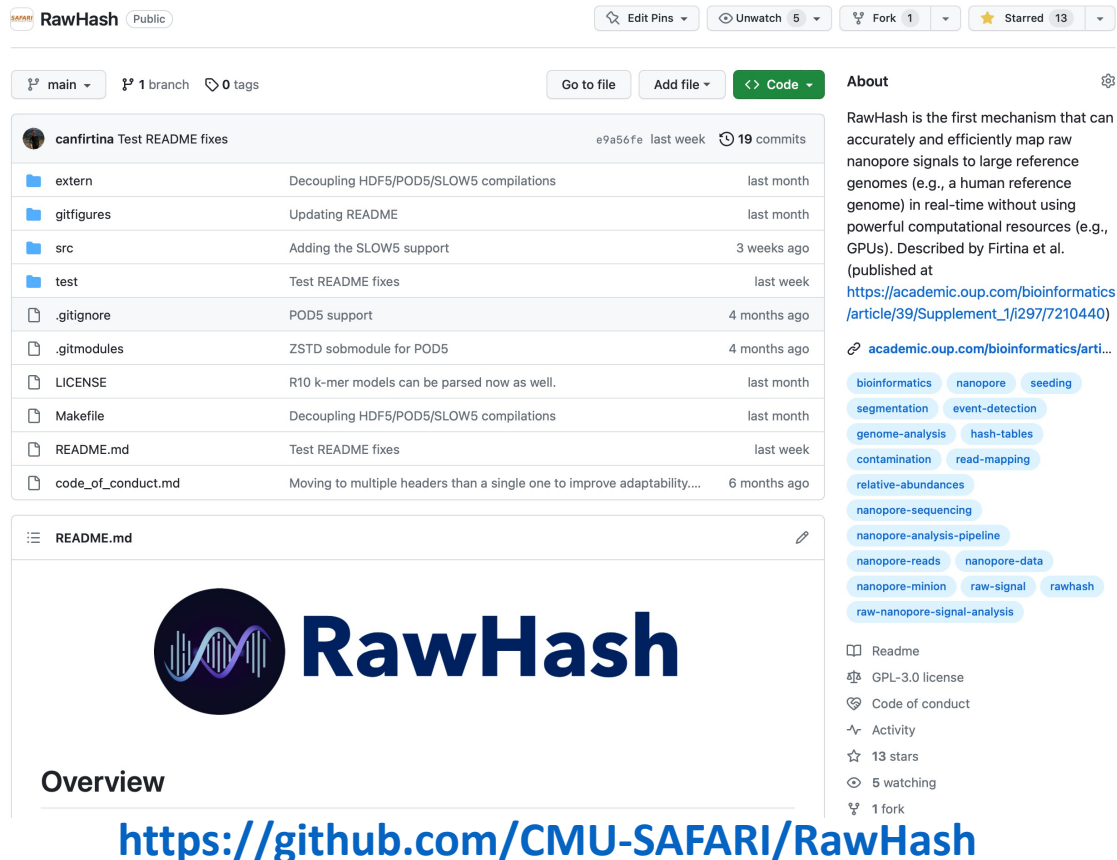
**Can Firtina** <sup>1,\*</sup>, **Nika Mansouri Ghiasi** <sup>1</sup>, **Joel Lindegger** <sup>1</sup>, **Gagandeep Singh** <sup>1</sup>,  
**Meryem Banu Cavlak** <sup>1</sup>, **Haiyu Mao** <sup>1</sup>, **Onur Mutlu** <sup>1,\*</sup>

<sup>1</sup>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](mailto:firtinac@ethz.ch) (C.F.), [omutlu@ethz.ch](mailto: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



RawHash


Public

Edit Pins Unwatch 5 Fork 1 Starred 13

main 1 branch 0 tags Go to file Add file Code

File/Folder	Description	Last Commit
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 submodule 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

