



Real-time Sequence Analysis

from Raw Nanopore Electrical Signals
with Accurate Hash-based Search Mechanisms

Can Firtina

canfirtina@gmail.com

<https://cfirtina.com>

10 May 2024

The Broad Institute of MIT and Harvard

SAFARI

ETH zürich

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/>
- **Contact me:** canfirtina@gmail.com
- **Website:** <https://cfirtina.com>
- **Twitter (aka X):** <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/>
- ❑ omutlu@gmail.com (Best way to reach)
- ❑ <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

SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

SAFARI

<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

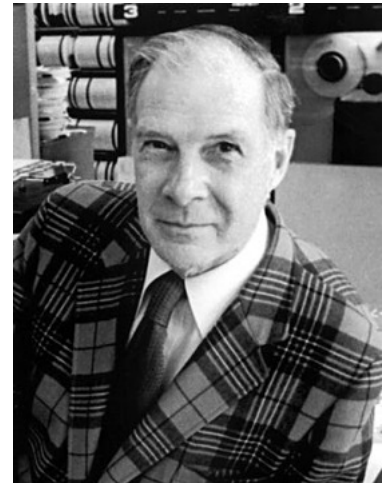
- Enabling Fast and Accurate Sequence Analysis
 - Real-Time Analysis Nanopore Raw Signal Analysis
 - Fuzzy seed matching for accurate sequence analysis

- Conclusion

The Goal of Computing: Beyond Numbers

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

Richard Hamming

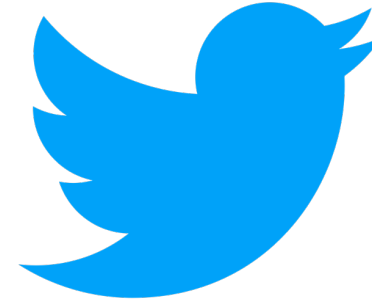


We need to gain insights
and observations
much more efficiently
than ever before

Big Data is Everywhere



Astronomy
25 zetta-bytes/year



Twitter (now X)
0.5-15 billion tweets/year

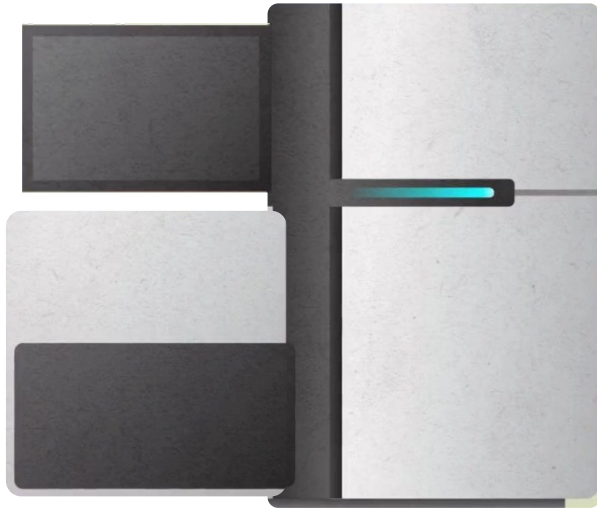


YouTube
500-900 million hours/year



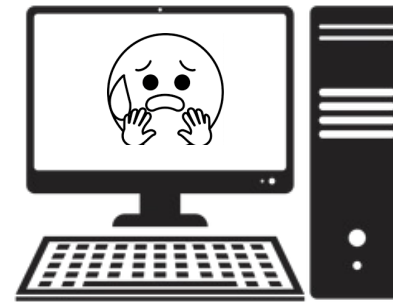
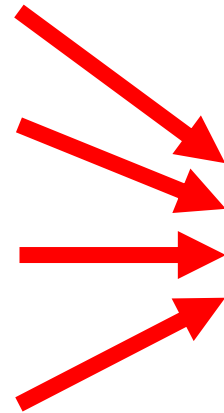
Genomics
1 zetta-bases/year

Problems with Data Analysis Today



Special-Purpose Machine
for **Data Generation**

FAST



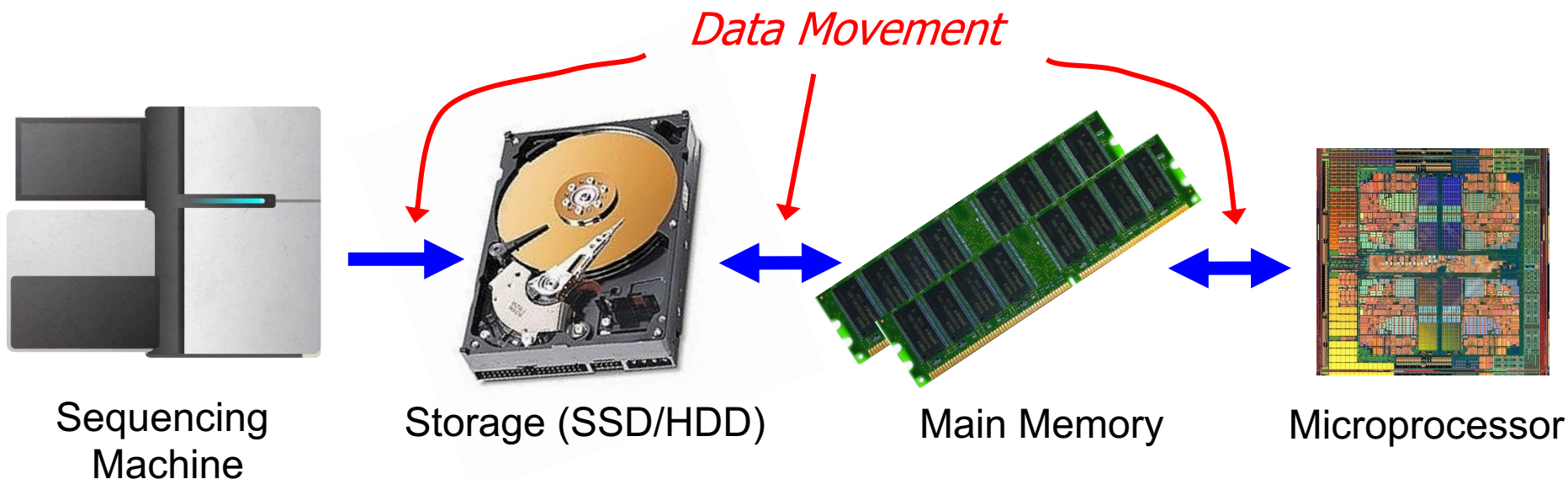
General-Purpose Machine
for **Data Analysis**

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

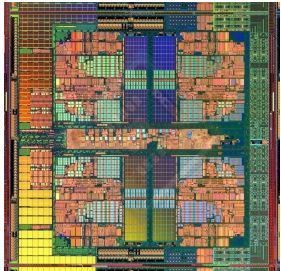
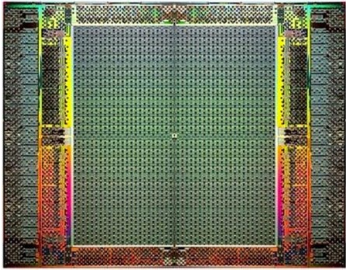
* Pandiyan and Wu, "Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms," IISWC 2014

We need intelligent algorithms
and intelligent architectures
that handle data well

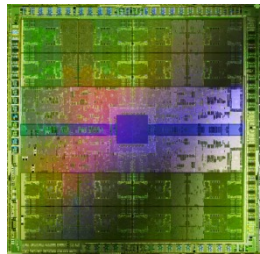
Pushing Towards New Architectures

Modern systems

FPGAs



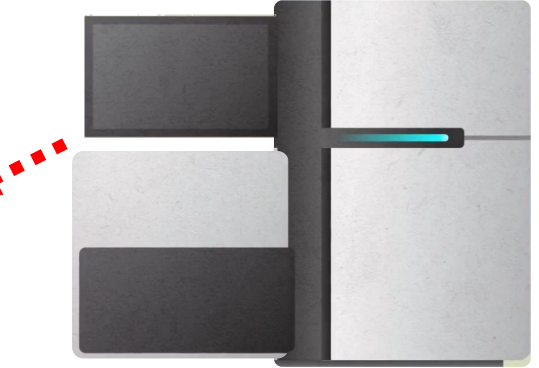
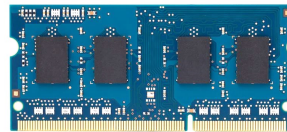
Heterogeneous Processors and Accelerators



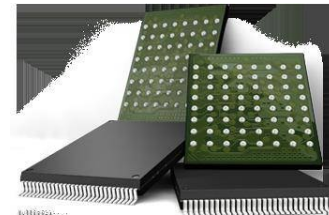
(General Purpose) GPUs



Hybrid Main Memory



Sequencing Machine

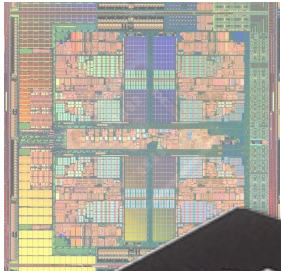
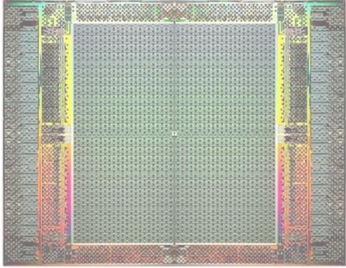


Persistent Memory/Storage

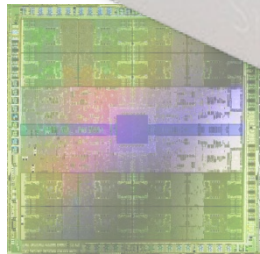
Pushing Towards New Architectures

Modern systems

FPGAs

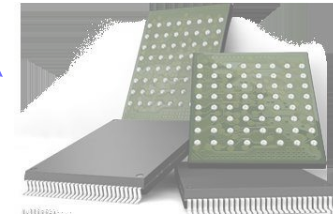


Hetero
Pro
Ac



(General Purpose) GPUs

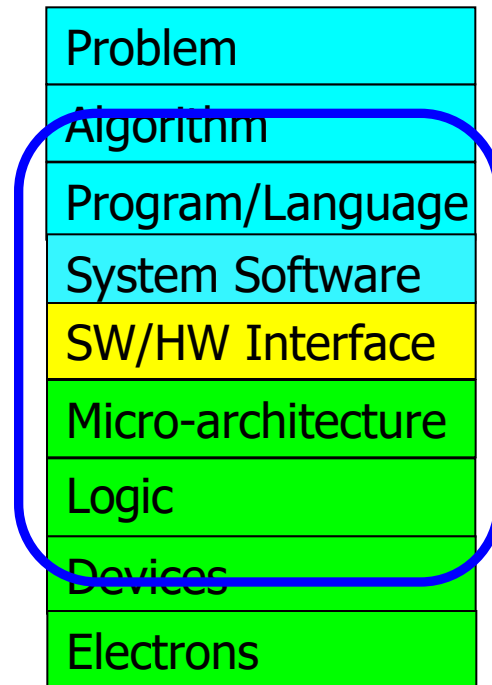
Sequencing
Machine



Persistent Memory/Storage

Algorithm-Arch-Device Co-Design is Critical

**Computer Architecture
(expanded view)**



Accelerating Genome Analysis [DAC 2023]

- Onur Mutlu and Can Firtina,
"Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
Invited Special Session Paper in Proceedings of the 60th Design Automation Conference (DAC), San Francisco, CA, USA, July 2023.
[[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

Many opportunities exist
with the improvements in
sequencing technologies
and emerging hardware

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

OXFORD

RawHash: enabling fast and accurate real-time analysis of raw nanopore signals for large genomes

Can Firtina ^{1,*}, **Nika Mansouri Ghiasi** ¹, **Joel Lindegger** ¹, **Gagandeep Singh** ¹,
Meryem Banu Cavlak ¹, **Haiyu Mao** ¹, **Onur Mutlu** ^{1,*}

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"
*Preprint on **arXiv**, September 2023.*
[\[arXiv version\]](#)
[\[RawHash2 Source Code\]](#)

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

Machine Learning in Genomics

- 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 Cavlak¹ Gagandeep Singh¹ Mohammed Alser¹ Can Firtina¹ Joël Lindegger¹
Mohammad Sadrosadati¹ Nika Mansouri Ghiasi¹ Can Alkan² Onur Mutlu¹

¹*ETH Zürich*

²*Bilkent University*

Genome Similarity Identification [NARGAB 2023]

- 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"

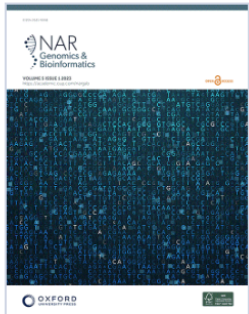
NAR Genomics and Bioinformatics, March 2023.

[[Online link at NAR Genomics and Bioinformatics Journal](#)]

[[arXiv preprint](#)]

[[biorXiv preprint](#)]



[[BLEND Source Code](#)]



Volume 5, Issue 1
March 2023

JOURNAL ARTICLE

BLEND: a fast, memory-efficient and accurate mechanism to find fuzzy seed matches in genome analysis

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

NAR Genomics and Bioinformatics, Volume 5, Issue 1, March 2023, lqad004,

New Applications: Frequent Database Updates

- Jeremie S. Kim*, Can Firtina*, M. Banu Cavlak, Damla Senol Cali, Nastaran Hajinazar, Mohammed Alser, Can Alkan, and Onur Mutlu,
"AirLift: A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes"
Proceedings of the 21st Asia Pacific Bioinformatics Conference (APBC),
Changsha, China, April 2023.
[[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^{1†}, Can Firtina^{1†}, Meryem Banu Cavlak², Damla Senol Cali³, Nastaran Hajinazar^{1,4}, Mohammed Alser¹, Can Alkan² and Onur Mutlu^{1,2,3*}

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

FREE

[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** ▼

Accelerating ML & Genome Graphs [ACM TACO '24]

- 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, **"ApHMM: Accelerating Profile Hidden Markov Models for Fast and Energy-Efficient Genome Analysis"** **ACM TACO**, Feb 2024.
[[Online link at ACM TACO](#)]
[[arXiv preprint](#)]
[[ApHMM Source Code](#)]

ApHMM: Accelerating Profile Hidden Markov Models for Fast and Energy-Efficient Genome Analysis

Just Accepted

Authors:  [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](#),  [Onur Mutlu](#) ([Less](#)) [Authors Info & Claims](#)

ACM Transactions on Architecture and Code Optimization • Accepted on October 2023 • <https://doi.org/10.1145/3632950>

Published: 28 December 2023 [Publication History](#)



Accelerating String Matching [MICRO 2020]

- Damla Senol Cali, Gurpreet S. Kalsi, Zülal 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, and Onur Mutlu, **"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"**
Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.
[[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[✕] Zülal 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

Accelerating Genome Graphs [ISCA 2022]

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika MansouriGhiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,
"SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
Proceedings of the 49th International Symposium on Computer Architecture (ISCA), New York, June 2022.
[\[arXiv version\]](#)

SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping

Damla Senol Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Zülal Bingöl³
Gurpreet S. Kalsi⁴ Ziyi Zuo⁵ Can Firtina² Meryem Banu Cavlak² Jeremie Kim²
Nika Mansouri Ghiasi² Gagandeep Singh² Juan Gómez-Luna² Nour Almadhoun Alserr²
Mohammed Alser² Sreenivas Subramoney⁴ Can Alkan³ Saugata Ghose⁶ Onur Mutlu²

¹Bionano Genomics ²ETH Zürich ³Bilkent University ⁴Intel Labs
⁵Carnegie Mellon University ⁶University of Illinois Urbana-Champaign

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 Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu, **["GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"](#)**
Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, February-March 2022.
[[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 Alserr¹
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
¹*ETH Zürich* ²*Bionano Genomics*

Basecalling using PIM [MICRO 2023]

- Taha Shahroodi, Gagandeep Singh, Mahdi Zahedi, Haiyu Mao, Joel Lindegger, 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 \(pptx\) \(pdf\)\]](#)
[\[arXiv version\]](#)

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

Taha Shahroodi¹ Gagandeep Singh^{2,3} Mahdi Zahedi¹ Haiyu Mao³ Joel Lindegger³ Can Firtina³
Stephan Wong¹ Onur Mutlu³ Said Hamdioui¹

¹TU Delft ²AMD Research ³ETH Zürich

Accelerating Genome Analysis [DAC 2023]

- Onur Mutlu and Can Firtina,
"Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
Invited Special Session Paper in Proceedings of the 60th Design Automation Conference (DAC), San Francisco, CA, USA, July 2023.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#)] (38 minutes, including Q&A)
[[Related Invited Paper](#)]
[[arXiv version](#)]

