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
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 to achieve reduced sequencing time (and cost).
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

Raw Signal Analysis Pipeline

Nanopore Sequencer

Raw Signal

Basecaller

Basecalled Read

...AGTACT...

Basecalled Read Mapper

Raw Signal Mapper

Mapping Locations

Chr. 7 Pos. 4157

Downstream Analysis
Existing Solutions Nanopore Signal Analysis

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

   Real-Time Analysis
   
   Basecalling ➔ Read Mapping
   
   Less noisy analysis from basecalled sequences

   Costly and power-hungry computational requirements

2. Mapping signals to reference genomes without basecalling

   Real-Time Analysis
   
   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

Raw Signal
The Problem – Mapping Raw Signals

Existing solutions are inaccurate or inefficient for large genomes
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

14

RawAlign Overview

1. Seeding
- Anchors

2. Chaining
- Anchors
- Chain

3. Alignment
- Chain
- Alignment

Coarse-Grained Fast

Fine-Grained Accurate
Alignment Algorithms

**Needleman-Wunsch**

Compare Basecalled Sequences

\[
\begin{align*}
dp[i,j] &= \min(dp[i-1,j-1] + \begin{cases} 
0 & \text{if } \text{read}[i] = \text{ref}[j] \\
1 & \text{otherwise}
\end{cases}, \\
& \quad \quad \quad \quad \quad \quad \text{or } \text{unmatched symbol} \\
& \quad \quad \quad \quad \quad \quad \text{or deleted symbol} \\
& \quad \quad \quad \quad \quad \quad \text{or inserted symbol}) \\
& + dp[i,j-1] + 1 \\
& + dp[i-1,j] + 1 \\
\end{align*}
\]

**Dynamic Time Warping**

Compare Raw Signal Sequences

\[
\begin{align*}
\text{Numeric Signal Values} \\
\text{Nucleotide Bases}
\end{align*}
\]
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 \textit{efficiently} integrates \textit{alignment} through

1. Pre-alignment \textit{filtering} (chaining)
2. \textit{Early termination} (branch-and-bound)
3. Anchor-guided alignment
4. Banding/windowing
5. \textit{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§
Mohammad Sadrosadati§
Can Firtina§
Nika Mansouri Ghiasi§
Mohammed Alser§
Onur Mutlu§

§ETH Zürich

SAFARI
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>Read Mapping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d1  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>d2  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>d3  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>d4  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>d5  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>
<tr>
<td>Relative Abundance Estimation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1-D5</td>
<td>2,118,047</td>
<td>6,257M</td>
<td>d1-d5</td>
<td>d1-d5</td>
<td>3,246M</td>
<td></td>
</tr>
<tr>
<td>Contamination Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1 and D5</td>
<td>1,651,523</td>
<td>2,178M</td>
<td>d1 and d5</td>
<td>d1</td>
<td>29,903</td>
<td></td>
</tr>
</tbody>
</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
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

<table>
<thead>
<tr>
<th></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><strong>d1 SARS-CoV-2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uncalled</td>
<td>0.250</td>
<td>6,573,310</td>
<td>29.244</td>
<td>0.410</td>
<td>0.972</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>
</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>
</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>
</tr>
<tr>
<td><strong>d2 E.coli</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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>
</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>
</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>
</tr>
<tr>
<td><strong>d3 Yeast</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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>
</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>
</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>
</tr>
<tr>
<td><strong>d4 Green Algae</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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>608.898</td>
<td>5.804</td>
<td>0.938</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>
</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>
</tr>
<tr>
<td><strong>d5 Human</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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>
</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>
</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>
</tr>
<tr>
<td><strong>Contamination</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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>
</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>
</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>
</tr>
<tr>
<td><strong>Relative Abundance</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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>
</tr>
<tr>
<td>RawHash</td>
<td>60.760</td>
<td>596,740</td>
<td>2,264,014</td>
<td>3.816</td>
<td>0.439</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>
</tr>
</tbody>
</table>
# Relative Abundance 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><code>minimap2</code></td>
<td>0.613</td>
<td>0.163</td>
<td>0.025</td>
<td>0.053</td>
<td>0.147</td>
<td><strong>0.050</strong></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>Simgap</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><strong>RawAlign</strong></td>
<td><strong>0.565</strong></td>
<td><strong>0.248</strong></td>
<td><strong>0.002</strong></td>
<td><strong>0.050</strong></td>
<td><strong>0.136</strong></td>
<td><strong>0.123</strong></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§
Mohammed Alser§
Nika Mansouri Ghiasi§
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 Sadrosadati  Mohammed Alser  Onur Mutlu
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