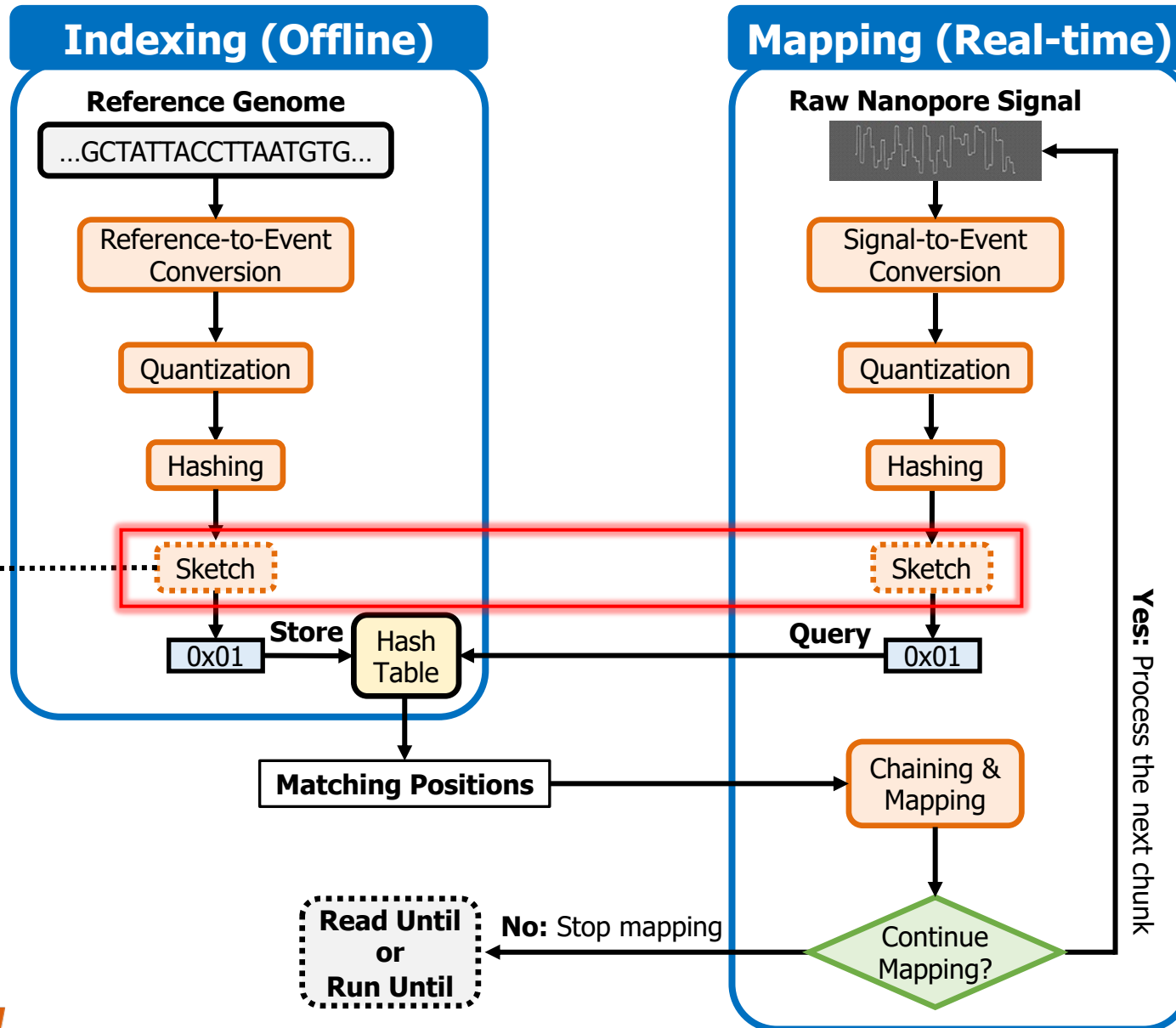
<https://github.com/CMU-SAFARI/RawHash>

bioinformatics nanopore seeding segmentation event-detection genome-analysis hash-tables contamination read-mapping relative-abundances nanopore-sequencing nanopore-analysis-pipeline nanopore-reads nanopore-data nanopore-minion raw-signal rawhash raw-nanopore-signal-analysis

Readme GPL-3.0 license Code of conduct Activity 13 stars 5 watching 1 fork



# Sketching with Hash-based Indexing





# Outline

Background

RawHash

Evaluation

Conclusion

# Conclusion

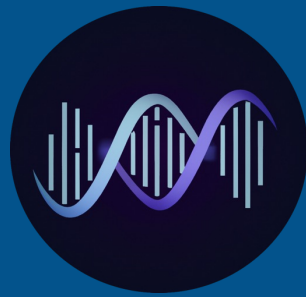
## Key Contributions:

- 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



# RawHash

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

**Can Firtina**

Nika Mansouri Ghiasi

Joel Lindegger

Gagandeep Singh

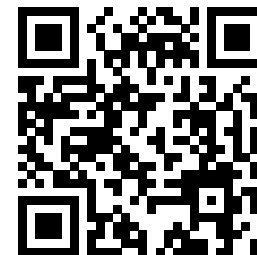
Meryem Banu Cavlak

Haiyu Mao

Onur Mutlu



[Paper](#)



