RawAlign

Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment

Joël Lindegger

Can Firtina Nika Mansouri Ghiasi

Mohammad Sadrosadati Mohammed Alser Onur Mutlu

SAFARI

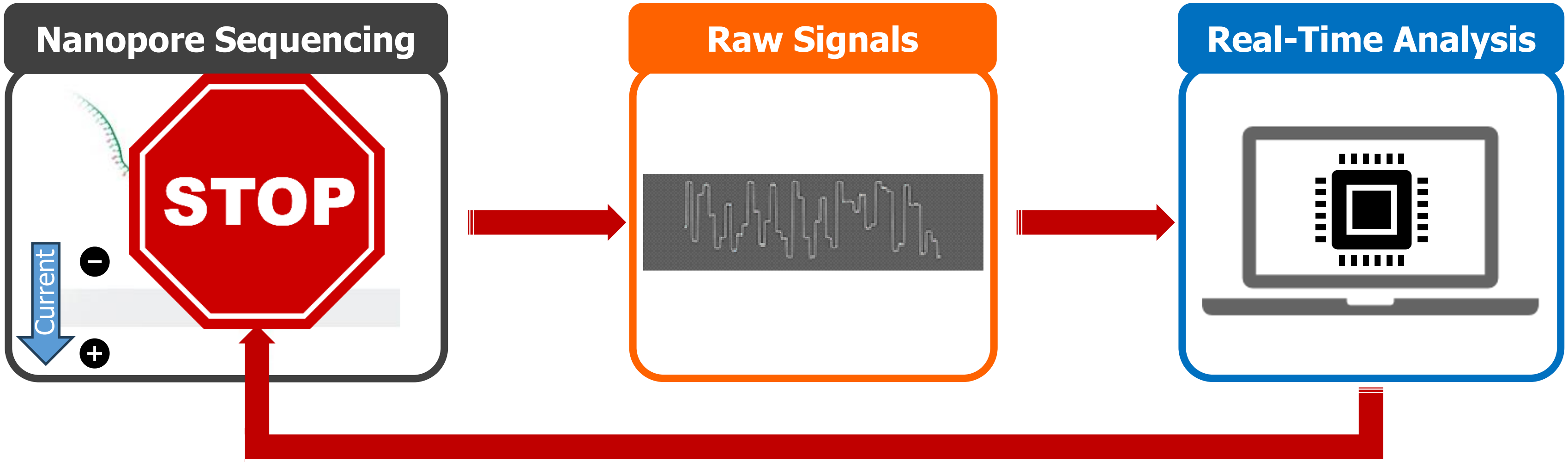
ETH zürich

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



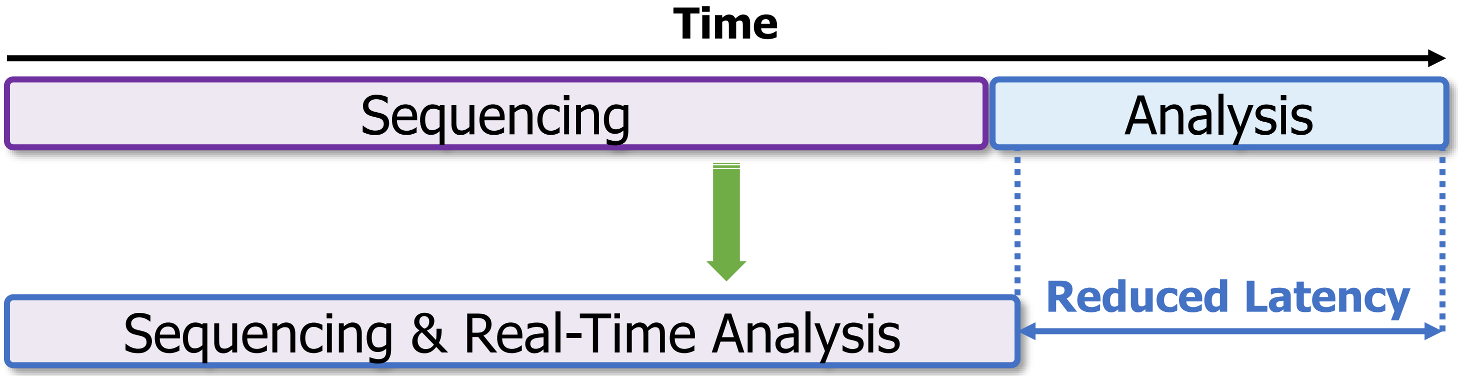
Raw Signals: Ionic current measurements generated at a certain **throughput**

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

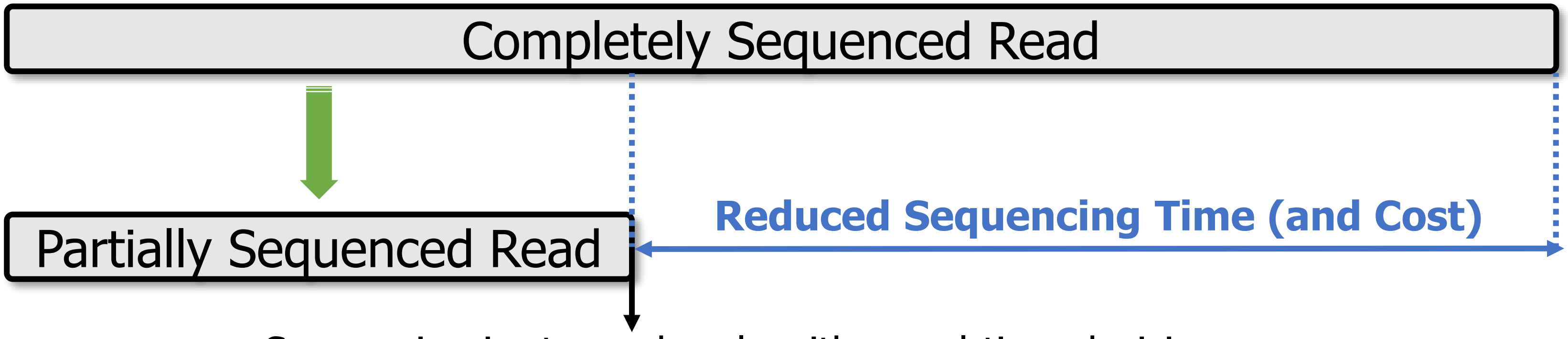
Real-Time Decisions: Stopping sequencing **early** based on real-time analysis

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

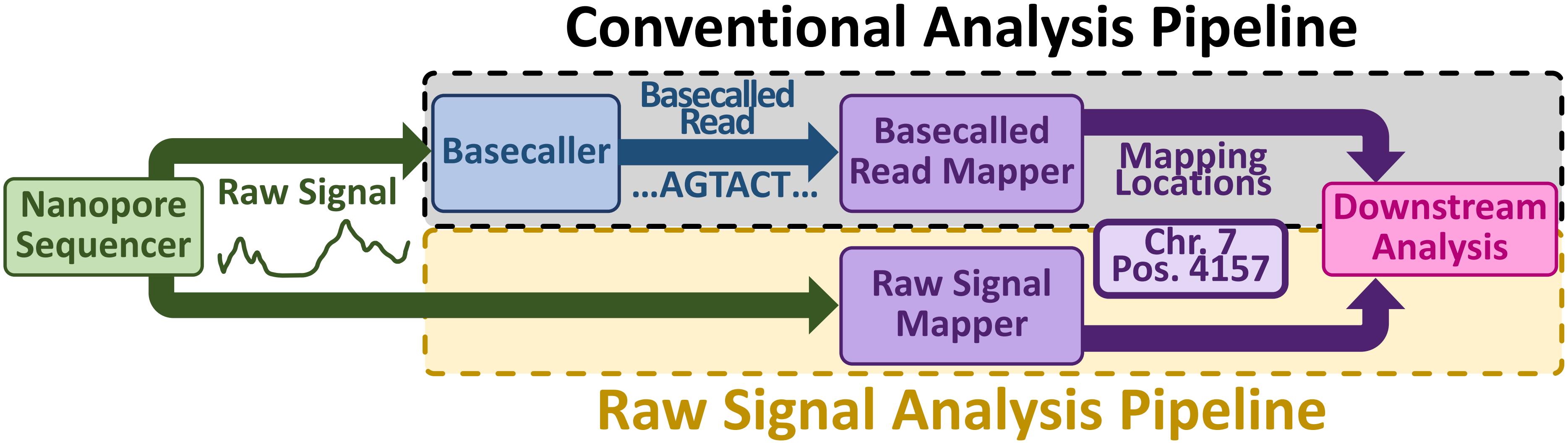
Problem: Real-time analysis of nanopore raw signals **fails to scale** to large reference databases (e.g., the human genome)

Goal: Analyze raw nanopore signals with
- high accuracy
- high throughput
- low latency
- low memory usage
- needing few bases to be sequenced
for a **wide range of reference database size**

RawAlign: The **first Seed-Filter-Align mapper** for raw nanopore signals

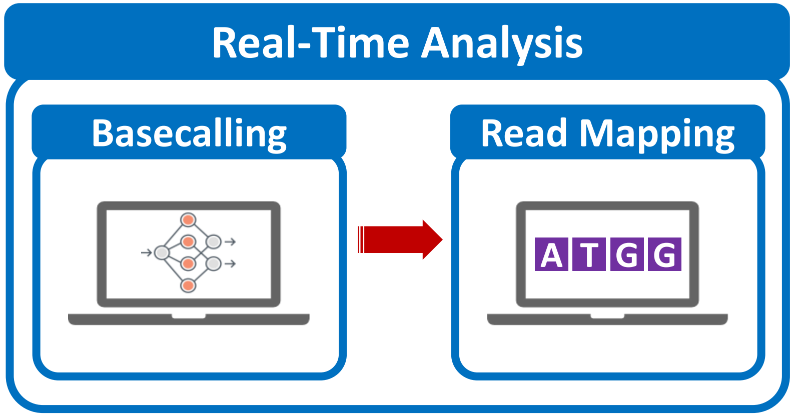
Key Results:
- Only tool to map raw nanopore signals to **large reference databases** with **high accuracy**
- Generalizes to all kinds of **reference database sizes**
- Compared to **RawHash**: **similar throughput** (between 0.80×-1.08×) while **improving accuracy** on all datasets (between 1.02×-1.64× F-1 score)

Nanopore Signal Analysis Overview



Existing Solutions Nanopore Signal Analysis

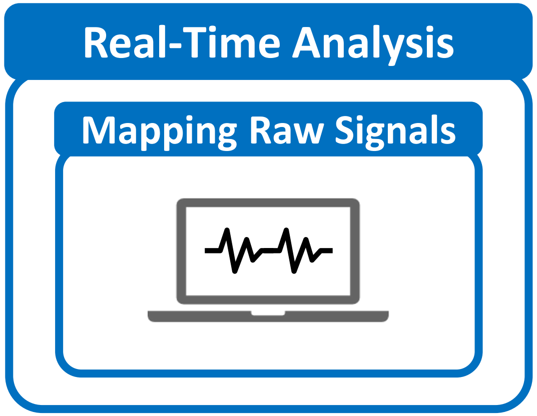
1. Deep neural networks (**DNNs**) for translating **signals** to **bases**



