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

SAFARI
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**

**Conventional Analysis Pipeline**

- **Nanopore Sequencer** → **Basecaller** (Basecalled Read: AGTACT) → **Basecalled Read Mapper** → **Mapping Locations** (Chr. 7 Pos. 4157) → **Downstream Analysis**

**Raw Signal Analysis Pipeline**

- **Nanopore Sequencer** → **Basecaller** (Basecalled Read: AGTACT) → **Basecalled Read Mapper** → **Mapping Locations** (Chr. 7 Pos. 4157) → **Downstream Analysis**
Existing Solutions Nanopore Signal Analysis

1. Deep neural networks (DNNs) for translating \textbf{signals} to \textbf{bases}

   - **Basecalling**
   - **Read Mapping**

   

   - Less noisy analysis from basecalled sequences
   - \textbf{Costly and power-hungry} computational requirements

2. Mapping \textbf{signals} to reference genomes \textbf{without} basecalling

   - **Mapping Raw Signals**

   - Raw signals contain richer information than bases
   - Efficient analysis with better scalability and portability
The Problem – Mapping Raw Signals

Small Reference Genome
- Fewer candidate regions in small genomes
- Accurate mapping
- High throughput

Large Reference Genome (Human)
- Substantially larger number of regions to check per read as the genome size increases
- Problem: Probabilistic mechanisms on many regions → inaccurate mapping
- Problem: Distance calculation on many regions → reduced throughput
The Problem – Mapping Raw Signals

- Small Reference Genome
- Large Reference Genome (Human)

Fewer candidate regions in small genomes

- Accurate mapping
- High throughput

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

Problem:
- Probabilistic mechanisms on many regions → inaccurate mapping
- Distance calculation on many regions → reduced throughput

Existing solutions are inaccurate or inefficient for large genomes
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+]

**RawHash Overview**  

RawAlign Overview

1. Seeding
2. Chaining
3. Alignment

Coarse-Grained: Fast
Fine-Grained: Accurate
Alignment Algorithms

**Needleman-Wunsch**

*Compare Basecalled Sequences*

\[
\begin{align*}
    dp[i,j] &= \min \left( 
    dp[i-1,j-1] + (\text{read}[i] == \text{ref}[j]) \ ? 0 : 1, \\
    dp[i-1,j] + 1, \\
    dp[i,j-1] + 1 
    \right) 
\end{align*}
\]

**Dynamic Time Warping**

*Compare Raw Signal Sequences*

\[
\begin{align*}
    dp[i,j] &= \min \left( 
    dp[i-1,j-1] + \text{abs}(|\text{read}[i] - \text{ref}[j]|), \\
    dp[i-1,j] + 1, \\
    dp[i,j-1] + 1 
    \right) 
\end{align*}
\]
Challenges in Integrating Alignment to Mapping

1. Alignment Algorithms Called Frequently

2. Each Call to Alignment Algorithm is Expensive
Recall: RawAlign Overview

1. Seeding

2. Chaining

3. Alignment

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:**

<table>
<thead>
<tr>
<th>Organism</th>
<th>Flow Cell Version</th>
<th>Reads (#)</th>
<th>Bases (#)</th>
<th>SRA Accession</th>
<th>Reference Genome</th>
<th>Genome Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SARS-CoV-2</td>
<td>R9.4</td>
<td>1,382,016</td>
<td>594M</td>
<td>CADDE Centre</td>
<td>GCF_009858895.2</td>
<td>29,903</td>
</tr>
<tr>
<td>E. coli</td>
<td>R9.4</td>
<td>353,317</td>
<td>2,364M</td>
<td>ERR9127551</td>
<td>GCA_000007445.1</td>
<td>5M</td>
</tr>
<tr>
<td>Yeast</td>
<td>R9.4</td>
<td>49,992</td>
<td>380M</td>
<td>SRR8648503</td>
<td>GCA_000146045.2</td>
<td>12M</td>
</tr>
<tr>
<td>Green Algae</td>
<td>R9.4</td>
<td>63,215</td>
<td>1,335M</td>
<td>ERR3237140</td>
<td>GCF_000002595.2</td>
<td>111M</td>
</tr>
<tr>
<td>Human HG001</td>
<td>R9.4</td>
<td>269,507</td>
<td>1,584M</td>
<td>FAB42260 Nanopore WGS</td>
<td>T2T-CHM13 (v2)</td>
<td>3,117M</td>
</tr>
</tbody>
</table>

**Read Mapping**

<table>
<thead>
<tr>
<th>D1-D5</th>
<th>2,118,047</th>
<th>6,257M</th>
<th>d1-d5</th>
<th>d1-d5</th>
<th>3,246M</th>
</tr>
</thead>
</table>