[Code](#)

# Backup Slides

# Practical Similarity Identification Seeds

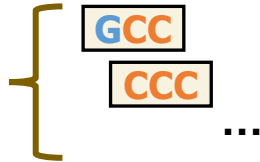
Reference



Read



K-mers



K-mers Locations

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

Index (Hash Table)

Seeding

Determine potential matching regions (seeds) in the reference genome

Seed Filtering  
(e.g., Chaining)

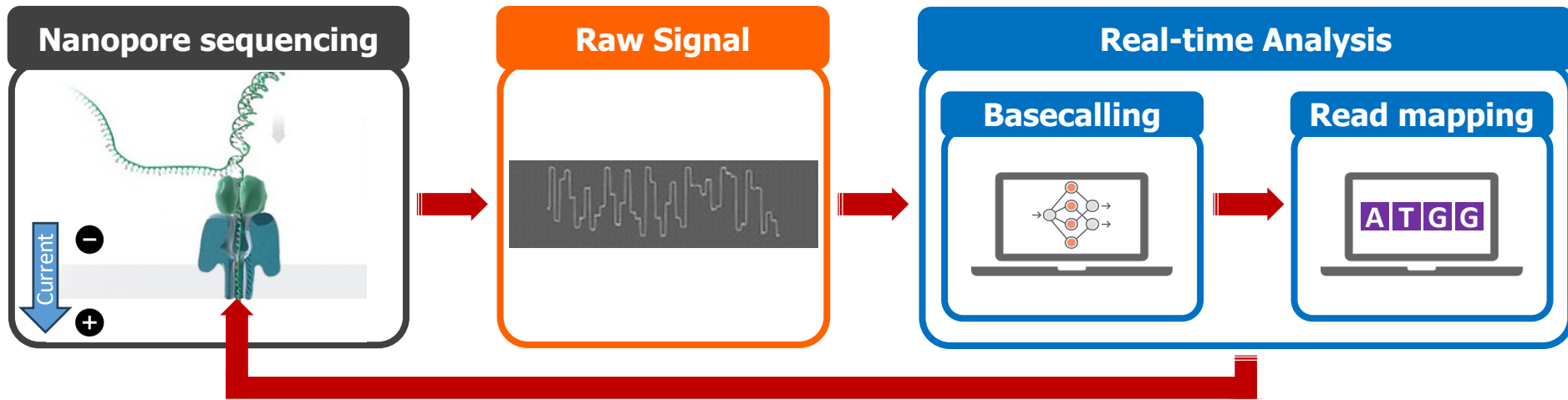
Prune some seeds in the reference genome

