To appear in the proceedings of 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

Preprint

Source Code

SAFARI

ETH zürich

Executive Summary

Problem

Performing real-time genome analysis is inaccurate and inefficient for large genomes, causing serious barriers in fully exploiting the opportunities in real-time genome analysis

Goal

Enable efficient and accurate analysis for large genomes while the raw sequencing data is generated in real-time

RawHash

- Encodes the raw sequencing data into hash values to accurately and efficiently **identify similarities by matching their hash values**
- Makes **real-time decisions** that can stop sequencing a DNA molecule without fully sequencing it
- Proposes **Sequence Until** that can accurately and dynamically **stop the entire sequencing** of all DNA molecules at once

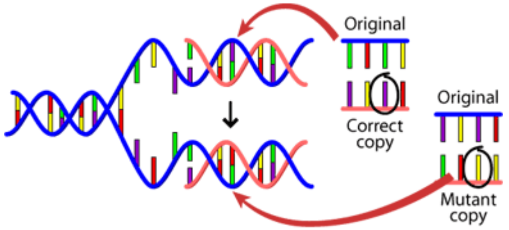
Key Results

- Up to **2x more accurate** mapping results compared to the state-of-the-art works
- **25.8x and 3.4x better average throughput** compared to UNCALLED and Sigmap, respectively
- The Sequence Until techniques enables **reducing the sequencing time and cost by 15x**

Genome Analysis

Genome Sequencing: Enables us to determine the order of the DNA sequence in an organism's genome

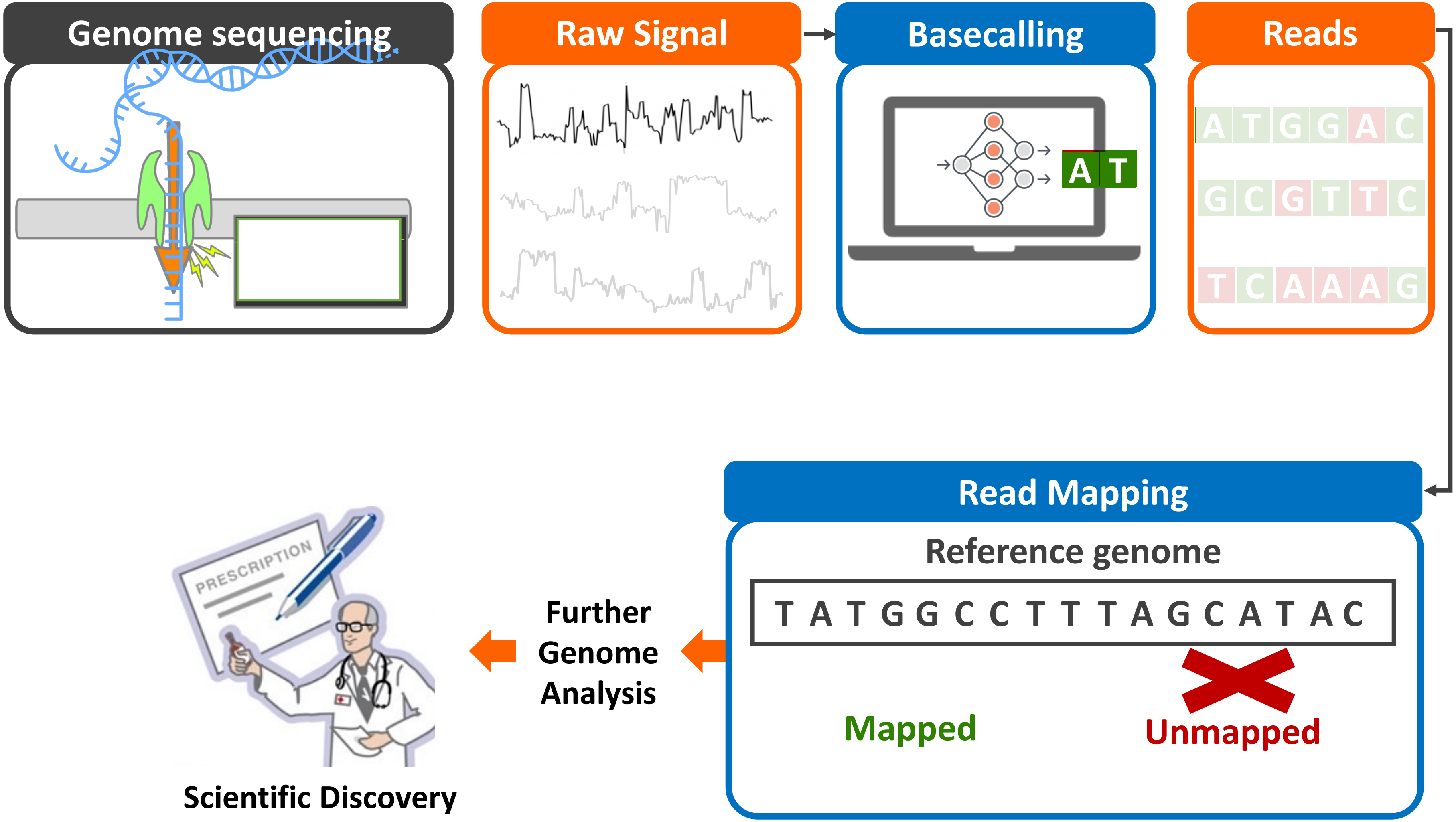
- Plays a pivotal role in:
  - Precision medicine
  - Outbreak tracing
  - Understanding of evolution



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

- Can sequence large fragments of DNA (i.e., 10Kbp - 2Mbp)
- Has high throughput
- Low cost
- Provides unique features

Traditional Genome Analysis Pipeline



Genome sequencing

Raw Signal

Basecalling

Reads

Read Mapping

Reference genome

T A T G G C C T T T A G C A T A C

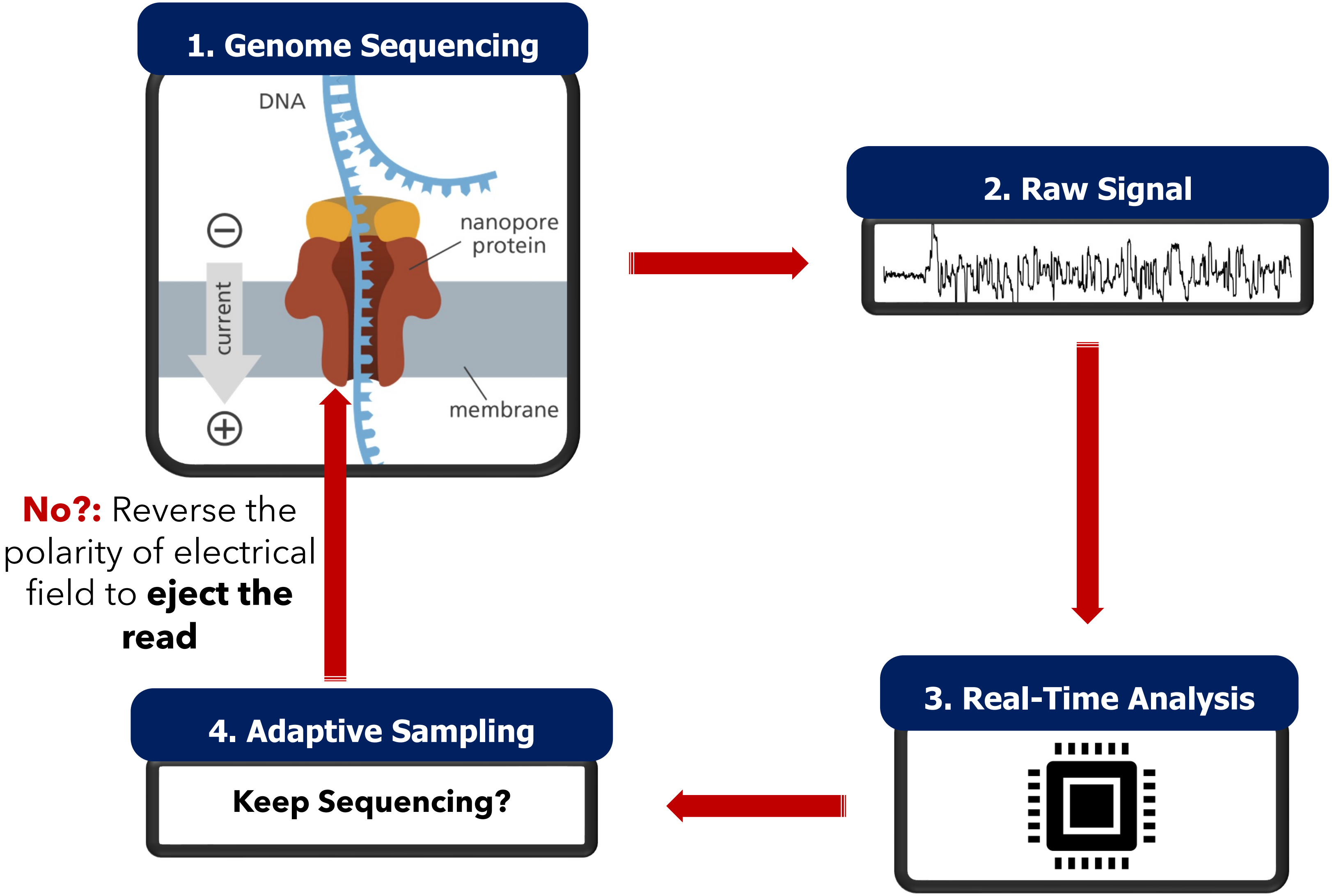
Mapped

Unmapped

Further Genome Analysis

Scientific Discovery

Real-Time Genome Analysis



1. Genome Sequencing

2. Raw Signal

3. Real-Time Analysis

4. Adaptive Sampling

Keep Sequencing?