Less noisy analysis from basecalled sequences

Costly and power-hungry computational requirements

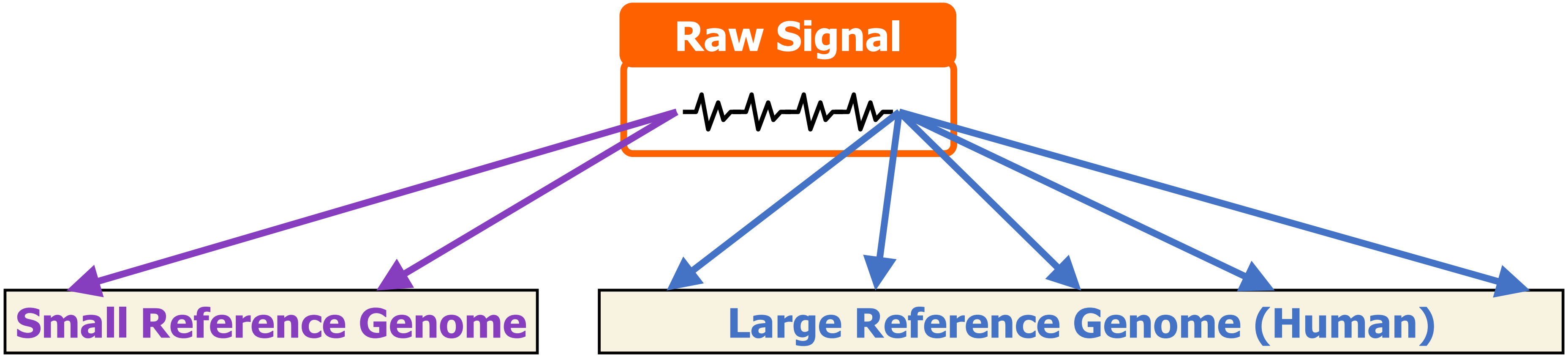
2. Mapping **signals** to reference genomes **without** basecalling



Raw signals contain richer information than bases

Efficient analysis with better scalability and portability

The Problem – Mapping Raw Signals



Raw Signal

Small Reference Genome

Large Reference Genome (Human)

Fewer candidate regions in **small genomes**

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

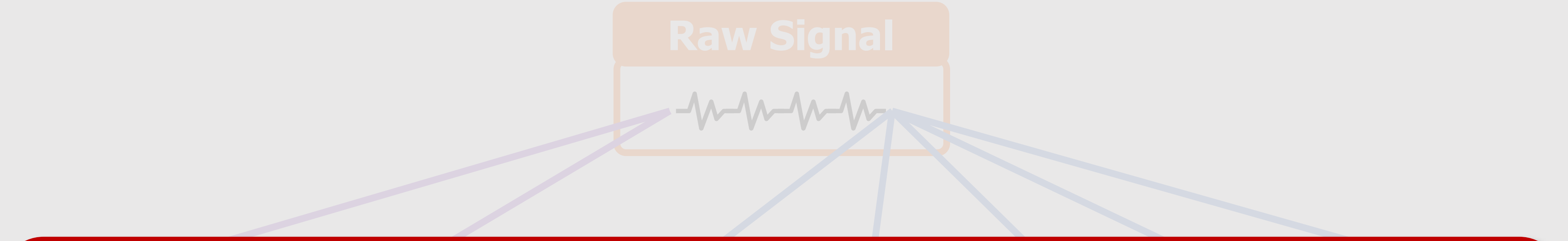
Accurate mapping

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

RawAlign

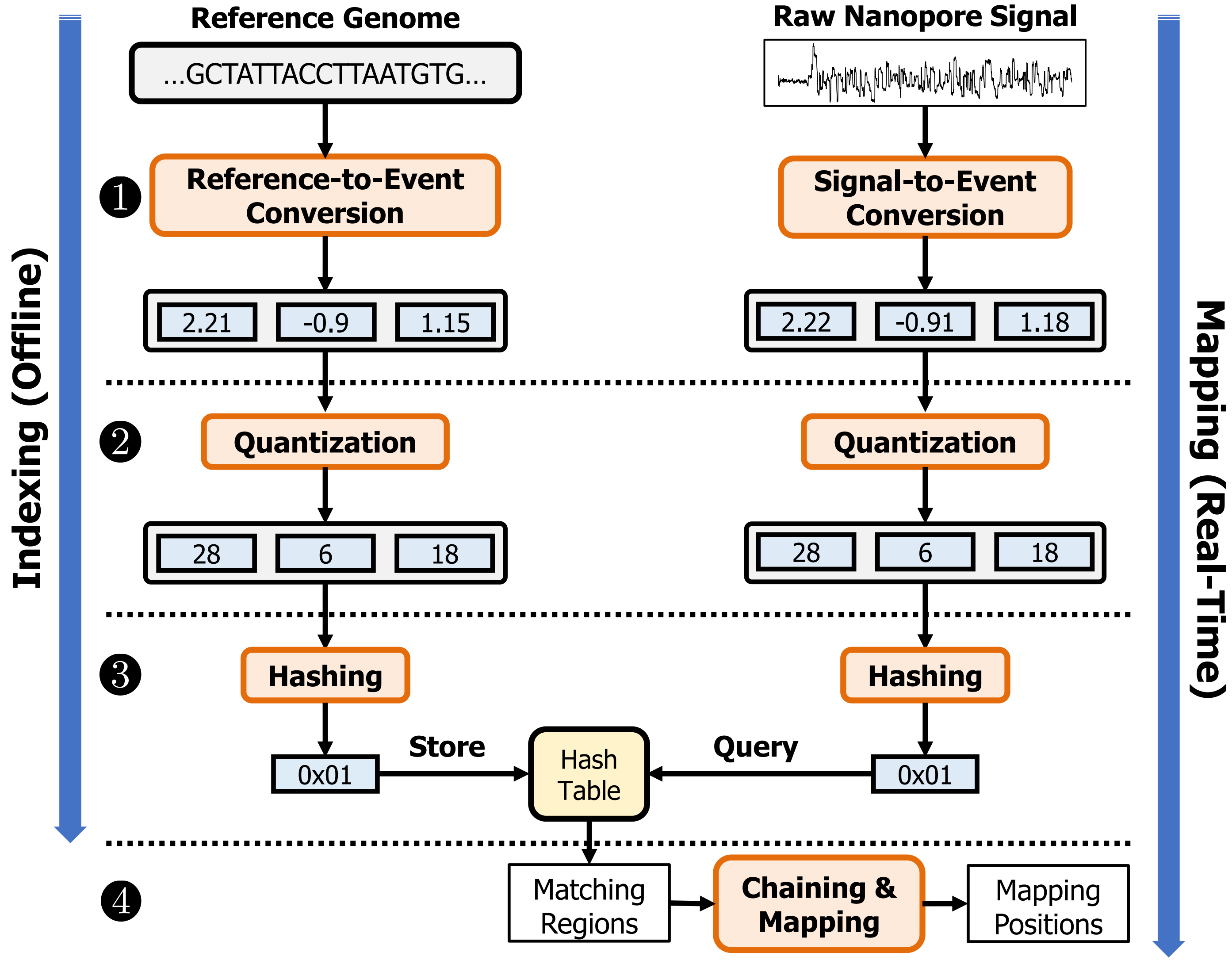
Evaluation

Conclusion

Goal

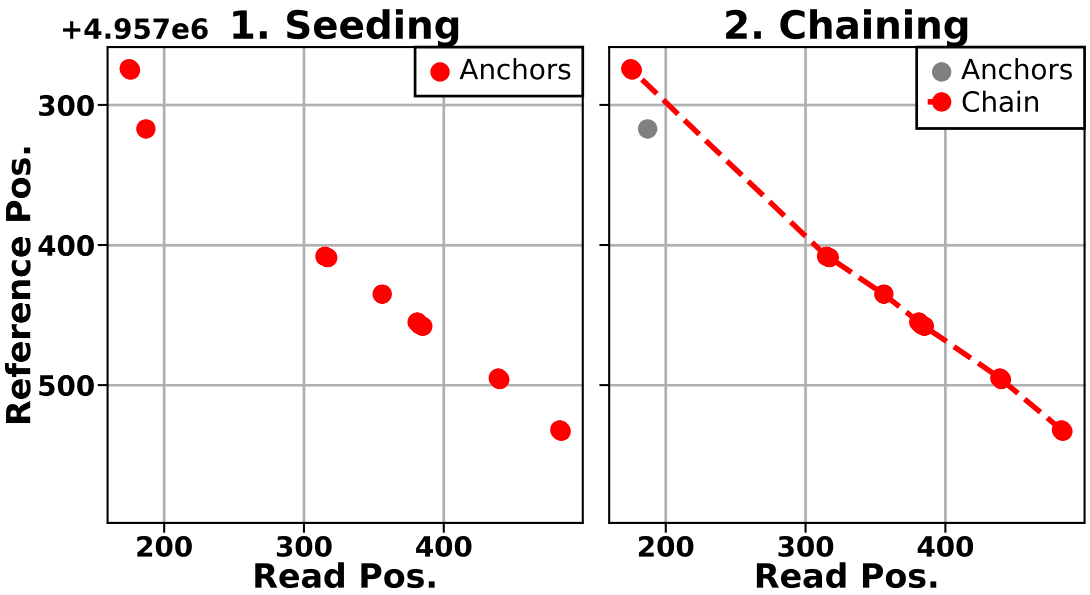
Analyze raw nanopore signals with
- high accuracy
- high throughput
- low latency
- low memory usage
- needing few bases to be sequenced
for a **wide range of reference database size**

RawHash Overview [Firtina+]



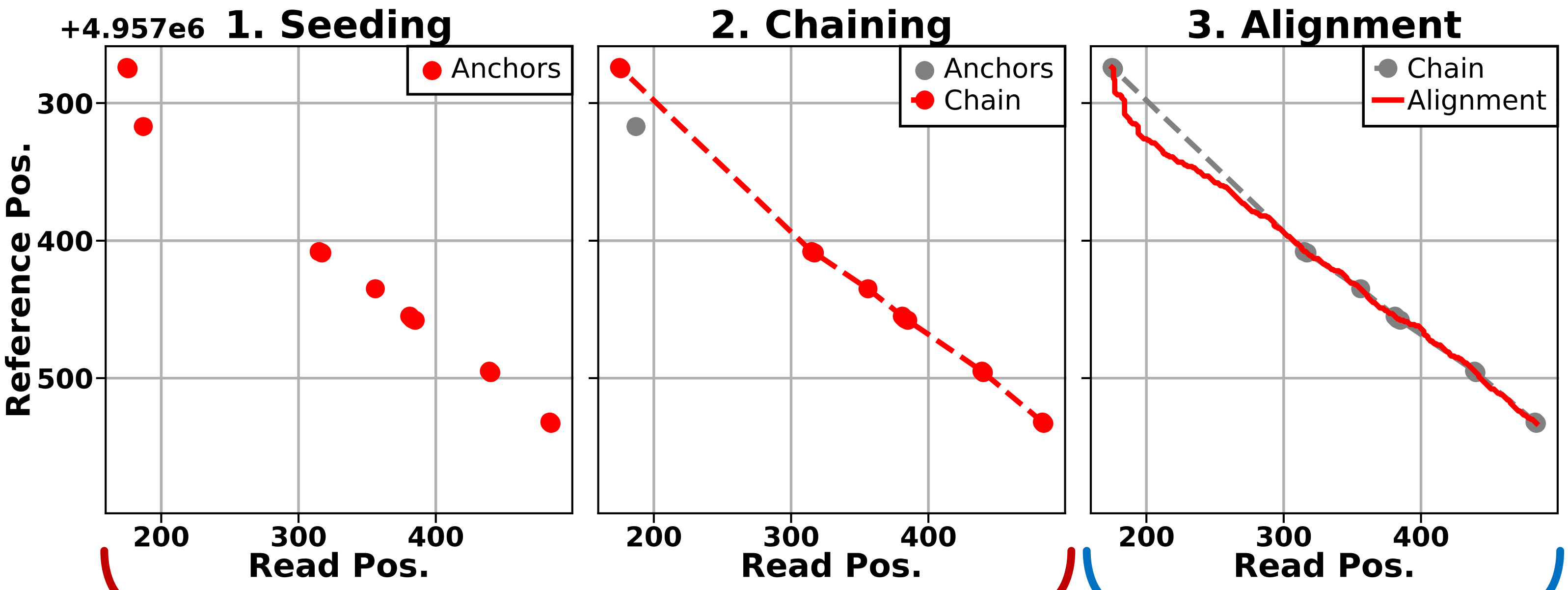
Firtina+, "RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes", Bioinformatics, 2023