Alignment

Determine the exact differences between the read and the reference genome

# Existing Solutions – Real-time Basecalling

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

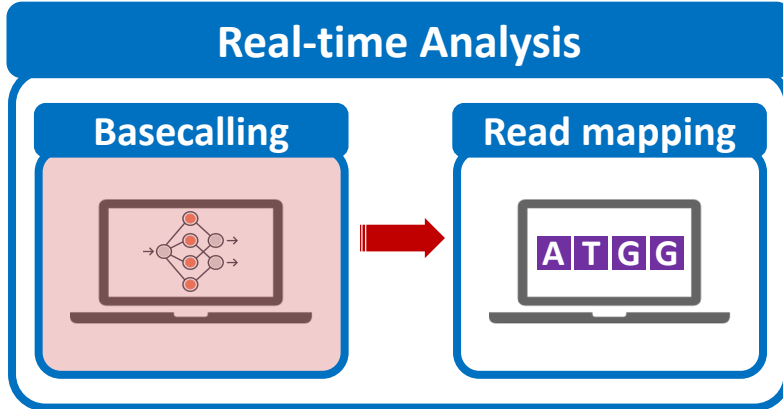


DNNs provide **less noisy analysis** from basecalled sequences

**Costly and power-hungry** computational requirements

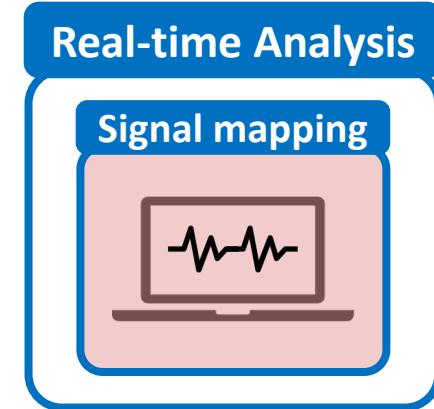
# The Problem

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



**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

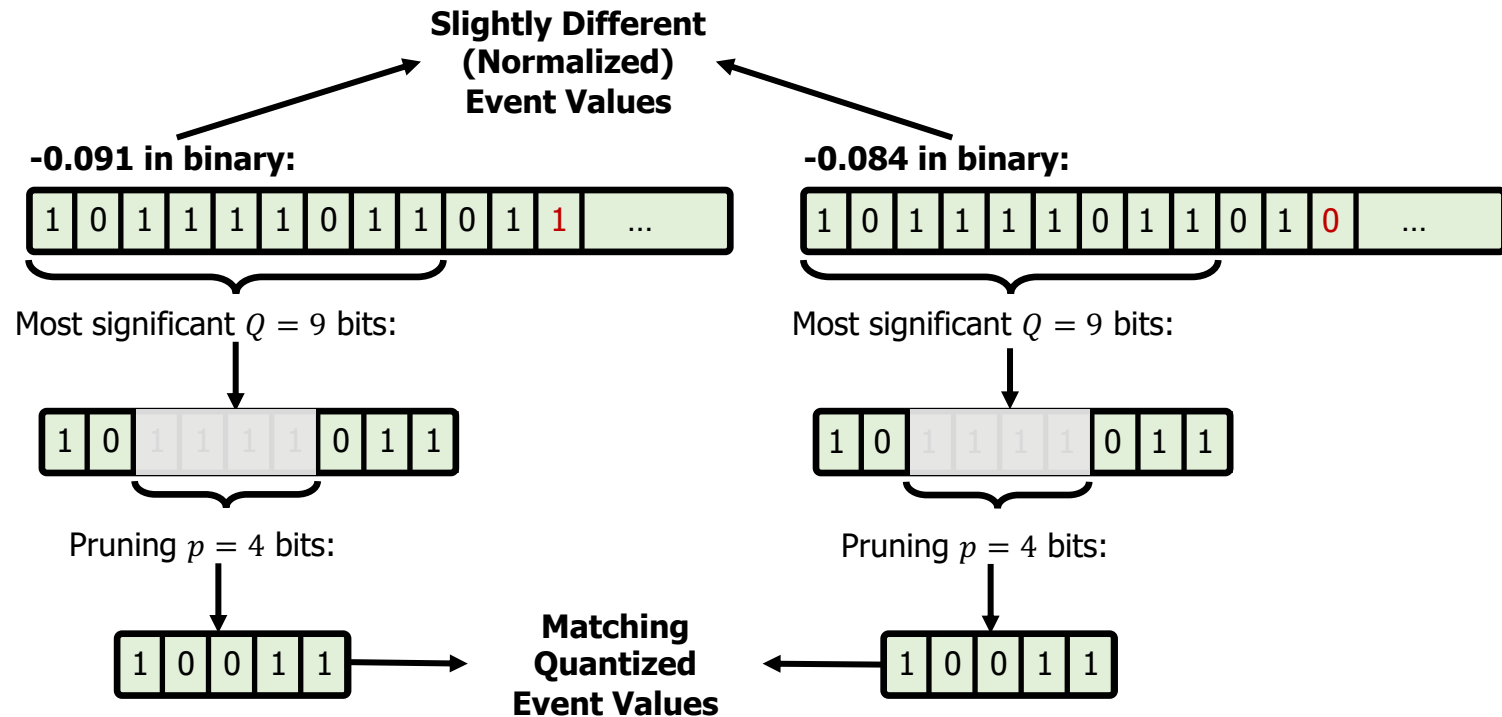
- Stopping when reads reach to a particular depth of coverage
- Stopping when the abundance of all genomes reach a particular threshold

**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**



# Average Sequenced Bases and Chunks

Tool	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>
Average sequenced base length per read					
UNCALLED	<b>184.51</b>	<b>580.52</b>	<b>1,233.20</b>	5,300.15	6,060.23
RawHash	513.95	1,376.14	2,565.09	<b>4,760.59</b>	<b>4,773.58</b>
Average sequenced number of chunks per read					
Sigmap	<b>1.01</b>	<b>2.11</b>	<b>4.14</b>	<b>5.76</b>	<b>10.40</b>
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 (%)				
	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>
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	<i>Contamination</i>	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>	<i>Relative Abundance</i>
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