Accelerating Genome Analysis via Algorithm-Architecture Co-Design

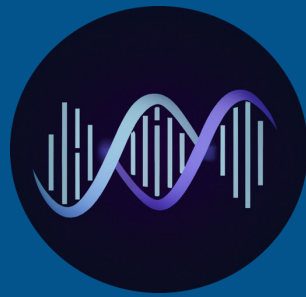
Onur Mutlu Can Firtina
ETH Zürich

Agenda for Today

- Cutting-edge in Accelerating Genome Analysis

- Enabling Fast and Accurate Sequence Analysis
 - Real-Time Analysis Nanopore Raw Signal Analysis
 - Fuzzy seed matching for accurate sequence analysis

- Conclusion



RawHash

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

Can Firtina

Nika Mansouri Ghiasi

Joel Lindegger

Gagandeep Singh

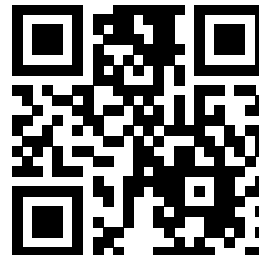
Meryem Banu Cavlak

Haiyu Mao

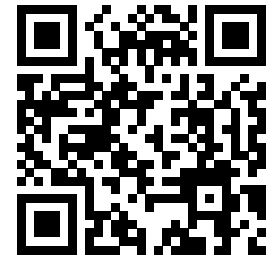
Onur Mutlu



[RawHash](#)



[RawHash2](#)



[Code](#)

Outline

Background

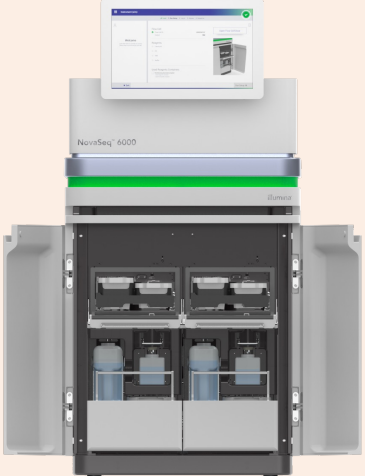
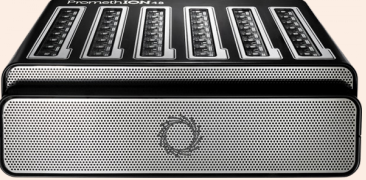

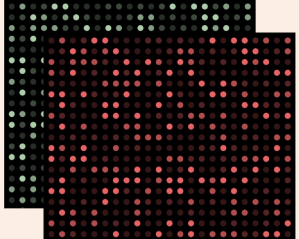
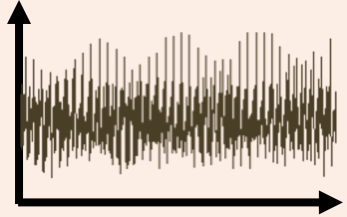
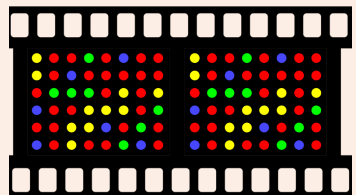



RawHash

RawHash2

Evaluation

Conclusion

Different Raw Sequencing Data

Illumina	Nanopore	PacBio
		
		
Multiple images  .BCL/.CBCL	Electrical Signal  .POD5	30-hour movie  .BAM

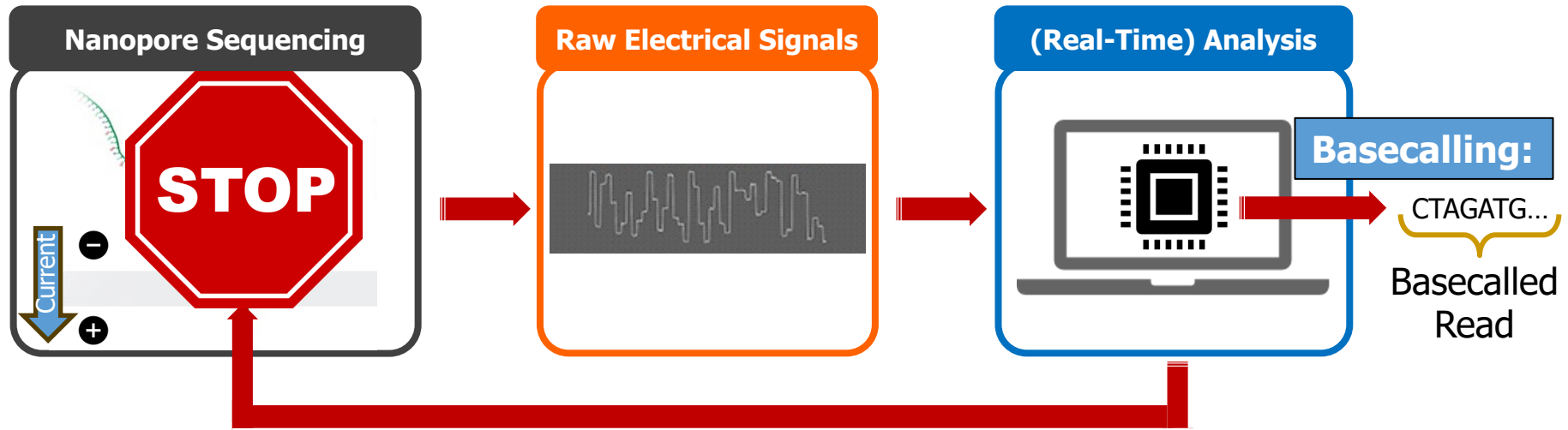
Nanopore Sequencing

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



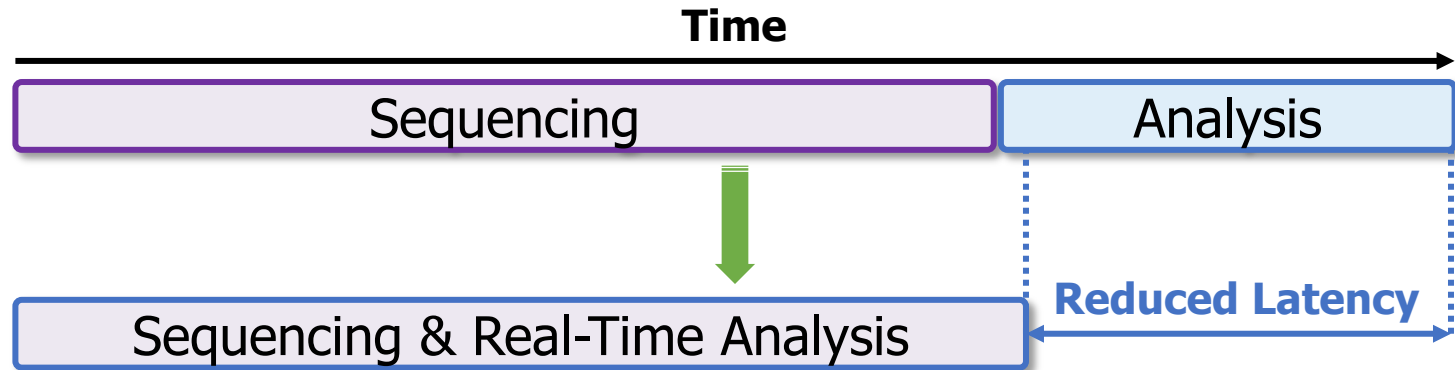
Raw Signals: Ionic current measurements generated as DNA passes through the **nanopore at a certain speed**

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

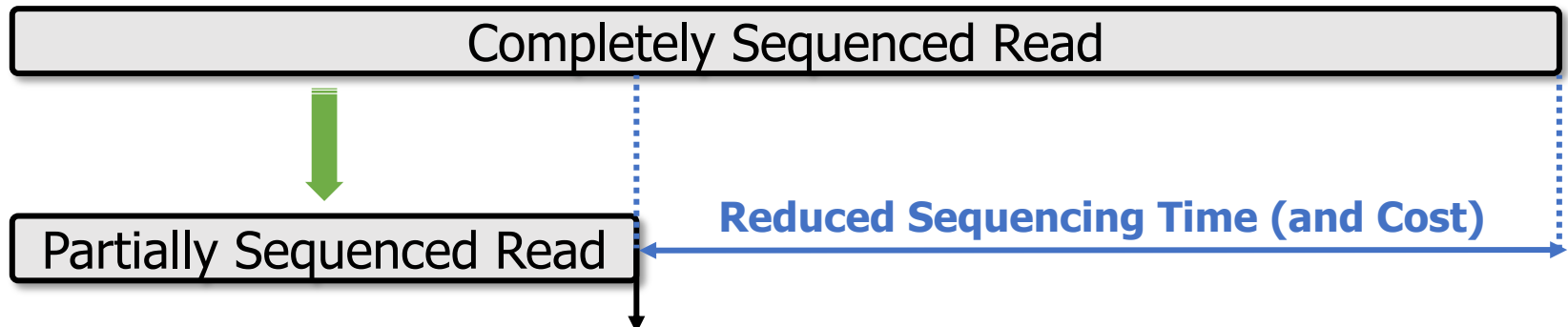
Real-Time Decisions: Stopping sequencing **early** based on real-time analysis

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

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

Outline

Background

RawHash

RawHash2

Evaluation

Conclusion

Executive Summary

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

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

Key Contributions:

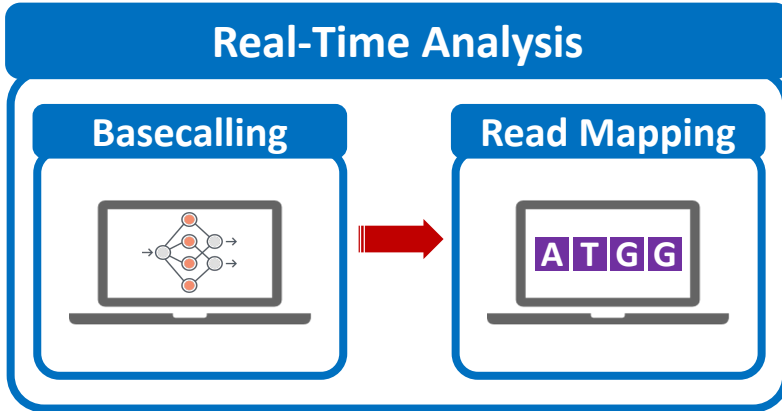
- 1) The **first hash-based mechanism** for mapping raw nanopore signals
- 2) The novel **Sequence Until** technique can accurately and **dynamically stop the entire sequencing of all reads at once** if further sequencing is not necessary

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

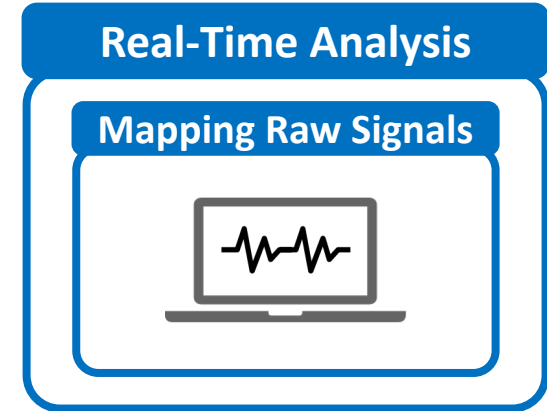
1. Deep neural networks (**DNNs**) for translating **signals** to **bases**



Less noisy analysis from basecalled sequences

Costly and power-hungry computational requirements

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

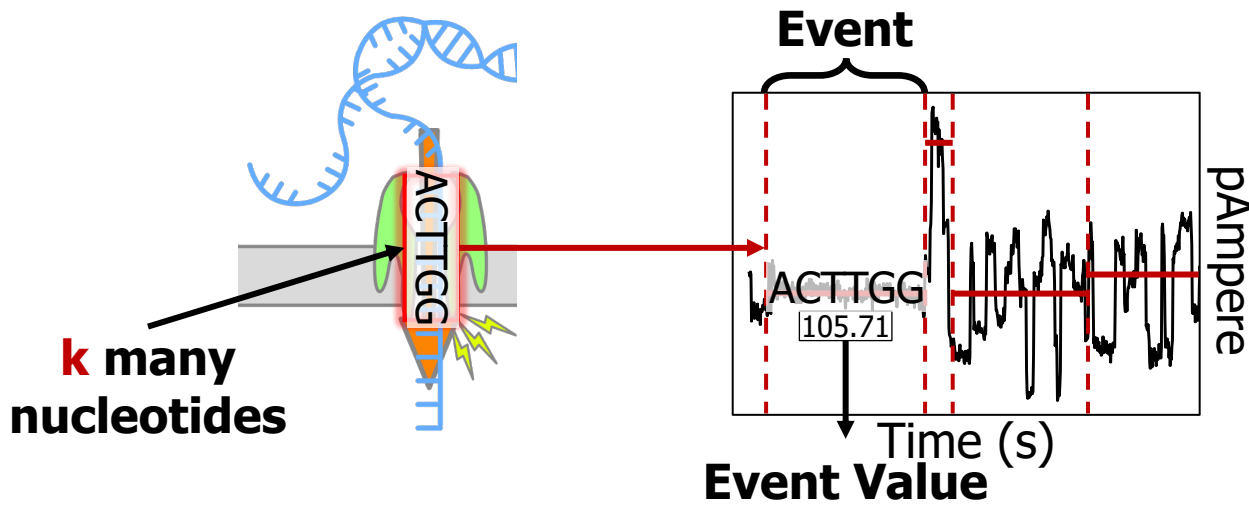


Raw signals contain richer information than bases

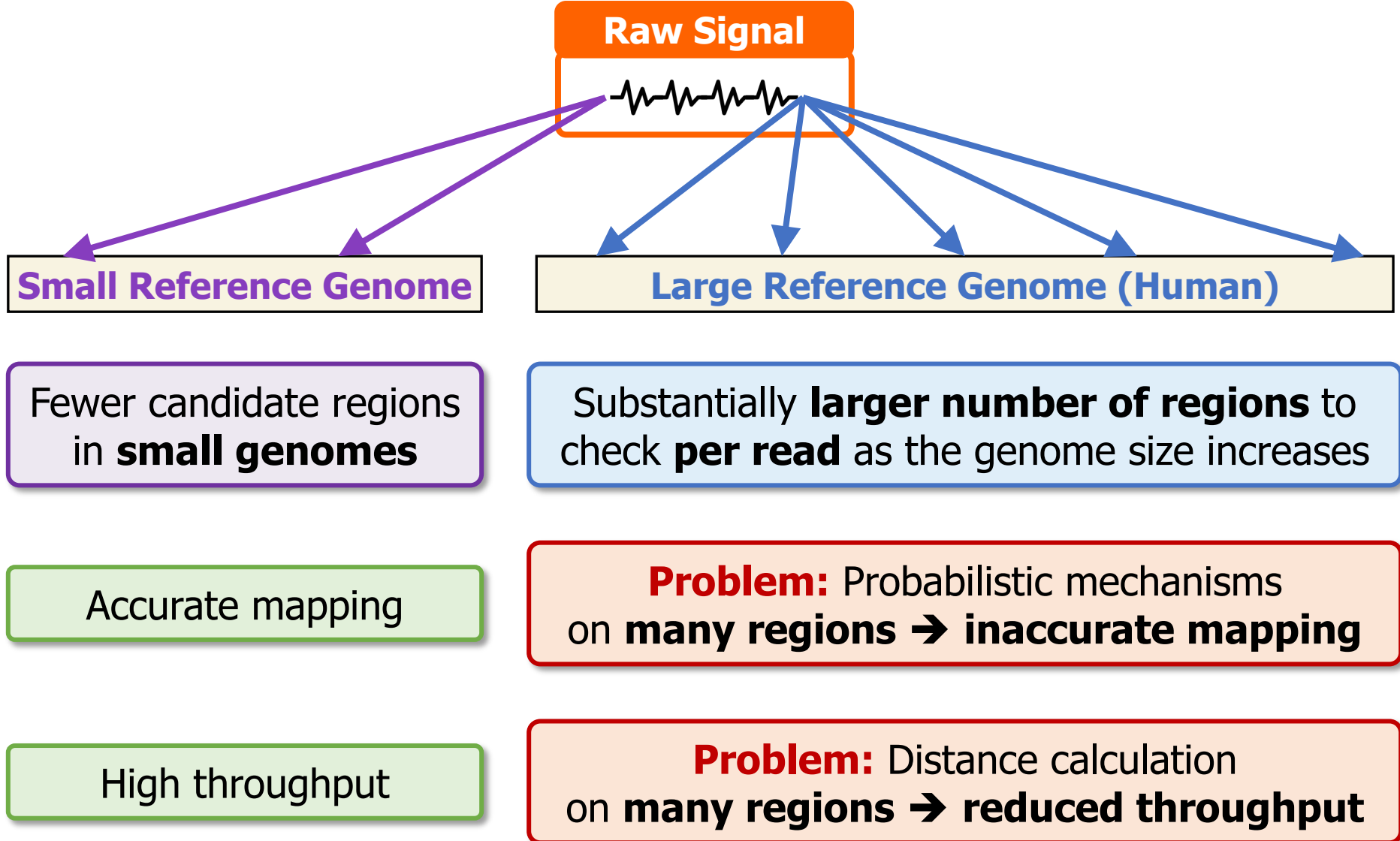
Efficient analysis with better scalability and portability