RawHash Overview [Firtina+]



Firtina+, "RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes", Bioinformatics, 2023

RawAlign Overview


1. Seeding

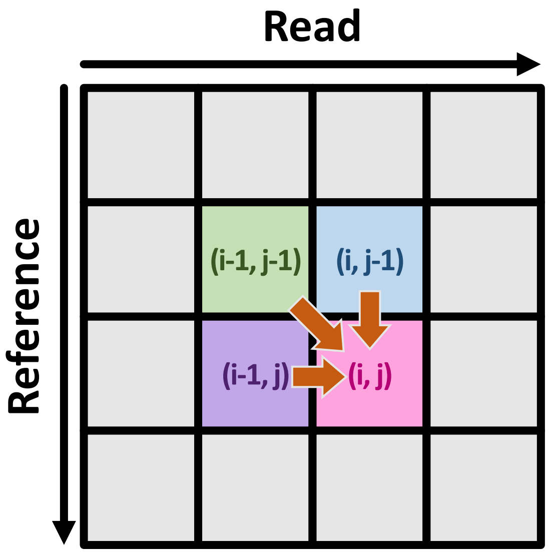
2. Chaining

3. Alignment

Coarse-Grained
Fast

Fine-Grained
Accurate

Alignment Algorithms



Needleman-Wunsch

Compare Basecalled Sequences

Nucleotide Bases

$$dp[i,j] = \min \begin{Bmatrix} dp[i-1,j-1] + (\boxed{read[i] == ref[j]})\ ?\ 0 : 1 \\ dp[i-1,j] + 1 \\ dp[i,j-1] + 1 \end{Bmatrix}$$

Dynamic Time Warping

Compare Raw Signal Sequences

Numeric Signal Values

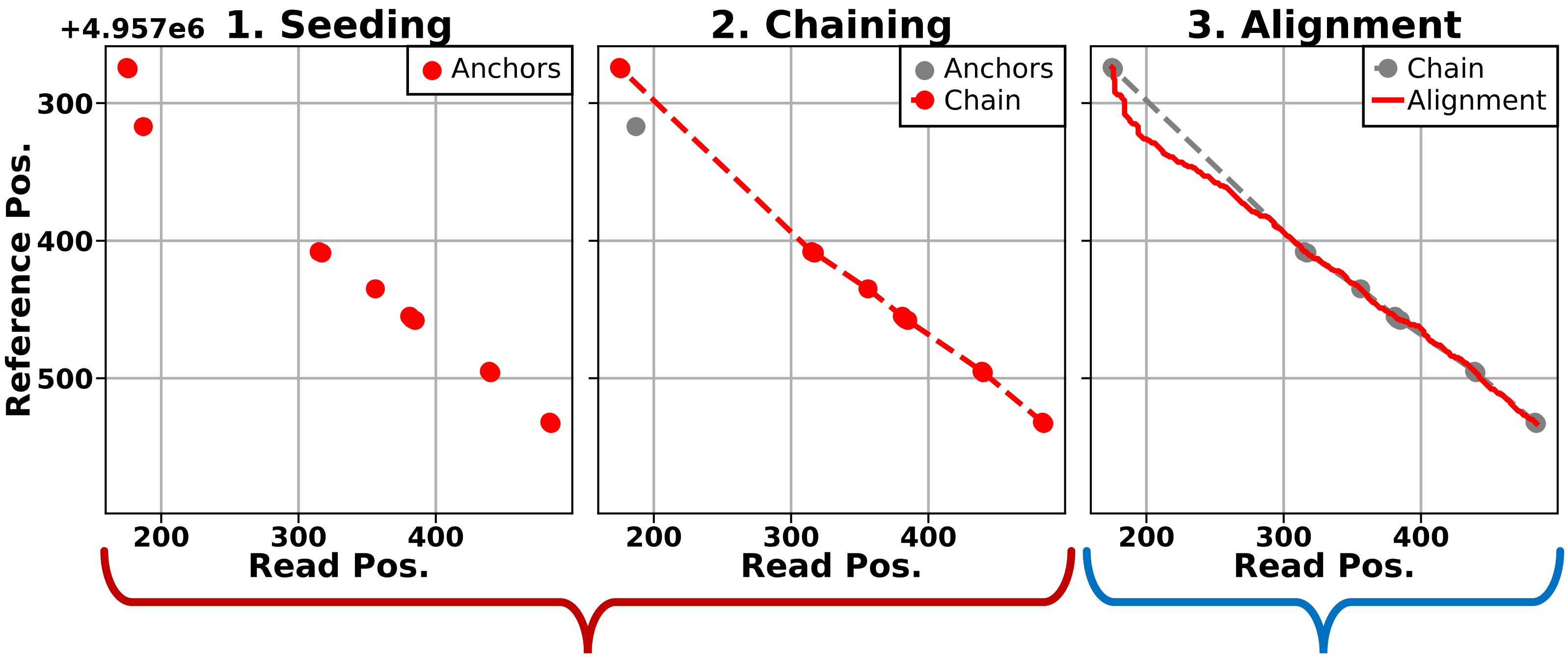
$$dp[i,j] = \min \begin{Bmatrix} dp[i-1,j-1] \\ dp[i-1,j] \\ dp[i,j-1] \end{Bmatrix} + abs(\boxed{read[i] - ref[j]})$$

Challenges in Integrating Alignment to Mapping

1. Alignment Algorithms **Called Frequently**

2. **Each Call** to Alignment Algorithm is **Expensive**

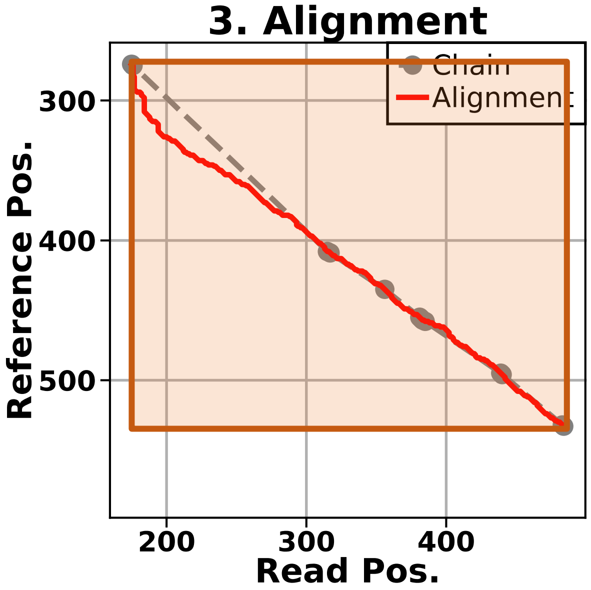
Recall: RawAlign Overview



Coarse-Grained
Fast

Fine-Grained
Accurate
Slow

Alignment is Expensive



Dynamic programming table
scales with the **square** of the **read length**

Efficient Alignment

RawAlign **efficiently** integrates **alignment** through

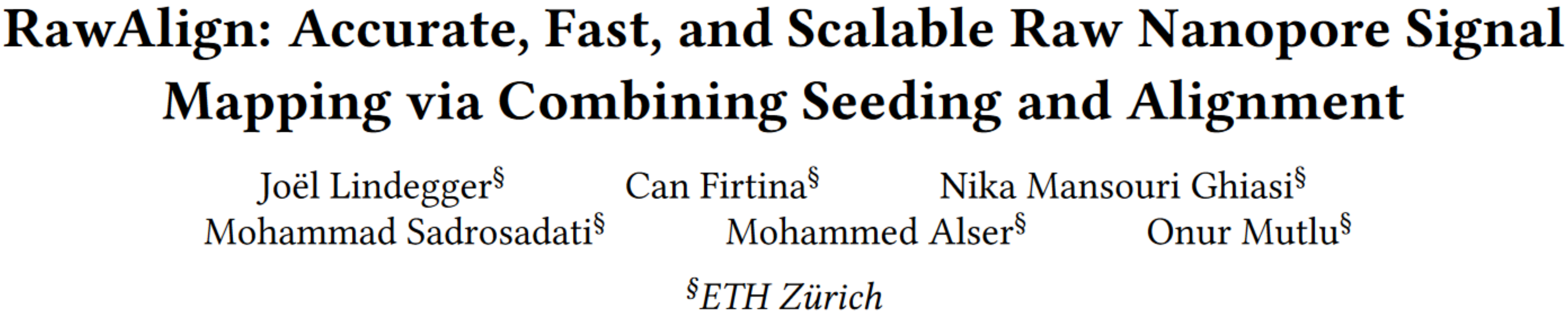
1. Pre-alignment **filtering** (chaining)
2. **Early termination** (branch-and-bound)
3. **Anchor-guided alignment**
4. **Banding/windowing**
5. **Vectorization** (SIMD)

More in The Paper

RawAlign **efficiently** integrates **alignment** through

1. Pre-alignment **filtering** (chaining)
2. **Early termination** (branch-and-bound)
3. **Anchor-guided alignment**
4. **Banding/windowing**
5. **Vectorization** (SIMD)

All Details in the Paper



RawAlign: Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment

Joël Lindegger[§] Can Firtina[§] Nika Mansouri Ghiasi[§]
Mohammad Sadrosadati[§] Mohammed Alser[§] Onur Mutlu[§]

[§]*ETH Zürich*

Outline

Background

RawAlign

Evaluation

Conclusion

Evaluation Methodology

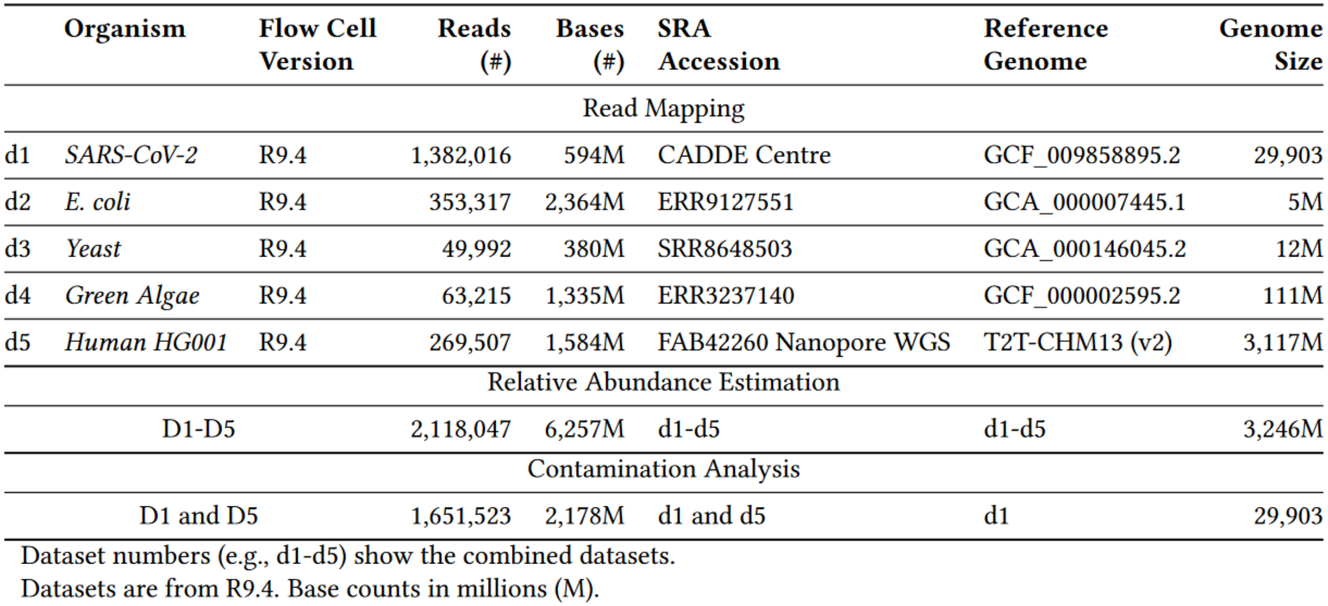
- Compared to **UNCALLED** [Kovaka+, Nat. Biotech. 2021]
  Sigmap [Zhang+, ISMB/ECCB 2021]
  and **RawHash** [Firtina+, Bioinformatics 2023]
  - **CPU baseline:** Intel Xeon Gold 6226R @2.9GHz
  - **64 threads** for each tool