**Relative Abundance Estimation**

<table>
<thead>
<tr>
<th>D1 and D5</th>
<th>1,651,523</th>
<th>2,178M</th>
<th>d1 and d5</th>
<th>d1</th>
<th>29,903</th>
</tr>
</thead>
</table>

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

Larger Area is Better
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

### Table 2: Numeric Results

<table>
<thead>
<tr>
<th>Tool</th>
<th>d1 SARS-CoV-2</th>
<th>Memory Footprint (GB)</th>
<th>Throughput (bp/s)</th>
<th>Analysis Latency (ms)</th>
<th>Sequencing Latency (Chunks)</th>
<th>Accuracy (F-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncalled</td>
<td>0.280</td>
<td>6,575.310</td>
<td>29.244</td>
<td>0.410</td>
<td>0.972</td>
<td></td>
</tr>
<tr>
<td>Sigmap</td>
<td>28.250</td>
<td>350,565.180</td>
<td>1.111</td>
<td>1.005</td>
<td>0.711</td>
<td></td>
</tr>
<tr>
<td>RawHash</td>
<td>4.210</td>
<td>502,043.190</td>
<td>0.942</td>
<td>1.238</td>
<td>0.925</td>
<td></td>
</tr>
<tr>
<td>RawAlign</td>
<td>4.520</td>
<td>438,089.990</td>
<td>1.070</td>
<td>1.126</td>
<td>0.939</td>
<td></td>
</tr>
<tr>
<td>d2 E.coli</td>
<td>Uncalled</td>
<td>0.800</td>
<td>5,174.050</td>
<td>115.787</td>
<td>1.290</td>
<td>0.973</td>
</tr>
<tr>
<td>Sigmap</td>
<td>111.170</td>
<td>19,215.930</td>
<td>34.441</td>
<td>2.111</td>
<td>0.967</td>
<td></td>
</tr>
<tr>
<td>RawHash</td>
<td>4.270</td>
<td>49,559.740</td>
<td>19.754</td>
<td>3.200</td>
<td>0.928</td>
<td></td>
</tr>
<tr>
<td>RawAlign</td>
<td>0.000</td>
<td>53,693.170</td>
<td>13.323</td>
<td>1.995</td>
<td>0.968</td>
<td></td>
</tr>
<tr>
<td>d3 Yeast</td>
<td>Uncalled</td>
<td>0.580</td>
<td>5,151.670</td>
<td>159.304</td>
<td>2.773</td>
<td>0.941</td>
</tr>
<tr>
<td>Sigmap</td>
<td>14.710</td>
<td>15,217.010</td>
<td>67.602</td>
<td>4.139</td>
<td>0.947</td>
<td></td>
</tr>
<tr>
<td>RawHash</td>
<td>4.530</td>
<td>17,996.930</td>
<td>77.586</td>
<td>5.826</td>
<td>0.906</td>
<td></td>
</tr>
<tr>
<td>RawAlign</td>
<td>4.530</td>
<td>17,854.670</td>
<td>48.394</td>
<td>3.071</td>
<td>0.963</td>
<td></td>
</tr>
<tr>
<td>d4 Green Algae</td>
<td>Uncalled</td>
<td>1.260</td>
<td>8,174.320</td>
<td>440.815</td>
<td>11.790</td>
<td>0.840</td>
</tr>
<tr>
<td>Sigmap</td>
<td>53.710</td>
<td>2,251.370</td>
<td>608.989</td>
<td>5.804</td>
<td>0.938</td>
<td></td>
</tr>
<tr>
<td>RawHash</td>
<td>14.060</td>
<td>5,429.580</td>
<td>700.304</td>
<td>10.646</td>
<td>0.824</td>
<td></td>
</tr>
<tr>
<td>RawAlign</td>
<td>12.200</td>
<td>5,871.450</td>
<td>276.094</td>
<td>4.514</td>
<td>0.932</td>
<td></td>
</tr>
<tr>
<td>d5 Human</td>
<td>Uncalled</td>
<td>13.170</td>
<td>5,612.920</td>
<td>1,077.536</td>
<td>12.959</td>
<td>0.320</td>
</tr>
<tr>
<td>Sigmap</td>
<td>313.400</td>
<td>195.180</td>
<td>16,296.435</td>
<td>10.401</td>
<td>0.327</td>
<td></td>
</tr>
<tr>
<td>RawHash</td>
<td>56.940</td>
<td>1,298.520</td>
<td>6,318.984</td>
<td>10.695</td>
<td>0.557</td>
<td></td>
</tr>
<tr>
<td>RawAlign</td>
<td>80.350</td>
<td>956.310</td>
<td>3,510.682</td>
<td>6.321</td>
<td>0.703</td>
<td></td>
</tr>
<tr>
<td>Contamination</td>
<td>Uncalled</td>
<td>1.060</td>
<td>6,607.850</td>
<td>199.283</td>
<td>3.557</td>
<td>0.964</td>
</tr>
<tr>
<td>Sigmap</td>
<td>111.650</td>
<td>405,956.490</td>
<td>1.206</td>
<td>2.062</td>
<td>0.650</td>
<td></td>
</tr>
<tr>
<td>RawHash</td>
<td>4.280</td>
<td>524,042.570</td>
<td>1.139</td>
<td>2.409</td>
<td>0.872</td>
<td></td>
</tr>
<tr>
<td>RawAlign</td>
<td>4.500</td>
<td>455,376.380</td>
<td>2.004</td>
<td>3.227</td>
<td>0.938</td>
<td></td>
</tr>
<tr>
<td>Relative Abundance</td>
<td>Uncalled</td>
<td>10.870</td>
<td>6,721.770</td>
<td>309.079</td>
<td>4.921</td>
<td>0.218</td>
</tr>
<tr>
<td>Sigmap</td>
<td>506.340</td>
<td>181.880</td>
<td>5,670.365</td>
<td>3.338</td>
<td>0.406</td>
<td></td>
</tr>
<tr>
<td>RawHash</td>
<td>60.760</td>
<td>596.740</td>
<td>2,264.014</td>
<td>3.816</td>
<td>0.459</td>
<td></td>
</tr>
<tr>
<td>RawAlign</td>
<td>83.760</td>
<td>480.050</td>
<td>1,652.162</td>
<td>2.336</td>
<td>0.754</td>
<td></td>
</tr>
</tbody>
</table>