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



The Problem – Mapping Raw Signals



Existing solutions are
inaccurate or inefficient
for large genomes

Accurate mapping

on many regions → inaccurate mapping

High throughput

Problem: Distance calculation
on many regions → reduced throughput

Goal

Enable **fast and accurate real-time analysis**
of raw nanopore signals **for large genomes**



RawHash

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



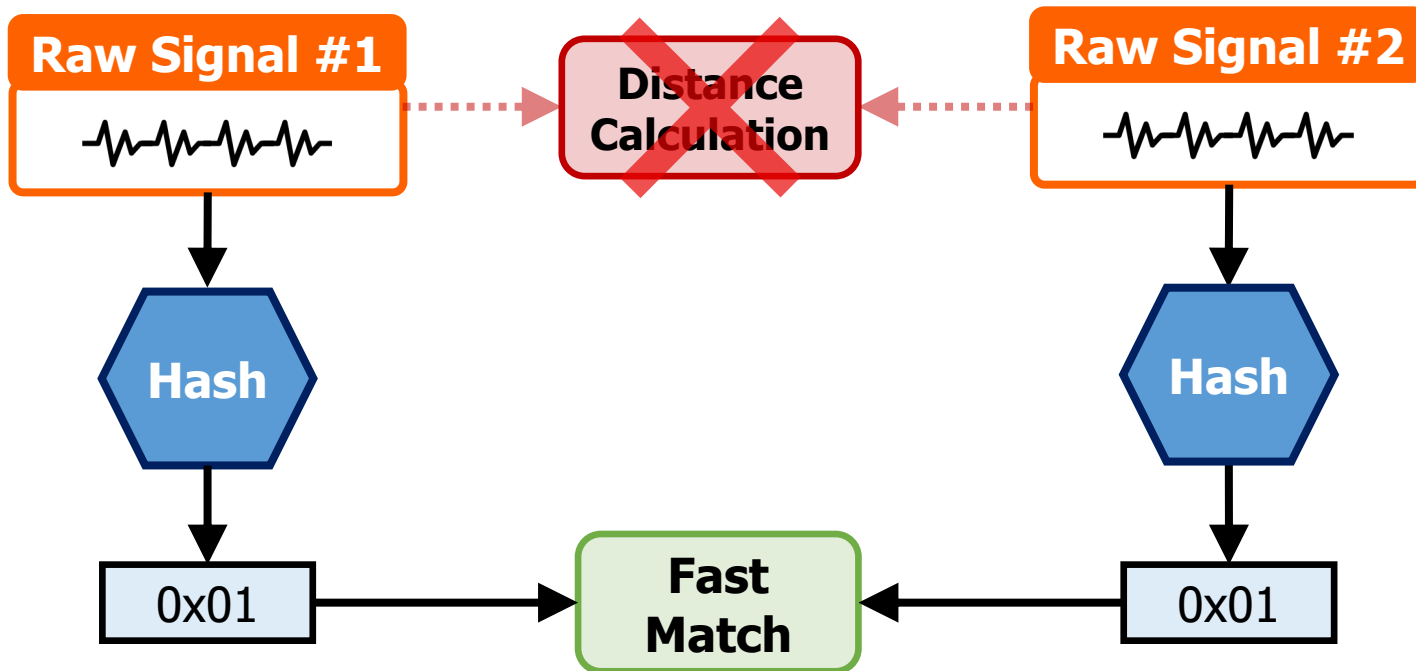
RawHash

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

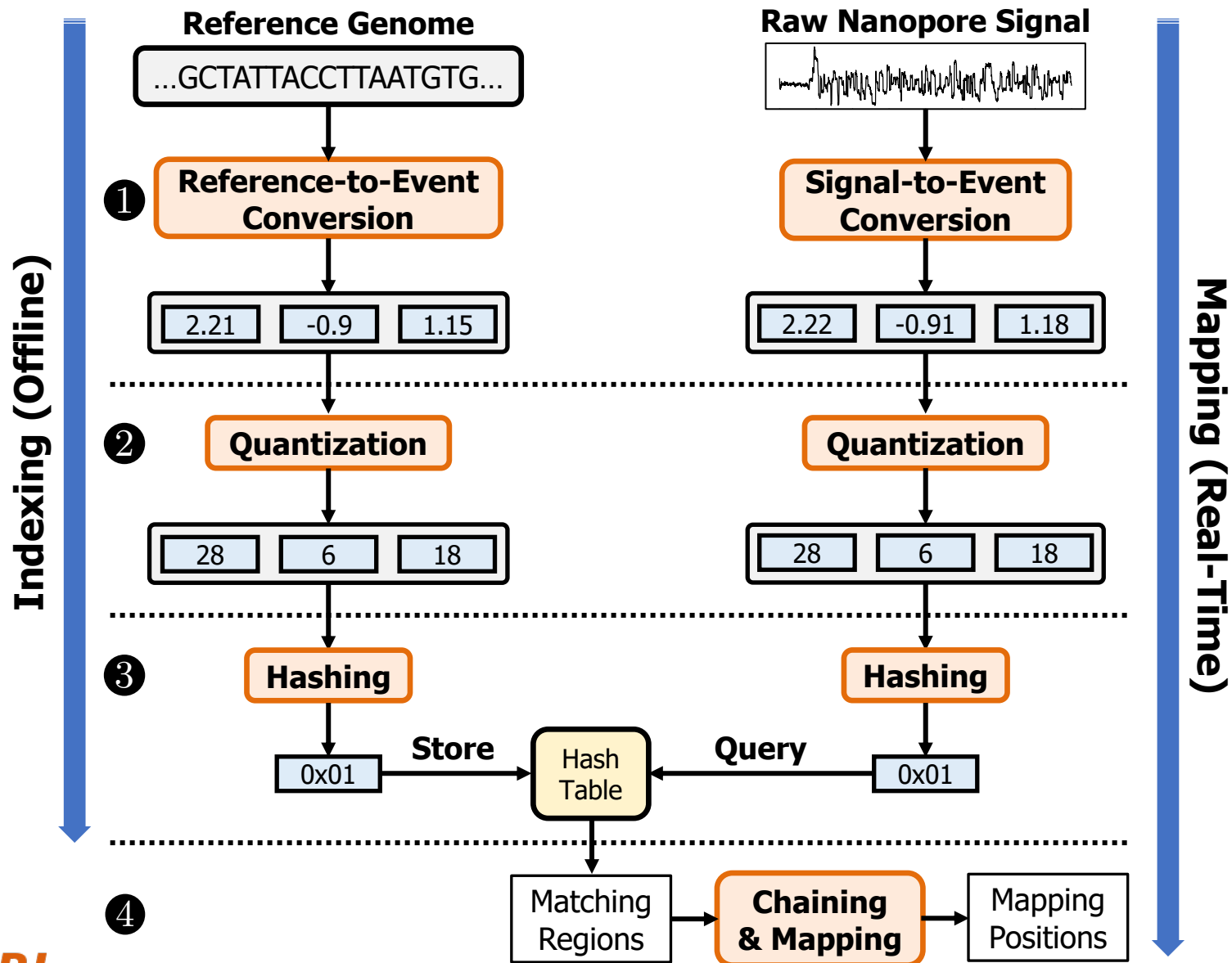
Key Observation: **Identical** nucleotides generate **similar** raw signals



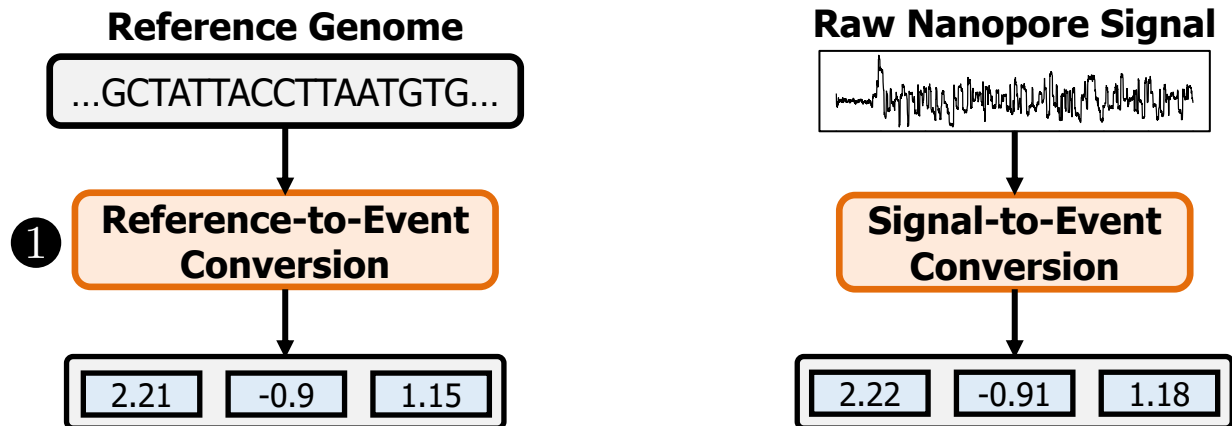
Challenge #1: Generating the **same** hash value for **similar enough** signals

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

RawHash Overview

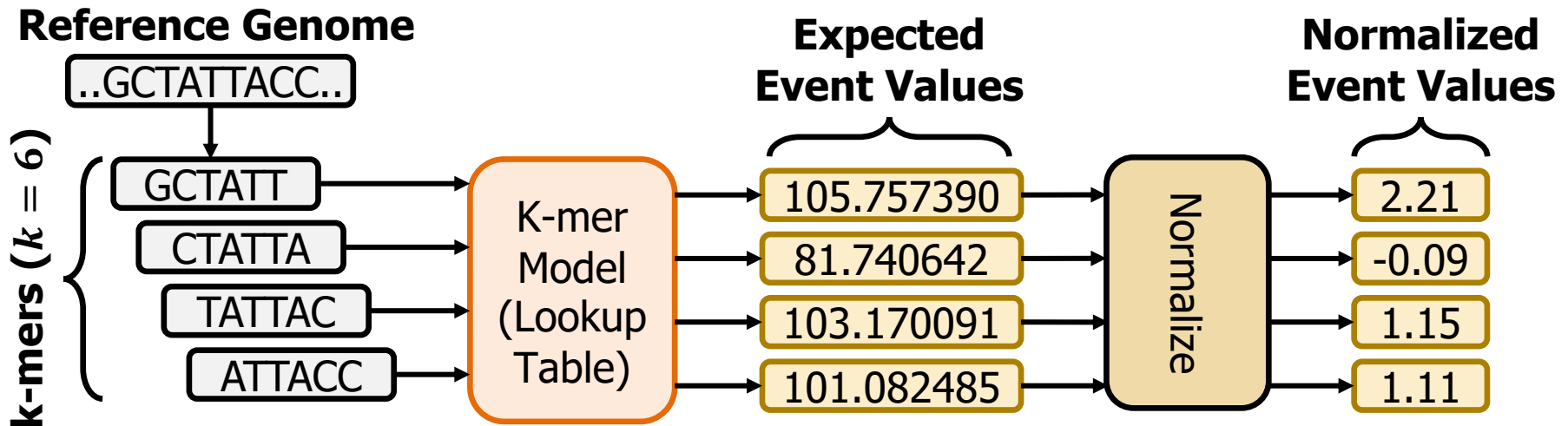


RawHash Overview



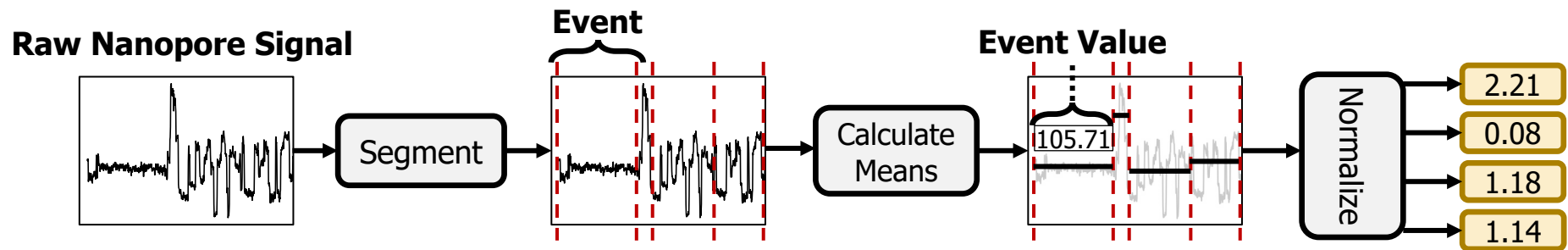
Reference-to-Event Conversion

- **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



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

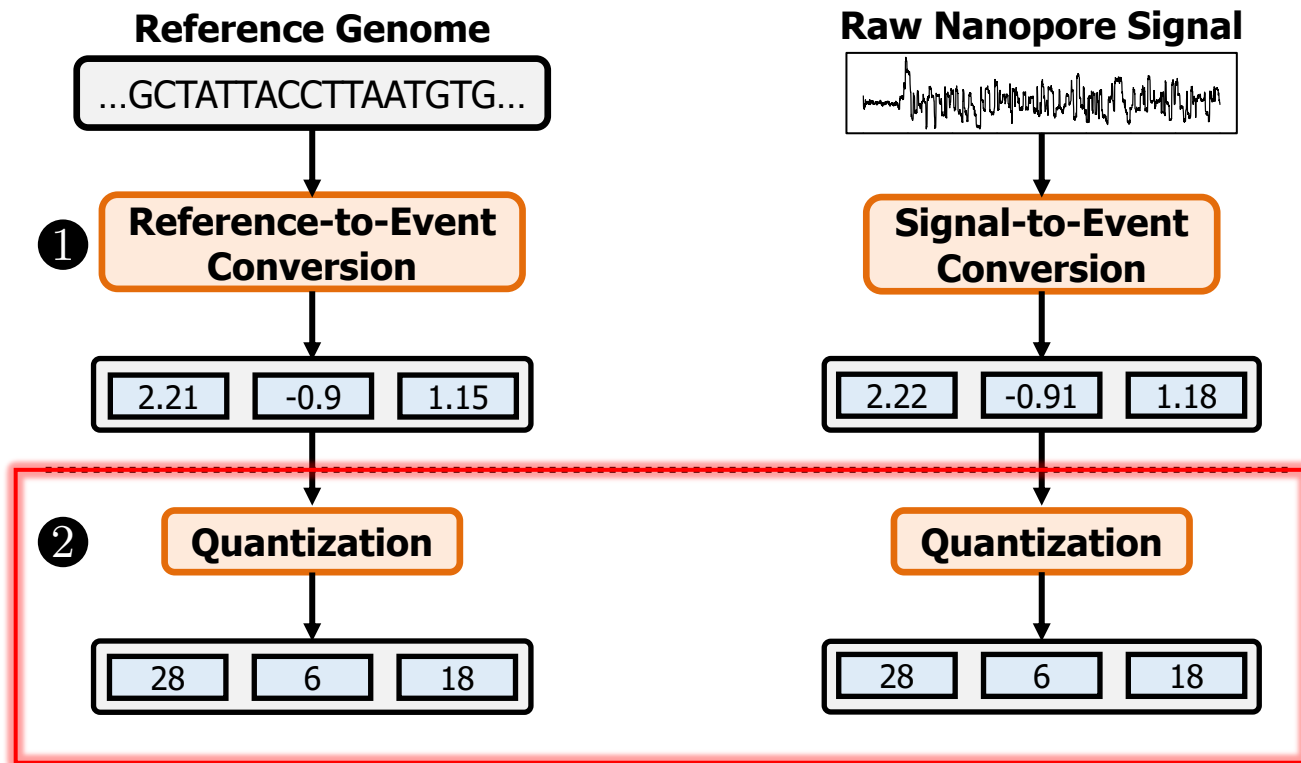
Signal-to-Event Conversion

- **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?