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

Evaluation Methodology

- Evaluation metrics:
  - **Memory footprint (GB)**
  - Mean **throughput (bp/s)** per thread
  - Mean **analysis latency (ms)**
  - Mean **sequencing latency (chunks)**
  - **Accuracy (F-1 score)**

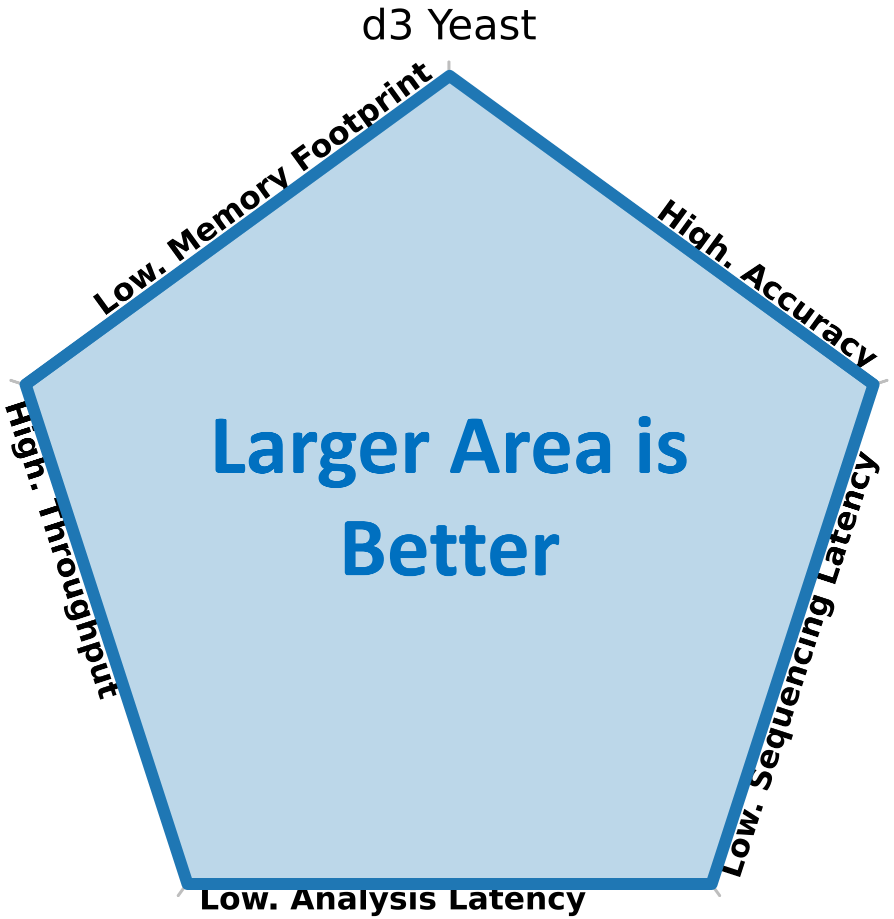
- **Datasets:**



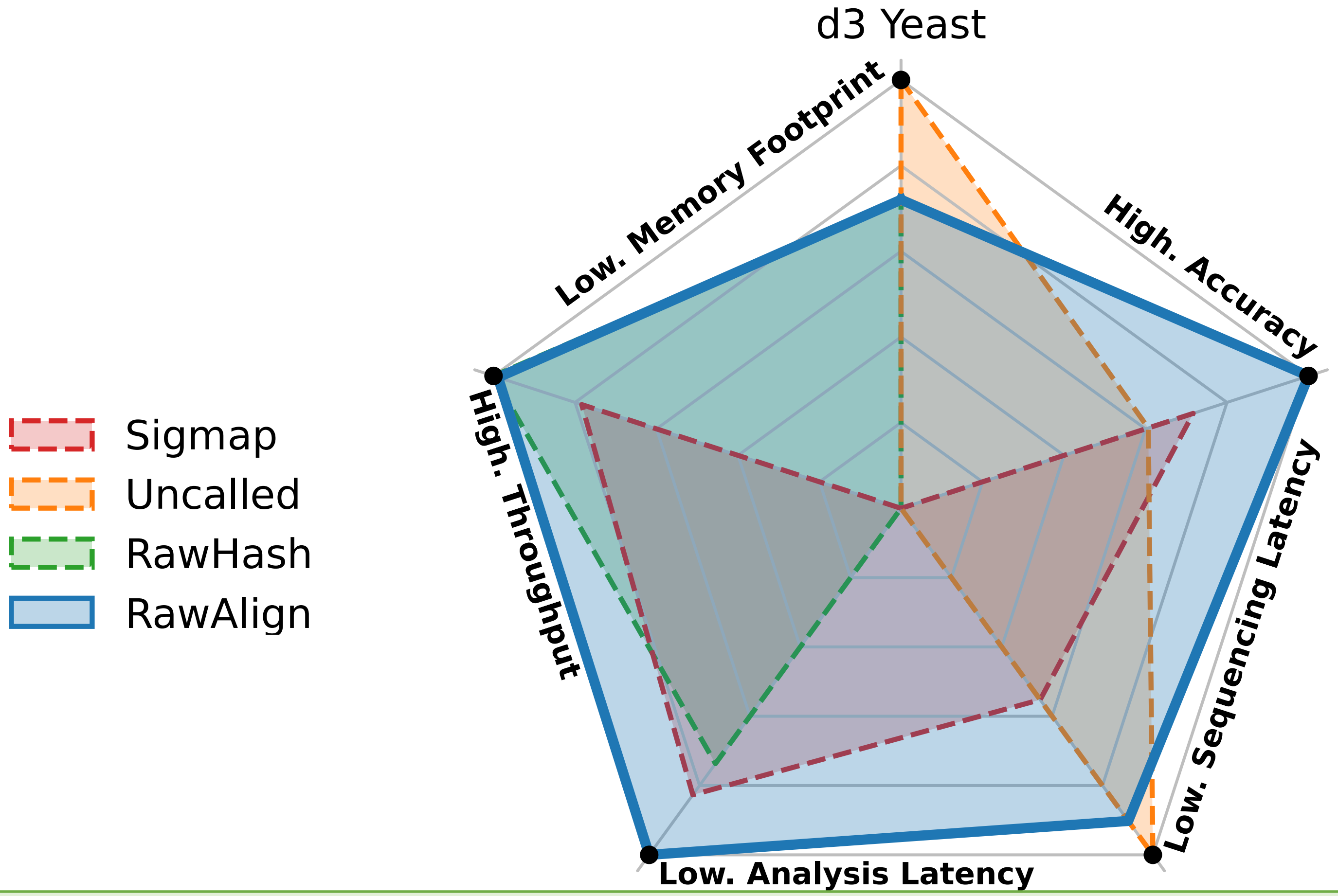
	Organism	Flow Cell Version	Reads (#)	Bases (#)	SRA Accession	Reference Genome	Genome Size
				Read Mapping			
d1	*SARS-CoV-2*	R9.4	1,382,016	594M	CADDE Centre	GCF_009858895.2	29,903
d2	*E. coli*	R9.4	353,317	2,364M	ERR9127551	GCA_000007445.1	5M
d3	*Yeast*	R9.4	49,992	380M	SRR8648503	GCA_000146045.2	12M
d4	*Green Algae*	R9.4	63,215	1,335M	ERR3237140	GCF_000002595.2	111M
d5	*Human HG001*	R9.4	269,507	1,584M	FAB42260 Nanopore WGS	T2T-CHM13 (v2)	3,117M
				Relative Abundance Estimation			
	D1-D5		2,118,047	6,257M	d1-d5	d1-d5	3,246M
				Contamination Analysis			
	D1 and D5		1,651,523	2,178M	d1 and d5	d1	29,903

Dataset numbers (e.g., d1-d5) show the combined datasets.
Datasets are from R9.4. Base counts in millions (M).