No?: Reverse the polarity of electrical field to **eject the read**

Objectives in Real-Time Genome Analysis

Fast analysis that can match the throughput of sequencer

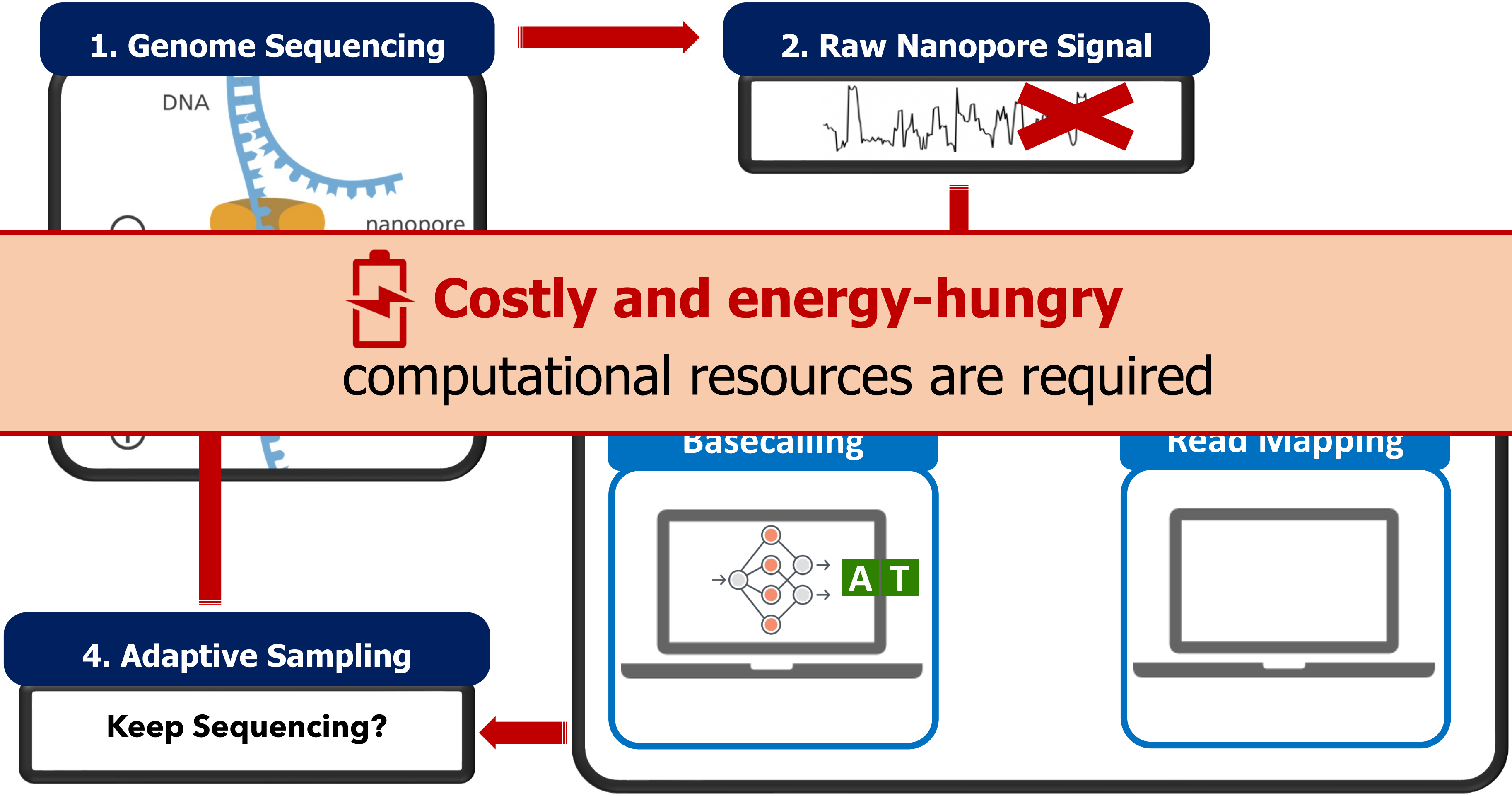
Fast decision to reduce the sequencing time and cost with effective use of adaptive sampling

Accurate analysis from noisy raw signal data

Low-power to enable portable sequencing and better scalability

Solutions for Real-Time Analysis

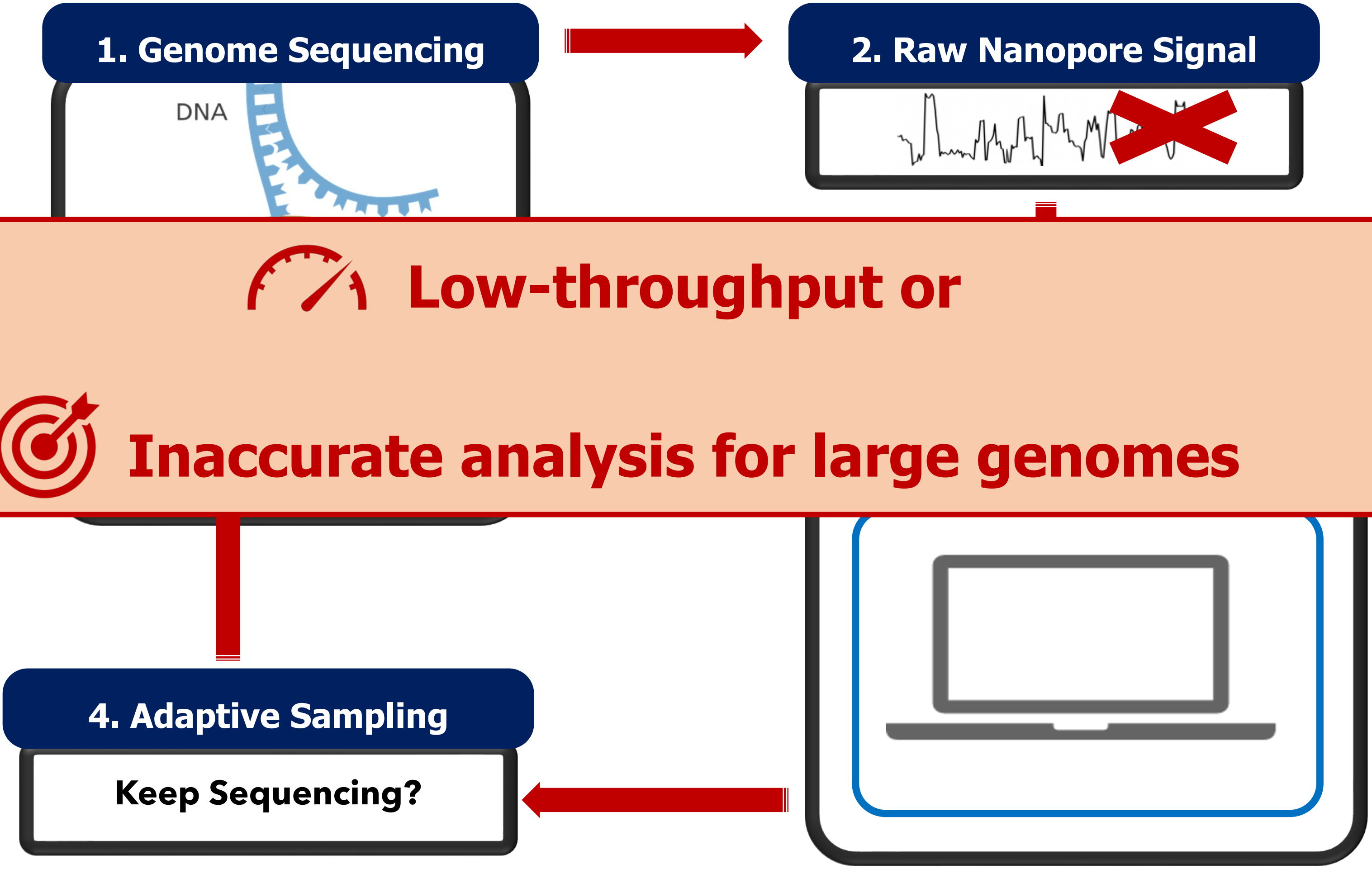
1. Using deep neural networks (DNNs) to basecall and map reads