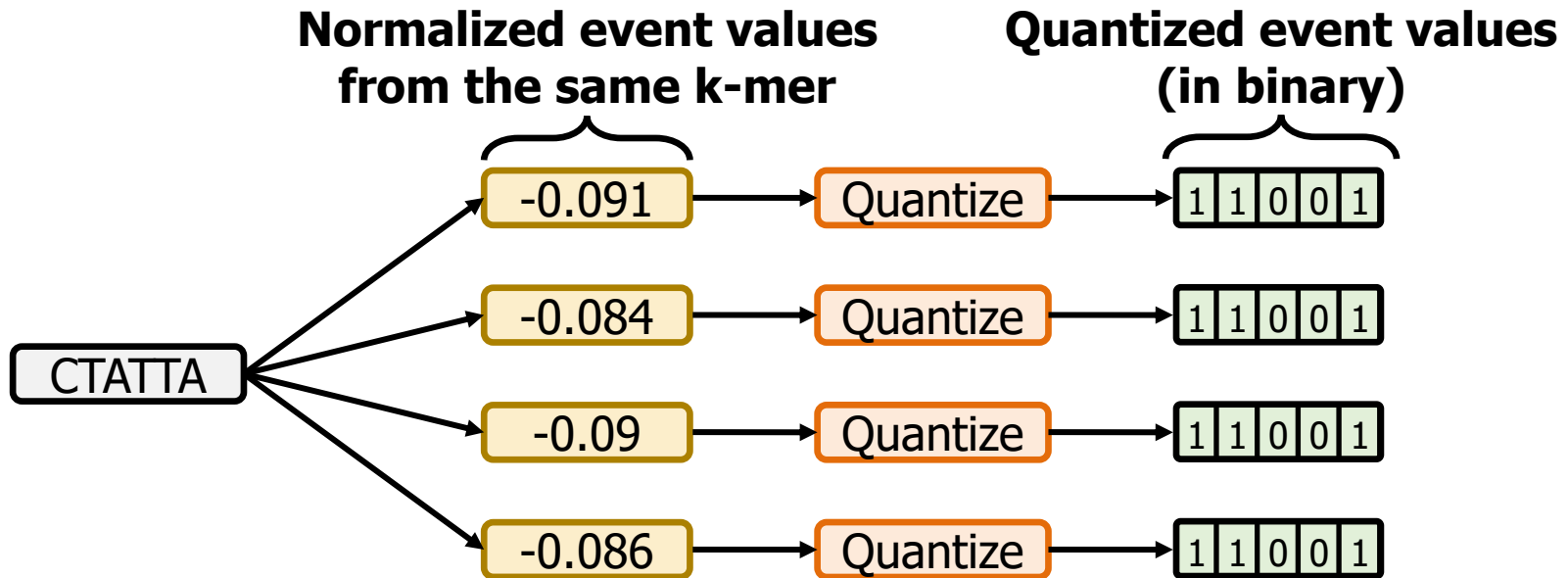
- Consecutive events → consecutive k-mers

