Real-time Analysis of Genomic Sequences from Nanopore Electrical Signals by Fast and Accurate Hash-based Search

Can Firtina
canfirtina@gmail.com
https://cfirtina.com

6 May 2024
The Jackson Laboratory
Brief Self Introduction

- **Can Firtina**
  - Senior Ph.D. student in the SAFARI Research Group at ETH Zurich

- **Research interests:** Bioinformatics & Computer Architecture
  - Real-time genome analysis
  - Similarity search in a large space of genomic data
  - Hardware-Algorithm co-design to accelerate genome analysis
  - Genome editing
  - Error correction

- Get to know our group and our research
  - **Group website:** [https://safari.ethz.ch/](https://safari.ethz.ch/)
  - **Contact me:** canfirtina@gmail.com
  - **Website:** [https://cfirtina.com](https://cfirtina.com)
  - **Twitter (aka X):** [https://twitter.com/FirtinaC](https://twitter.com/FirtinaC)
Professor Mutlu

Onur Mutlu

- Full Professor @ ETH Zurich ITET (INFK), since September 2015
- Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-...
- PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
- [https://people.inf.ethz.ch/omutlu/](https://people.inf.ethz.ch/omutlu/)
- omutlu@gmail.com (Best way to reach)
- [https://people.inf.ethz.ch/omutlu/projects.htm](https://people.inf.ethz.ch/omutlu/projects.htm)

Research and Teaching in:

- Computer architecture, computer systems, hardware security, bioinformatics
- Memory and storage systems
- Hardware security, safety, predictability
- Fault tolerance
- Hardware/software cooperation
- Architectures for bioinformatics, health, medicine
- ...

...
SAFARI Research Group

Computer architecture, HW/SW, systems, bioinformatics, security, memory

40+ Researchers

Think BIG, Aim HIGH!

https://safari.ethz.ch
Four Key Current Directions

- Fundamentally Secure/Reliable/Safe Architectures

- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures

- Fundamentally Low-Latency and Predictable Architectures

- Algorithms & Architectures for AI/ML, Genomics, Medicine
Agenda for Today

- Cutting-edge in Accelerating Genome Analysis

- Enabling Fast and Accurate Real-time Analysis
  - RawHash and RawHash2

- Conclusion
“The purpose of computing is [to gain] insight, not numbers”

Richard Hamming

We need to gain insights and observations much more efficiently than ever before
Big Data is Everywhere

- **Astronomy**: 25 zetta-bytes/year
- **Twitter (now X)**: 0.5-15 billion tweets/year
- **YouTube**: 500-900 million hours/year
- **Genomics**: 1 zetta-bases/year

Problems with Data Analysis Today

**Special-Purpose Machine** for Data Generation

**General-Purpose Machine** for Data Analysis

**FAST**

**SLOW**

Slow and inefficient processing capability
Large amounts of data movement
Data Movement Dominates Performance

- **Data movement** dominates performance and is a **major** system **energy bottleneck** (accounting for 40%-62%)

Single memory request *consumes* >160x-800x more energy compared to performing an addition operation

---

* Boroumand et al., “Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks,” ASPLOS 2018
* Kestor et al., “Quantifying the Energy Cost of Data Movement in Scientific Applications,” IISWC 2013
* Pandiyan and Wu, “Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms,” IISWC 2014
We need **intelligent algorithms** and **intelligent architectures** that **handle data well**
Pushing Towards New Architectures

Modern systems

FPGAs

Heterogeneous Processors and Accelerators

Hybrid Main Memory

(General Purpose) GPUs

Sequencing Machine

Persistent Memory/Storage
Pushing Towards New Architectures

Modern systems

FPGAs

Heterogeneous Processors and Accelerators

(General Purpose) GPUs

Persistent Memory/Storage

SAFARI

https://nanoporetech.com/products/smidgion
Algorithm-Arch-Device Co-Design is Critical
Onur Mutlu and Can Firtina,

"Accelerating Genome Analysis via Algorithm-Architecture Co-Design"


[Slides (pptx) (pdf)]
[Talk Video (38 minutes, including Q&A)]
[Related Invited Paper]
[arXiv version]
Applications are only limited by our imagination
Genome Editing

The Nobel Prize in Chemistry 2020
awarded "for the development of a method of genome editing"

DNA Computing

Massive parallelism to solve (hard) problems!
Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

Damla Senol Cali, Jeremie S Kim, Saugata Ghose, Can Alkan, Onur Mutlu

Briefings in Bioinformatics, bby017, https://doi.org/10.1093/bib/bby017
Published: 02 April 2018 Article history ▼

[Open arxiv.org version]
New Frontiers: Raw Signal Analysis [ISMB 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

[Bioinformatics Journal version]
[Slides (pptx) (pdf)]
[RawHash Source Code]

Bioinformatics, 2023, 39, i297–i307
https://doi.org/10.1093/bioinformatics/btad272

ISMB/ECCB 2023

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,*

1Department 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.)
Fast and Accurate Real-Time Genome Analysis

- Can Firtina, Melina Soysal, Joel Lindegger, and Onur Mutlu,
  "RawHash2: Mapping Raw Nanopore Signals Using Hash-Based Seeding and Adaptive Quantization"
  [arXiv version]
  [RawHash2 Source Code]

RawHash2: Mapping Raw Nanopore Signals Using Hash-Based Seeding and Adaptive Quantization

Can Firtina  Melina Soysal  Joël Lindegger  Onur Mutlu
ETH Zürich
Fast and Accurate Real-Time Genome Analysis

- Joel Lindegger, Can Firtina, Nika Mansouri Ghiasi, Mohammad Sadrosadati, Mohammed Alser, and Onur Mutlu,

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

Preprint on arXiv, October 2023.

[arXiv version]
[RawAlign Source Code]

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
M. Banu Cavlak, Gagandeep Singh, Mohammed Alser, Can Firtina, Joel Lindegger, Mohammad Sadrosadati, Nika Mansouri Ghiasi, Can Alkan, and Onur Mutlu, "TargetCall: Eliminating the Wasted Computation in Basecalling via Pre-Basecalling Filtering" 
Proceedings of the 21st Asia Pacific Bioinformatics Conference (APBC), Changsha, China, April 2023. 
[TargetCall Source Code] 
[arxiv.org Version] 
[Talk Video at BIO-Arch 2023 Workshop] 

TargetCall: Eliminating the Wasted Computation in Basecalling via Pre-Basecalling Filtering 
Meryem Banu Cavlak\textsuperscript{1} Gagandeep Singh\textsuperscript{1} Mohammed Alser\textsuperscript{1} Can Firtina\textsuperscript{1} Joël Lindegger\textsuperscript{1} 
Mohammad Sadrosadati\textsuperscript{1} Nika Mansouri Ghiasi\textsuperscript{1} Can Alkan\textsuperscript{2} Onur Mutlu\textsuperscript{1} 
\textsuperscript{1}ETH Zürich \hspace{1cm} \textsuperscript{2}Bilkent University

Can Firtina, Jisung Park, Mohammed Alser, Jeremie S. Kim, Damla Senol Cali, Taha Shahroodi, Nika Mansouri Ghiasi, Gagandeep Singh, Konstantinos Kanellopoulos, Can Alkan, and Onur Mutlu,

"BLEND: A Fast, Memory-Efficient, and Accurate Mechanism to Find Fuzzy Seed Matches in Genome Analysis"


[Online link at NAR Genomics and Bioinformatics Journal]
[arXiv preprint]
[biorXiv preprint]
[BLEND Source Code]
New Applications: Frequent Database Updates