Costly and energy-hungry computational resources are required

Solutions for Real-Time Analysis

2. Mapping signals without basecalling



Low-throughput or

Inaccurate analysis for large genomes

Outline

Background

Goal and Key Ideas

RawHash

Evaluation

Conclusions

Goal

Fast analysis that can scale to large genomes

Fast decisions for adaptive sampling to reduce sequencing time and cost

Accurate analysis for large genomes

Low-power analysis that can be used with portable devices

RawHash

The first mechanism that can **efficiently and accurately map** raw signals to **large genomes** using an efficient **hash-based search**

Proposes **Sequence Until**, a novel mechanism that can **decide in runtime** if further sequencing of reads is needed to **stop the *entire* sequencing process**

Outline

Background

Goal and Key Ideas

RawHash

Evaluation

Conclusions

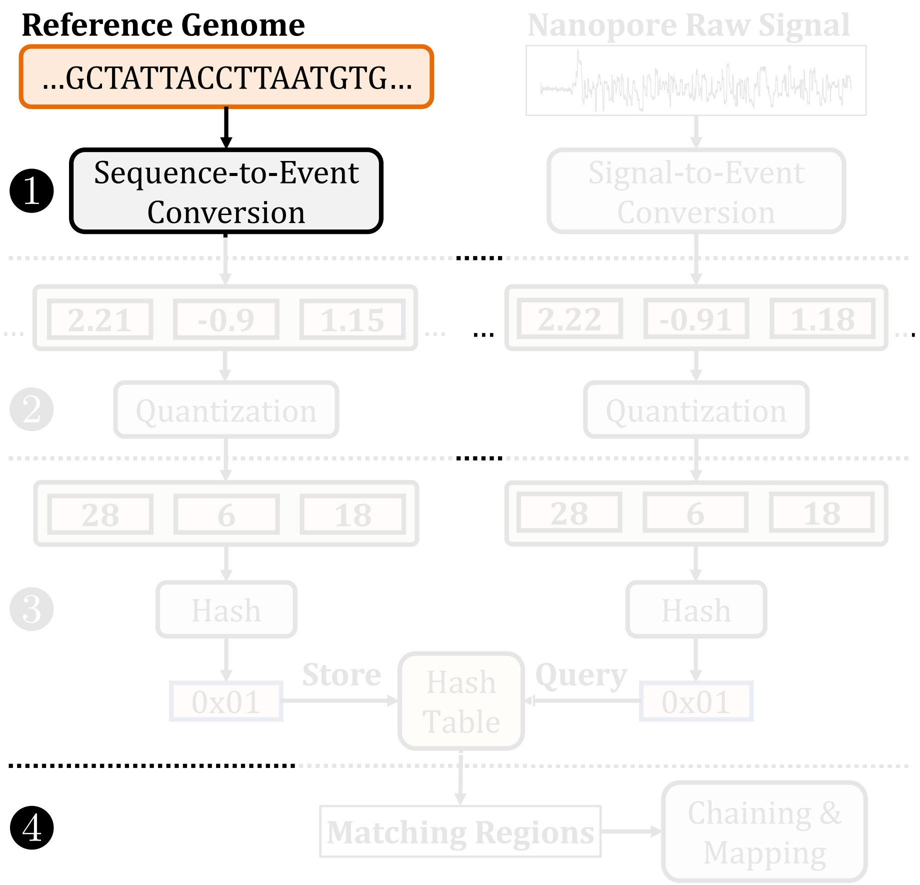
RawHash Overview

1. Indexing (offline):

1. Convert the reference genome to its signal representation
2. Generate hash values from signals
3. Store hash values and their positions in a hash table

2. Mapping (real-time):

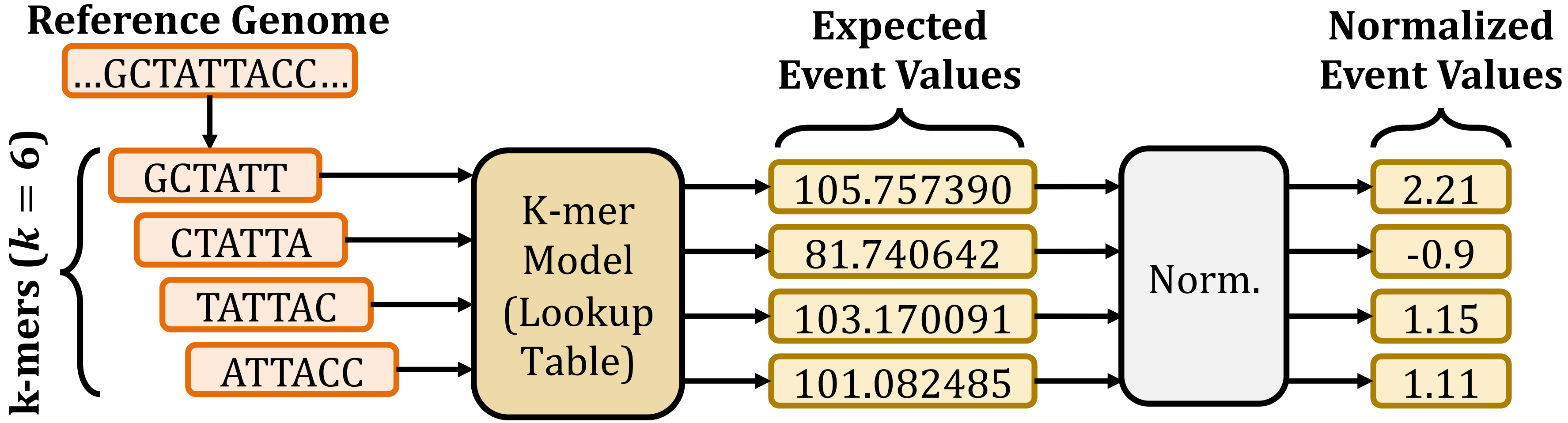
1. Generate hash values from raw nanopore signals
2. Use the hash table to find matching hash values between a reference genome and the nanopore raw signal
3. Mapping regions: Regions with a certain number of hash value matches



Converting the Reference Sequences to Signals

- To offload the translation costs to the offline indexing step
- To enable utilizing the rich information in raw nanopore signals

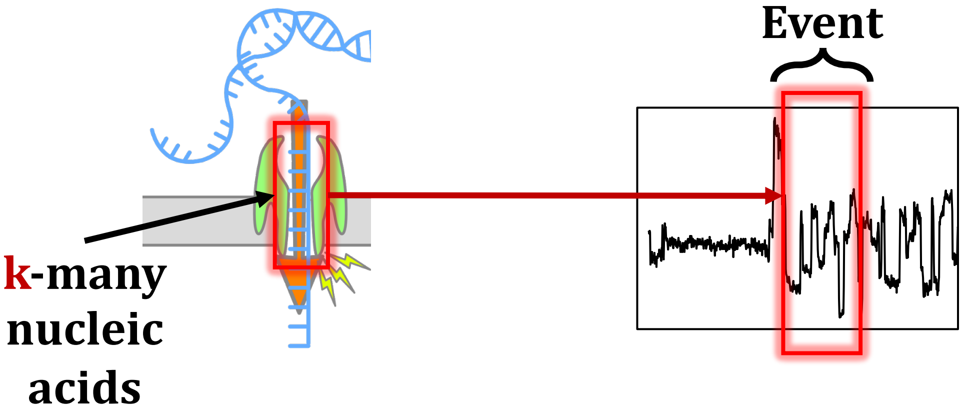
- Key Steps:
  - **K-mer model: Expected** current readings after sequencing a **fixed** **k** number of nucleic acids (**k**-mers)
  - Utilize the lookup table to convert all **k**-mers of a reference genome to their expected values (**events**)