RawHash Overview

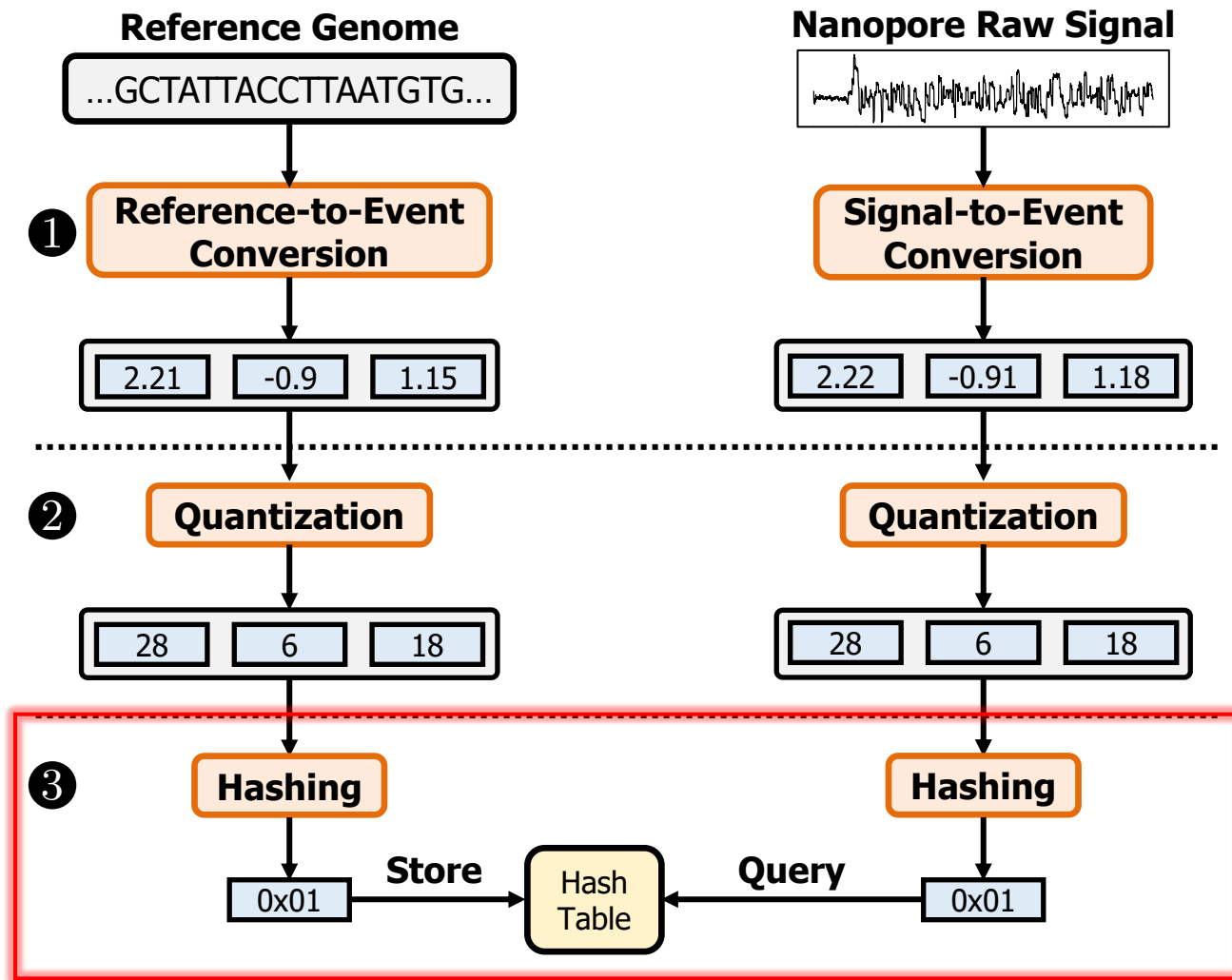


Quantizing the Event Values

- **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**

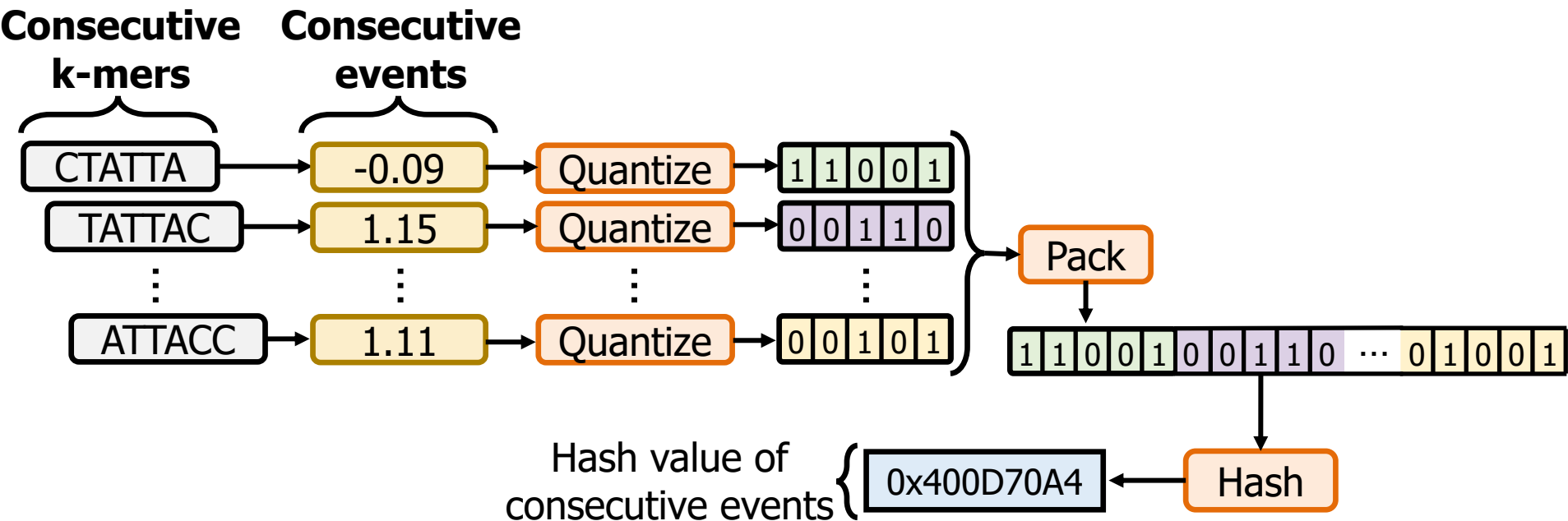


RawHash Overview

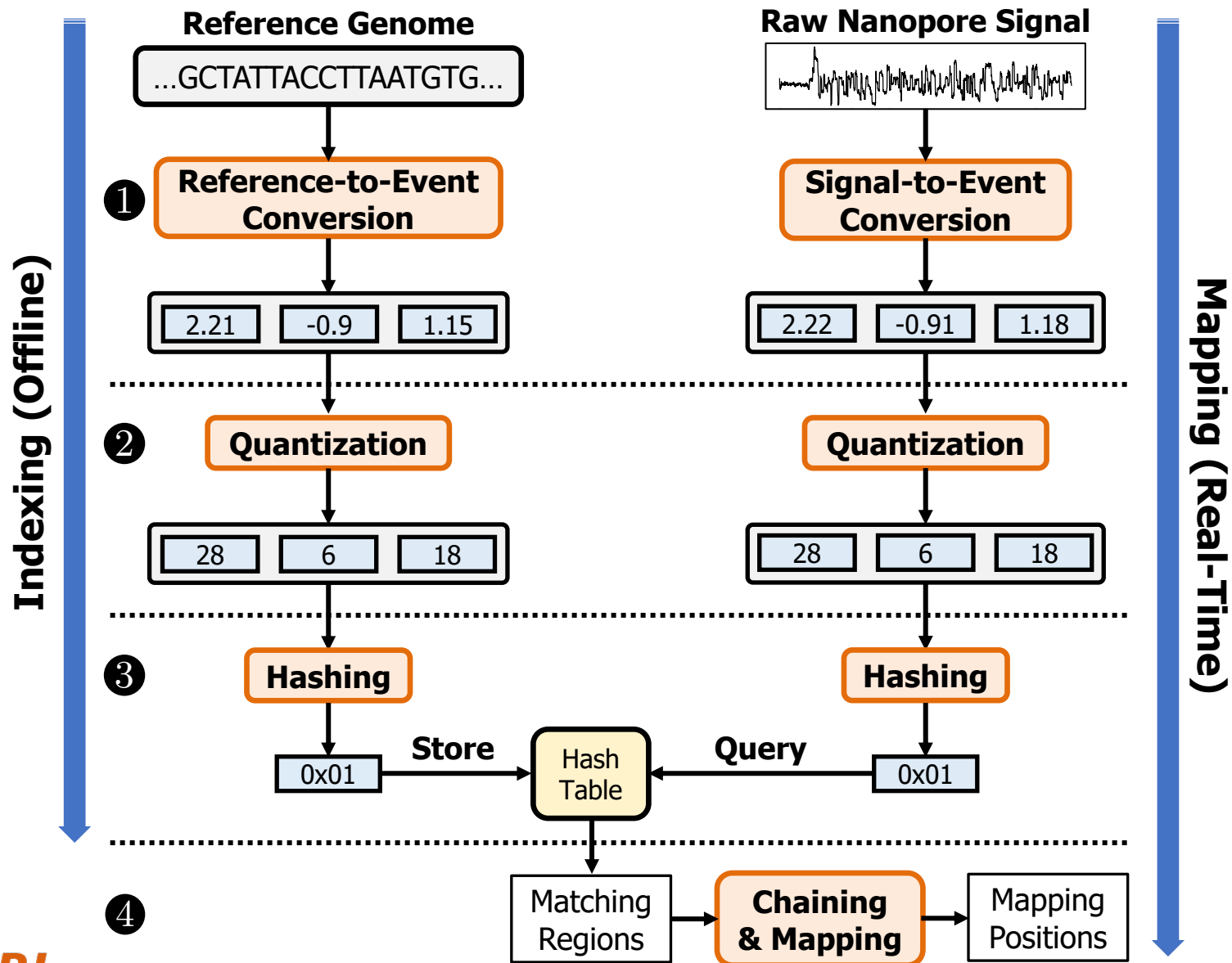


Hashing for Fast Similarity Search

- 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

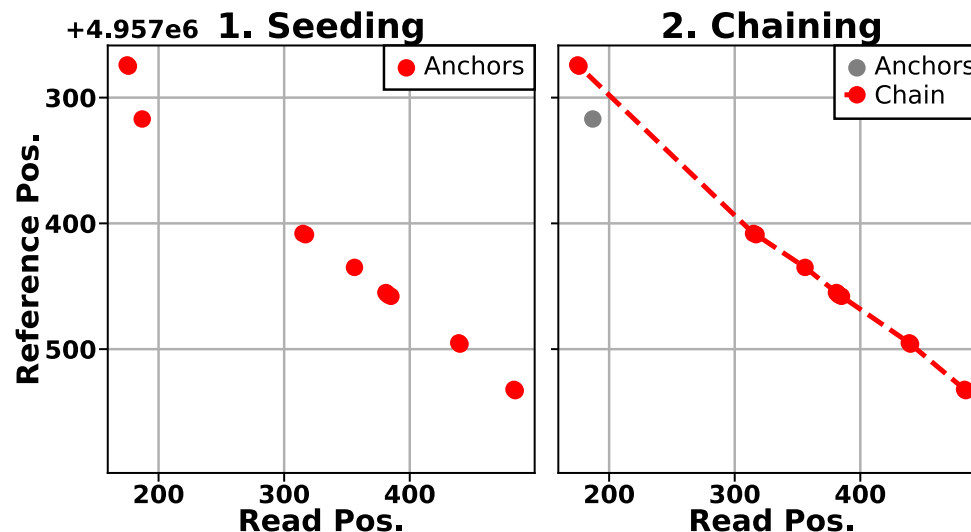


RawHash Overview

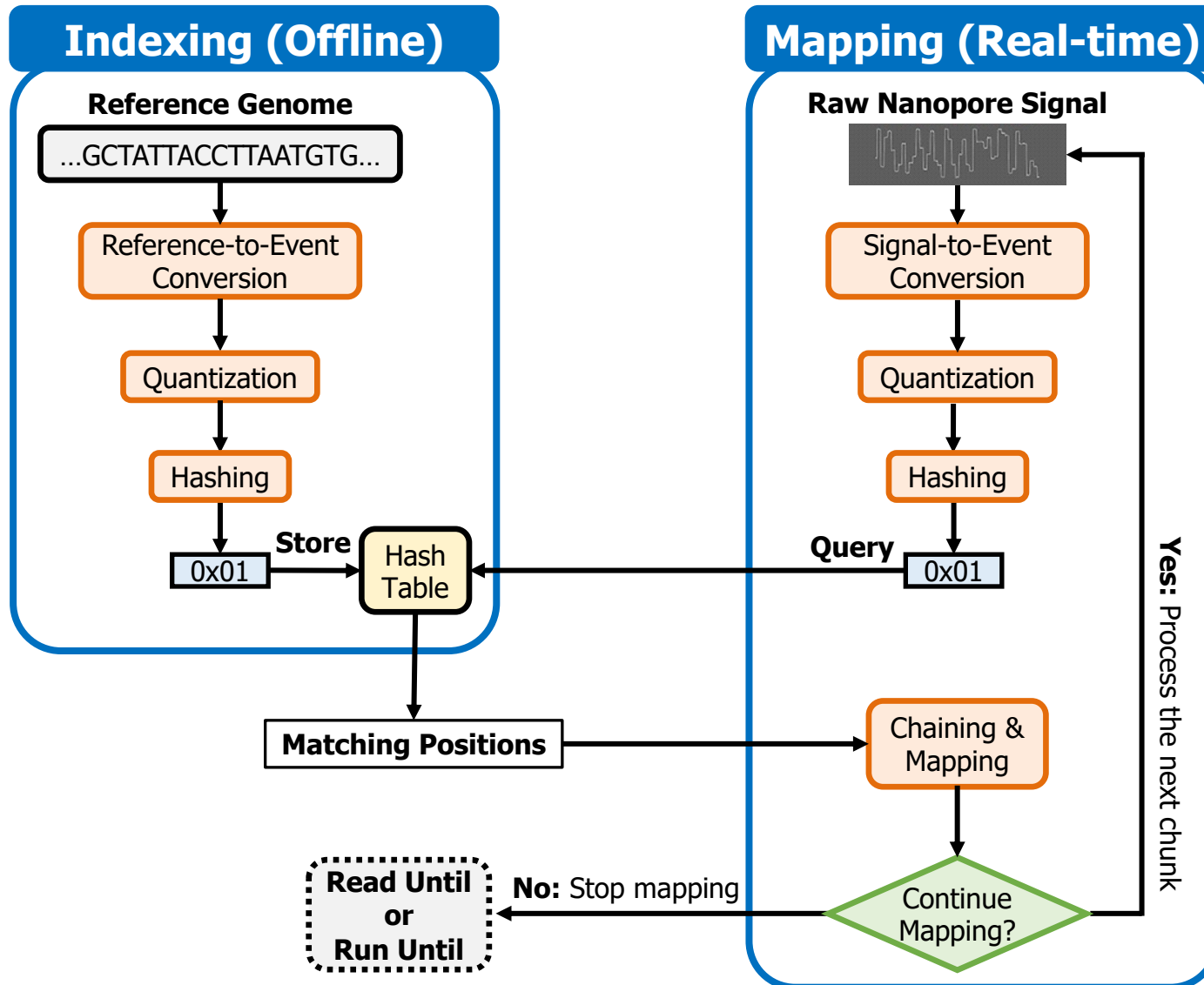


The Chaining Algorithm

- **Exact hash value matches:** Needed for finding matching regions between a reference genome and a read
- What if there are mutations or errors?
 - **No hash (seed) match** will occur in such positions
- The chaining algorithm links **exact matches in a proximity** even though there are gaps (no seed matches) between them



Real-Time Mapping using Hash-based Indexing





RawHash

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

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

Outline

Background

RawHash

RawHash2

Evaluation

Conclusion

Key Contributions in RawHash2

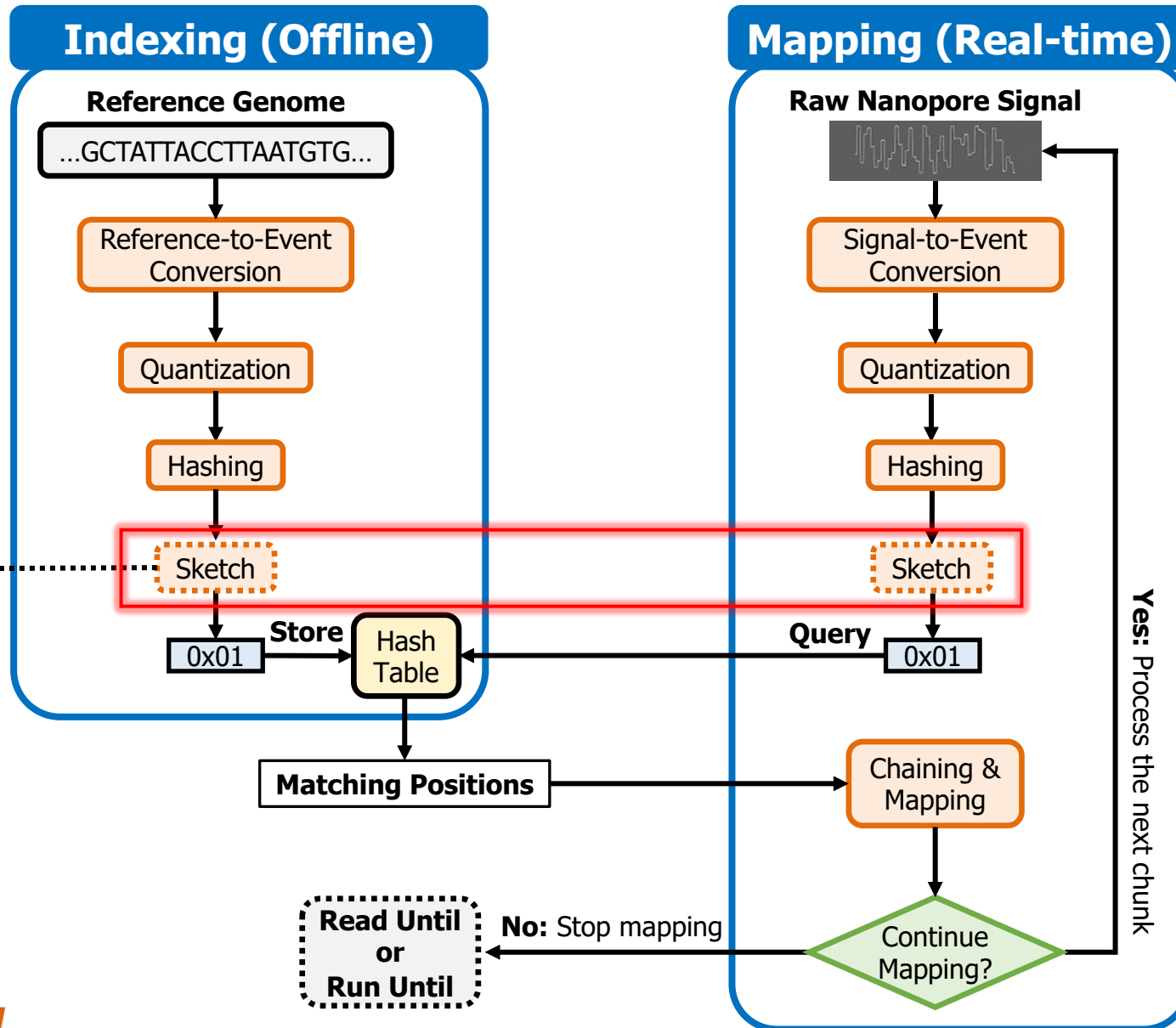
A new adaptive quantization to better fit the expected nanopore signal pattern to achieve **high accuracy**

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



Outline

Background

RawHash

RawHash2

Evaluation

Conclusion

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

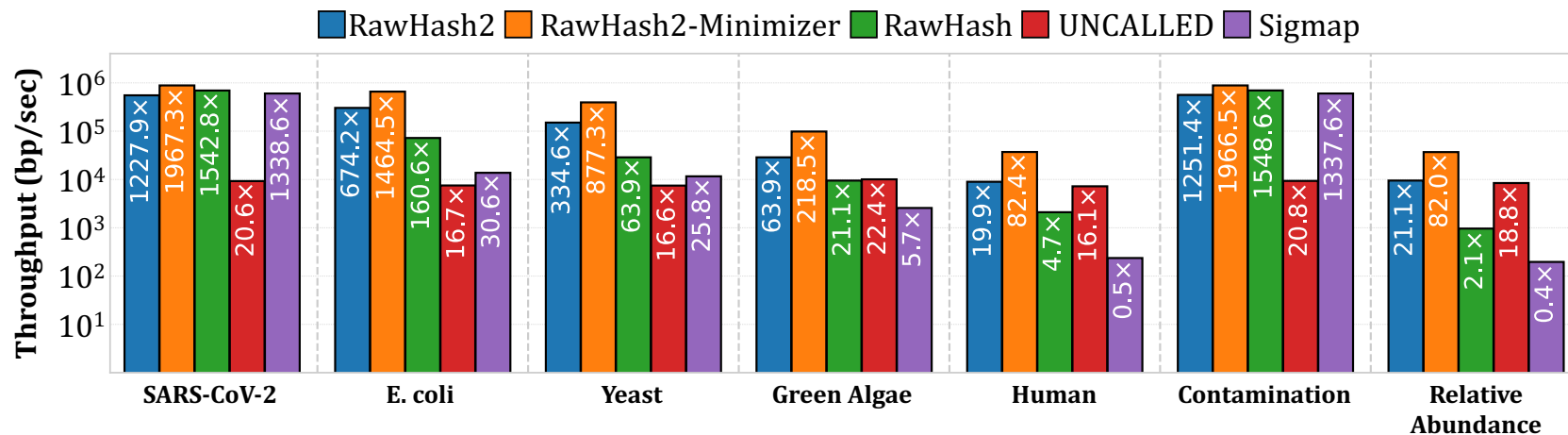
- Evaluation metrics:
 - **Throughput** (bases processed per second per CPU thread)
 - Potential reduction in **sequencing time and cost**
 - **Accuracy**
 - **Baseline:** Mapping basecalled reads using minimap2
 - Precision, recall, and F1 scores
 - Relative abundance estimation distance to ground truth

- **Datasets:**

	Organism	Reads (#)	Bases (#)	Genome Size
Read Mapping				
D1	<i>SARS-CoV-2</i>	1,382,016	594M	29,903
D2	<i>E. coli</i>	353,317	2,365M	5M
D3	<i>Yeast</i>	49,989	380M	12M
D4	<i>Green Algae</i>	29,933	609M	111M
D5	<i>Human HG001</i>	269,507	1,584M	3,117M
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 bp/sec (data generation speed)**

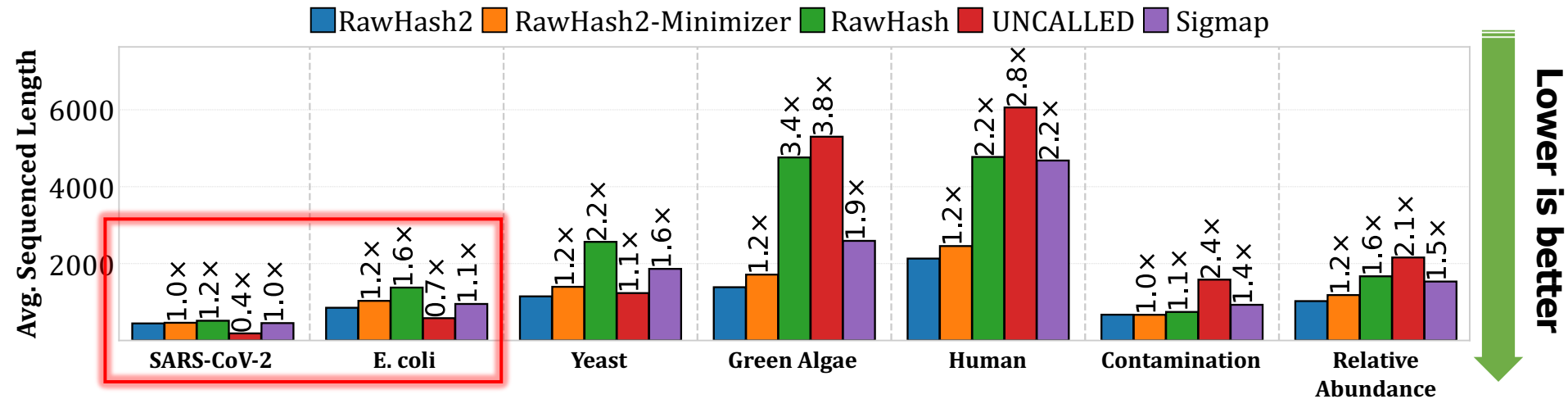


RawHash2: 27×, 19×, and 4× better average throughput compared to **UNCALLED**, **Sigmap** and **RawHash**, respectively

RawHash2-Minimizer further improves the throughput by **2.5×** compared to **RawHash2**

Average Sequenced Length

- Fewer bases to sequence → Less unnecessary sequencing



RawHash2 **reduces sequencing time and cost**

on average by **1.9x** compared to UNCALLED 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)

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

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

Tool	Estimated Relative Abundance Ratios in 50,000 Random Reads					
	<i>SARS-CoV-2</i>	<i>E. coli</i>	<i>Yeast</i>	<i>Green Algae</i>	<i>Human</i>	Distance
RawHash (100%)	0.0270	0.3636	0.3062	0.1951	0.1081	N/A
RawHash + Sequence Until (7%)	0.0283	0.3539	0.3100	0.1946	0.1133	0.0118

Sequence Until 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

Outline

Background

RawHash

RawHash2

Evaluation

Conclusion

Conclusion

Key Contributions:

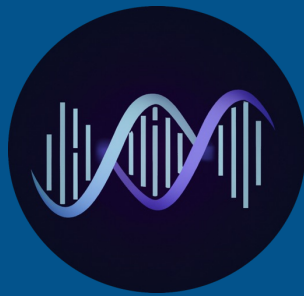
- 1) The **first hash-based mechanism** for mapping raw nanopore signals
- 2) The novel **Sequence Until** technique can accurately and **dynamically stop the entire sequencing of all reads at once** if further sequencing is not necessary

Key Results: Across 3 use cases and 5 genomes of varying sizes

- **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×**

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 Lindegger

Gagandeep Singh

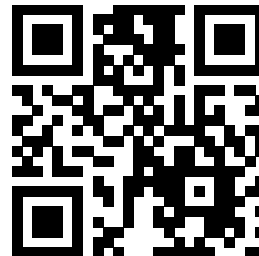
Meryem Banu Cavlak

Haiyu Mao

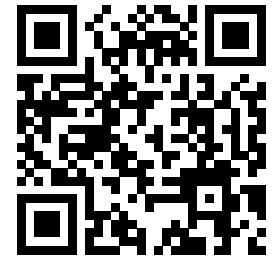
Onur Mutlu



[RawHash](#)



[RawHash2](#)



[Code](#)

Agenda for Today

- Cutting-edge in Accelerating Genome Analysis
- Enabling Fast and Accurate Sequence Analysis
 - Real-Time Analysis Nanopore Raw Signal Analysis
 - Fuzzy seed matching for accurate sequence analysis
- Conclusion



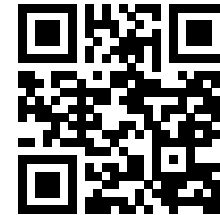
BLEND

A Fast, Memory-efficient and Accurate Mechanism to Find Fuzzy Seed Matches in Genome Analysis

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



[Paper \(NARGAB\)](#)



[Source Code](#)