  [AirLift Source Code]
  [arxiv.org Version (pdf)]
  [Talk Video at BIO-Arch 2023 Workshop]

**METHOD**

AirLift: A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes

Jeremie S. Kim†, Can Firtina†, Meryem Banu Cavlak², Damla Senol Cali³, Nastaran Hajinazar¹,⁴, Mohammed Alser¹, Can Alkan² and Onur Mutlu¹,²,³*

SAFARI *Equal contribution
Error Correction using ML [Bioinform. 2020]

- Can Firtina, Jeremie S. Kim, Mohammed Alser, Damla Senol Cali, A. Ercument Cicek, Can Alkan, and Onur Mutlu,
  "Apollo: A Sequencing-Technology-Independent, Scalable, and Accurate Assembly Polishing Algorithm"
  Bioinformatics, June 2020.
  [Source Code]
  [Online link at Bioinformatics Journal]

Apollo: a sequencing-technology-independent, scalable and accurate assembly polishing algorithm

Can Firtina, Jeremie S Kim, Mohammed Alser, Damla Senol Cali, A Ercument Cicek, Can Alkan ✉, Onur Mutlu ✉️

Bioinformatics, Volume 36, Issue 12, 15 June 2020, Pages 3669–3679,
https://doi.org/10.1093/bioinformatics/btaa179

Published: 13 March 2020  Article history ▼
Can Firtina, Kamlesh Pillai, Gurpreet S. Kalsi, Bharathwaj Suresh, Damla Senol Cali, Jeremie S. Kim, Taha Shahroodi, Meryem Banu Cavlak, Joël Lindegger, Mohammed Alser, Juan Gómez Luna, Sreenivas Subramoney, and Onur Mutlu,


*ACM TACO*, Feb 2024.

[Online link at ACM TACO]

[arXiv preprint]

[ApHMM Source Code]
Accelerating String Matching [MICRO 2020]

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,

"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"


[Lightning Talk Video (1.5 minutes)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (18 minutes)]
[Slides (pptx) (pdf)]

---

**GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis**

Damla Senol Cali†‡☑, Gurpreet S. Kalsi†‡☑, Züal Bingöl☐, Can Firtina☐, Lavanya Subramanian☐†, Jeremie S. Kim☐‡, Rachata Ausavarungnirun☐, Mohammed Alser☐, Juan Gomez-Luna☐, Amirali Boroumand☐†, Anant Nori☐, Allison Scibisz☐†, Sreenivas Subramoney☐†, Can Alkan☐, Saugata Ghose☐†, Onur Mutlu☐‡

†Carnegie Mellon University ☑Processor Architecture Research Lab, Intel Labs ☐Bilkent University ☞ETH Zürich
☐Facebook ☜King Mongkut’s University of Technology North Bangkok ☠University of Illinois at Urbana–Champaign

SAFARI
In-Storage Genome Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alser, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,

"GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"


[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]

GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi¹  Jisung Park¹  Harun Mustafa¹  Jeremie Kim¹  Ataberk Olgun¹
Arvid Gollwitzer¹  Damla Senol Cali²  Can Firtina¹  Haiyu Mao¹  Nour Almadhoun Alser¹
Rachata Ausavarungnirun³  Nandita Vijaykumar⁴  Mohammed Alser¹  Onur Mutlu¹

¹ETH Zürich  ²Bionano Genomics  ³KMUTNB  ⁴University of Toronto
Genome Analysis via PIM [MICRO 2022]

- Haiyu Mao, Mohammed Alser, Mohammad Sadrosadati, Can Firtina, Akanksha Baranwal, Damla Senol Cali, Aditya Manglik, Nour Almadhoun Alserr, and Onur Mutlu,

"GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping"

Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.

[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Lecture Video (25 minutes)]
[arXiv version]

GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping

Haiyu Mao¹  Mohammed Alser¹  Mohammad Sadrosadati¹  Can Firtina¹  Akanksha Baranwal¹  Damla Senol Cali²  Aditya Manglik¹  Nour Almadhoun Alserr¹  Onur Mutlu¹

¹ETH Zürich  ²Bionano Genomics

Taha Shahroodi, Gagandeep Singh, Mahdi Zahedi, Haiyu Mao, Joel Lindeger, Can Firtina, Stephan Wong, Onur Mutlu, and Said Hamdioui,

"Swordfish: A Framework for Evaluating Deep Neural Network-based Basecalling using Computation-In-Memory with Non-Ideal Memristors"

Proceedings of the 56th International Symposium on Microarchitecture (MICRO), Toronto, ON, Canada, November 2023.

[Slides (pptx) (pdf)]
[arXiv version]
Agenda for Today

- Cutting-edge in Accelerating Genome Analysis

- Enabling Fast and Accurate Real-time Analysis
  - RawHash and RawHash2

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

Background

RawHash

RawHash2

Evaluation

Conclusion
Different Raw Sequencing Data

**Illumina**
- Multiple images
- .BCL/.CBCL

**Nanopore**
- Electrical Signal
- .POD5

**PacBio**
- 30-hour movie
- .BAM
Nanopore Sequencing: a widely used sequencing technology

- **Long** reads (up to >2 million nucleotides)
- **Portable** sequencing
- Cost-effective
- More unique features: **real-time analysis**
Nanopore Sequencing & Real-time Analysis

**Raw Signals:** Ionic current measurements generated as DNA passes through the nanopore at a certain speed

**(Real-Time) Analysis:** Translating to bases or directly analyzing raw signals

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

Completely Sequenced Read

Partially Sequenced Read

Reduced Sequencing Time (and Cost)
Challenges in Real-Time Analysis

- **Rapid analysis** to match the nanopore sequencer throughput
- **Timely decisions** to stop sequencing as early as possible
- **Accurate analysis** from noisy raw signal data
- **Power-efficient** computation for scalability and portability
Executive Summary

Problem: Real-time analysis of nanopore raw signals is inaccurate and inefficient for large genomes

Goal: Enable fast and accurate real-time analysis of raw nanopore signals

Key Contributions:
1) The first hash-based mechanism for mapping raw nanopore signals
2) The novel Sequence Until technique can accurately and dynamically stop the entire sequencing of all reads at once if further sequencing is not necessary

Key Results: Across 3 use cases and 5 genomes of varying sizes
- 27× 19×, and 4× better average throughput compared to the state-of-the-art works
- Most accurate raw signal mapper for all datasets
- Sequence Until reduces the sequencing time and cost by 15×
Existing Solutions

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

   - **Less noisy analysis from basecalled sequences**
   - **Costly and power-hungry computational requirements**

2. Mapping signals to reference genomes without basecalling

   - **Raw signals contain richer information than bases**
   - **Efficient analysis with better scalability and portability**
Enabling Analysis From Electrical Signals

- **K** many nucleotides (k-mers) sequenced at a time
- **Event:** A **segment** of the raw signal
  - Corresponds to a **particular** k-mer
  - Abrupt signal changes show sequencing of a new k-mer
  - **Statistical methods** can find these abrupt changes
  - **Event value:** average of signals **within an event**

- **Observation:** **Identical** k-mers generate **similar** event values during sequencing
The Problem – Mapping Raw Signals

**Raw Signal**

- **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

- Raw Signal

Existing solutions are **inaccurate or inefficient for large genomes**
Enable **fast and accurate real-time analysis** of raw nanopore signals **for large genomes**
The **first hash-based search mechanism** to quickly and accurately map raw nanopore signals to reference genomes

**Sequence Until** can accurately and **dynamically stop** the entire sequencing run at once if further sequencing is unnecessary
The **first hash-based search mechanism** to quickly and accurately map raw nanopore signals to reference genomes.