Events in Raw Nanopore Signals

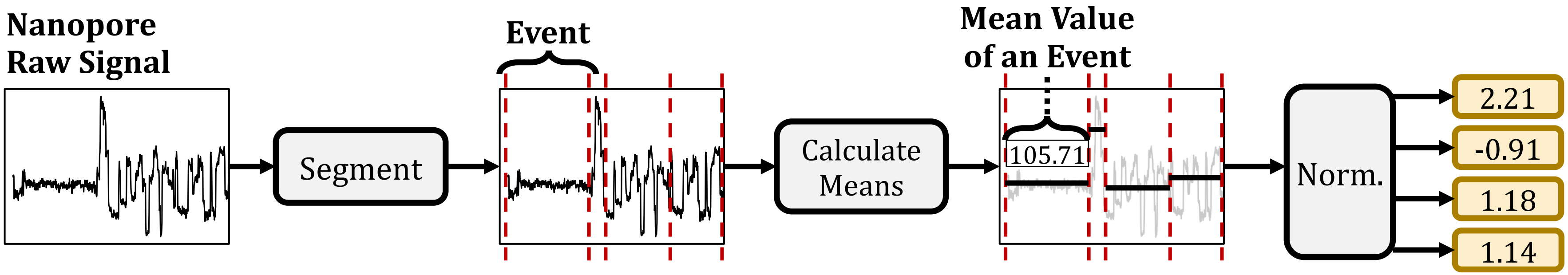
- **Event:** Series of current readings
  - Generated when sequencing a particular k-mer
  - **Next event:** DNA molecule is shifted by one nucleic acid, creating the next k-mer

- **Event detection** identifies regions of signals corresponding to the sequencing of certain **k**-mers in the DNA molecule
  - **Next event:** Abrupt signal changes between two consecutive k-mers



Event Detection in Raw Nanopore Signals

- **Event detection** identifies regions of signals corresponding to the sequencing of certain k-mers in the DNA molecule
  - By performing a statistical test **(segmentation)** to identify the **abrupt changes** in the signal generated as molecules move through nanopores



- Observation: Nanopore sequencers **do not** generate **exactly the same signals** when sequencing the **same k-mer**
  - However, the signals are still **slightly similar** to each other
  - How can we leverage this?

Quantizing the Event Values

- **Goal:** Assign the same bucket (i.e., quantized values) to the similar event values

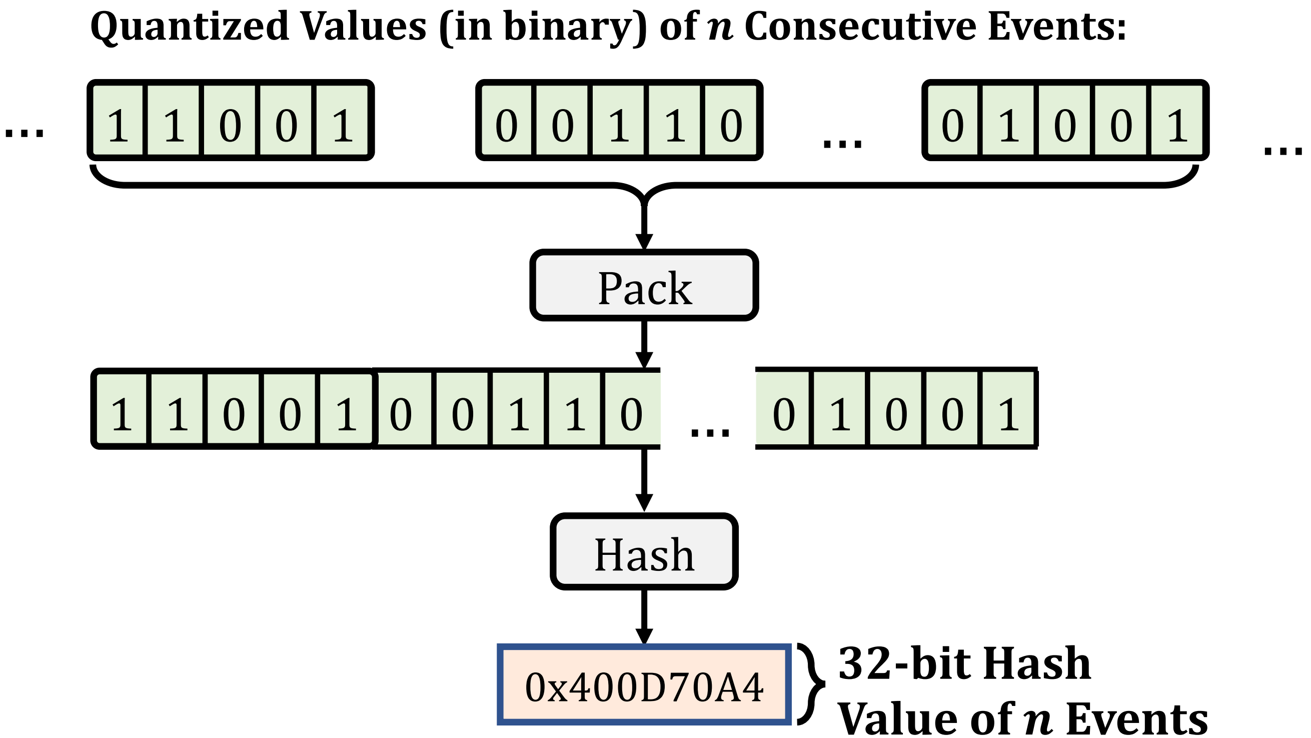
- **Key Steps:**

1. Use the binary representations of event values (floating-point)
2. Take the most significant Q bits (to quantize)
3. Ignore the p bits in the middle (does not add much value)

Hashing for Efficient Search

- **Goal:** Enable finding efficient similarity detection by accurately matching hash values between signals

1. Pack the quantized values of *some* consecutive k-mers
2. Hash the packed value to generate a hash value
3. Use efficient data structures (e.g., hash tables) to identify regions with the similar event values by matching their hash values



Quantized Values (in binary) of n Consecutive Events:

… 1 1 0 0 1 0 0 1 1 0 … 0 1 0 0 1 …

1 1 0 0 1 0 0 1 1 0 … 0 1 0 0 1

0x400D70A4

32-bit Hash Value of n Events

Outline

Background

Goal and Key Ideas

RawHash

Evaluation

Conclusions

Evaluation Methodology

- Datasets from very small (viral) to large genomes (human and metagenomics)

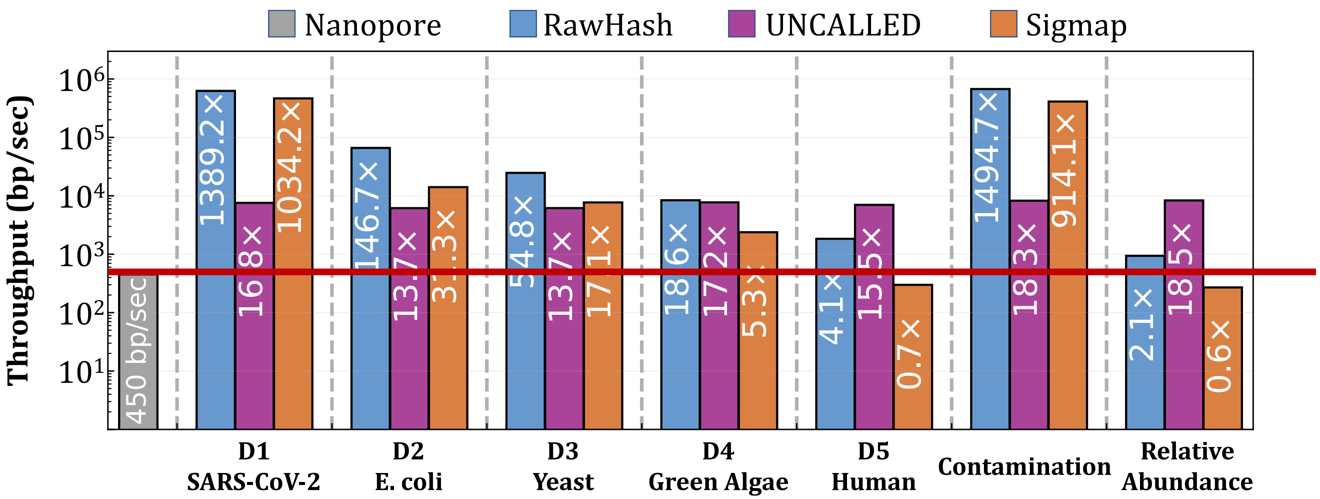
- Compared with UNCALLED and Sigmap
  - RawHash, UNCALLED, and Sigmap do not require powerful computational resources (e.g., GPUs) to achieve efficient and portable genome analysis

- Use cases
  1. Read mapping
  2. Relative abundance estimation
  3. Contamination analysis

- Benefits of Sequence Until

Performance

- Throughput (bases per second)
  - Throughput of a nanopore sequencer: 450 bp/sec