Read Mapping Results



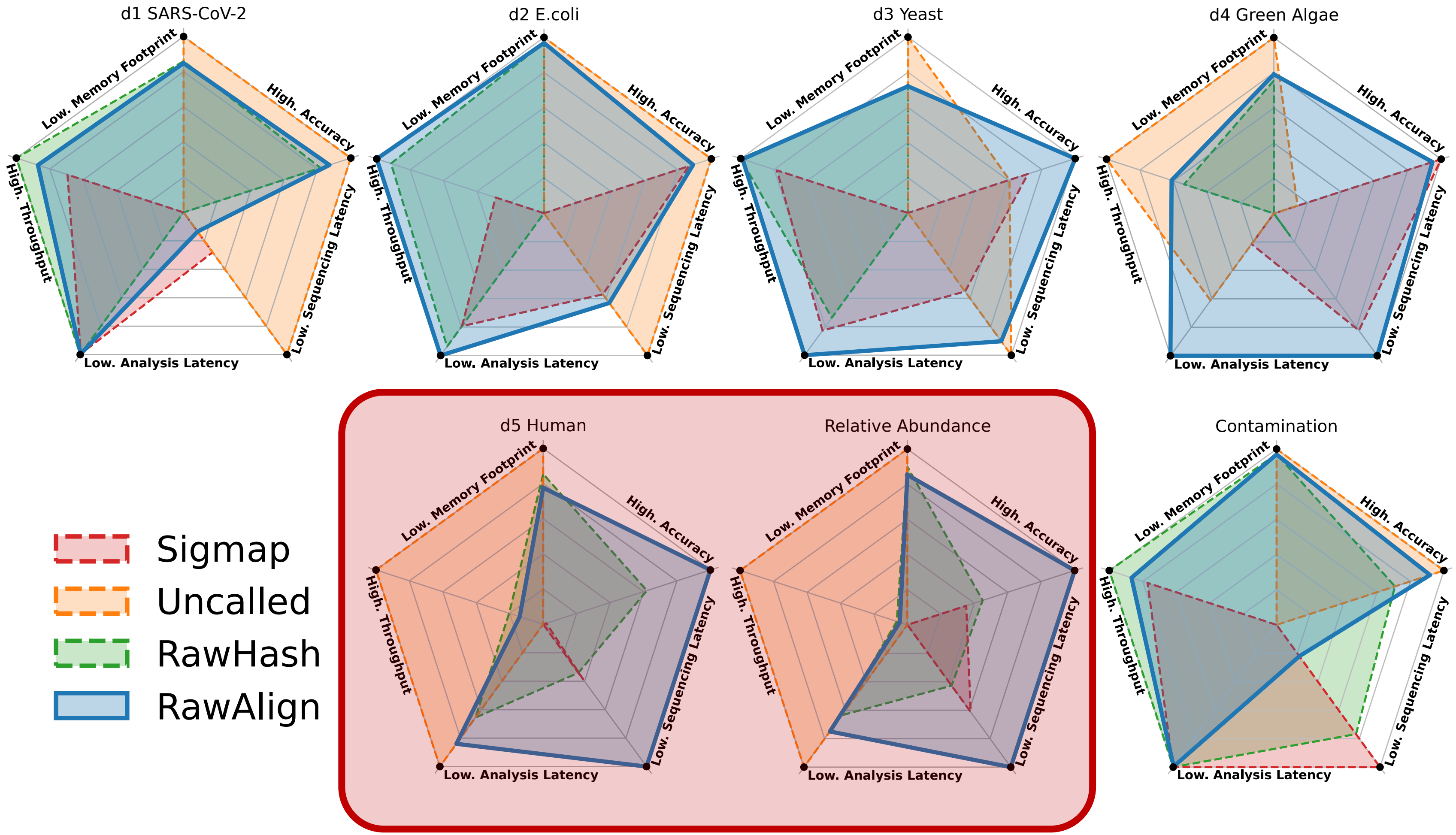
Read Mapping Results



RawAlign is the **only tool** to do **well in all metrics**

and has the **highest accuracy and throughput**

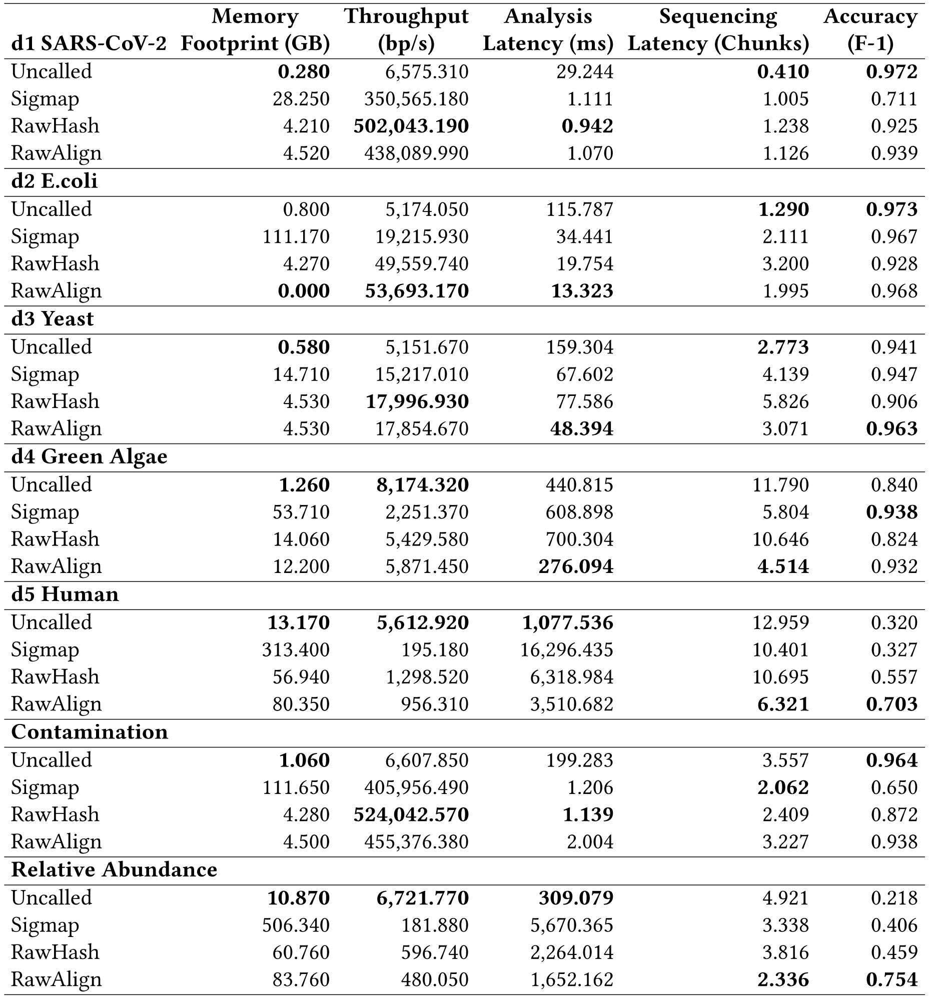
Read Mapping Results



Large Reference Databases

“Difficult” Datasets

Read Mapping Results



d1 SARS-CoV-2	Memory Footprint (GB)	Throughput (bp/s)	Analysis Latency (ms)	Sequencing Latency (Chunks)	Accuracy (F-1)
Uncalled	**0.280**	6,575.310	29.244	**0.410**	**0.972**
Sigmap	28.250	350,565.180	1.111	1.005	0.711
RawHash	4.210	**502,043.190**	**0.942**	1.238	0.925
RawAlign	4.520	438,089.990	1.070	1.126	0.939
d2 E.coli					
Uncalled	0.800	5,174.050	115.787	**1.290**	**0.973**
Sigmap	111.170	19,215.930	34.441	2.111	0.967
RawHash	4.270	49,559.740	19.754	3.200	0.928
RawAlign	**0.000**	**53,693.170**	**13.323**	1.995	0.968
d3 Yeast					
Uncalled	**0.580**	5,151.670	159.304	**2.773**	0.941
Sigmap	14.710	15,217.010	67.602	4.139	0.947
RawHash	4.530	**17,996.930**	77.586	5.826	0.906
RawAlign	4.530	17,854.670	**48.394**	3.071	**0.963**
d4 Green Algae					
Uncalled	**1.260**	**8,174.320**	440.815	11.790	0.840
Sigmap	53.710	2,251.370	608.898	5.804	**0.938**
RawHash	14.060	5,429.580	700.304	10.646	0.824
RawAlign	12.200	5,871.450	**276.094**	**4.514**	0.932
d5 Human					
Uncalled	**13.170**	**5,612.920**	**1,077.536**	12.959	0.320
Sigmap	313.400	195.180	16,296.435	10.401	0.327
RawHash	56.940	1,298.520	6,318.984	10.695	0.557
RawAlign	80.350	956.310	3,510.682	**6.321**	**0.703**
Contamination					
Uncalled	**1.060**	6,607.850	199.283	3.557	**0.964**
Sigmap	111.650	405,956.490	1.206	**2.062**	0.650
RawHash	4.280	**524,042.570**	**1.139**	2.409	0.872
RawAlign	4.500	455,376.380	2.004	3.227	0.938
Relative Abundance					
Uncalled	**10.870**	**6,721.770**	**309.079**	4.921	0.218
Sigmap	506.340	181.880	5,670.365	3.338	0.406
RawHash	60.760	596.740	2,264.014	3.816	0.459
RawAlign	83.760	480.050	1,652.162	**2.336**	**0.754**

Relative Abundance Results

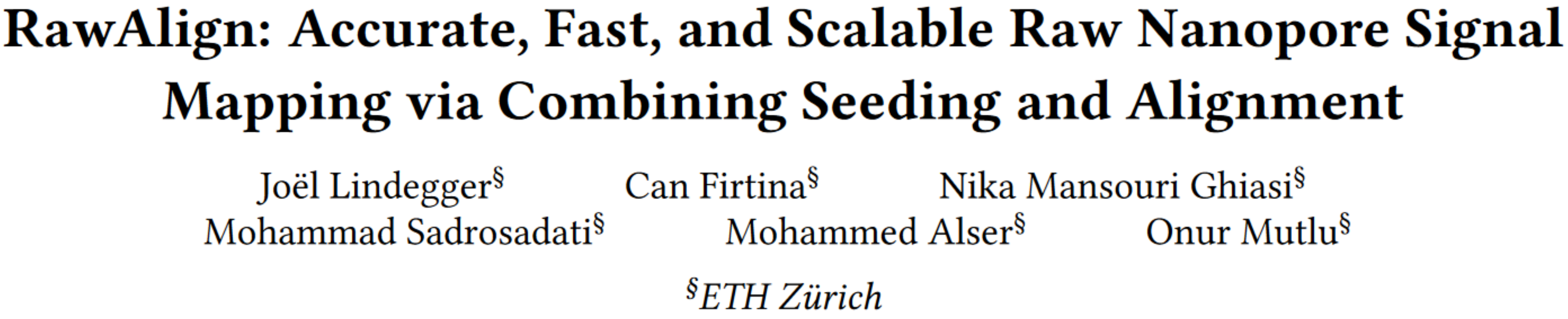
Tool	SARS-CoV-2	E.coli	Yeast	Green Algae	Human	Distance
Ground Truth	0.652	0.167	0.024	0.030	0.127	-
minimap2	0.613	0.163	0.025	0.053	0.147	**0.050**
Uncalled	0.072	0.466	0.001	0.150	0.312	0.689
Sigmap	0.201	0.446	0.002	0.123	0.229	0.549
RawHash	0.309	0.440	0.000	0.073	0.178	0.445
RawAlign	0.565	0.248	0.002	0.050	0.136	0.123

State-of-the-art basecalling baseline

RawAlign

RawAlign approaches the accuracy of the **state-of-the-art basecalling-based** analysis pipeline (using minimap2)

All Details in the Paper

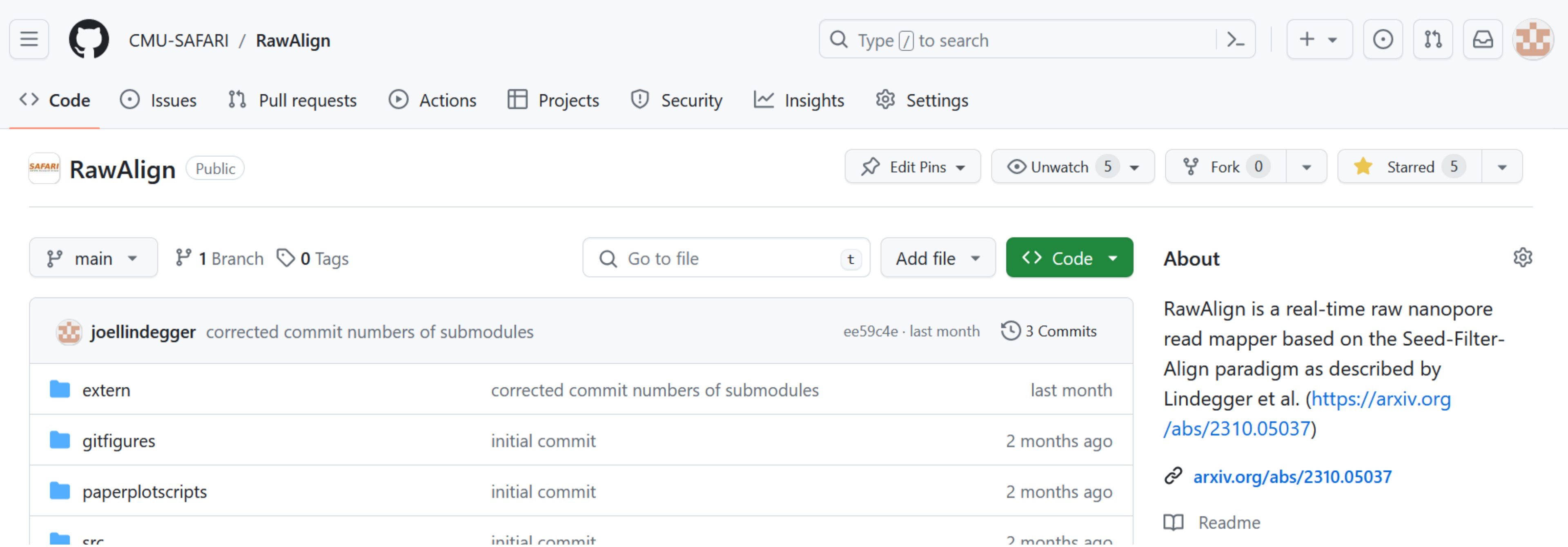


RawAlign: Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment

Joël Lindegger[§] Can Firtina[§] Nika Mansouri Ghiasi[§]
Mohammad Sadrosadati[§] Mohammed Alser[§] Onur Mutlu[§]