Sequence Until can accurately and **dynamically stop** the entire sequencing run at once if further sequencing is unnecessary.
RawHash – Key Idea

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

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

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

SAFARI
RawHash Overview

1. Reference-to-Event Conversion
   - Reference Genome
     - GCTATTACCTTAATGTG...
   - Signal-to-Event Conversion
     - Raw Nanopore Signal
   - 2.21 - 0.9 - 1.15

2. Quantization
   - 28 - 6 - 18

3. Hashing
   - Hashing
   - 0x01
   - Store
   - Query
   - Hash Table

4. Matching Regions
   - Chaining & Mapping
   - Mapping Positions

Indexing (Offline)
Mapping (Real-Time)
RawHash Overview

Reference Genome

...GCTATTACCTTAATGTG...

Reference-to-Event Conversion

2.21 -0.9 1.15

Raw Nanopore Signal

Signal-to-Event Conversion

2.22 -0.91 1.18
Reference-to-Event Conversion

- **K-mer model:** Provides **expected** event values for each k-mer
  - Preconstructed based on nanopore sequencer characteristics

- Use the **k-mer model** to convert all k-mers of a reference genome to their **expected** event values

<table>
<thead>
<tr>
<th>Reference Genome</th>
<th>Expected Event Values</th>
<th>Normalized Event Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>..GCTATTACC..</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCTATT</td>
<td>105.757390</td>
<td>2.21</td>
</tr>
<tr>
<td>CTATTA</td>
<td>81.740642</td>
<td>-0.09</td>
</tr>
<tr>
<td>TATTAC</td>
<td>103.170091</td>
<td>1.15</td>
</tr>
<tr>
<td>ATTACC</td>
<td>101.082485</td>
<td>1.11</td>
</tr>
</tbody>
</table>

**K-mer Model (Lookup Table)**
Signal-to-Event Conversion

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

- Consecutive events ➔ consecutive k-mers
Signal-to-Event Conversion

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

Can we directly match signals to each other?

- Consecutive events → consecutive k-mers
RawHash Overview

Reference Genome

...GCTATTACCTTAATGTG...

Reference-to-Event Conversion

1

2.21 -0.9 1.15

Quantization

2

28 6 18

Raw Nanopore Signal

Signal-to-Event Conversion

2.22 -0.91 1.18

Quantization

28 6 18
Quantizing the Event Values

- **Observation:** Slight differences in raw signals from identical k-mers
  - **Challenge:** Direct event value matching is not feasible and accurate

- **Key Idea:** Quantize the event values
  - Enables assigning **identical quantized values** to similar event values

**Diagram:**

- **Normalized event values from the same k-mer**
  - CTATTA
  - -0.091
  - -0.084
  - -0.09
  - -0.086

- **Quantized event values (in binary)**
  - 11001
  - 11001
  - 11001
  - 11001
RawHash Overview

1. Reference-to-Event Conversion
   - Reference Genome
     - GCTATTACCTTAATGTG...
   - Nanopore Raw Signal
     - Signal-to-Event Conversion
     - 2.22 -0.91 1.18
   - 2.21 -0.9 1.15

2. Quantization
   - 28 6 18
   - 28 6 18

3. Hashing
   - Hash Table
     - Store 0x01
     - Query 0x01
Hashing for Fast Similarity Search

- Each event usually represents a very small k-mer (6 to 9 characters)
  - **Challenge**: Short k-mers are likely to appear in many locations

- **Key Idea**: Create longer k-mers from many consecutive events
- **Key Benefit**: Directly match hash values to quickly identify similarities

```
<table>
<thead>
<tr>
<th>Consecutive k-mers</th>
<th>Consecutive events</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTATTA</td>
<td>-0.09</td>
</tr>
<tr>
<td>TATTAC</td>
<td>1.15</td>
</tr>
<tr>
<td>ATTACC</td>
<td>1.11</td>
</tr>
</tbody>
</table>
```

- **Quantize**
  - CTATTA: 0.09
  - TATTAC: 1.15
  - ATTACC: 1.11
- **Quantize**
  - CTATTA: 11001
  - TATTAC: 00110
  - ATTACC: 00101
- **Pack**
  - 1100100110 ...
- **Hash**
  - 0x400D70A4

Hash value of consecutive events
RawHash Overview

**Reference Genome**

...GCTATTACCTTAATGTG...

1. Reference-to-Event Conversion

   - 2.21
   - -0.9
   - 1.15

2. Quantization

   - 28
   - 6
   - 18

3. Hashing

   - Store
     - 0x01

   - Query
     - 0x01

4. Matching Regions

   - Chaining & Mapping

   - Mapping Positions

**Raw Nanopore Signal**

1. Signal-to-Event Conversion

   - 2.22
   - -0.91
   - 1.18

2. Quantization

   - 28
   - 6
   - 18
Real-Time Mapping using Hash-based Indexing

**Indexing (Offline)**
- Reference Genome
  - Reference-to-Event Conversion
  - Quantization
  - Hashing
  - Store: Hash Table
- Matching Positions

**Mapping (Real-time)**
- Raw Nanopore Signal
  - Signal-to-Event Conversion
  - Quantization
  - Hashing
- Query: 0x01
- Chaining & Mapping
- Continue Mapping?

**Read Until or Run Until**
- Yes: Process the next chunk
- No: Stop mapping
The **first hash-based search mechanism** to quickly and accurately map raw nanopore signals to reference genomes.

Sequence Until can accurately and dynamically stop the entire sequencing run at once if further sequencing is unnecessary.
The first hash-based search mechanism to quickly and accurately map raw nanopore signals to reference genomes.

**Sequence Until** can accurately and **dynamically stop** the entire sequencing run at once if further sequencing is unnecessary.
The Sequence Until Mechanism

- **Problem:**
  - Unnecessary sequencing waste time, power and money

- **Key Idea:**
  - **Dynamically** decide if further sequencing of the entire sample is necessary to achieve high accuracy
  - Stop sequencing early without sacrificing accuracy

- **Potential Benefits:**
  - Significant *reduction in sequencing time and cost*

- Example real-time genome analysis use case:
  - Relative abundance estimation
The Sequence Until Mechanism

- **Key Steps:**
  1. Continuously generate relative abundance estimation after every $n$ reads
  2. Keep the last $t$ estimation results
  3. **Detect outliers** in the results via **cross-correlation** of the recent $t$ results
  4. Absence of outliers indicates **consistent results**
     - Further sequencing **is likely** to generate consistent results ➔ Stop the sequencing
Outline

Background

RawHash

RawHash2

Evaluation

Conclusion
Key Contributions in RawHash2

- **A new adaptive quantization** to better fit the expected nanopore signal pattern to achieve high accuracy

- **Improved chaining algorithm** with sensitive penalty scores

- **Weighted decision making** for more robust mapping

- **Frequency filter** and **minimizer sketching** to reduce seed matches for faster and space-efficient mapping
Sketching with Hash-based Indexing

### Indexing (Offline)

- **Reference Genome**
  - `...GCTATTACCTTAATGTG...`
- **Reference-to-Event Conversion**
- **Quantization**
- **Hashing**
- **Sketch**
- **Store**
  - 0x01
  - **Hash Table**
- **Matching Positions**

**All k-mers, Minimizers, Strobemers, BLEND, …**

### Mapping (Real-time)

- **Raw Nanopore Signal**
- **Signal-to-Event Conversion**
- **Quantization**
- **Hashing**
- **Sketch**
- **Query**
  - 0x01
- **Chaining & Mapping**
- **Continue Mapping?**