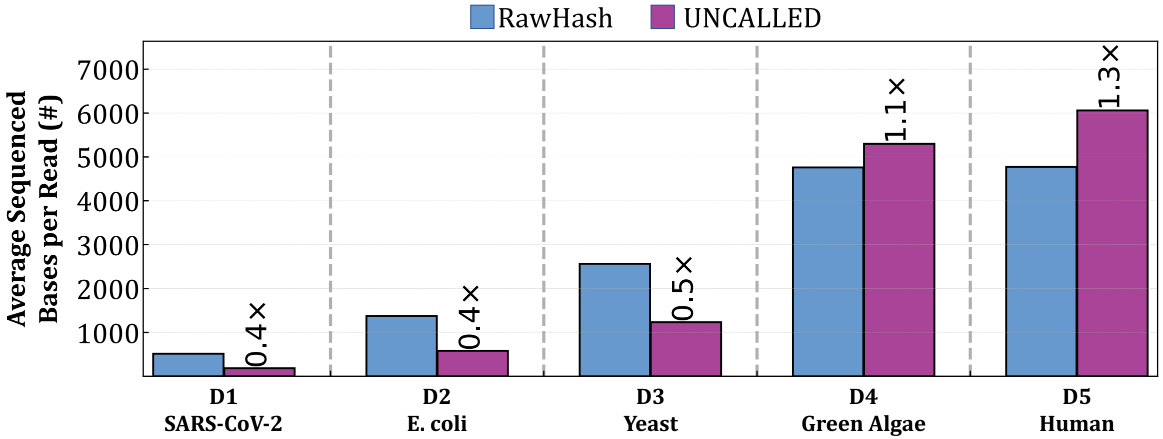


Fast Analysis: Both RawHash and UNCALLED can match the throughput of nanopore

Sigmap falls behind the throughput of nanopores for larger genomes

Sequencing Time and Cost

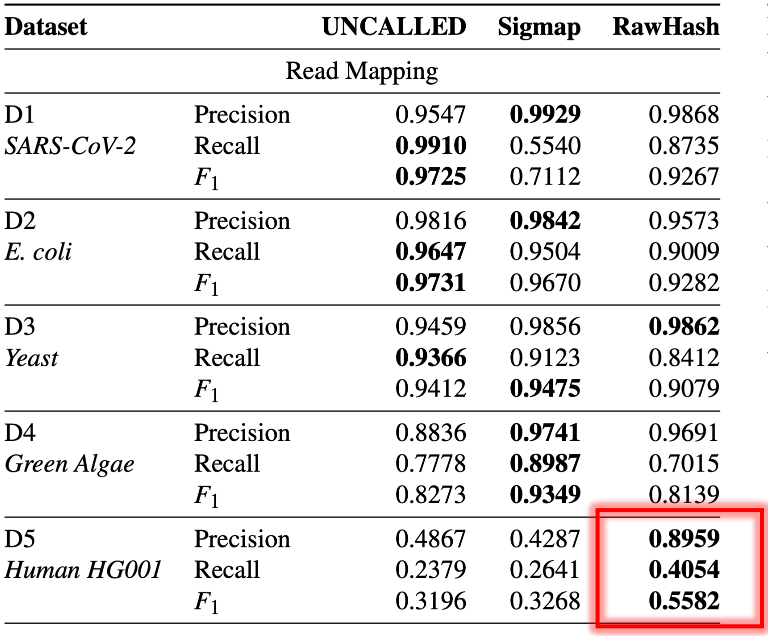
- Number of bases that needs to be sequenced before making a decision to eject the read
  - Lower is better (cheaper and faster sequencing)



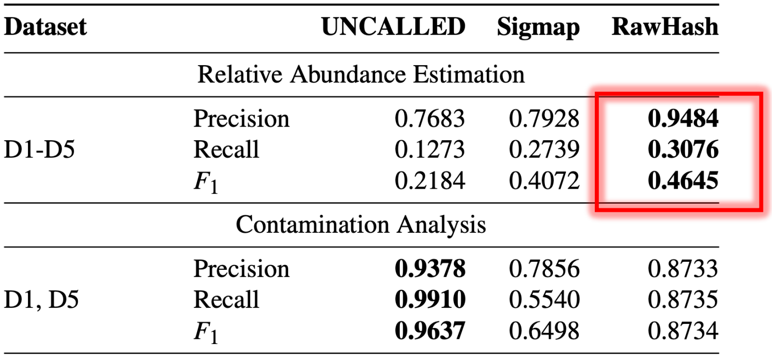
Fast Decision: RawHash reduces the sequencing time and cost for large genomes than UNCALLED

Accuracy of Mapping

- Accuracy of genome analysis when mapping reads for three use cases



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**

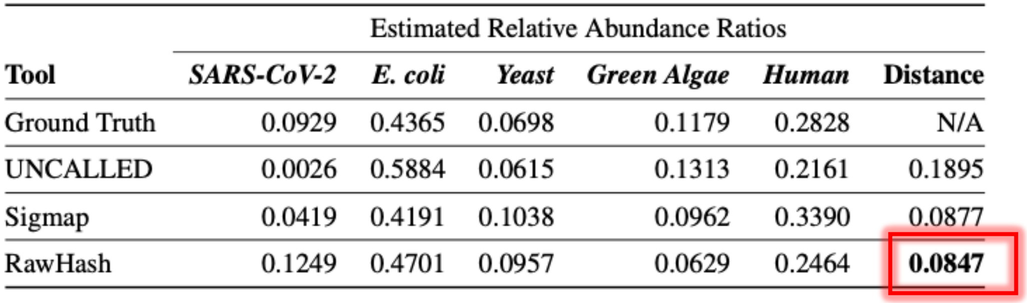


Dataset		UNCALLED	Sigmap	RawHash
		Relative Abundance Estimation		
D1-D5	Precision	0.7683	0.7928	**0.9484**
	Recall	0.1273	0.2739	**0.3076**
	F_1	0.2184	0.4072	**0.4645**
		Contamination Analysis		
D1, D5	Precision	**0.9378**	0.7856	0.8733
	Recall	**0.9910**	0.5540	0.8735
	F_1	**0.9637**	0.6498	0.8734

Accurate Analysis: RawHash provides the best accuracy for large genomes

Relative Abundance Estimations

- Estimating the relative abundance of each genome compared to the baseline as generated by minimap2
  - Distance: Euclidean distance (L2-norm) compared to the ground truth distance



	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**

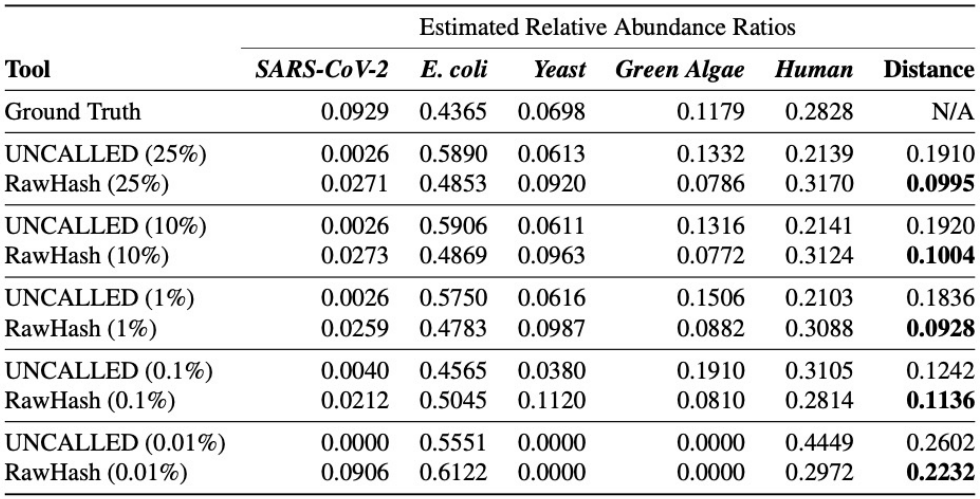
Accurate Analysis: RawHash provides the relative abundance estimations closest to the ground truth

The Sequence Until Mechanism

- **Key Insight:** Do we need to keep sequencing **the entire sample** for all applications in genome analysis?

- **Use case example:** Can we predict the relative abundance estimation by sequencing only a portion of the sample and still provide accurate results?