[§]*ETH Zürich*

RawAlign Source Code



CMU-SAFARI / RawAlign

joellindegger corrected commit numbers of submodules ee59c4e · last month 3 Commits

extern	corrected commit numbers of submodules	last month
gitfigures	initial commit	2 months ago
paperplotscripts	initial commit	2 months ago
src	initial commit	2 months ago

About

RawAlign is a real-time raw nanopore read mapper based on the Seed-Filter-Align paradigm as described by Lindegger et al. (https://arxiv.org/abs/2310.05037)

arxiv.org/abs/2310.05037

Readme

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

Outline

Background

RawAlign

Evaluation

Conclusion

Conclusion

Problem: Real-time analysis of nanopore raw signals **fails to scale** to large reference databases (e.g., the human genome)

Goal: Analyze raw nanopore signals with
- high accuracy
- high throughput
- low latency
- low memory usage
- needing few bases to be sequenced
for a **wide range of reference database size**

RawAlign: The **first Seed-Filter-Align mapper** for raw nanopore signals

Key Results:
- Only tool to map raw nanopore signals to **large reference databases** with **high accuracy**
- **Generalizes** to all kinds of **reference database sizes**
- Compared to **RawHash**: **similar throughput** (between 0.80×-1.08×) while **improving accuracy** on all datasets (between 1.02×-1.64× F-1 score)

RawAlign

Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment

Joël Lindegger

Can Firtina Nika Mansouri Ghiasi

Mohammad Sadrosadati Mohammed Alser Onur Mutlu

SAFARI

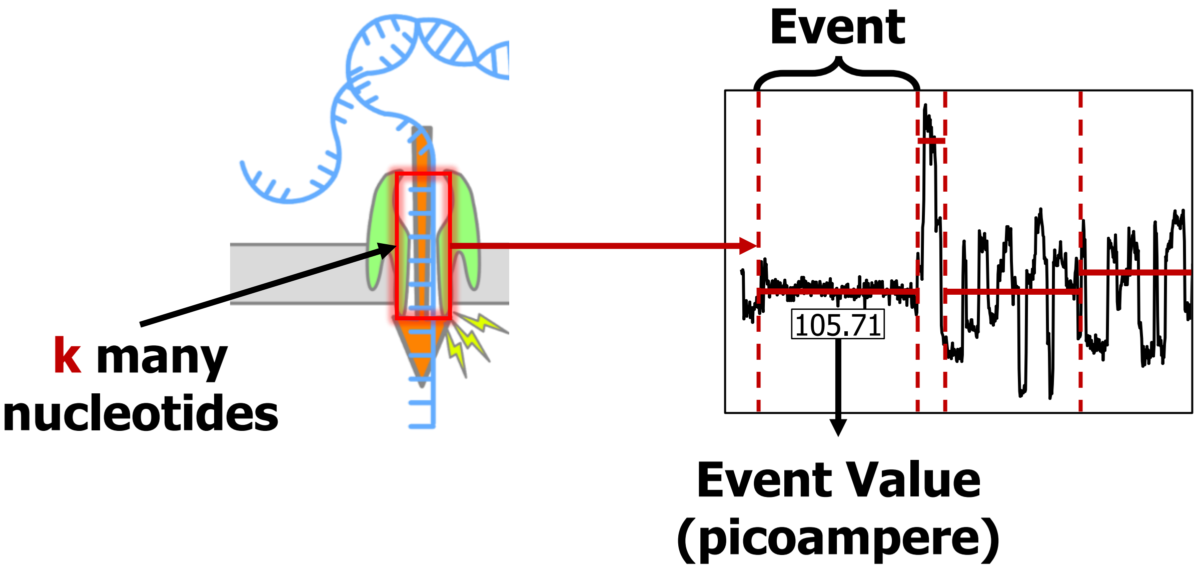
ETH zürich

Backup Slides

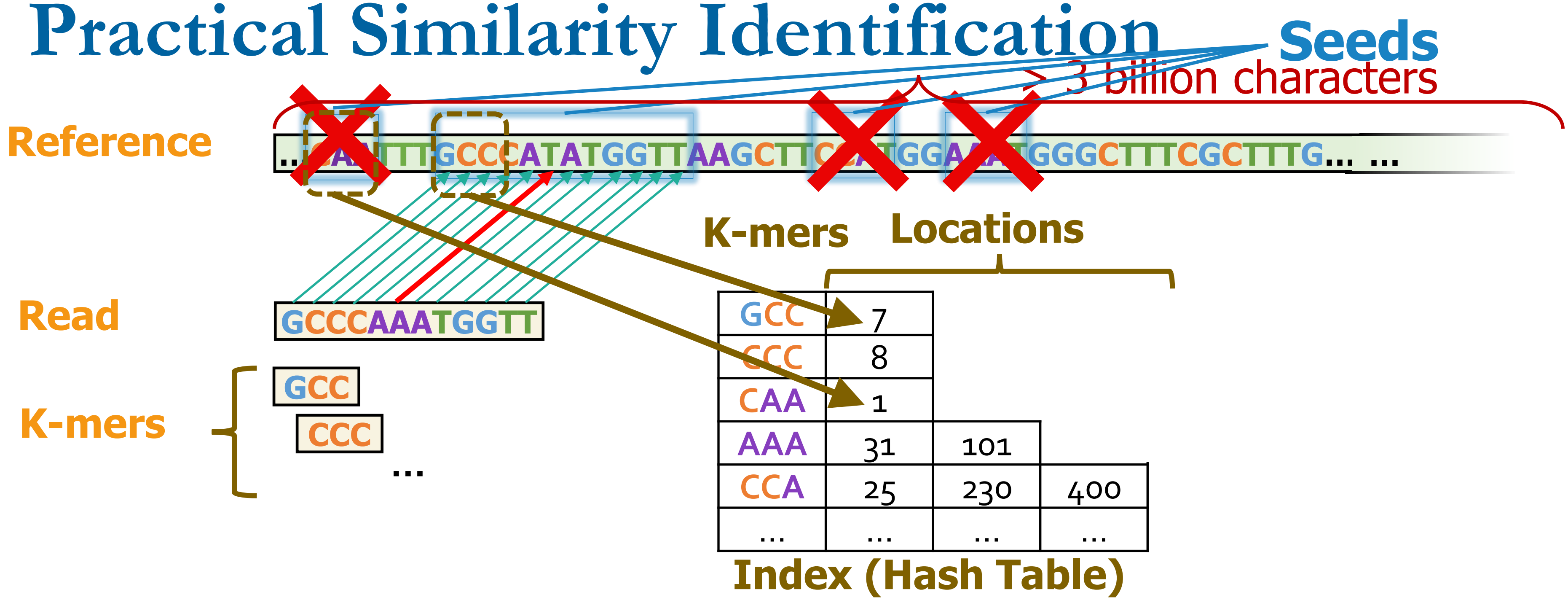
Events in Raw Nanopore Signals

- **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**



Practical Similarity Identification



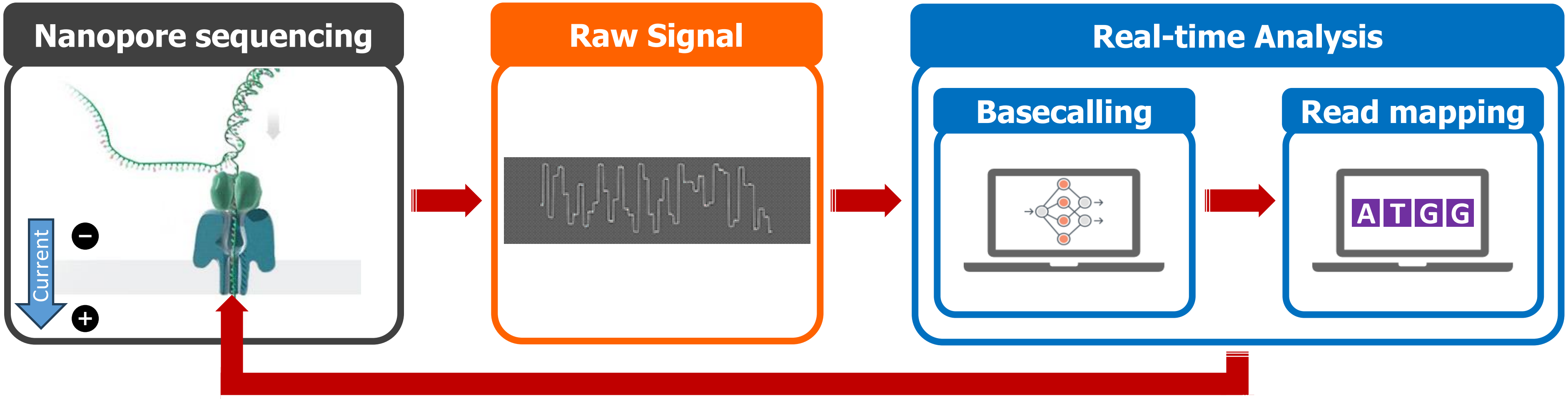
K-mers	Locations		
GCC	7		
CCC	8		
CAA	1		
AAA	31	101	
CCA	25	230	400
...	...	...	...

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

Existing Solutions – Real-time Basecalling

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

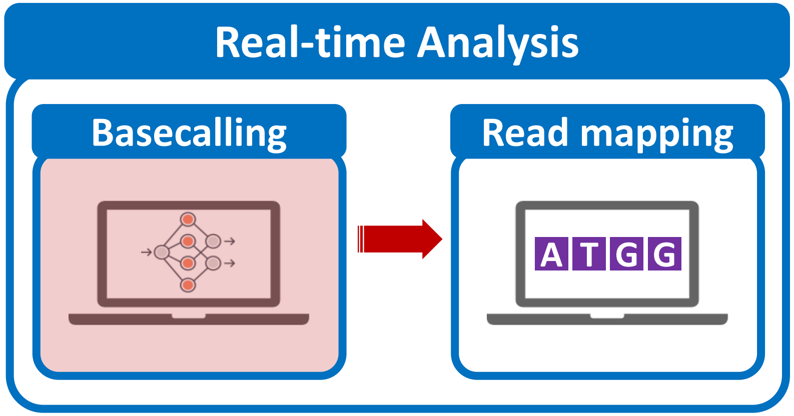


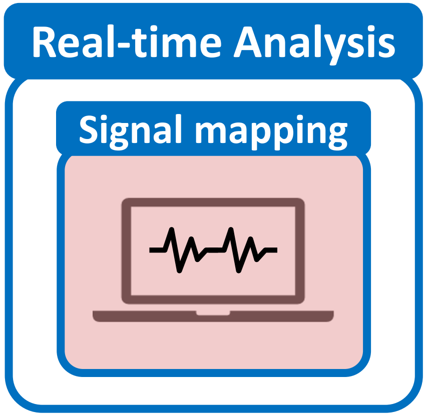
DNNs provide **less noisy analysis** from basecalled sequences