**Yes: Process the next chunk**

**No: Stop mapping**

**Continue Mapping?**

**All k-mers, Minimizers, Strobemers, BLEND, …**
Evaluation Methodology

- Two settings for RawHash2:
  - **RawHash2**: All hash values without sampling
  - **RawHash2-Minimizer**: Minimizer sketching

- Compared to **UNCALLED** [Kovaka+, Nat. Biotech. '21], **Sigmap** [Zhang+, ISMB '21] and **RawHash** [Firtina+, ISMB '23]

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

- Evaluation metrics:
  - **Throughput** (bases processed per second per CPU thread)
  - Potential reduction in **sequencing time and cost**

- **Accuracy**
  - **Baseline**: Mapping basecalled reads using minimap2
  - Precision, recall, and F1 scores
  - Relative abundance estimation distance to ground truth

- **Datasets:**

<table>
<thead>
<tr>
<th>Organism</th>
<th>Reads (#)</th>
<th>Bases (#)</th>
<th>Genome Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Read Mapping</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1 SARS-CoV-2</td>
<td>1,382,016</td>
<td>594M</td>
<td>29,903</td>
</tr>
<tr>
<td>D2 E. coli</td>
<td>353,317</td>
<td>2,365M</td>
<td>5M</td>
</tr>
<tr>
<td>D3 Yeast</td>
<td>49,989</td>
<td>380M</td>
<td>12M</td>
</tr>
<tr>
<td>D4 Green Algae</td>
<td>29,933</td>
<td>609M</td>
<td>111M</td>
</tr>
<tr>
<td>D5 Human HG001</td>
<td>269,507</td>
<td>1,584M</td>
<td>3,117M</td>
</tr>
</tbody>
</table>

  |                |           |           |             |
  | **Relative Abundance Estimation** |           |           |             |
  | D1-D5          | 2,084,762 | 5,531M    | 3,246M      |

  |                |           |           |             |
  | **Contamination Analysis** |           |           |             |
  | D1 and D5      | 1,651,523 | 2,178M    | 29,903      |
Throughput

- **Real-time analysis requires** faster throughput than sequencer
  - Throughput from a single pore: \(~450\ \text{bp/sec (data generation speed)}\)

**RawHash2**: \(27\times, 19\times, \text{and } 4\times\) better average throughput compared to **UNCALLED, Sigmap** and **RawHash**, respectively

**RawHash2-Minimizer** further improves the throughput by \(2.5\times\) compared to **RawHash2**
Average Sequenced Length

- Fewer bases to sequence ➔ Less unnecessary sequencing

RawHash2 reduces sequencing time and cost
on average by $1.9 \times$ compared to UNCALLLED and RawHash

RawHash2 leads to sequencing the least amount of bases
for larger genomes
Accuracy

- Read mapping, contamination, and relative abundance estimation accuracy (baseline: basecalled mapping)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>RH2</th>
<th>RH2-Min.</th>
<th>RH</th>
<th>UNCALLED</th>
<th>Sigmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>SARS-CoV-2</td>
<td>F1</td>
<td>0.9867</td>
<td>0.9691</td>
<td>0.9252</td>
<td>0.9725</td>
<td>0.7112</td>
</tr>
<tr>
<td>E. coli</td>
<td>F1</td>
<td>0.9748</td>
<td>0.9631</td>
<td>0.9280</td>
<td>0.9731</td>
<td>0.9670</td>
</tr>
<tr>
<td>Yeast</td>
<td>F1</td>
<td>0.9602</td>
<td>0.9472</td>
<td>0.9060</td>
<td>0.9407</td>
<td>0.9469</td>
</tr>
<tr>
<td>Green Algae</td>
<td>F1</td>
<td>0.9351</td>
<td>0.9191</td>
<td>0.8114</td>
<td>0.8277</td>
<td>0.9350</td>
</tr>
<tr>
<td>Human</td>
<td>F1</td>
<td>0.7599</td>
<td>0.6699</td>
<td>0.5574</td>
<td>0.3197</td>
<td>0.3269</td>
</tr>
<tr>
<td>Contamination</td>
<td>Precision</td>
<td>0.9595</td>
<td>0.9424</td>
<td>0.8702</td>
<td>0.9378</td>
<td>0.7856</td>
</tr>
<tr>
<td>Rel. Abundance</td>
<td>Distance</td>
<td>0.2678</td>
<td>0.4243</td>
<td>0.4385</td>
<td>0.6812</td>
<td>0.5430</td>
</tr>
</tbody>
</table>

Best results are **highlighted**.

RawHash2 provides the **most accurate read mapping**

RawHash2-Minimizer provides an **on-par accuracy with RawHash2** while improving **the throughput substantially**
Benefits of Sequence Until

- Running RawHash with and without Sequence Until

<table>
<thead>
<tr>
<th>Tool</th>
<th>Estimated Relative Abundance Ratios in 50,000 Random Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SARS-CoV-2</td>
</tr>
<tr>
<td>RawHash (100%)</td>
<td>0.0270</td>
</tr>
<tr>
<td>RawHash + Sequence Until (7%)</td>
<td>0.0283</td>
</tr>
</tbody>
</table>

Sequence Until enables sequencing only 7% (~1/15) of the entire sample with high accuracy.

UNCALLED and RawHash benefit from Sequence Until significantly by up to 100× reductions in sequencing.
Conclusion

Key Contributions:
1) The first hash-based mechanism for mapping raw nanopore signals
2) The novel Sequence Until technique can accurately and dynamically stop the entire sequencing of all reads at once if further sequencing is not necessary

Key Results: Across 3 use cases and 5 genomes of varying sizes
- 27x 19x, and 4x better average throughput compared to the state-of-the-art works
- Most accurate raw signal mapper for all datasets
- Sequence Until reduces the sequencing time and cost by 15x

Many opportunities for analyzing raw nanopore signals in real-time:
- Many hash-based sketching techniques can now be used for raw signals
- Indexing is very cheap: Many future use cases with the on-the-fly index construction
- We should rethink the algorithms to perform downstream analysis fully using raw signals
RawHash

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

Can Firtina

Nika Mansouri Ghiasi Joel Lindegerg Gagandeep Singh
Meryem Banu Cavlak Haiyu Mao Onur Mutlu

RawHash RawHash2 Code

SAFARI ETH Zürich
Agenda for Today

- Cutting-edge in Accelerating Genome Analysis

- Enabling Fast and Accurate Real-time Analysis
  - RawHash and RawHash2

- Conclusion
Future is Bright
for Raw Signal Analysis
Raw Signal Alignment Coupled with Mapping

Joel Lindegger, Can Firtina, Nika Mansouri Ghiasi, Mohammad Sadrosadati, Mohammed Alser, and Onur Mutlu,
"RawAlign: Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment"
Preprint on arXiv, October 2023.
[arXiv version]
[RawAlign Source Code]

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
Rawsamble: Overlapping and Assembling Raw Nanopore Signals using a Hash-based Seeding Mechanism

Can Firtina¹  Maximilian Mordig¹,²  Joël Lindegger¹  Harun Mustafa¹,³,⁴  Sayan Goswami¹  Stefano Mercogliano¹  Yan Zhu¹,⁵  Andre Kahles¹,³,⁴  Onur Mutlu¹

¹ETH Zurich  ²Max Planck Institute for Intelligent Systems  ³University Hospital Zurich  ⁴Swiss Institute of Bioinformatics  ⁵University of Toronto
Real-time \textit{de novo} Assembly Construction