Reference

Read

K-mers

K-mers

Locations

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*

---

-0.091 in binary:

```
1 1 1 1 0 1 1 0 1 1 ...
```

Most significant \( Q = 9 \) bits:

```
1 0 1 1 1 0 1 1
```

Pruning \( p = 4 \) bits:

```
1 0 0 1 1
```

Matching Quantized Event Values

-0.084 in binary:

```
1 1 1 1 0 1 1 0 1 0 ...
```

Most significant \( Q = 9 \) bits:

```
1 0 1 1 1 0 1 1
```

Pruning \( p = 4 \) bits:

```
1 0 0 1 1
```
## Average Sequenced Bases and Chunks

<table>
<thead>
<tr>
<th>Tool</th>
<th><strong>SARS-CoV-2</strong></th>
<th><strong>E. coli</strong></th>
<th><strong>Yeast</strong></th>
<th><strong>Green Algae</strong></th>
<th><strong>Human</strong></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>Fraction of entire runtime (%)</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>
<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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></td>
</tr>
</tbody>
</table>
## Required Computation Resources in Indexing

<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>UNCALLLED</td>
<td>8.72</td>
<td>9.00</td>
<td>11.08</td>
<td>18.62</td>
<td>285.88</td>
<td>4,148.10</td>
<td>4,382.38</td>
</tr>
<tr>
<td>Sigmap</td>
<td>0.02</td>
<td>0.04</td>
<td>8.66</td>
<td>24.57</td>
<td>449.29</td>
<td>36,765.24</td>
<td>40,926.76</td>
</tr>
<tr>
<td>RawHash</td>
<td>0.18</td>
<td>0.13</td>
<td>2.62</td>
<td>4.48</td>
<td>34.18</td>
<td>1,184.42</td>
<td>788.88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tool</th>
<th>CPU Time (sec)</th>
<th>Real time (sec)</th>
<th>Peak memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNCALLLED</td>
<td>1.01</td>
<td>1.04</td>
<td>0.07</td>
</tr>
<tr>
<td>Sigmap</td>
<td>0.13</td>
<td>0.25</td>
<td>0.01</td>
</tr>
<tr>
<td>RawHash</td>
<td>0.14</td>
<td>0.10</td>
<td>0.01</td>
</tr>
</tbody>
</table>

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

RawHash requires **larger memory space** than UNCALLLED.
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></td>
<td>CPU Time (sec)</td>
<td>Real time (sec)</td>
<td>Peak memory (GB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNCALLED</td>
<td>265,902.26</td>
<td>20,628.57</td>
<td>0.65</td>
<td>0.19</td>
<td>0.52</td>
<td>285.42</td>
<td>19,409.71</td>
</tr>
<tr>
<td>Sigmap</td>
<td>4,573.18</td>
<td>6,725.26</td>
<td>1,544.68</td>
<td>1,544.68</td>
<td>2,067.02</td>
<td>2,138.91</td>
<td>158,904.69</td>
</tr>
<tr>
<td>RawHash</td>
<td>3,721.62</td>
<td>3,917.49</td>
<td>1,949.53</td>
<td>1,949.53</td>
<td>957.13</td>
<td>1,804.96</td>
<td>65,411.43</td>
</tr>
</tbody>
</table>

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>E. coli</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>Sigmoid</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 Usage

<table>
<thead>
<tr>
<th>Preset (-x)</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>
<tr>
<td>Minimap2</td>
<td>2.24</td>
<td><a href="https://github.com/lh3/minimap2/releases/tag/v2.24">https://github.com/lh3/minimap2/releases/tag/v2.24</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

SAFARI