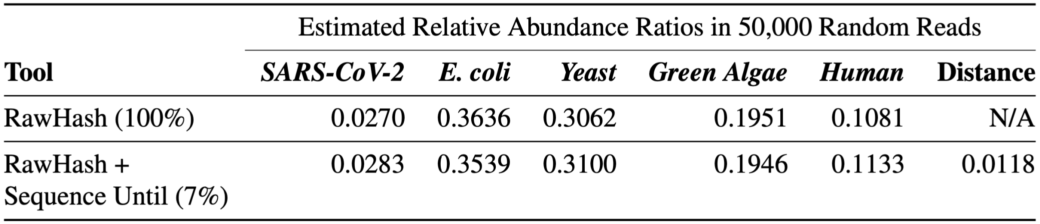
- **Potential Benefits:** Reduced sequencing time and costs by avoiding full sequencing



	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**

Benefits of Sequence Until

- Sequence Until mechanism **dynamically** analyzes the results of a genome analysis use case **to find outliers** in the analysis
- **If no outlier** in the previous estimations
  - Further sequencing is unlikely to change the analysis significantly
  - Stop the **entire sequencing**: Significant reduction in sequencing time and cost



	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 dynamically stops the entire sequencing after sequencing **only 7% of the entire sample while providing high accuracy**

Sequencing only a portion of the sample significantly **reduces sequencing time and cost (~15x reduction)**

Outline

Background

Goal and Key Ideas

RawHash

Evaluation

Conclusions

RawHash Summary

Problem

Performing real-time genome analysis is inaccurate and inefficient for large genomes, causing serious barriers in fully exploiting the opportunities in real-time genome analysis

Goal

Enable efficient and accurate similarity identification between raw signals

RawHash

- Encodes the similar signal values into the same quantized value to alleviate the noise issues in raw signals
- Generates hash values from quantized values to efficiently identify similarities between signals based on hash value matches
- Proposes Sequence Until that can accurately and dynamically stop the entire sequencing

Key Results

- Up to **2x more accurate** mapping results
- **25.8x and 3.4x better average throughput** compared to UNCALLED and Sigmap, respectively
- The Sequence Until techniques enables **reducing the**

RawHash

To appear in the proceedings of ISMB/ECCB 2023

- 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, **00**, i1–i11
https://doi.org/10.1093/bioinformatics/btad272
ISMB/ECCB 2023

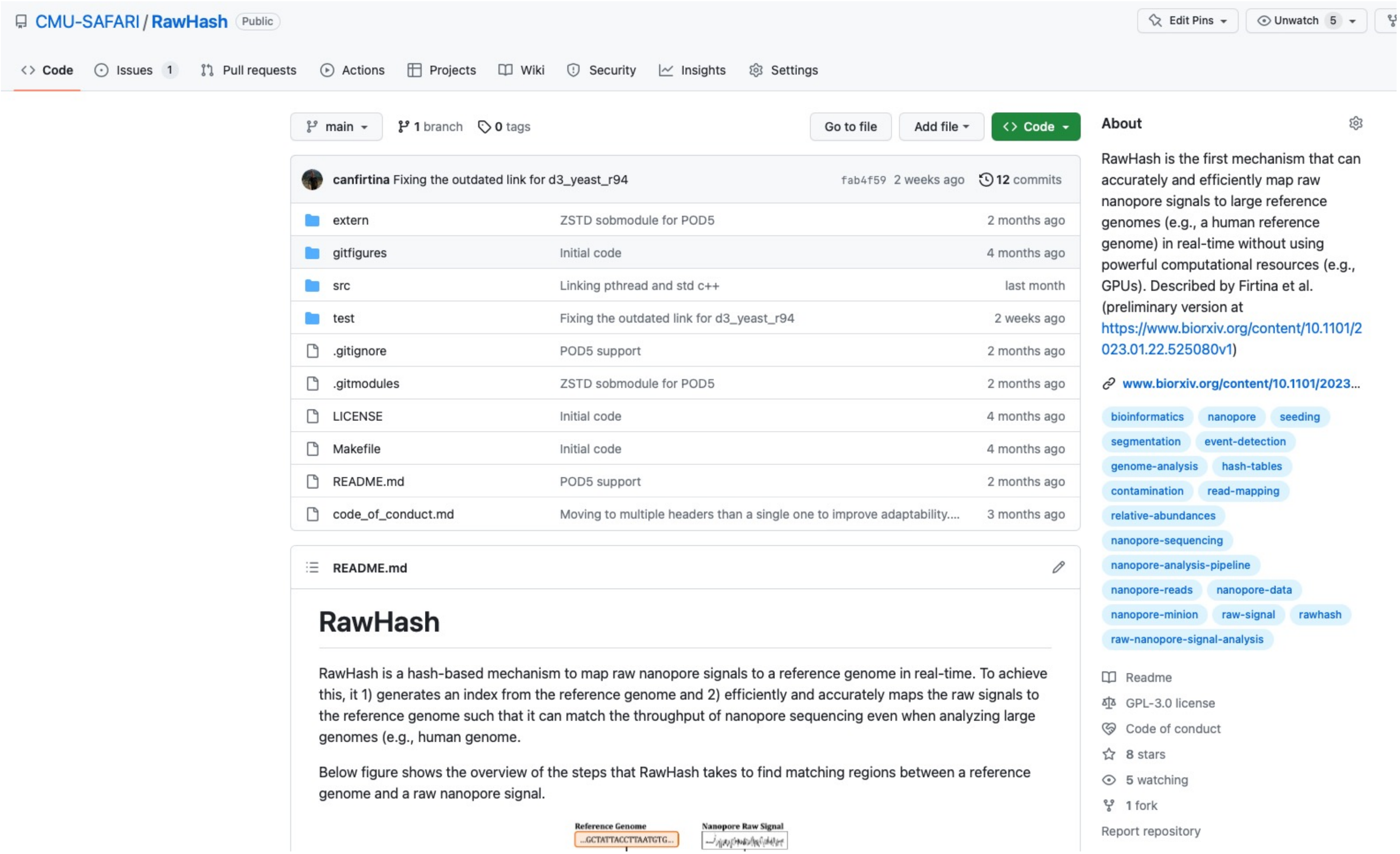
OXFORD

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



canfirtina Fixing the outdated link for d3_yeast_r94 fab4f59 2 weeks ago 12 commits

extern	ZSTD sobmodule for POD5	2 months ago
gitfigures	Initial code	4 months ago
src	Linking pthread and std c++	last month
test	Fixing the outdated link for d3_yeast_r94	2 weeks ago
.gitignore	POD5 support	2 months ago
.gitmodules	ZSTD sobmodule for POD5	2 months ago
LICENSE	Initial code	4 months ago
Makefile	Initial code	4 months ago
README.md	POD5 support	2 months ago
code_of_conduct.md	Moving to multiple headers than a single one to improve adaptability....	3 months ago

README.md

RawHash

RawHash is a hash-based mechanism to map raw nanopore signals to a reference genome in real-time. To achieve this, it 1) generates an index from the reference genome and 2) efficiently and accurately maps the raw signals to the reference genome such that it can match the throughput of nanopore sequencing even when analyzing large genomes (e.g., human genome.

Below figure shows the overview of the steps that RawHash takes to find matching regions between a reference genome and a raw nanopore signal.

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. (preliminary version at https://www.biorxiv.org/content/10.1101/2023.01.22.525080v1)

www.biorxiv.org/content/10.1101/2023...

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
8 stars
5 watching
1 fork
Report repository