SAFARI

ETH zürich

Carnegie Mellon

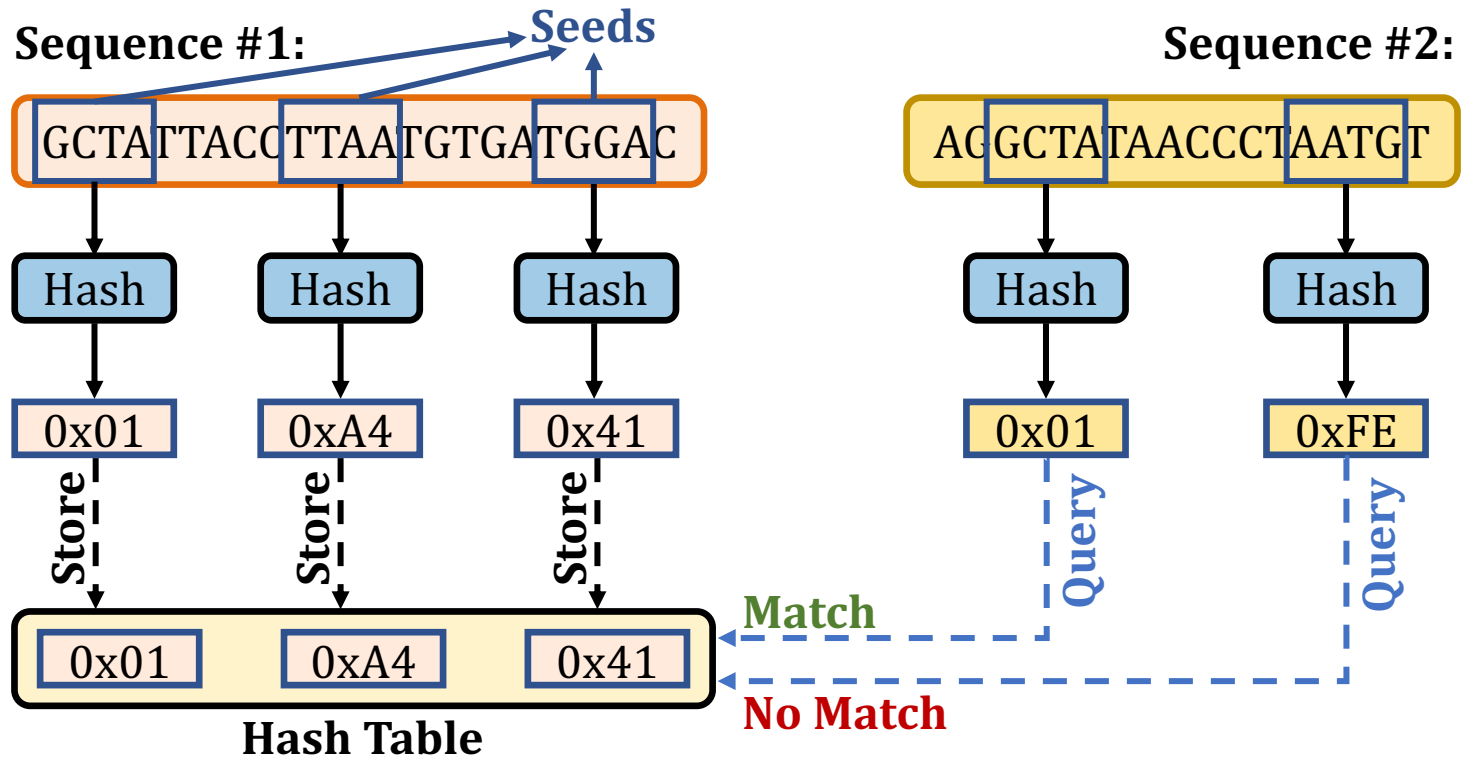

TU Delft

Delft University of Technology



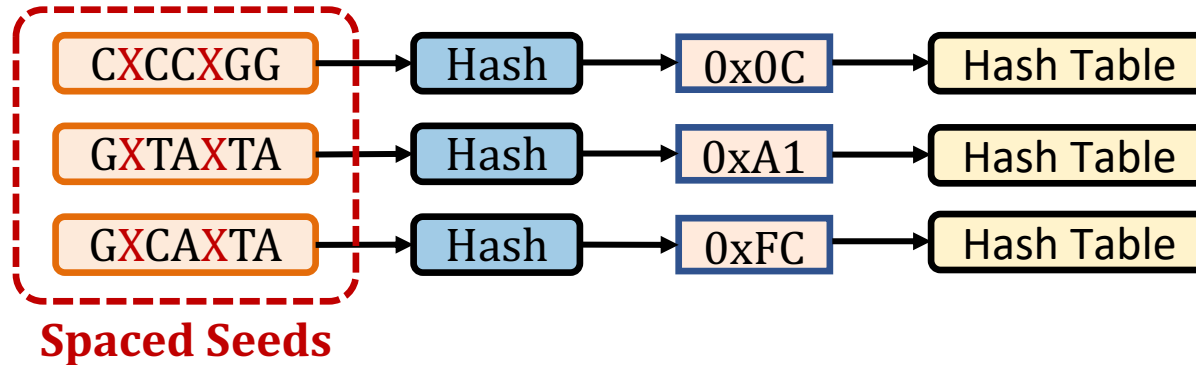
Bilkent University

Finding Seed Matches in Hash Tables



Hashing the Seeds

- Seeds are hashed
 - To find seed matches with a **direct lookup of hash values**

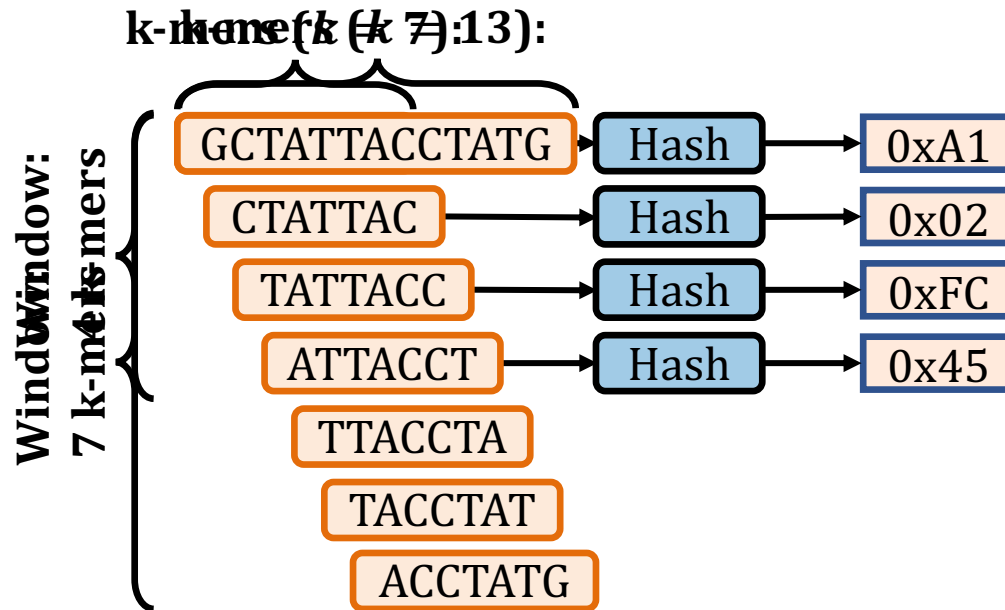


- **Low collision hash functions**

- Seeds must **match exactly** to generate the **same hash value**
- **Advantage:** **Dissimilar seeds** are **unlikely** to match
- **Limitation:** **Highly similar seeds (fuzzy seeds)** are **unlikely** to match

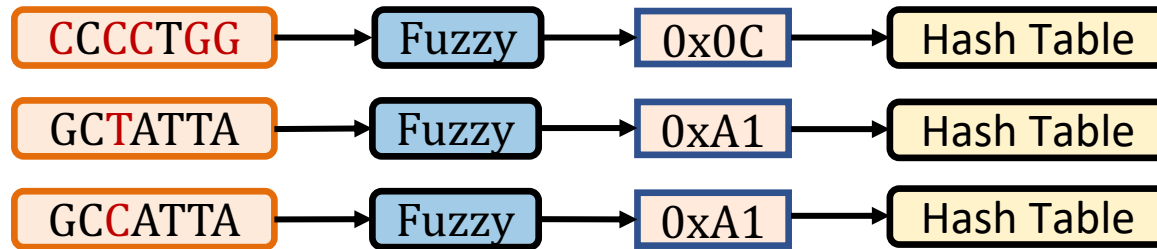
Challenges - Exact-matching seeds

- **Limitations** when adjusting many **seeding parameters**
 - K-mer size
 - Window length in minimizers (sampling ratio)
 - Sensitivity
- Trade-off between **performance, memory, and accuracy**



Opportunities - Fuzzy matches of seeds

- A mechanism for **finding fuzzy seed matches** enables
 - Assigning the **same** hash values to **highly similar seeds** with mismatches at **any arbitrary positions**
 - **Different** hash values for **dissimilar seeds**



- **Better accuracy:** Finding **useful and novel seed matches** that cannot be identified when finding only exact-matching seeds
- **Rethinking the seeding parameters** to achieve better trade-off between performance, memory, and accuracy

Our Goal

Enable finding **fuzzy matches of seeds**
as well as **exact-matching seeds**
with a **single lookup** of hash values of seeds



BLEND

Uses a hashing mechanism, **SimHash**, that can generate **the same hash values** for **similar sets**

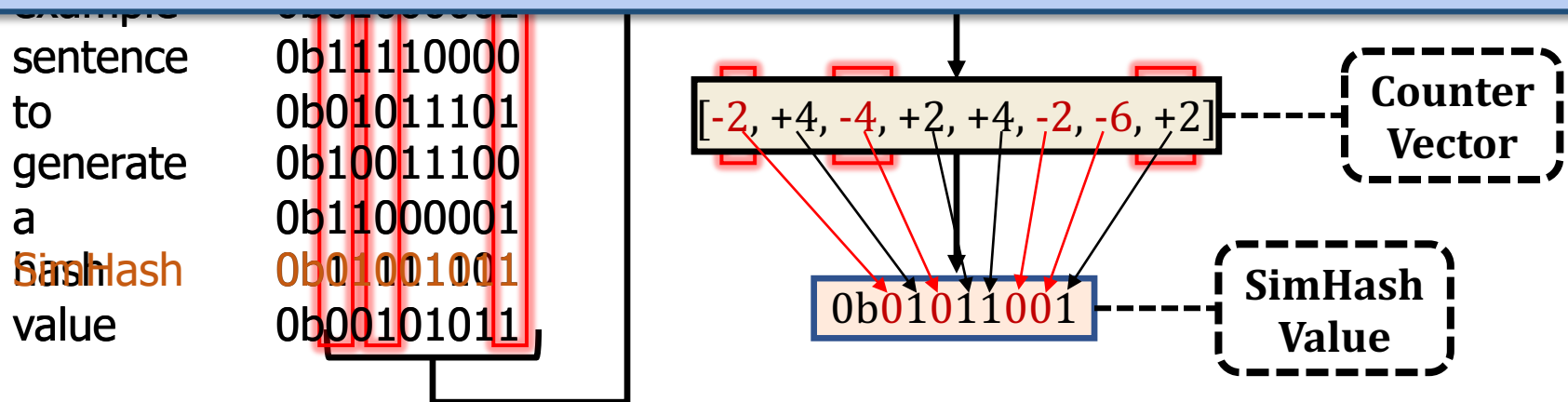
Provides the mechanisms for accurately and efficiently **converting seed sequences into set of items**

The SimHash Technique – Example

- Goal: Generate the **same hash value** for **similar set of items**
 - **Example input:** A sentence (a set of items)
 - **Items:** Words in a sentence (hash values of items)

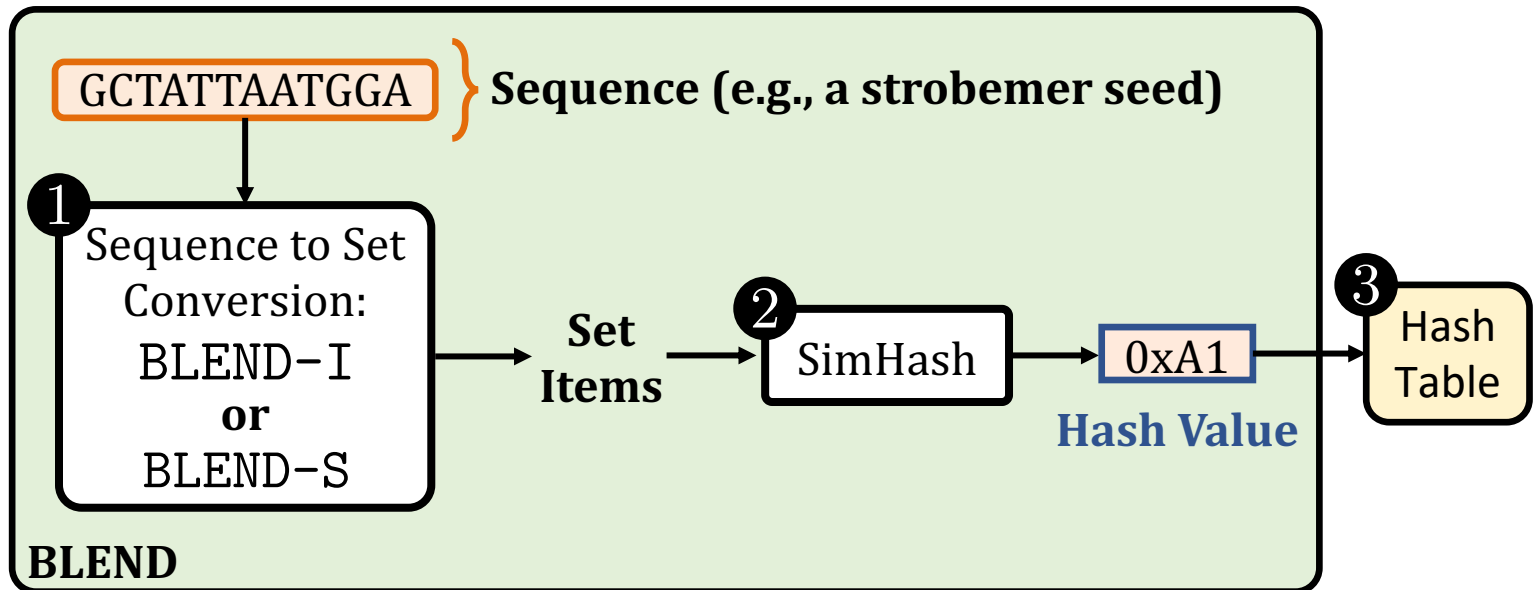
• Count the net difference between 0s and 1s at each position

Challenge: Efficiently and accurately
convert seeds to set of items
to use with SimHash



BLEND Overview

- **Goal:** Efficiently find fuzzy seed matches with a single lookup
 - **Input:** A fixed-length sequence (seed sequence)
- 1. Efficiently and accurately **convert the seed** to its **set of items**
 - Two conversion mechanisms: **BLEND-I** and **BLEND-S**
- 2. Generate the SimHash value of the seed
- 3. Efficiently identify fuzzy seed matches by matching the SimHash values using a hash table

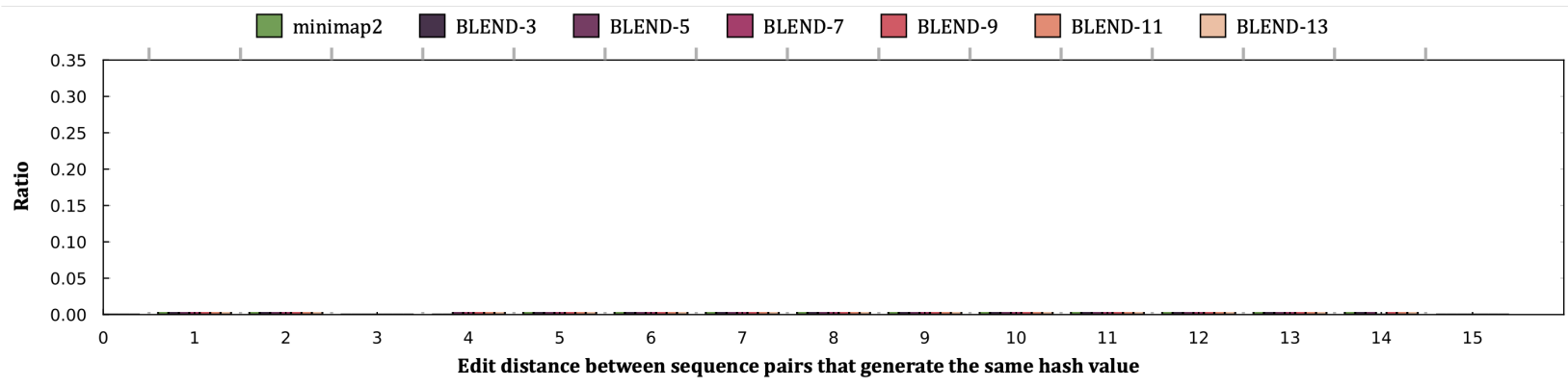


Evaluation Methodology

- **Integrated into minimap2** to perform end-to-end mapping
- Real and simulated datasets from
 - PacBio (HiFi and CLR), ONT, and Illumina reads
 - Human CHM13 and HG002, Fruit fly, Yeast, and E. coli genomes
- Use case 1: **Read overlapping** (all-vs-all overlapping)
 - Evaluated the **accuracy, completeness, and contiguity** of *de novo* assemblies generated from overlaps
- Use case 2: **Read mapping** to a reference genome
 - Mapping accuracy from simulated reads
 - Structural variant calling