- All-vs-all overlapping using raw signals

<table>
<thead>
<tr>
<th>Organism</th>
<th>Shared Overlaps (%)</th>
<th>Unique to Rawsamble (%)</th>
<th>Unique to Minimap2 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1 E. coli</td>
<td>44.57</td>
<td>15.08</td>
<td>40.35</td>
</tr>
<tr>
<td>D2 Yeast</td>
<td>47.07</td>
<td>35.62</td>
<td>17.32</td>
</tr>
<tr>
<td>D3 Human</td>
<td>19.73</td>
<td>27.56</td>
<td>52.71</td>
</tr>
</tbody>
</table>

- Building \textit{de novo} assemblies directly from raw signals

<table>
<thead>
<tr>
<th>Organism</th>
<th>Tool</th>
<th>No. of contigs</th>
<th>Avg. Contig Length</th>
<th>Max. Contig Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1 E. coli</td>
<td>Rawsamble</td>
<td>39</td>
<td>373,594</td>
<td>1,431,572</td>
</tr>
<tr>
<td></td>
<td>minimap2</td>
<td>4</td>
<td>2,611,044</td>
<td>5,210,938</td>
</tr>
<tr>
<td>D2 Yeast</td>
<td>Rawsamble</td>
<td>431</td>
<td>47,250</td>
<td>256,116</td>
</tr>
<tr>
<td></td>
<td>minimap2</td>
<td>278</td>
<td>82,757</td>
<td>386,005</td>
</tr>
<tr>
<td>D3 Human</td>
<td>Rawsamble</td>
<td>59</td>
<td>16,376</td>
<td>66,163</td>
</tr>
<tr>
<td></td>
<td>minimap2</td>
<td>53</td>
<td>10,572</td>
<td>42,654</td>
</tr>
</tbody>
</table>
Storage-Centric Design for Raw Signal Analysis

RawGAINS: A Heterogeneous Storage-Centric Processing System for Raw Signal Genome Analysis
Opportunities for New Applications

- Improving the basecalling accuracy using the overlapping information between signals
- Full downstream analysis fully using raw nanopore signals
- Cooperating the raw signal analysis and basecalled sequence analysis together
- Many, many more keeping the hardware design in mind
Things Are Happening In Industry
Illumina DRAGEN Bio-IT Platform (2018)

- Processes whole genome at 34x coverage in ~30 minutes with hardware support for data compression

[Image of FPGA board(s)]

ema.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html
Nova/NextSeq with Analysis Capability

Scale your studies with ease

Process high-throughput data quickly with hardware acceleration.

With four field-programmable gate arrays (FPGAs) onboard you have the most powerful DRAGEN analysis ever, enabling you to process NovaSeq X System data easily. Perform up to four simultaneous applications per flow cell in a single run.

Reduce data footprint, manage and store data easily with lower costs and lower energy consumption, with built-in compression that reduces FASTQ file sizes by up to 80%.

Stream data directly to Illumina Connected Analytics or BaseSpace Sequence Hub on the cloud for scalable data management, analysis, and aggregation.


NVIDIA Clara Parabricks (2020)

A University of Michigan startup in 2018 joined NVIDIA in 2020

GPU board(s)

PERFORMANCE COMPARISON
Germline End-to-End Secondary Analysis

1,200 minutes

<table>
<thead>
<tr>
<th></th>
<th>CPU/GATK</th>
<th>8X T4</th>
<th>8X V100</th>
<th>8X A100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>52 minutes</td>
<td>35 minutes</td>
<td>23 minutes</td>
<td></td>
</tr>
</tbody>
</table>
NVIDIA is claiming a 7x improvement in dynamic programming algorithm (DPX instructions) performance on a single H100 versus naïve execution on an A100.

We are accelerating the transformation in how we analyze the human genome!

Bionano & NVIDIA: Accelerating Analysis for Fast Time to Results

- Technological solution to support higher throughput
- New high-performance algorithms from Bionano
- Powered by NVIDIA RTX™ 6000 Ada Generation GPUs
- Analysis of highly complex cancer whole genomes in less than 2 hours
- Workflow tailored for a small lab and IT footprint
Cerebras’s Wafer Scale Engine (2021)

- The largest ML accelerator chip (2021)
- 850,000 cores

Cerebras WSE-2
2.6 Trillion transistors
46,225 mm²

Largest GPU
54.2 Billion transistors
826 mm²
NVIDIA Ampere GA100

https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning
https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/
UPMEM Processing-in-DRAM Engine (2019)

- **Processing in DRAM Engine**
- Includes **standard DIMM modules**, with a large number of **DPU processors** combined with DRAM chips.

- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth

---


Onur Mutlu, Computer Architecture Lecture 2b, Fall 2019, ETH Zurich
The vision of BioPIM is the realization of *cheap, ultra-fast and ultra-low energy mobile genomics* that eliminates the current dependence of sequence analysis on large and power-hungry computing clusters/data-centers.
Fast Genome Analysis…

- Onur Mutlu,
  "Accelerating Genome Analysis: A Primer on an Ongoing Journey"
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 37 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]
More on Fast Genome Analysis…

- Onur Mutlu, 
  "Accelerating Genome Analysis"
  Invited Talk at the Barcelona Supercomputing Center (BSC), Barcelona, Spain, 6 September 2022.
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 35 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]
  [Related Invited Paper (at Computational and Structural Biology Journal, 2022)]
More on Accelerating Genome Analysis

- Can Firtina,
  "Enabling Accurate, Fast, and Memory-Efficient Genome Analysis via Efficient and Intelligent Algorithms"
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 6 minutes)]
More on Real-Time Genome Analysis

- Can Firtina,
  "RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes"
  *Proceedings Talk at ISMB-ECCB, Lyon, France, 25 July 2023.*
  [Slides (pptx) (pdf)]
  [Talk Video (18 minutes)]
Accelerating Genome Analysis [DAC 2023]

Onur Mutlu and Can Firtina,
"Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
[Slides (pptx) (pdf)]
[Talk Video (38 minutes, including Q&A)]
[Related Invited Paper]
[arXiv version]

Accelerating Genome Analysis via Algorithm-Architecture Co-Design

Onur Mutlu    Can Firtina
ETH Zürich
BIO-Arch Workshop at RECOMB 2023

April 14, 2023

BIO-Arch: Workshop on Hardware Acceleration of Bioinformatics Workloads

About

BIO-Arch is a new forum for presenting and discussing new ideas in accelerating bioinformatics workloads with the co-design of hardware & software and the use of new computer architectures. Our goal is to discuss new system designs tailored for bioinformatics. BIO-Arch aims to bring together researchers in the bioinformatics, computational biology, and computer architecture communities to strengthen the progress in accelerating bioinformatics analysis (e.g., genome analysis) with efficient system designs that include hardware acceleration and software systems tailored for new hardware technologies.

Venue

BIO-Arch will be held in The Social Facilities of Istanbul Technical University on April 14. Detailed information about how to arrive at the venue location with various transportation options can be found on the RECOMB website.

Our panel discussion will be held in conjunction with the main RECOMB conference. The panel discussion will be held in Marriott Şişli on April 17 at 17:00. You can find

https://www.youtube.com/watch?v=2rCsb4-nLmg
https://safari.ethz.ch/recomb23-arch-workshop/
Genomics Course (Fall 2023)

- **Fall 2023 Edition:**

- **Spring 2023 Edition:**

- **Youtube Livestream (Fall 2023):**
  - [https://youtube.com/playlist?list=PL5Q2soXY2Zi_O0wyOjiMShG4t2QPZoeE3](https://youtube.com/playlist?list=PL5Q2soXY2Zi_O0wyOjiMShG4t2QPZoeE3)

- Project course
  - Taken by Bachelor’s students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Conclusion

- We covered various recent ideas to
  - Accelerate genome analysis
  - Analyze genomes in ways that were not possible before

- Enabling cost-effective, portable, fast, and accurate genome analysis has many implications
  - What are the new applications to enable with these unique benefits?