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

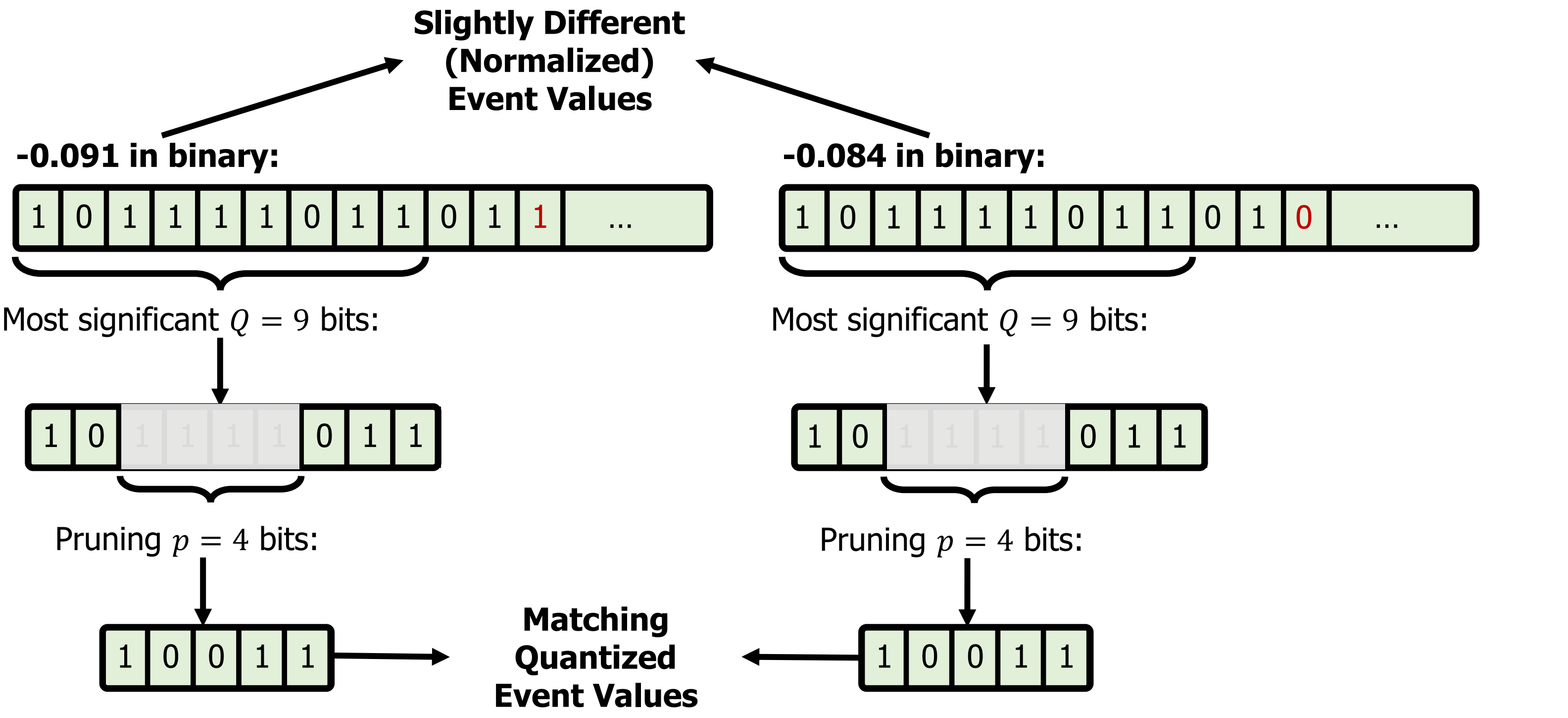
Sequence Until: Stopping the sequencing based on information decision

- 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**



Slightly Different (Normalized) Event Values

-0.091 in binary:

1 0 1 1 1 1 0 1 1 0 1 1 …

Most significant $Q = 9$ bits:

1 0 1 1 1 1 0 1 1

Pruning $p = 4$ bits:

1 0 0 1 1

-0.084 in binary:

1 0 1 1 1 1 0 1 1 0 1 0 …

Most significant $Q = 9$ bits:

1 0 1 1 1 1 0 1 1

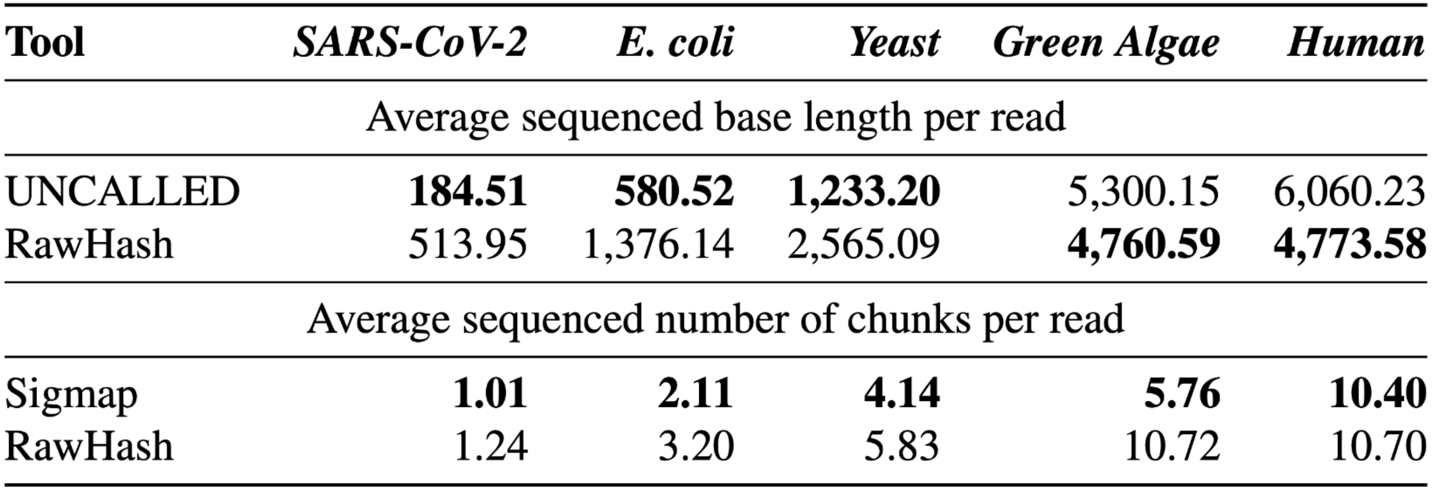
Pruning $p = 4$ bits:

1 0 0 1 1

Matching Quantized Event Values

SAFARI

Average Sequenced Bases and Chunks



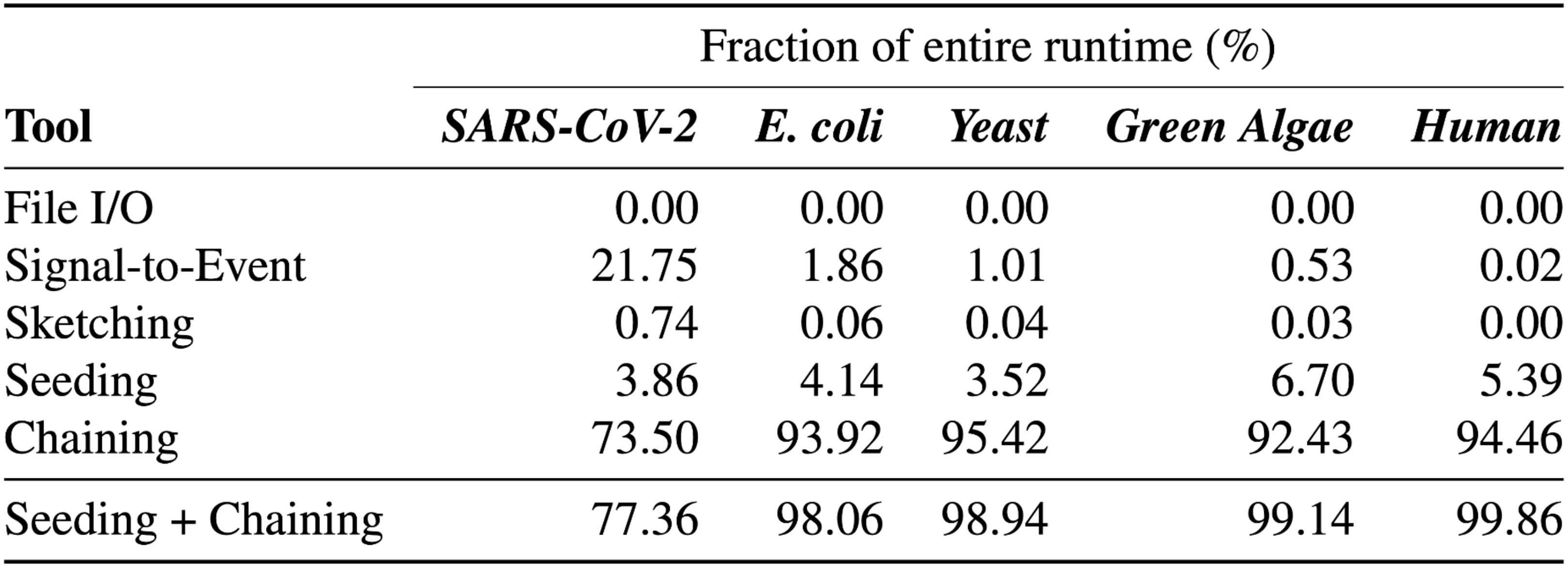
Tool	*SARS-CoV-2*	*E. coli*	*Yeast*	*Green Algae*	*Human*
Average sequenced base length per read					
UNCALLED	**184.51**	**580.52**	**1,233.20**	5,300.15	6,060.23
RawHash	513.95	1,376.14	2,565.09	**4,760.59**	**4,773.58**
Average sequenced number of chunks per read					
Sigmap	**1.01**	**2.11**	**4.14**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