Emprical Analysis on Fuzzy Seed Matching

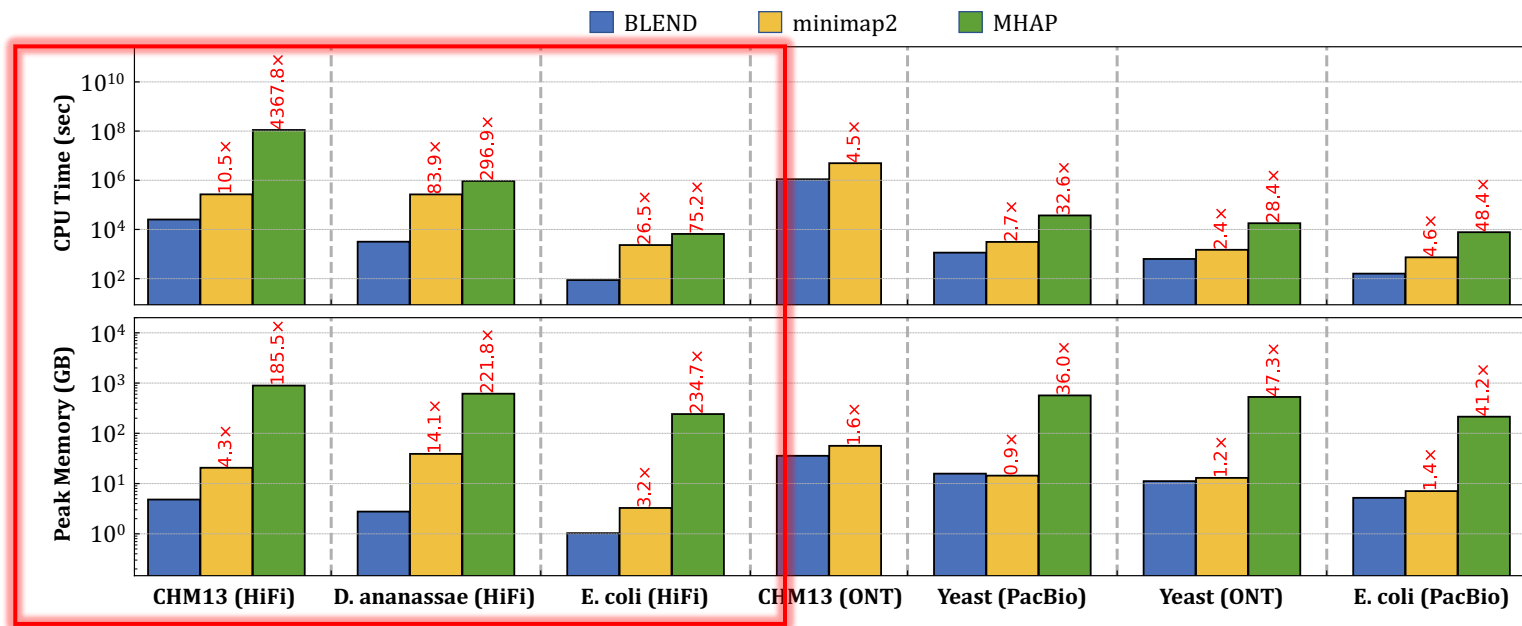
- Edit distance between sequence pairs with a hash collision
 - **Goal in fuzzy seed matching:** Increase the collision rate for highly similar seeds



BLEND **increases the collision rate** for **highly similar seeds**

while keeping a **collision rate similar to minimap2** for **dissimilar seeds**

Read Overlapping – Performance & Memory



For HiFi: Average **speedup** of 40.3x (minimap2)

Reducing the **memory** footprint by 7.2x

Improving critical parameters without hurting the accuracy:

Window length (200) and **seed length** (31-mers)

Read Overlapping – Assembly Evaluation

Dataset	Tool	Average Identity (%)	Genome Fraction (%)	<i>k</i> -mer Compl. (%)	Largest Contig (Mb)	NGA50 (kb)	NG50 (kb)
<i>CHM13</i> (HiFi)	BLEND	99.8526	98.4847	90.15	22.8397	5442.25	5442.31
	minimap2	99.7421	97.1493	83.05	47.1387	7133.43	7134.31
	MHAP	N/A	N/A	N/A	N/A	N/A	N/A
	Reference	100	100	100	248.387	154 260	154 260
<i>D. ananassae</i> (HiFi)	BLEND	99.7856	97.2308	86.43	6.23256	792.407	798.913
	minimap2	99.7044	96.3190	72.33	4.43396	273.398	278.775
	MHAP	99.5551	0.7276	0.21	0.028586	N/A	N/A
	Reference	100	100	100	30.6728	26 427.4	26 427.4
<i>E. coli</i> (HiFi)	BLEND	99.8320	99.8801	87.91	3.41699	3416.99	3416.99
	minimap2	99.7064	99.8748	79.27	3.08849	3087.05	3087.05
	MHAP	N/A	N/A	N/A	N/A	N/A	N/A
	Reference	100	100	100	4.94446	4944.46	4944.46

For HiFi: More accurate and complete assemblies

Better contiguity in most cases

Fuzzy seed matches can lead to finding **novel and useful overlaps**

Read Mapping – SV Calling

- Structural variant (SV) calling using read mappings from each tool
 - Sniffles2 to call SVs from HG002 long read mappings
 - Truvari to compare the resulting SVs with the benchmarking SV set (Tier 1 set from GIAB)

Tool	HG002 SVs (high-confidence tier 1 SV set)					
	TP (#)	FP (#)	FN (#)	Precision	Recall	F_1
BLEND	9229	855	412	0.9152	0.9573	0.9358
minimap2	9222	915	419	0.9097	0.9565	0.9326
LRA	9155	830	486	0.9169	0.9496	0.9329
Winnomap2	9170	1029	471	0.8991	0.9511	0.9244

Best overall accuracy in downstream analysis

BLEND Summary

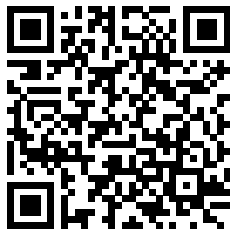
Problem	Finding exact-matching seeds introduce limitations in further improving the performance and accuracy of genome analysis
Goal	Enable finding the fuzzy seed matches as well as the exact-matching seeds accurately and efficiently
BLEND	<ul style="list-style-type: none">• Provides effective mechanisms for converting seeds into set of items to use with the SimHash technique• Uses the SimHash technique to enable generating the same hash value for highly similar seeds to find fuzzy seed matches with a directly lookup of hash values
Key Results	<ul style="list-style-type: none">• Significant speedups and lower memory footprint especially when using HiFi reads• Improves the accuracy of important applications in genome analysis



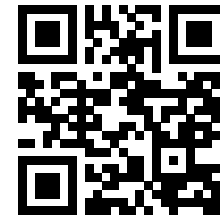
BLEND

A Fast, Memory-efficient and Accurate Mechanism to Find Fuzzy Seed Matches in Genome Analysis

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



[Paper \(NARGAB\)](#)



[Source Code](#)

SAFARI

ETH zürich

Carnegie Mellon


TU Delft

Delft University of Technology



Bilkent University

Agenda for Today

- Cutting-edge in Accelerating Genome Analysis

- Enabling Fast and Accurate Sequence Analysis
 - Real-Time Analysis Nanopore Raw Signal Analysis
 - Fuzzy seed matching for accurate sequence analysis

- 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

Overlapping Raw Signals in Real-Time

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

Can Firtina¹ Maximilian Mordig^{1,2} Joël Lindegger¹ Harun Mustafa^{1,3,4} Sayan Goswami¹
Stefano Mercogliano¹ Yan Zhu^{1,5} Andre Kahles^{1,3,4} Onur Mutlu¹
¹*ETH Zurich* ²*Max Planck Institute for Intelligent Systems* ³*University Hospital Zurich*
⁴*Swiss Institute of Bioinformatics* ⁵*University of Toronto*

Real-time *de novo* Assembly Construction

- All-vs-all overlapping using raw signals

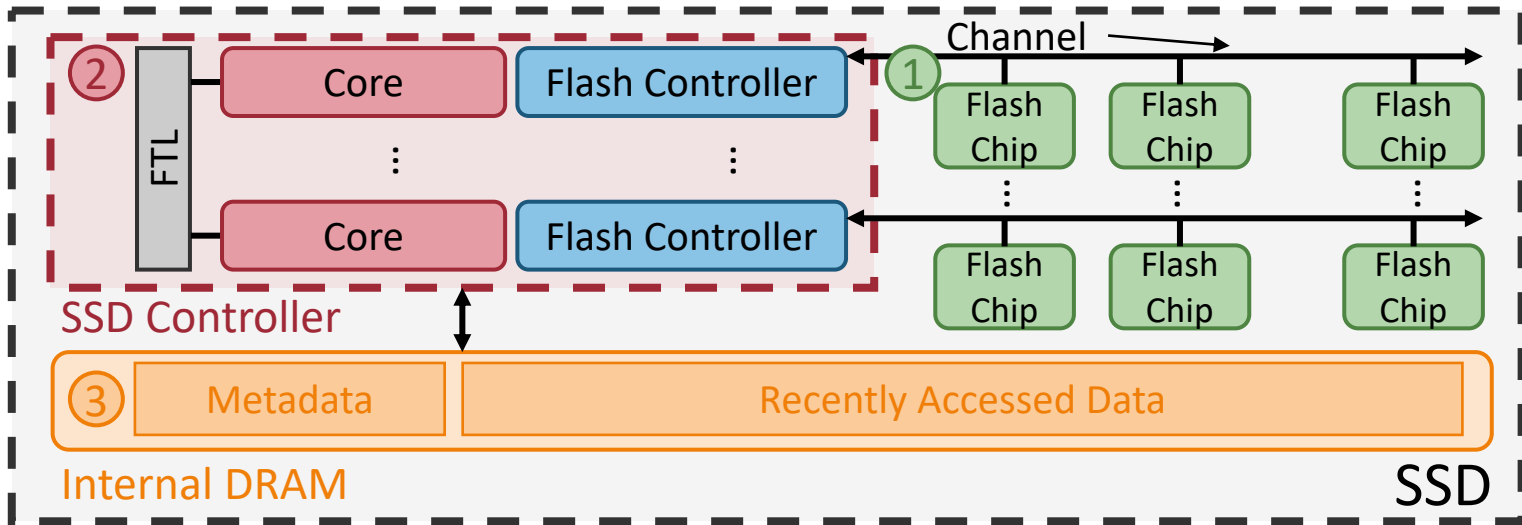
	Organism	Shared Overlaps (%)	Unique to Rawsamble (%)	Unique to Minimap2 (%)
D1	<i>E. coli</i>	44.57	15.08	40.35
D2	<i>Yeast</i>	47.07	35.62	17.32
D3	<i>Human</i>	19.73	27.56	52.71

- Building *de novo* assemblies directly from raw signals

Organism	Tool	No. of contigs	Avg. Contig Length	Max. Contig Length
D1	Rawsamble	39	373,594	1,431,572
<i>E. coli</i>	minimap2	4	2,611,044	5,210,938
D2	Rawsamble	431	47,250	256,116
<i>Yeast</i>	minimap2	278	82,757	386,005
D3	Rawsamble	59	16,376	66,163
<i>Human</i>	minimap2	53	10,572	42,654

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
- Full downstream analysis fully using raw nanopore signals
- Cooperating the raw signal analysis and basecalled sequence analysis together
 - While also utilizing fuzzy seed matching for better sensitivity
- 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



emea.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html
emea.illumina.com/company/news-center/press-releases/2018/2349147.html
illumina.com/content/dam/illumina/gcs/assembled-assets/marketing-literature/dragen-bio-it-data-sheet-m-gl-00680/dragen-bio-it-data-sheet-m-gl-00680.pdf

Nova/NextSeq with Analysis Capability



Scale your studies with ease

Process high-throughput data quickly with hardware acceleration

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.

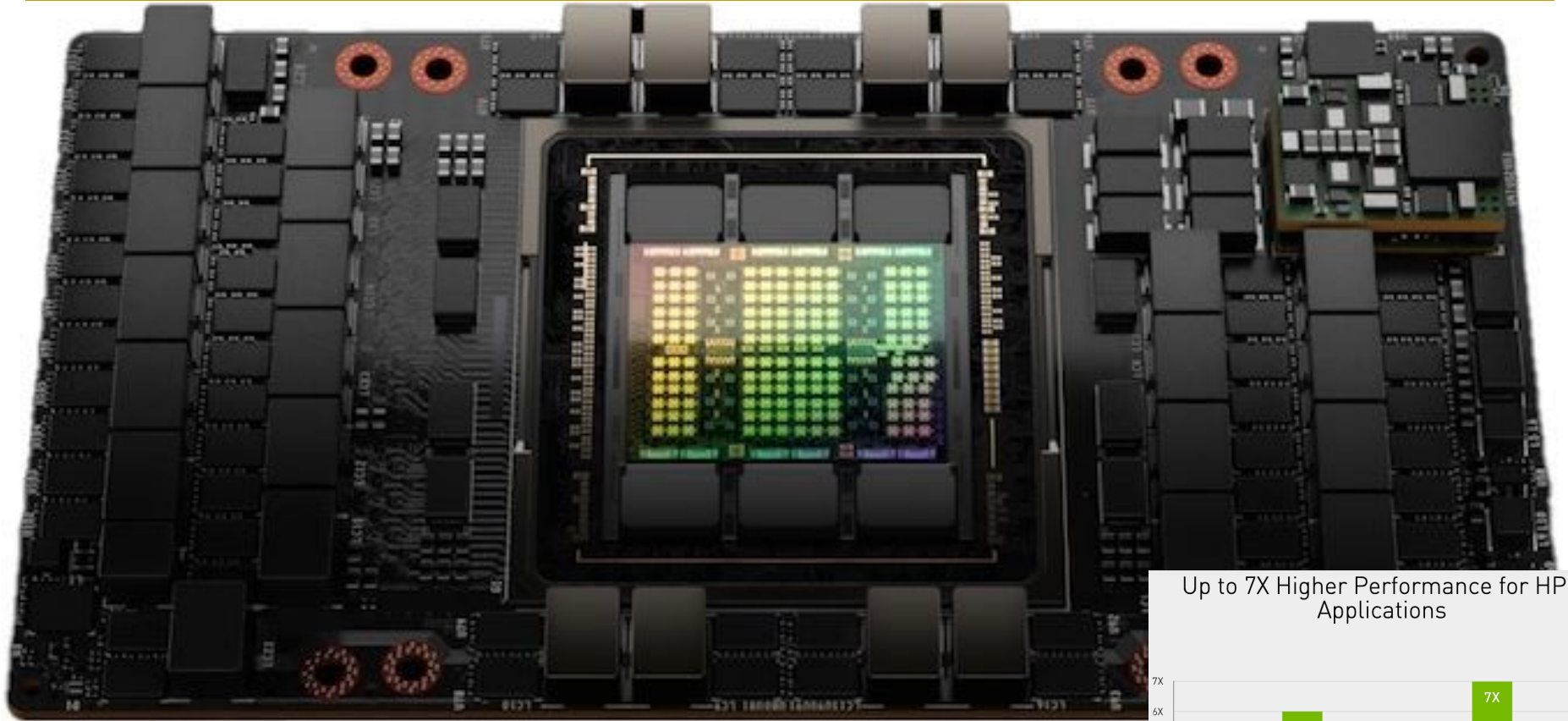
Reduce data footprint, manage and store data easily with lower costs and lower energy consumption, with built-in compression that reduces FASTQ file sizes by up to 80%.

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

<https://www.illumina.com/content/dam/illumina/gcs/assembled-assets/marketing-literature/dragen-bio-it-data-sheet-m-gl-00680/dragen-bio-it-data-sheet-m-gl-00680.pdf>

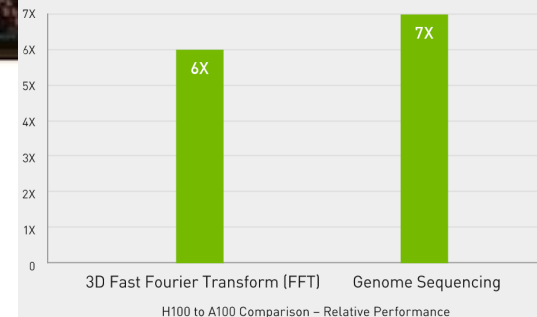
<https://emea.illumina.com/systems/sequencing-platforms/novaseq-x-plus/products-services/software.html>

NVIDIA H100 (2022)

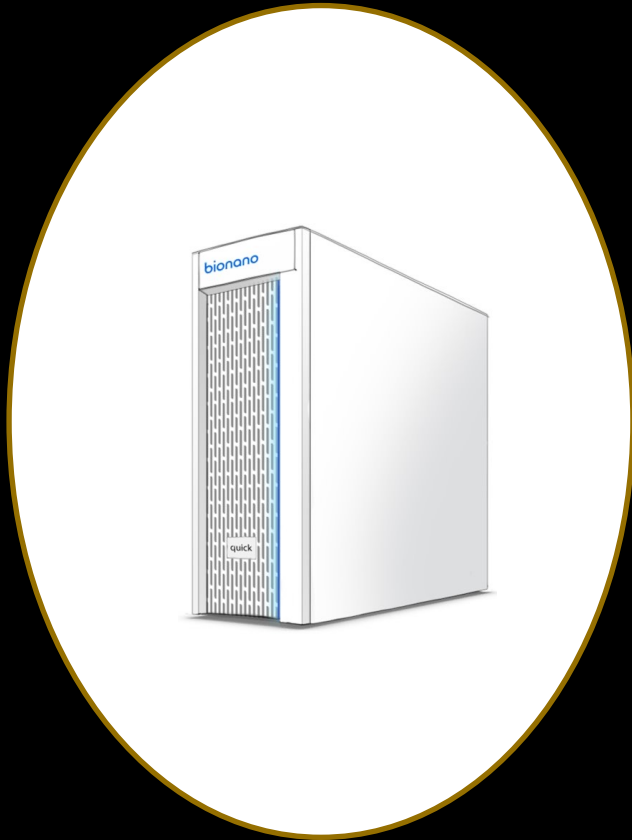


NVIDIA is claiming a **7x improvement** in dynamic programming algorithm (**DPX instructions**) performance on a single H100 versus naïve execution on an A100.