We conclude that RawAlign can be directly used to estimate relative abundances as an end-to-end use case with high accuracy and without the need for basecalling.

### 3.4. Band Width Parameter Sweep

To minimize the computational overhead of alignment, we use banded dynamic time warping (see §2.6.3). If the width of the band is chosen too narrow, accuracy will be degraded, if too wide, performance will suffer. We parameter sweep the band width as a fraction of the length of the read fragment to be aligned on the d2 E.coli dataset, and plot accuracy and throughput. Fig. 5 shows the results. We observe that a band width of 20% of the length of the read fragment is sufficient to achieve the best accuracy (in terms of F-1 score), while simultaneously maximizing throughput. The reason that throughput is also maximized
### Relative Abundance Results

#### Table 2: Numeric Results

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

We conclude that RawAlign can be directly used to estimate relative abundances as an end-to-end use case with high accuracy and without the need for basecalling.

3.4. **Band Width Parameter Sweep**

To minimize the computational overhead of alignment, we use banded dynamic time warping (see §2.6.3). If the width of the band is chosen too narrow, accuracy will be degraded; if too wide, performance will suffer. We parameter sweep the band width as a fraction of the length of the read fragment to be aligned on the d2 E.coli dataset, and plot accuracy and throughput. Fig. 5 shows the results. We observe that a band width of 20% of the length of the read fragment is sufficient to achieve the best accuracy (in terms of F-1 score), while simultaneously maximizing throughput. The reason that throughput is also maximized is...

#### Table 3: Read Ratio Relative Abundances

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

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§
Mohammad Sadrosadati§
Can Firtina§
Nika Mansouri Ghiasi§
Mohammed Alser§
Onur Mutlu§

§ETH Zürich

SAFARI
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 Sadrosadat  Mohammed Alser  Onur Mutlu
P&S Genomics
Lecture 12b: RawAlign

Joël Lindegger
ETH Zürich
Fall 2023
10 January 2024
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**
  
  ![Diagram showing event detection and event value calculation](image)
Practical Similarity Identification

Reference

Read

K-mers

Index (Hash Table)

Seeding

Seed Filtering (e.g., Chaining)

Alignment

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

SAFARI
Existing Solutions – Real-time Basecalling

Deep neural networks (DNNs) for translating \textit{signals} to \textit{bases}

DNNs provide \textbf{less noisy analysis} from basecalled sequences

\textbf{Costly and power-hungry} computational requirements
The Problem

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

Real-time Analysis

<table>
<thead>
<tr>
<th>Basecalling</th>
<th>Read mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Basecalling" /></td>
<td><img src="image2" alt="Read mapping" /></td>
</tr>
</tbody>
</table>

**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*

---

-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
```
Matching Quantized Event Values

-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
```
Slightly Different (Normalized) Event Values