- Can we do even better?
  - Understanding and modifying the sequencing process for analyzing other types of biological data

- Many future opportunities exist
  - Especially with new sequencing technologies
  - Especially with new applications and use cases
Real-time Analysis of Genomic Sequences from Nanopore Electrical Signals by Fast and Accurate Hash-based Search

Can Firtina
canfirtina@gmail.com
https://cfirtina.com

6 May 2024
The Jackson Laboratory
Analysis is Bottlenecked in Read Mapping!!

48 Human whole genomes at 30× coverage in about 2 days

Illumina NovaSeq 6000

1 Human genome
32 CPU hours on a 48-core processor

### A Tsunami of Sequencing Data

A Tera-scale increase in sequencing production in the past 25 years

<table>
<thead>
<tr>
<th></th>
<th>1990</th>
<th>Kilo = 1,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genes &amp; Operons</td>
<td>1995</td>
<td>Mega = 1,000,000</td>
</tr>
<tr>
<td>Bacterial genomes</td>
<td>2000</td>
<td>Giga = 1,000,000,000</td>
</tr>
<tr>
<td>Human genome</td>
<td>2005</td>
<td>Tera = 1,000,000,000,000</td>
</tr>
<tr>
<td>Human microbiome</td>
<td>2015</td>
<td>Peta = 1,000,000,000,000,000</td>
</tr>
<tr>
<td>50K Microbiomes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>what is expected for the next 15 years (a Giga?)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200K Microbiomes</td>
<td>2030</td>
<td>Exa = 1,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1M Microbiomes</td>
<td>2025</td>
<td>Zetta = 1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>Earth Microbiome</td>
<td>2030</td>
<td>Yotta = 1,000,000,000,000,000,000,000,000</td>
</tr>
</tbody>
</table>

Source: [@kyrpides](https://twitter.com/kyrpides)

---

Efficient indexing of k-mer presence and abundance in sequencing datasets

Rayan Chikhi, VanBUG seminar 2020
Today’s Computing Systems

von Neumann model, 1945
where the CPU can access data stored in an off-chip main memory only through power-hungry bus

Storage (SSD/HDD) Main Memory Microprocessor

Burks, Goldstein, von Neumann, “Preliminary discussion of the logical design of an electronic computing instrument,” 1946.
The Problem

Data analysis is performed far away from the data
Did we realize the need for faster genome analysis?

Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021
Sequence Alignment in Unavoidable

- **Quadratic-time** dynamic-programming algorithm  **WHY?!**

  Enumerating all possible prefixes

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>E</th>
<th>T</th>
<th>H</th>
<th>E</th>
<th>R</th>
<th>L</th>
<th>A</th>
<th>N</th>
<th>D</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- NETHERLANDS x SWITZERLAND
- NETHERLANDS x S
- NETHERLANDS x SW
- NETHERLANDS x SWI
- NETHERLANDS x SWIT
- NETHERLANDS x SWITZ
- NETHERLANDS x SWITZE
- NETHERLANDS x SWITZER
- NETHERLANDS x SWITZERL
- NETHERLANDS x SWITZERLA
- NETHERLANDS x SWITZERLAN
- NETHERLANDS x SWITZERLAND

WHY?!
Sequence Alignment in Unavoidable

- **Quadratic-time** dynamic-programming algorithm
  - Enumerating all possible prefixes

- **Data dependencies** limit the computation parallelism
  - Processing row (or column) after another

- **Entire matrix** is computed even though strings can be dissimilar.
  - Number of differences is computed only at the backtracking step.
Metagenomics Analysis

Reads from different unknown donors at sequencing time are mapped to many known reference genomes.

genetic material recovered directly from environmental samples

Reads “text format”

Reference Database
Genomics vs. Metagenomics
Huge Demand for Performance & Efficiency

Exponential Growth of Neural Networks

1800x more compute
In just 2 years

Tomorrow, multi-trillion parameter models

Source: https://youtu.be/Bh13Idwcb0Q?t=283
Deeper and Larger Memory Hierarchies

**AMD Ryzen 5000, 2020**

- **Core Count:** 8 cores/16 threads
- **L1 Caches:** 32 KB per core
- **L2 Caches:** 512 KB per core
- **L3 Cache:** 32 MB shared

[Image of the chip with labeled sections]

AMD’s 3D Last Level Cache (2021)

AMD increases the L3 size of their 8-core Zen 3 processors from 32 MB to 96 MB

**Additional 64 MB L3 cache die stacked on top of the processor die**
- Connected using Through Silicon Vias (TSVs)
- Total of 96 MB L3 cache
Deeper and Larger Memory Hierarchies

IBM POWER10, 2020

**Cores:**
- 15-16 cores, 8 threads/core

**L2 Caches:**
- 2 MB per core

**L3 Cache:**
- 120 MB shared

Deeper and Larger Memory Hierarchies

Apple M1 Ultra System (2022)

https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php
Data Movement Overwhelms Modern Machines


62.7% of the total system energy is spent on **data movement**

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand$^1$
Rachata Ausavarungrunr$^1$
Aki Kuusela$^3$
Allan Knies$^3$

Saugata Ghose$^1$
Eric Shiu$^3$
Parthasarathy Ranganathan$^3$

Youngsok Kim$^2$
Rahul Thakur$^3$

Daehyun Kim$^{4,3}$
Onur Mutlu$^{5,1}$

SAFARI
Processing-in-Memory Landscape Today

And, many other experimental chips and startups
Communication Dominates Arithmetic

Dally, HiPEAC 2015

- 64-bit DP: 20 pJ
- 256-bit buses
- 256-bit access: 8 kB SRAM
- DRAM Rd/Wr: 16 nJ
- Efficient off-chip link: 500 pJ
- 50 pJ
- 1 nJ
- 26 pJ
- 256 pJ

20 mm
A memory access consumes $\sim 100$-1000X the energy of a complex addition.
Data Movement vs. Computation Energy

Energy for a 32-bit Operation (log scale)

- **Energy (pJ)**
- **ADD (int) Relative Cost**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy (pJ)</th>
<th>Relative Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD (int)</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>ADD (float)</td>
<td>0.9</td>
<td>3.1</td>
</tr>
<tr>
<td>Register File</td>
<td>1</td>
<td>3.7</td>
</tr>
<tr>
<td>MULT (int)</td>
<td>3.1</td>
<td>5</td>
</tr>
<tr>
<td>MULT (float)</td>
<td>3.7</td>
<td></td>
</tr>
<tr>
<td>SRAM Cache</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>DRAM</td>
<td>640</td>
<td></td>
</tr>
</tbody>
</table>

Data Movement vs. Computation Energy

A memory access consumes 6400X the energy of a simple integer addition.
Practical Similarity Identification

Reference

Read

K-mers

Seeds

3 billion characters

Anticipating GCC CAT AT GGG TT AAG GCT T C TGG AA T C G G G C T T T C G C T T T G ...