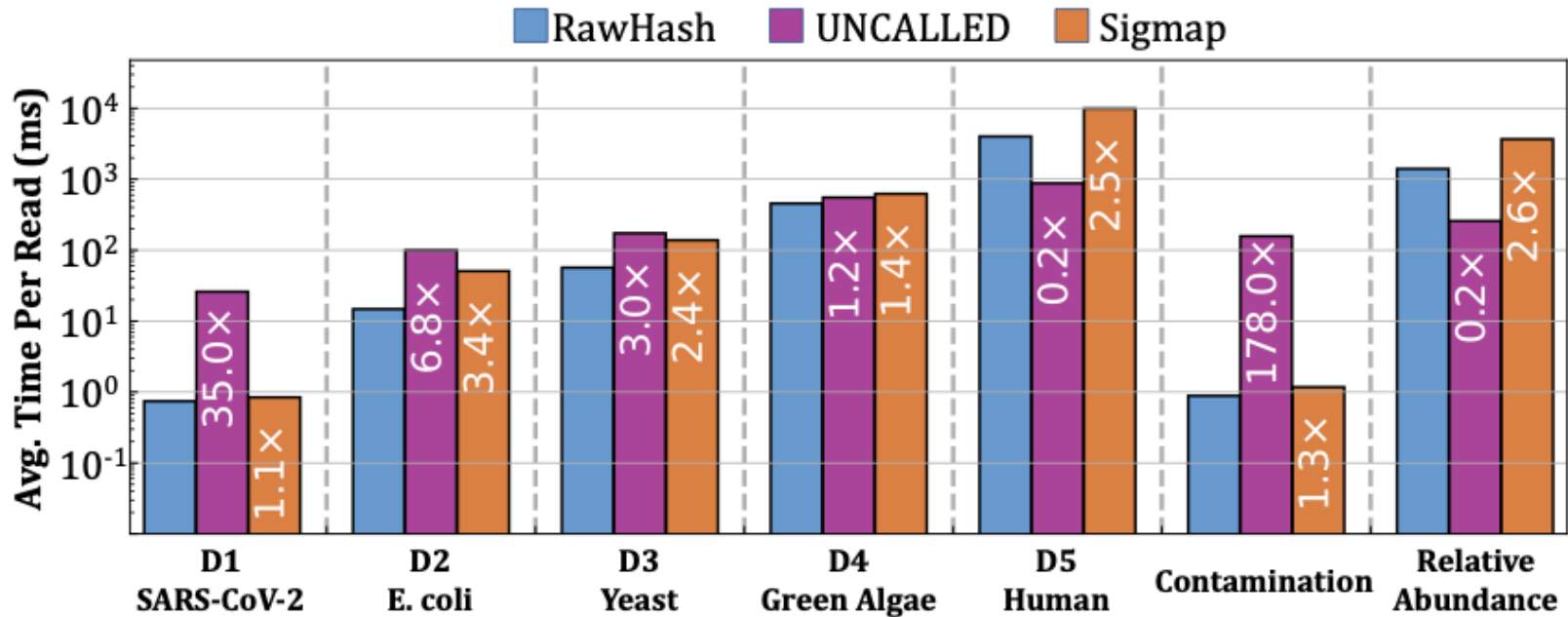
# Required Computation Resources in Mapping

Tool	<i>Contamination</i>	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>	<i>Relative Abundance</i>
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**

# Parameter Configurations

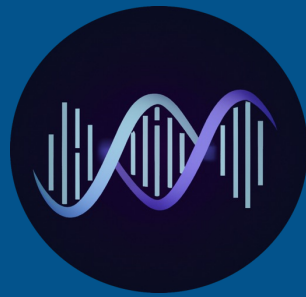
<b>Tool</b>	<b><i>Contamination</i></b>	<b><i>SARS-CoV-2</i></b>	<b><i>E. coli</i></b>	<b><i>Yeast</i></b>	<b><i>Green Algae</i></b>	<b><i>Human</i></b>	<b><i>Relative Abundance</i></b>
RawHash	-x viral -t 32	-x viral -t 32	-x sensitive -t 32	-x sensitive -t 32	-x fast -t 32	-x fast -t 32	-x fast -t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

<b>Preset (-x)</b>	<b>Corresponding parameters</b>	<b>Usage</b>
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

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



# RawHash

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

**Can Firtina**

Nika Mansouri Ghiasi

Joel Lindegger

Gagandeep Singh

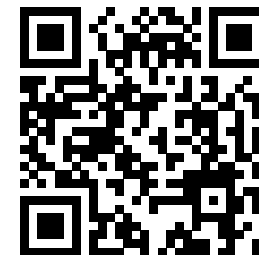
Meryem Banu Cavlak

Haiyu Mao

Onur Mutlu



[Paper](#)



[Code](#)