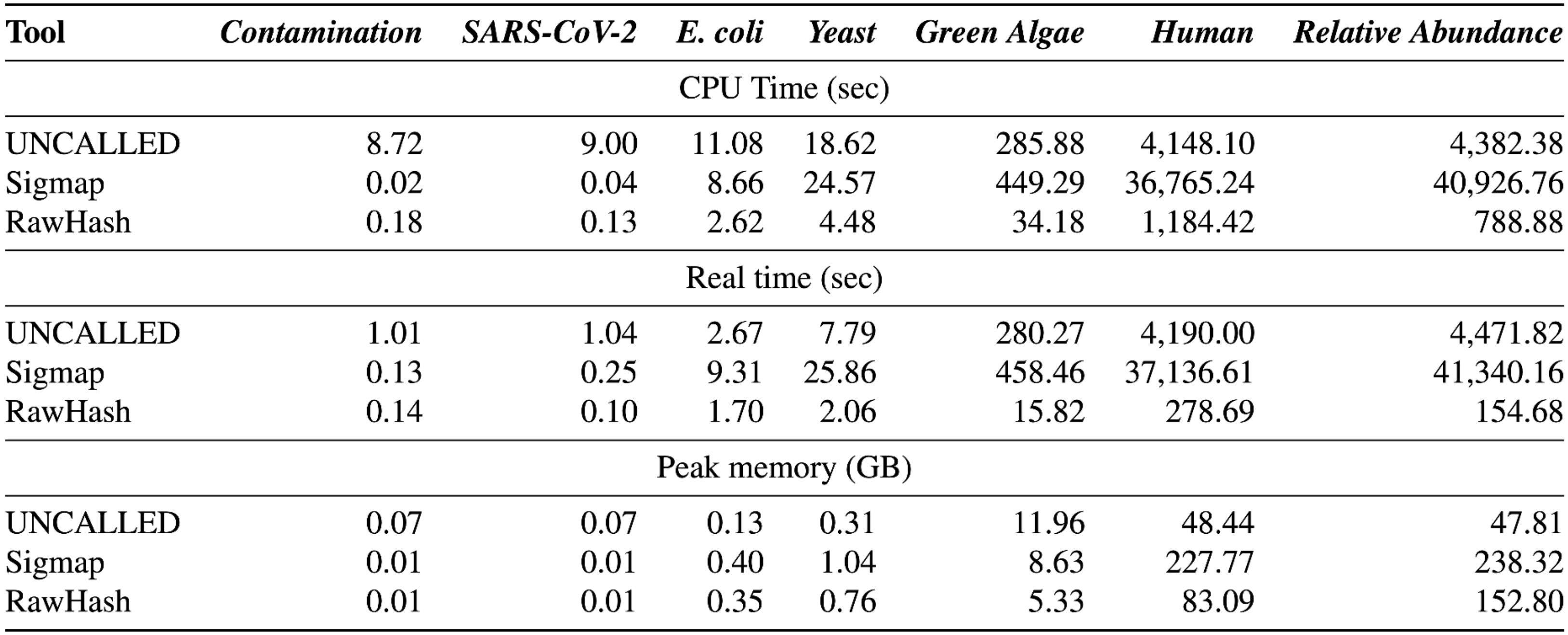
Breakdown Analysis of the RawHash Steps



	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

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

Required Computation Resources in Indexing

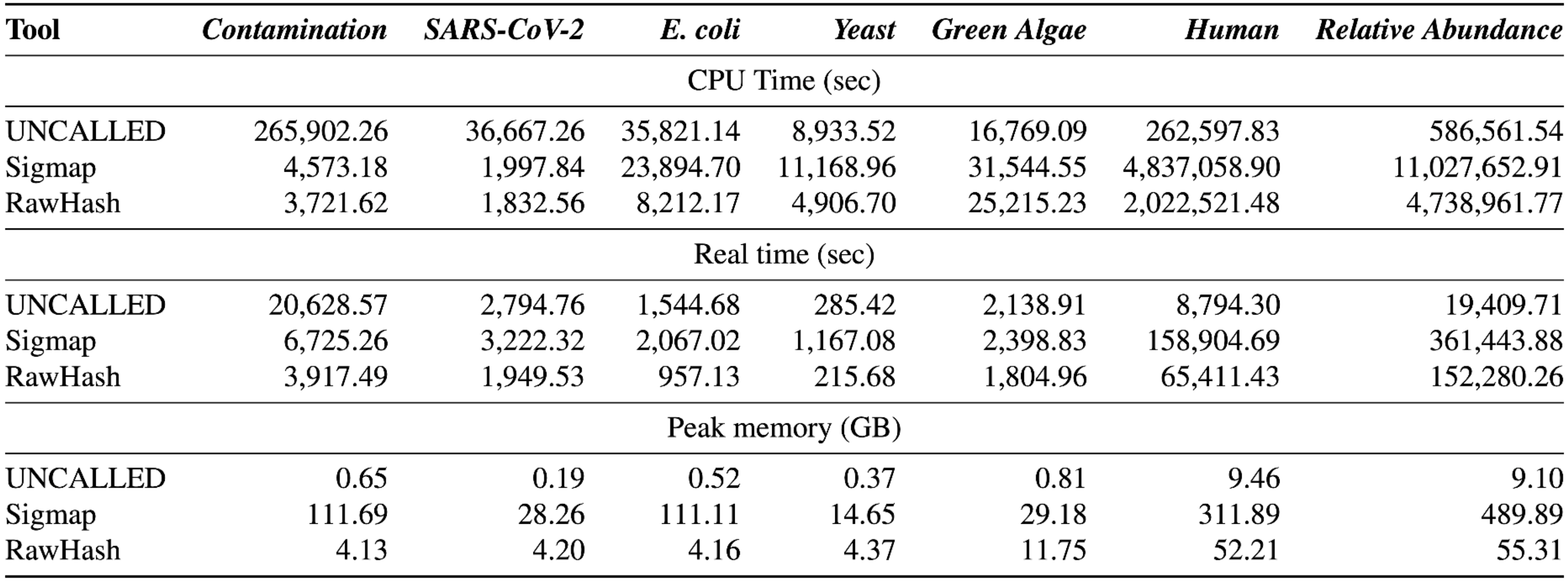


Tool	*Contamination*	*SARS-CoV-2*	*E. coli*	*Yeast*	*Green Algae*	*Human*	*Relative Abundance*
			CPU Time (sec)				
UNCALLED	8.72	9.00	11.08	18.62	285.88	4,148.10	4,382.38
Sigmap	0.02	0.04	8.66	24.57	449.29	36,765.24	40,926.76
RawHash	0.18	0.13	2.62	4.48	34.18	1,184.42	788.88
			Real time (sec)				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

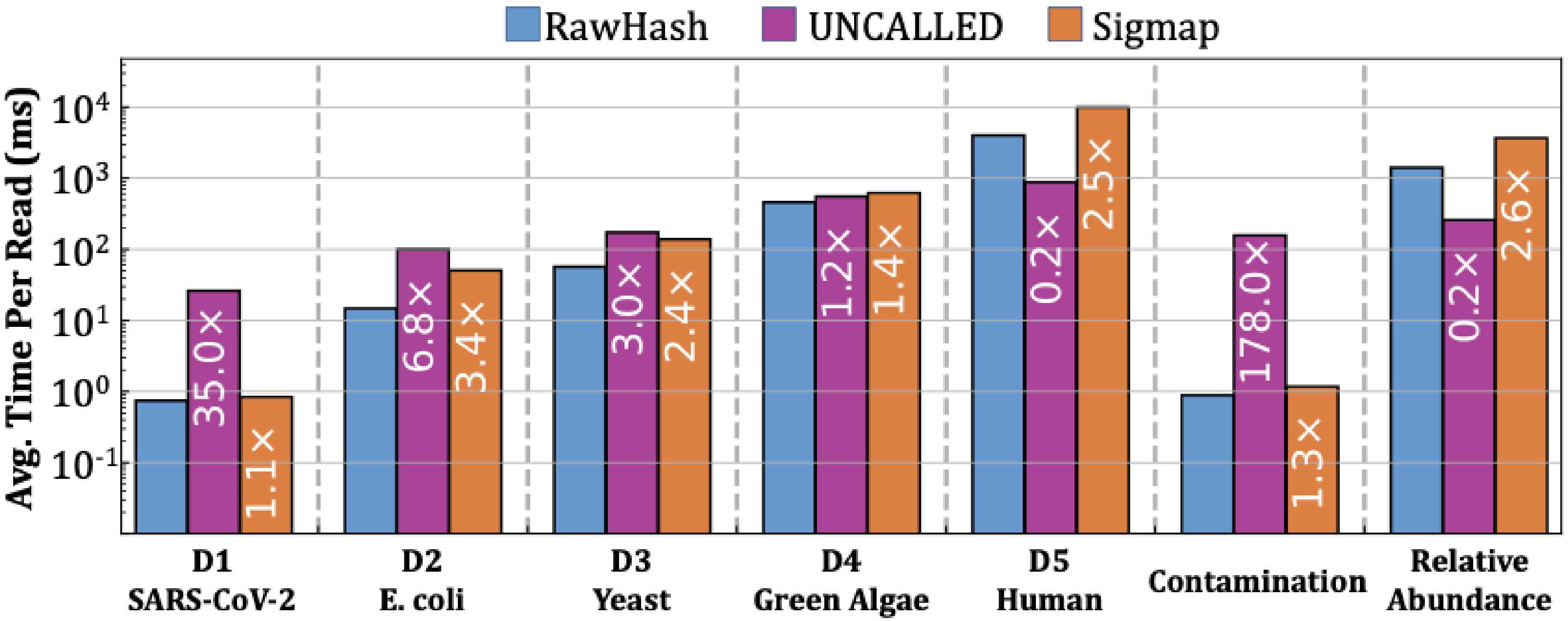


Tool	*Contamination*	*SARS-CoV-2*	*E. coli*	*Yeast*	*Green Algae*	*Human*	*Relative Abundance*
				CPU Time (sec)			
UNCALLED	265,902.26	36,667.26	35,821.14	8,933.52	16,769.09	262,597.83	586,561.54
Sigmap	4,573.18	1,997.84	23,894.70	11,168.96	31,544.55	4,837,058.90	11,027,652.91
RawHash	3,721.62	1,832.56	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**

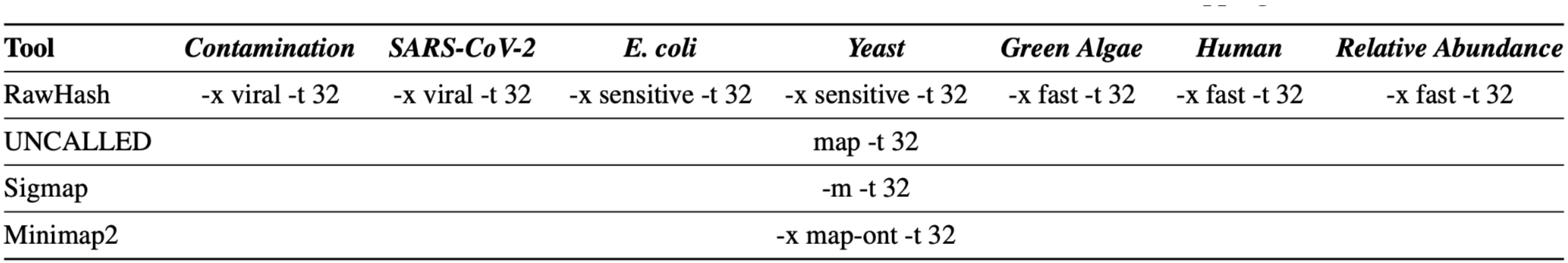
RawHash requires **larger memory space** than UNCALLED

Average Mapping Time per Read

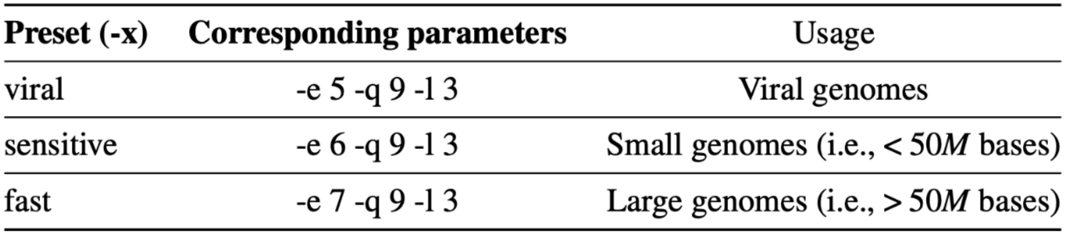


The mapping step of RawHash is **significantly faster than Sigmap** for all genomes, and **faster than UNCALLED for small genomes**

Parameter Configurations

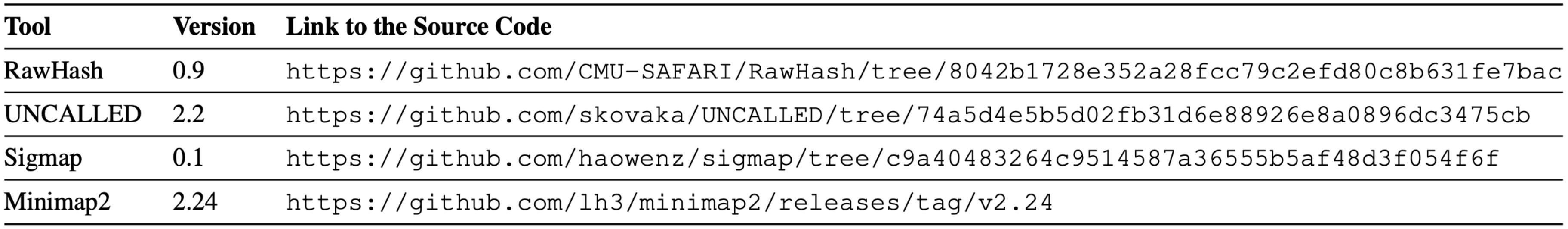


Tool	***Contamination***	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	***Relative Abundance***
RawHash	-x viral -t 32	-x viral -t 32	-x sensitive -t 32	-x sensitive -t 32	-x fast -t 32	-x fast -t 32	-x fast -t 32
UNCALLED	map -t 32						
Sigmap	-m -t 32						
Minimap2	-x map-ont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

RawAlign

Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment

Joël Lindegger

Can Firtina Nika Mansouri Ghiasi

Mohammad Sadrosadati Mohammed Alser Onur Mutlu

SAFARI

ETH zürich