K-mers

Locations

Index (Hash Table)

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

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 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**

**Real-time Analysis**

**Basecalling** → **Read mapping**

**Costly and energy-hungry computations to basecall each read:** Portable sequencing becomes challenging with resource-constrained devices

**Signal mapping**

**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*

![Diagram showing quantization and pruning of event values with binary representations.](image-url)
Breakdown Analysis of the RawHash Steps

<table>
<thead>
<tr>
<th>Tool</th>
<th>Fraction of entire runtime (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SARS-CoV-2</td>
</tr>
<tr>
<td>File I/O</td>
<td>0.00</td>
</tr>
<tr>
<td>Signal-to-Event</td>
<td>21.75</td>
</tr>
<tr>
<td>Sketching</td>
<td>0.74</td>
</tr>
<tr>
<td>Seeding</td>
<td>3.86</td>
</tr>
<tr>
<td>Chaining</td>
<td>73.50</td>
</tr>
<tr>
<td>Seeding + Chaining</td>
<td>77.36</td>
</tr>
</tbody>
</table>

The entire runtime is **bottlenecked by the chaining step**
# Full Read Mapping Accuracy

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>RH2</th>
<th>RH2-Min.</th>
<th>RH</th>
<th>UNCALLED</th>
<th>Sigmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>SARS-CoV-2</td>
<td>F1</td>
<td>0.9867</td>
<td>0.9691</td>
<td>0.9252</td>
<td>0.9725</td>
<td>0.7112</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.9939</td>
<td>0.9868</td>
<td>0.9832</td>
<td>0.9547</td>
<td>0.9929</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.9796</td>
<td>0.9521</td>
<td>0.8736</td>
<td><strong>0.9910</strong></td>
<td>0.5540</td>
</tr>
<tr>
<td>E. coli</td>
<td>F1</td>
<td>0.9748</td>
<td>0.9631</td>
<td>0.9280</td>
<td>0.9731</td>
<td>0.9670</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.9904</td>
<td>0.9865</td>
<td>0.9563</td>
<td>0.9817</td>
<td>0.9842</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.9597</td>
<td>0.9408</td>
<td>0.9014</td>
<td><strong>0.9647</strong></td>
<td>0.9504</td>
</tr>
<tr>
<td>Yeast</td>
<td>F1</td>
<td>0.9602</td>
<td>0.9472</td>
<td>0.9060</td>
<td>0.9407</td>
<td>0.9469</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.9553</td>
<td>0.9561</td>
<td>0.9852</td>
<td>0.9442</td>
<td><strong>0.9857</strong></td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.9652</td>
<td>0.9385</td>
<td>0.8387</td>
<td>0.9372</td>
<td>0.9111</td>
</tr>
<tr>
<td>Green Algae</td>
<td>F1</td>
<td>0.9351</td>
<td>0.9191</td>
<td>0.8114</td>
<td>0.8277</td>
<td>0.9350</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.9284</td>
<td>0.9280</td>
<td>0.9652</td>
<td>0.8843</td>
<td><strong>0.9743</strong></td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.9418</td>
<td>0.9104</td>
<td>0.6999</td>
<td>0.7779</td>
<td>0.8987</td>
</tr>
<tr>
<td>Human</td>
<td>F1</td>
<td>0.7599</td>
<td>0.6699</td>
<td>0.5574</td>
<td>0.3197</td>
<td>0.3269</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.8675</td>
<td>0.8511</td>
<td><strong>0.8943</strong></td>
<td>0.4868</td>
<td>0.4288</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.6760</td>
<td>0.5523</td>
<td>0.4049</td>
<td>0.2380</td>
<td>0.2642</td>
</tr>
<tr>
<td>Contamination</td>
<td>F1</td>
<td>0.9614</td>
<td>0.9317</td>
<td>0.8718</td>
<td><strong>0.9637</strong></td>
<td>0.6498</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.9595</td>
<td>0.9424</td>
<td>0.8702</td>
<td>0.9378</td>
<td>0.7856</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.9632</td>
<td>0.9212</td>
<td>0.8736</td>
<td><strong>0.9910</strong></td>
<td>0.5540</td>
</tr>
<tr>
<td>Rel. Abundance</td>
<td>F1</td>
<td>0.4659</td>
<td>0.3375</td>
<td>0.3045</td>
<td>0.1249</td>
<td>0.2443</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.4623</td>
<td>0.3347</td>
<td>0.3018</td>
<td>0.1226</td>
<td>0.2366</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.4695</td>
<td>0.3404</td>
<td>0.3071</td>
<td>0.1273</td>
<td>0.2525</td>
</tr>
</tbody>
</table>

Best results are **highlighted**.
## R10.4 Accuracy and Performance Results

<table>
<thead>
<tr>
<th>Flow Cell</th>
<th>Read Mapping Accuracy (E. coli)</th>
<th>RH2</th>
<th>RH2-Min.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R9.4</td>
<td>F1</td>
<td>0.9748</td>
<td>0.9631</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.9904</td>
<td>0.9865</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.9597</td>
<td>0.9408</td>
</tr>
<tr>
<td>R10.4</td>
<td>F1</td>
<td>0.8960</td>
<td>0.8389</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.9506</td>
<td>0.9325</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.8473</td>
<td>0.7623</td>
</tr>
<tr>
<td></td>
<td>Read Mapping Accuracy (S. aureus)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R10.4</td>
<td>F1</td>
<td>0.7749</td>
<td>0.6778</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.8649</td>
<td>0.8167</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.7018</td>
<td>0.5793</td>
</tr>
<tr>
<td></td>
<td>Performance (E. coli)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R9.4</td>
<td>Throughput [bp/sec]</td>
<td>303,382.45</td>
<td>659,013.57</td>
</tr>
<tr>
<td></td>
<td>Mean time per read [ms]</td>
<td>2.161</td>
<td>1.099</td>
</tr>
<tr>
<td>R10.4</td>
<td>Throughput [bp/sec]</td>
<td>175,351.94</td>
<td>480,471.75</td>
</tr>
<tr>
<td></td>
<td>Mean time per read [ms]</td>
<td>6.598</td>
<td>2.505</td>
</tr>
<tr>
<td></td>
<td>Performance (S. aureus)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R10.4</td>
<td>Throughput [bp/sec]</td>
<td>256,680.4</td>
<td>617,308.7</td>
</tr>
<tr>
<td></td>
<td>Mean time per read [ms]</td>
<td>5.478</td>
<td>2.243</td>
</tr>
</tbody>
</table>
Full Read Mapping Accuracy – Spider Plot

- **RawHash2**
- **RawHash2-Minimizer**
- **RawHash**
- **UNCALLED**
- **Sigmap**

### SARS-CoV-2
- **Precision**
- **Recall**
- **F1**

### E. coli
- **Precision**
- **Recall**
- **F1**

### Yeast
- **Precision**
- **Recall**
- **F1**

### Green Algae
- **Precision**
- **Recall**
- **F1**

### Human
- **Precision**
- **Recall**
- **F1**

### Contamination
- **Precision**
- **Recall**
- **F1**

### Relative Abundance
- **Precision**
- **Recall**
- **F1**
Different Metrics Combined – Spider Plot

- **SARS-CoV-2**
  - Throughput

- **E. coli**
  - Throughput

- **Yeast**
  - Throughput

- **Green Algae**
  - Throughput

- **Human**
  - Throughput

- **Contamination**
  - Throughput

- **Relative Abundance**
  - Throughput
### Required Computation Resources in Indexing