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

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

Preprint

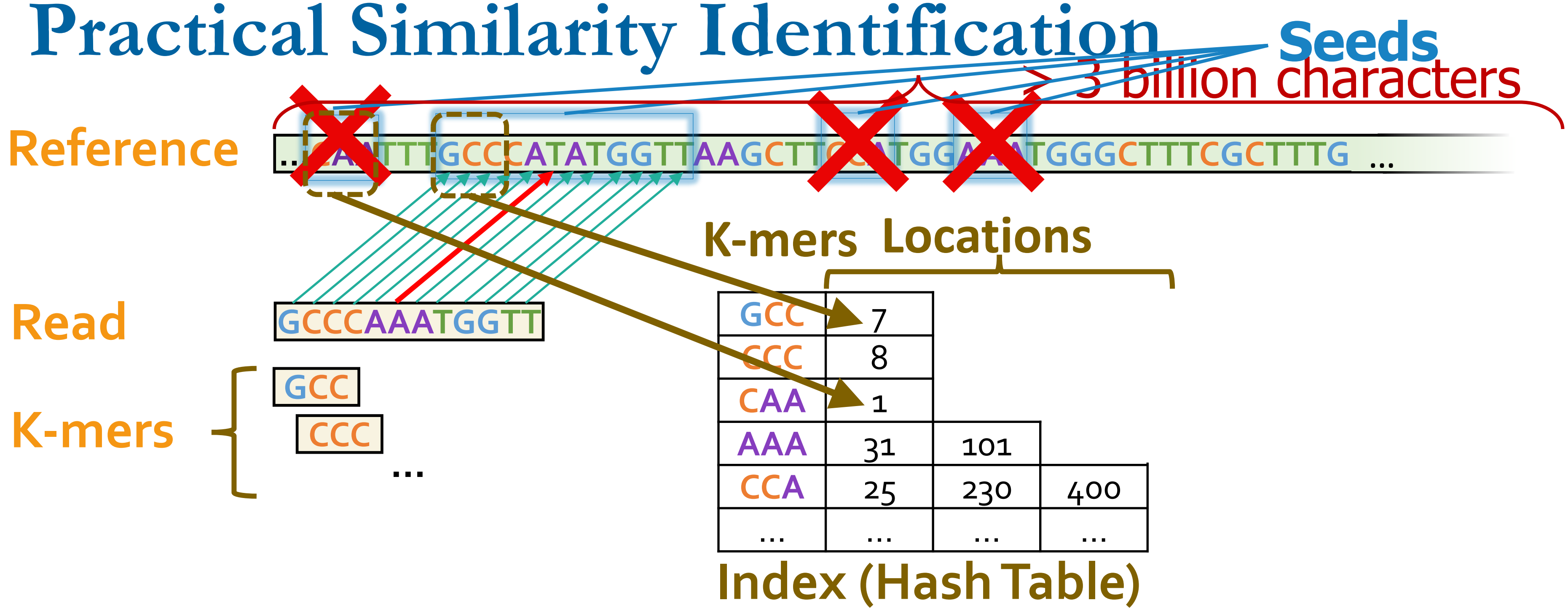
Source Code

SAFARI

ETH zürich

Backup Slides

Practical Similarity Identification

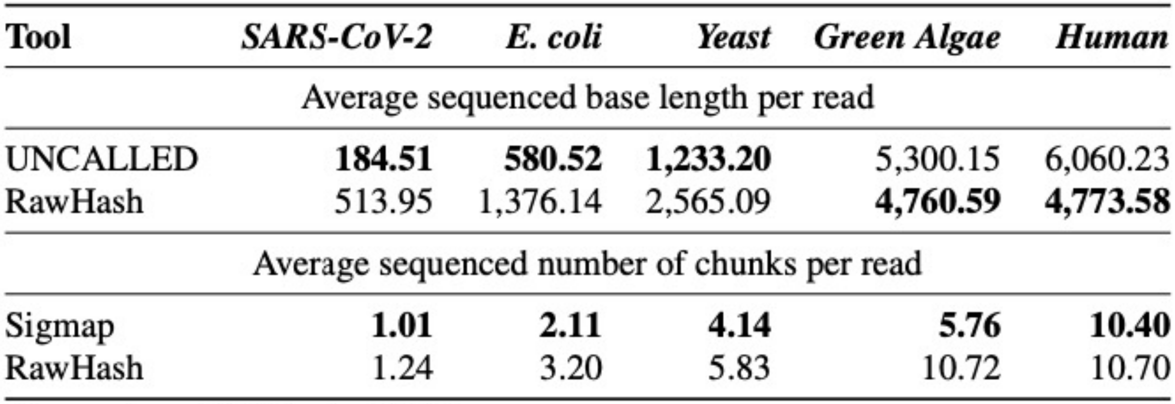


K-mers	Locations		
GCC	7		
CCC	8		
CAA	1		
AAA	31	101	
CCA	25	230	400
...	...	...	...

Index (Hash Table)

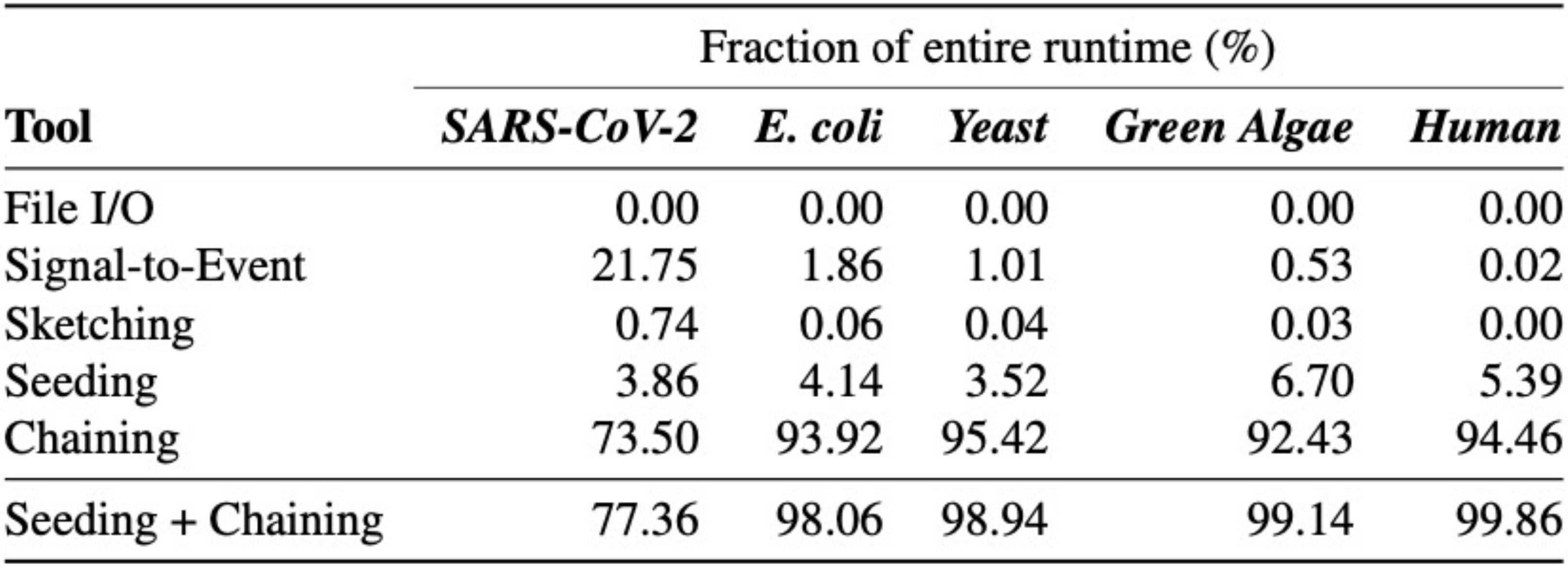
Seeding	Determine potential matching regions (seeds) in the reference genome
Seed Filtering (e.g., Chaining)	Prune some seeds in the reference genome
Alignment	Determine the exact differences between the read and the reference genome

Sequencing Time and Cost Reductions



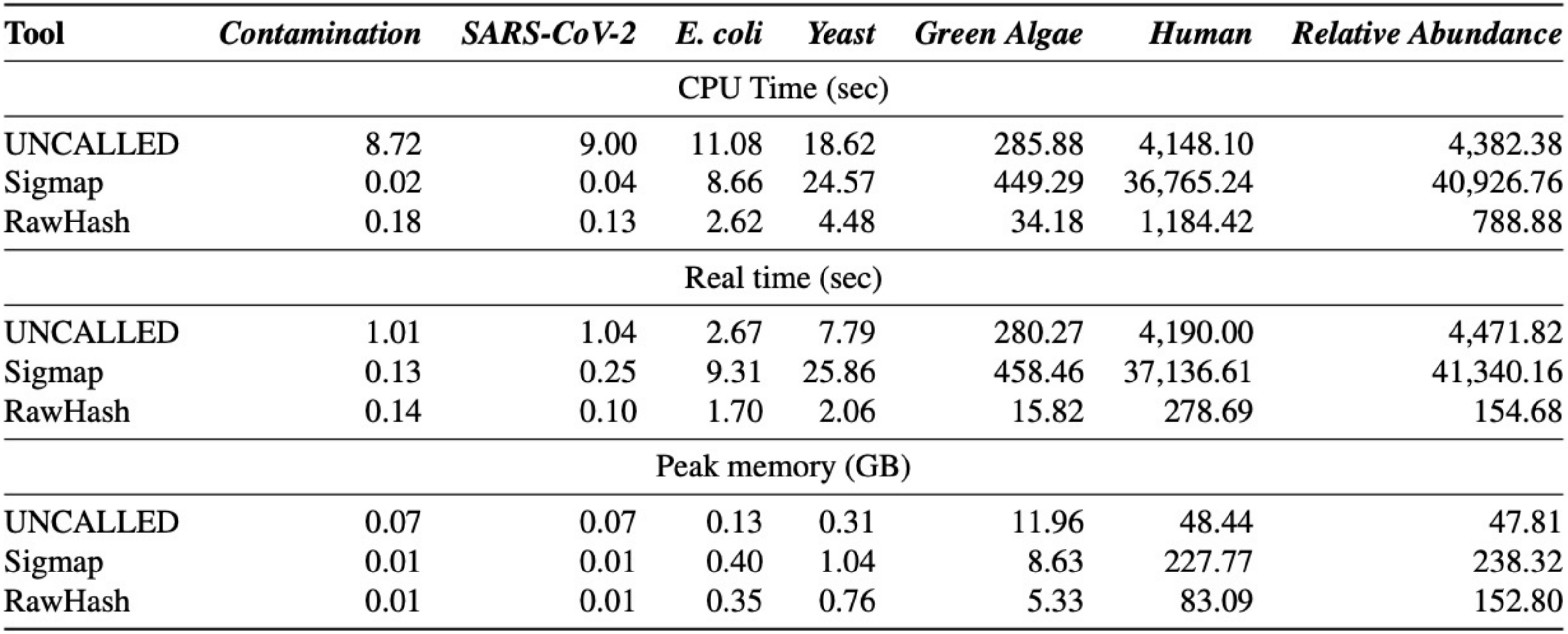
Tool	*SARS-CoV-2*	*E. coli*	*Yeast*	*Green Algae*	*Human*
		Average sequenced base length per read			
UNCALLED	**184.51**	**580.52**	**1,233.20**	5,300.15	6,060.23
RawHash	513.95	1,376.14	2,565.09	**4,760.59**	**4,773.58**
		Average sequenced number of chunks per read			
Sigmap	**1.01**	**2.11**	**4.14**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