---

SAFARI
### Average Sequenced Bases and Chunks

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

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

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

<table>
<thead>
<tr>
<th>Tool</th>
<th>SARS-CoV-2</th>
<th>E. coli</th>
<th>Yeast</th>
<th>Green Algae</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>File I/O</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Signal-to-Event</td>
<td>21.75</td>
<td>1.86</td>
<td>1.01</td>
<td>0.53</td>
<td>0.02</td>
</tr>
<tr>
<td>Sketching</td>
<td>0.74</td>
<td>0.06</td>
<td>0.04</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>Seeding</td>
<td>3.86</td>
<td>4.14</td>
<td>3.52</td>
<td>6.70</td>
<td>5.39</td>
</tr>
<tr>
<td>Chaining</td>
<td>73.50</td>
<td>93.92</td>
<td>95.42</td>
<td>92.43</td>
<td>94.46</td>
</tr>
<tr>
<td>Seeding + Chaining</td>
<td>77.36</td>
<td>98.06</td>
<td>98.94</td>
<td>99.14</td>
<td>99.86</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Tool</th>
<th>Contamination</th>
<th>SARS-CoV-2</th>
<th>E. coli</th>
<th>Yeast</th>
<th>Green Algae</th>
<th>Human</th>
<th>Relative Abundance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU Time (sec)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNCALLED</td>
<td>265,902.26</td>
<td>36,667.26</td>
<td>35,821.14</td>
<td>8,933.52</td>
<td>16,769.09</td>
<td>262,597.83</td>
<td>586,561.54</td>
</tr>
<tr>
<td>Sigmap</td>
<td>4,573.18</td>
<td>1,997.84</td>
<td>23,894.70</td>
<td>11,168.96</td>
<td>31,544.55</td>
<td>4,837,058.90</td>
<td>11,027,652.91</td>
</tr>
<tr>
<td>RawHash</td>
<td>3,721.62</td>
<td>1,832.56</td>
<td>8,212.17</td>
<td>4,906.70</td>
<td>25,215.23</td>
<td>2,022,521.48</td>
<td>4,738,961.77</td>
</tr>
</tbody>
</table>

| **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.
The mapping step of RawHash is significantly faster than Sigmap for all genomes, and faster than UNCALLED for small genomes.
## Parameter Configurations

<table>
<thead>
<tr>
<th>Tool</th>
<th>Contamination</th>
<th>SARS-CoV-2</th>
<th><em>E. coli</em></th>
<th>Yeast</th>
<th>Green Algae</th>
<th>Human</th>
<th>Relative Abundance</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawHash</td>
<td>-x viral -t 32</td>
<td>-x viral -t 32</td>
<td>-x sensitive -t 32</td>
<td>-x sensitive -t 32</td>
<td>-x fast -t 32</td>
<td>-x fast -t 32</td>
<td>-x fast -t 32</td>
</tr>
<tr>
<td>UNCALLED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>map -t 32</td>
</tr>
<tr>
<td>Simgmap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-m -t 32</td>
</tr>
<tr>
<td>Minimap2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-x map-ont -t 32</td>
</tr>
</tbody>
</table>

### Preset (-x) Corresponding parameters

<table>
<thead>
<tr>
<th>Preset</th>
<th>Corresponding parameters</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>viral</td>
<td>-e 5 -q 9 -l 3</td>
<td>Viral genomes</td>
</tr>
<tr>
<td>sensitive</td>
<td>-e 6 -q 9 -l 3</td>
<td>Small genomes (i.e., &lt; 50M bases)</td>
</tr>
<tr>
<td>fast</td>
<td>-e 7 -q 9 -l 3</td>
<td>Large genomes (i.e., &gt; 50M bases)</td>
</tr>
</tbody>
</table>
# Versions

<table>
<thead>
<tr>
<th>Tool</th>
<th>Version</th>
<th>Link to the Source Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawHash</td>
<td>0.9</td>
<td><a href="https://github.com/CMU-SAFARI/RawHash/tree/8042b1728e352a28fcc79c2efd80c8b631fe7bac">https://github.com/CMU-SAFARI/RawHash/tree/8042b1728e352a28fcc79c2efd80c8b631fe7bac</a></td>
</tr>
<tr>
<td>UNCALLED</td>
<td>2.2</td>
<td><a href="https://github.com/skovaka/UNCALLED/tree/74a5d4e5b5d02fb31d6e88926e8a0896dc3475cb">https://github.com/skovaka/UNCALLED/tree/74a5d4e5b5d02fb31d6e88926e8a0896dc3475cb</a></td>
</tr>
<tr>
<td>Sigmap</td>
<td>0.1</td>
<td><a href="https://github.com/haowenz/sigmap/tree/c9a40483264c9514587a36555b5af48d3f054f6f">https://github.com/haowenz/sigmap/tree/c9a40483264c9514587a36555b5af48d3f054f6f</a></td>
</tr>
</tbody>
</table>
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