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RH2</th>
<th>RH2-Min.</th>
<th>RH</th>
<th>UNCALLED</th>
<th>Sigmap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Indexing CPU Time (sec)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SARS-CoV-2</td>
<td>0.12</td>
<td>0.06</td>
<td>0.16</td>
<td>8.40</td>
<td>0.02</td>
</tr>
<tr>
<td>E. coli</td>
<td>2.48</td>
<td>1.61</td>
<td>2.56</td>
<td>10.57</td>
<td>8.86</td>
</tr>
<tr>
<td>Yeast</td>
<td>4.56</td>
<td>3.02</td>
<td>4.44</td>
<td>16.40</td>
<td>25.29</td>
</tr>
<tr>
<td>Green Algae</td>
<td>27.60</td>
<td>17.73</td>
<td>24.51</td>
<td>213.13</td>
<td>420.25</td>
</tr>
<tr>
<td>Human</td>
<td>1,093.56</td>
<td>588.30</td>
<td>809.08</td>
<td>3,496.76</td>
<td>41,993.26</td>
</tr>
<tr>
<td>Contamination</td>
<td>0.13</td>
<td>0.06</td>
<td>0.15</td>
<td>8.38</td>
<td>0.03</td>
</tr>
<tr>
<td>Rel. Abundance</td>
<td>747.74</td>
<td>468.14</td>
<td>751.67</td>
<td>3,666.14</td>
<td>36,216.87</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Indexing Peak Memory (GB)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SARS-CoV-2</td>
<td><strong>0.01</strong></td>
<td><strong>0.01</strong></td>
<td><strong>0.01</strong></td>
<td>0.06</td>
<td><strong>0.01</strong></td>
</tr>
<tr>
<td>E. coli</td>
<td>0.35</td>
<td>0.19</td>
<td>0.35</td>
<td>0.11</td>
<td>0.40</td>
</tr>
<tr>
<td>Yeast</td>
<td>0.75</td>
<td>0.39</td>
<td>0.76</td>
<td>0.30</td>
<td>1.04</td>
</tr>
<tr>
<td>Green Algae</td>
<td>5.11</td>
<td>2.60</td>
<td>5.33</td>
<td>11.94</td>
<td>8.63</td>
</tr>
<tr>
<td>Human</td>
<td>80.75</td>
<td>40.59</td>
<td>83.09</td>
<td>48.43</td>
<td>227.77</td>
</tr>
<tr>
<td>Contamination</td>
<td><strong>0.01</strong></td>
<td><strong>0.01</strong></td>
<td><strong>0.01</strong></td>
<td>0.06</td>
<td><strong>0.01</strong></td>
</tr>
<tr>
<td>Rel. Abundance</td>
<td>152.59</td>
<td>75.62</td>
<td>152.84</td>
<td><strong>47.80</strong></td>
<td>238.32</td>
</tr>
</tbody>
</table>

The indexing in RawHash is **orders of magnitude faster** than the indexing steps of UNCALLED and Sigmap, especially for **large genomes**.
### Required Computation Resources in Mapping

The mapping step of RawHash2 (and RawHash2-Minimizer) is significantly faster than all tools.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RH2</th>
<th>RH2-Min.</th>
<th>RH</th>
<th>UNCALLED</th>
<th>Sitemap</th>
</tr>
</thead>
<tbody>
<tr>
<td>SARS-CoV-2</td>
<td>1,705.43</td>
<td>1,227.05</td>
<td>1,539.64</td>
<td>29,282.90</td>
<td>1,413.32</td>
</tr>
<tr>
<td>E. coli</td>
<td>1,296.34</td>
<td>787.49</td>
<td>7,453.21</td>
<td>28,767.58</td>
<td>22,923.09</td>
</tr>
<tr>
<td>Yeast</td>
<td>545.77</td>
<td>246.37</td>
<td>4,145.38</td>
<td>7,181.44</td>
<td>7,146.32</td>
</tr>
<tr>
<td>Green Algae</td>
<td>2,135.83</td>
<td>657.63</td>
<td>22,103.03</td>
<td>12,593.01</td>
<td>26,778.44</td>
</tr>
<tr>
<td>Human</td>
<td>100,947.58</td>
<td>21,860.05</td>
<td>1,825,061.23</td>
<td>245,128.15</td>
<td>6,101,179.89</td>
</tr>
<tr>
<td>Contamination</td>
<td>3,783.69</td>
<td>2,332.28</td>
<td>3,480.43</td>
<td>234,199.60</td>
<td>3,011.78</td>
</tr>
<tr>
<td>Rel. Abundance</td>
<td>250,076.90</td>
<td>62,477.76</td>
<td>4,551,349.79</td>
<td>569,824.13</td>
<td>15,178,633.11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Mapping Peak Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SARS-CoV-2</td>
<td>4.15</td>
</tr>
<tr>
<td>E. coli</td>
<td>4.13</td>
</tr>
<tr>
<td>Yeast</td>
<td>4.38</td>
</tr>
<tr>
<td>Green Algae</td>
<td>6.11</td>
</tr>
<tr>
<td>Human</td>
<td>48.75</td>
</tr>
<tr>
<td>Contamination</td>
<td>4.16</td>
</tr>
<tr>
<td>Rel. Abundance</td>
<td>49.14</td>
</tr>
</tbody>
</table>

The mapping step of RawHash2 (and RawHash2-Minimizer) is significantly faster than all tools.
**Required CPU Threads For the Entire Flow Cell**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RH2</th>
<th>RH2-Min.</th>
<th>RH</th>
<th>UNCALLED</th>
<th>Sitemap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mapping Throughput (bp/sec)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SARS-CoV-2</td>
<td>552,561.25</td>
<td><strong>885,263.48</strong></td>
<td>694,274.92</td>
<td>9,260.31</td>
<td>602,380.96</td>
</tr>
<tr>
<td>E. coli</td>
<td>303,382.45</td>
<td><strong>659,013.57</strong></td>
<td>72,281.32</td>
<td>7,515.76</td>
<td>13,750.97</td>
</tr>
<tr>
<td>Yeast</td>
<td>150,547.61</td>
<td><strong>394,766.80</strong></td>
<td>28,757.15</td>
<td>7,471.48</td>
<td>11,624.82</td>
</tr>
<tr>
<td>Green Algae</td>
<td>28,742.46</td>
<td><strong>98,323.70</strong></td>
<td>9,488.79</td>
<td>10,069.41</td>
<td>2,569.89</td>
</tr>
<tr>
<td>Human</td>
<td>8,968.78</td>
<td><strong>37,086.38</strong></td>
<td>2,099.35</td>
<td>7,225.67</td>
<td>236.45</td>
</tr>
<tr>
<td>Contamination</td>
<td>563,129.81</td>
<td><strong>884,929.30</strong></td>
<td>696,873.20</td>
<td>9,343.95</td>
<td>601,936.49</td>
</tr>
<tr>
<td>Rel. Abundance</td>
<td>9,501.37</td>
<td><strong>36,919.79</strong></td>
<td>962.79</td>
<td>8,437.70</td>
<td>196.48</td>
</tr>
</tbody>
</table>

**CPU Threads Needed for the entire MinION Flowcell (512 pores)**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RH2</th>
<th>RH2-Min.</th>
<th>RH</th>
<th>UNCALLED</th>
<th>Sitemap</th>
</tr>
</thead>
<tbody>
<tr>
<td>SARS-CoV-2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>E. coli</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>31</td>
<td>17</td>
</tr>
<tr>
<td>Yeast</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>31</td>
<td>20</td>
</tr>
<tr>
<td>Green Algae</td>
<td>9</td>
<td>3</td>
<td>25</td>
<td>23</td>
<td>90</td>
</tr>
<tr>
<td>Human</td>
<td>26</td>
<td>7</td>
<td>110</td>
<td>32</td>
<td>975</td>
</tr>
<tr>
<td>Contamination</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>Rel. Abundance</td>
<td>25</td>
<td>7</td>
<td>240</td>
<td>28</td>
<td>1173</td>
</tr>
</tbody>
</table>

**RawHash2 (and RawHash2-Minimizer) requires the least amount of CPU threads to process the entire MinION flowcell.**