Profiling the RawHash Steps



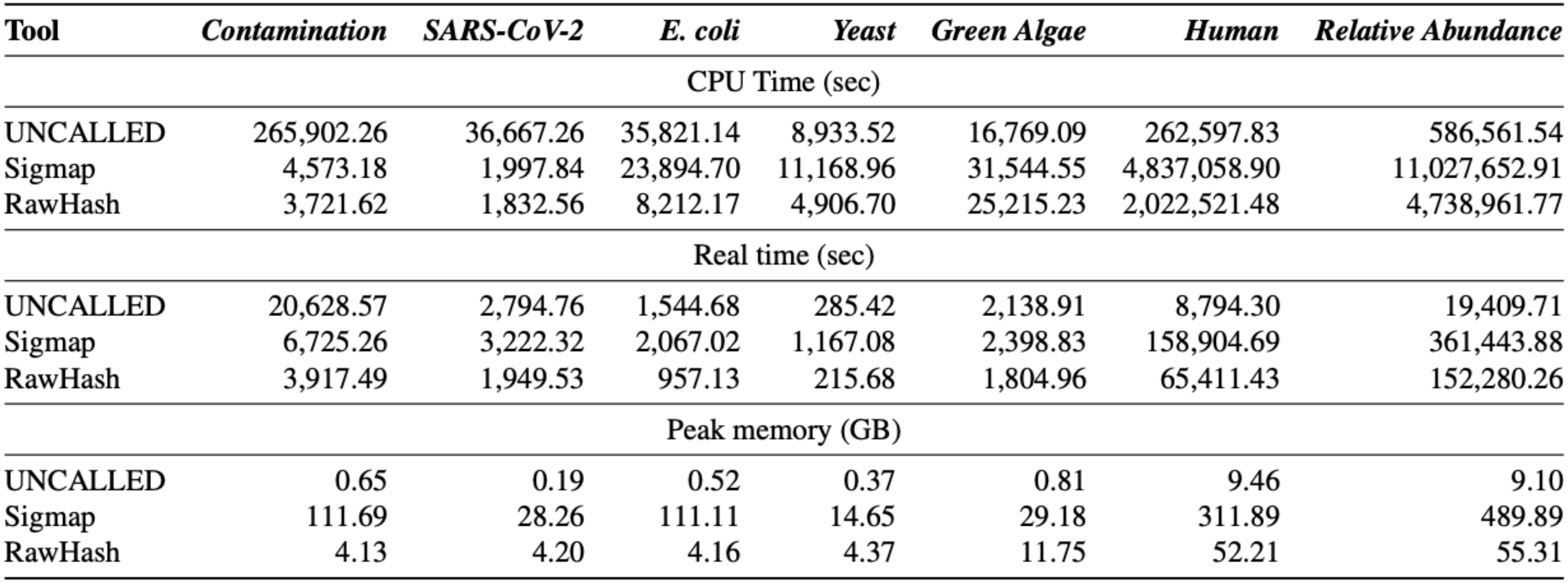
	Fraction of entire runtime (%)				
Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***
File I/O	0.00	0.00	0.00	0.00	0.00
Signal-to-Event	21.75	1.86	1.01	0.53	0.02
Sketching	0.74	0.06	0.04	0.03	0.00
Seeding	3.86	4.14	3.52	6.70	5.39
Chaining	73.50	93.92	95.42	92.43	94.46
Seeding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

Required Computation Resources in Indexing



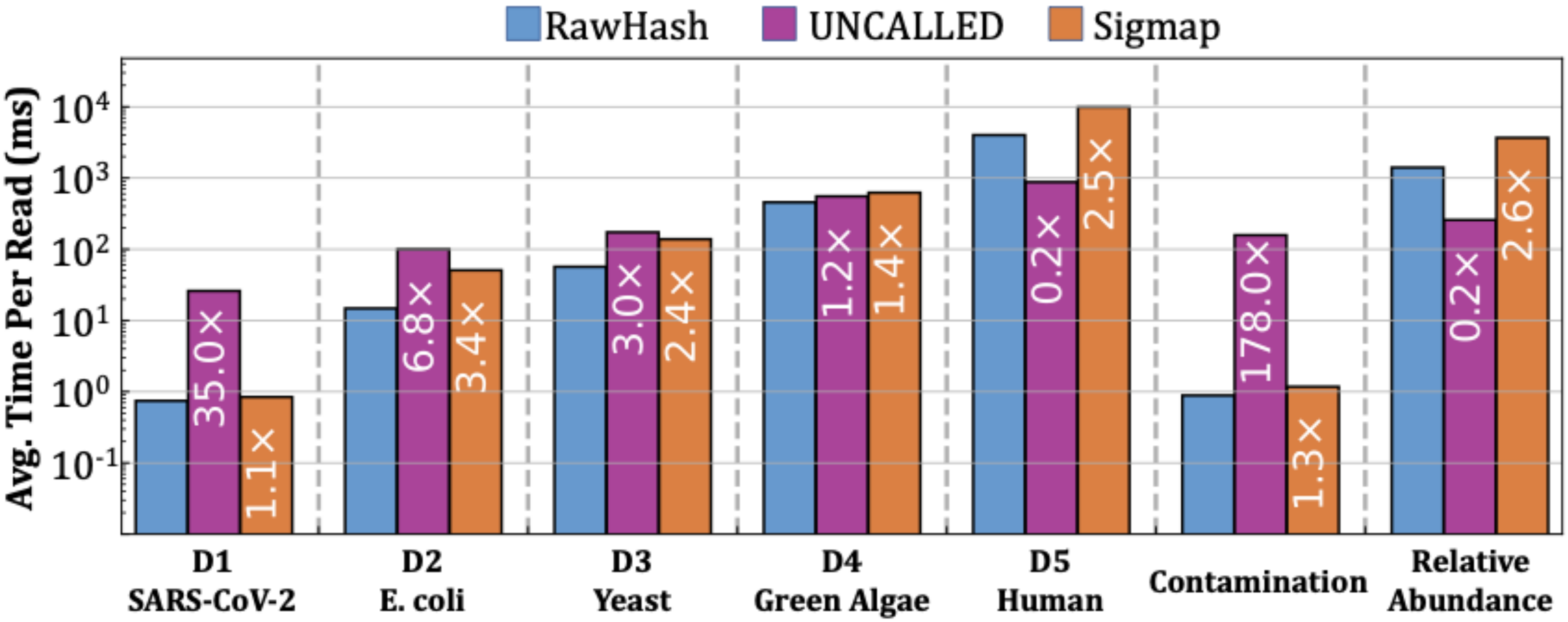
Tool	*Contamination*	*SARS-CoV-2*	*E. coli*	*Yeast*	*Green Algae*	*Human*	*Relative Abundance*
			CPU Time (sec)				
UNCALLED	8.72	9.00	11.08	18.62	285.88	4,148.10	4,382.38
Sigmap	0.02	0.04	8.66	24.57	449.29	36,765.24	40,926.76
RawHash	0.18	0.13	2.62	4.48	34.18	1,184.42	788.88
			Real time (sec)				
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

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

Average Mapping Time per Read


RawHash
UNCALLED
Sigmap
Avg. Time Per Read (ms)
10^4
10^3
10^2
10^1
10^0
10^{-1}
35.0×
1.1×
6.8×
3.4×
3.0×
2.4×
1.2×
1.4×
0.2×
2.5×
178.0×
1.3×
0.2×
2.6×
D1
SARS-CoV-2
D2
E. coli
D3
Yeast
D4
Green Algae
D5
Human
Contamination
Relative
Abundance