Up to 7X Higher Performance for HPC Applications



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



Bionano & NVIDIA:

Accelerating Analysis for Fast Time to Results



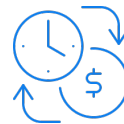
Technological solution to **support higher throughput**



New high-performance algorithms from Bionano



Powered by NVIDIA RTX™ 6000 Ada Generation GPUs

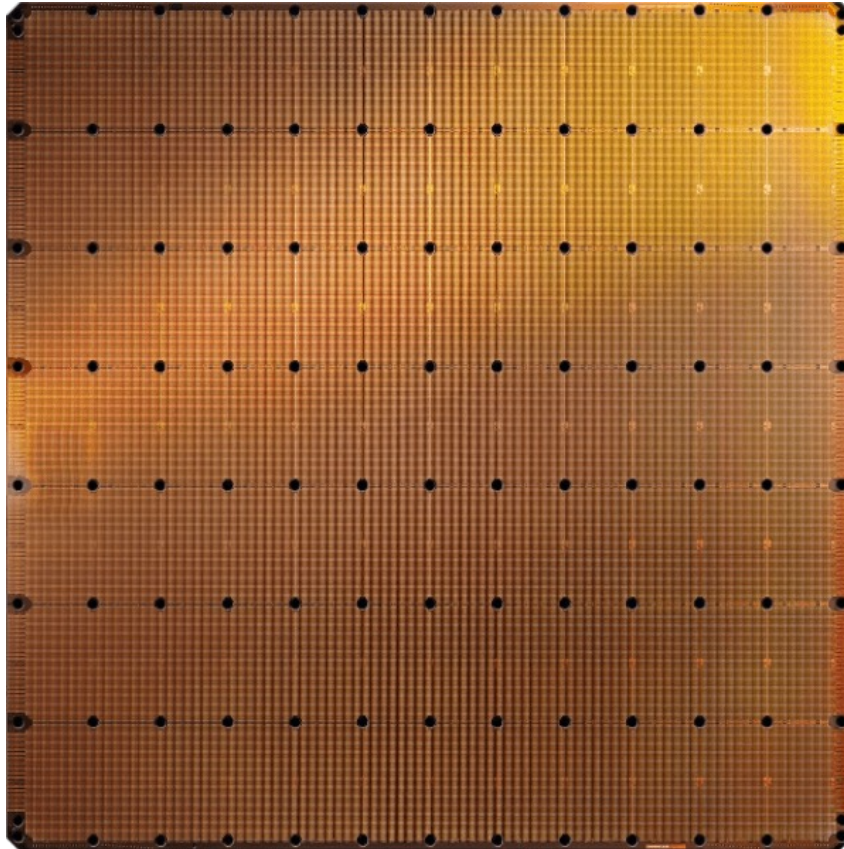


Analysis of highly complex cancer whole genomes in **less than 2 hours**



Workflow tailored for a **small lab and IT footprint**

Cerebras's Wafer Scale Engine (2021)



Cerebras WSE-2
2.6 Trillion transistors
46,225 mm²

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



Largest GPU
54.2 Billion transistors
826 mm²
NVIDIA Ampere GA100

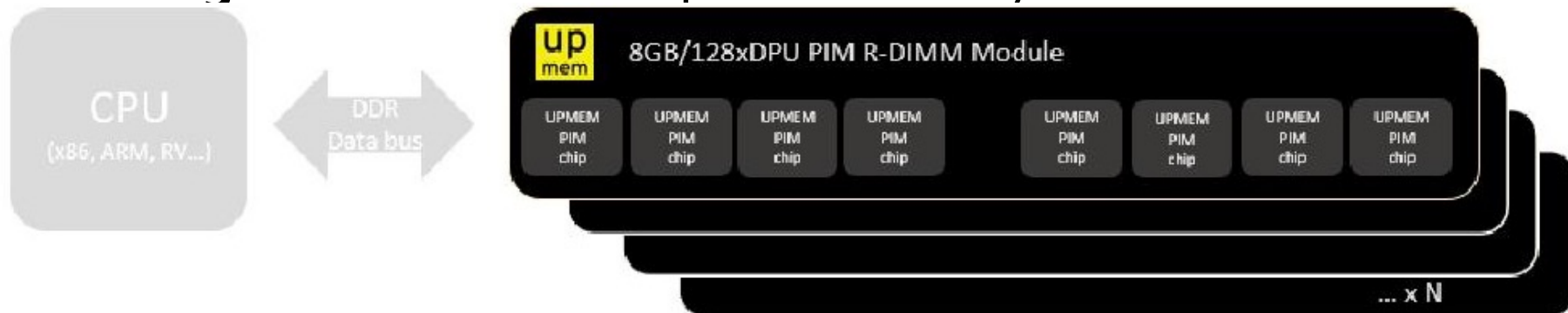
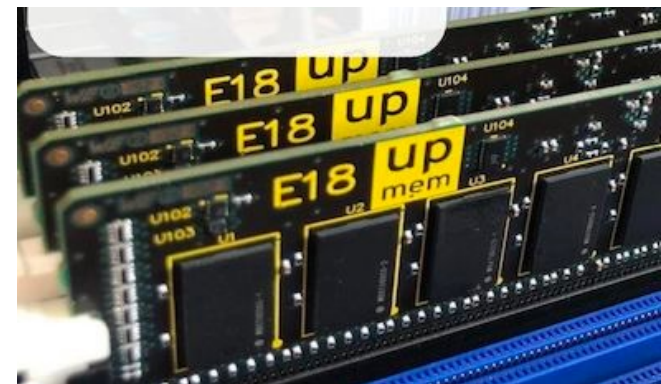
<https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning>

SAFARI

<https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/>

UPMEM Processing-in-DRAM Engine (2019)

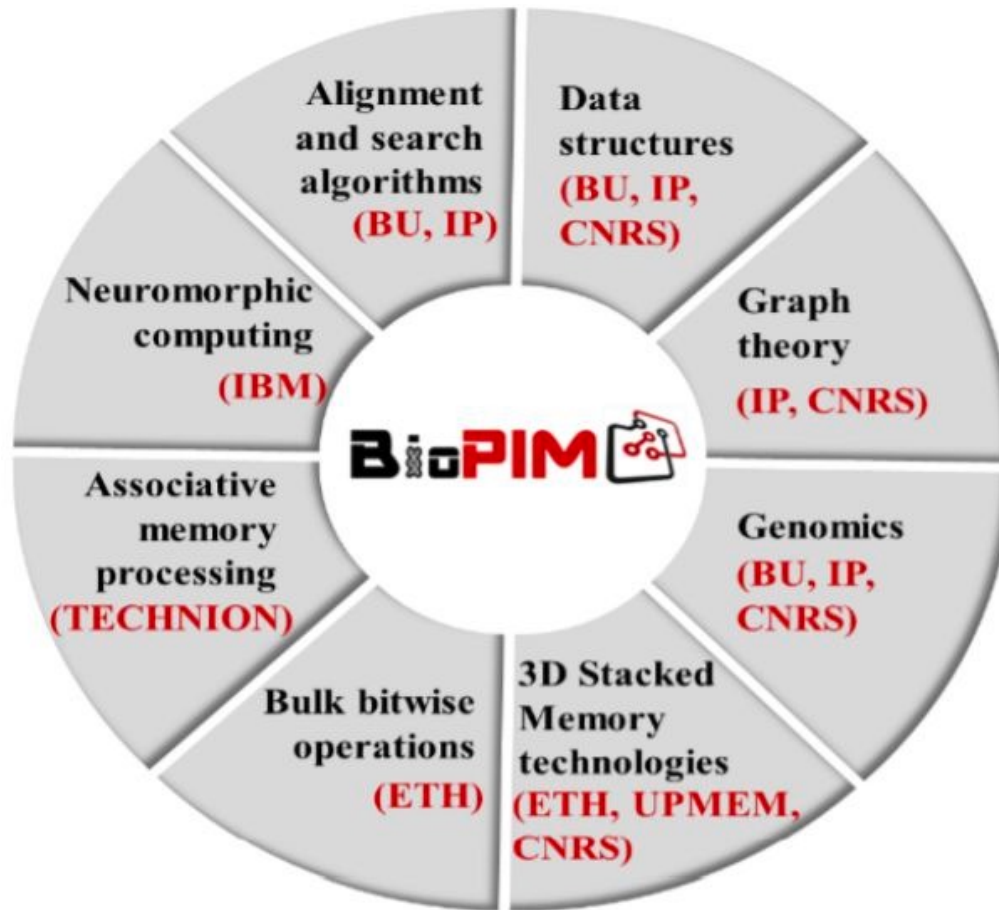
- **Processing in DRAM Engine**
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.
- Replaces **standard DIMMs**
 - DDR4 R-DIMM modules
 - 8GB+128 DPUs (16 PIM chips)
 - Standard 2x-nm DRAM process
 - **Large amounts of** compute & memory bandwidth



<https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem>

<https://www.upmem.com/video-upmem-presenting-its-true-processing-in-memory-solution-hot-chips-2019/>

BioPIM (2022)



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"
Invited Lecture at [Technion](#), Virtual, 26 January 2021.
[[Slides \(pptx\)](#) ([pdf](#))]
[[Talk Video](#) (1 hour 37 minutes, including Q&A)]
[[Related Invited Paper \(at IEEE Micro, 2020\)](#)]

Insight: Shifting a String Helps Similarity Search

7 matches 1 mismatch

ISTANBUL

ISTNBUL

ISTNBUL

81

46:08 / 1:37:37

Onur Mutlu

Onur Mutlu - Invited Lecture @Technion: Accelerating Genome Analysis: A Primer on an Ongoing Journey

566 views · Premiered Feb 6, 2021

31 0 SHARE SAVE ...

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 \(at Computational and Structural Biology Journal, 2022\)](#)]

A Bright Future for Intelligent Genome Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu
"Accelerating Genome Analysis: A Primer on an Ongoing Journey" IEEE Micro, August 2020.

Accelerating Genome Analysis: A Primer on an Ongoing Journey
Sept-Oct 2020, pp. 46-75, vol. 40
DOI Bookmark: 10.1109/MM.2020.3013726

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
July-Aug. 2021, pp. 38-48, vol. 41
DOI Bookmark: 10.1109/MM.2021.3068396

MinION from ONT

SmidgION from ONT

Accelerating Genome Analysis - Onur Mutlu's Invited Talk at the Barcelona Supercomputing Center

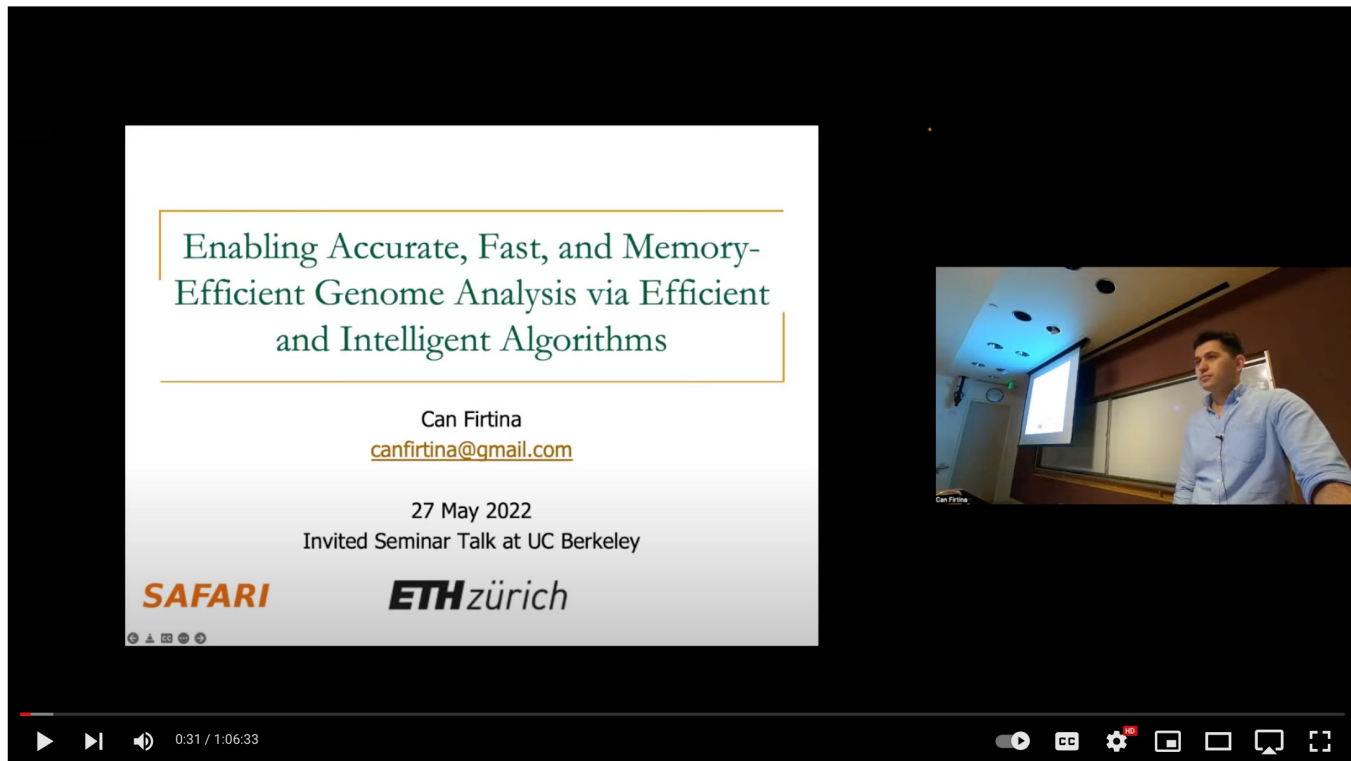
Onur Mutlu Lectures
36.6K subscribers

413 views Premiered Feb 2, 2023
Invited Talk at the Barcelona Supercomputing Center (BSC)
Accelerating Genome Analysis - A Primer on an Ongoing Journey

Presenter: Professor Onur Mutlu (<https://people.inf.ethz.ch/omutlu/>)
Date: September 6, 2022
1 hour 35 minutes (including Q&A)

More on Accelerating Genome Analysis

- Can Firtina,
"Enabling Accurate, Fast, and Memory-Efficient Genome Analysis via Efficient and Intelligent Algorithms"
Talk at UC Berkeley, Berkeley, CA, United States, May 27, 2022.
[\[Slides \(pptx\) \(pdf\)\]](#)
[\[Talk Video \(1 hour 6 minutes\)\]](#)



Enabling Accurate, Fast, and Memory-Efficient Genome Analysis - Can Firtina (Talk at UC Berkeley)

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)

RawHash – Key Idea

Key Observation: Identical nucleotides generate similar raw signals

The diagram illustrates the process: Raw Signal #1 and Raw Signal #2 are compared via Distance Calculation (marked with a red X). Both signals are processed through a Hash function to produce the same output, 0x01, which is then compared via Fast Match.

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

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

SAFARI 14

RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals | ISMB-ECCB 2023

Onur Mutlu Lectures
36.1K subscribers

Analytics Edit video

Share Save

294 views Premiered Aug 15, 2023
Talk of "RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes" at ISMB-ECCB 2023
Presenter: Can Firtina
Duration: 18:58 minutes

Accelerating Genome Analysis [DAC 2023]

- Onur Mutlu and Can Firtina,
"Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
Invited Special Session Paper in Proceedings of the 60th Design Automation Conference (DAC), San Francisco, CA, USA, July 2023.
[[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

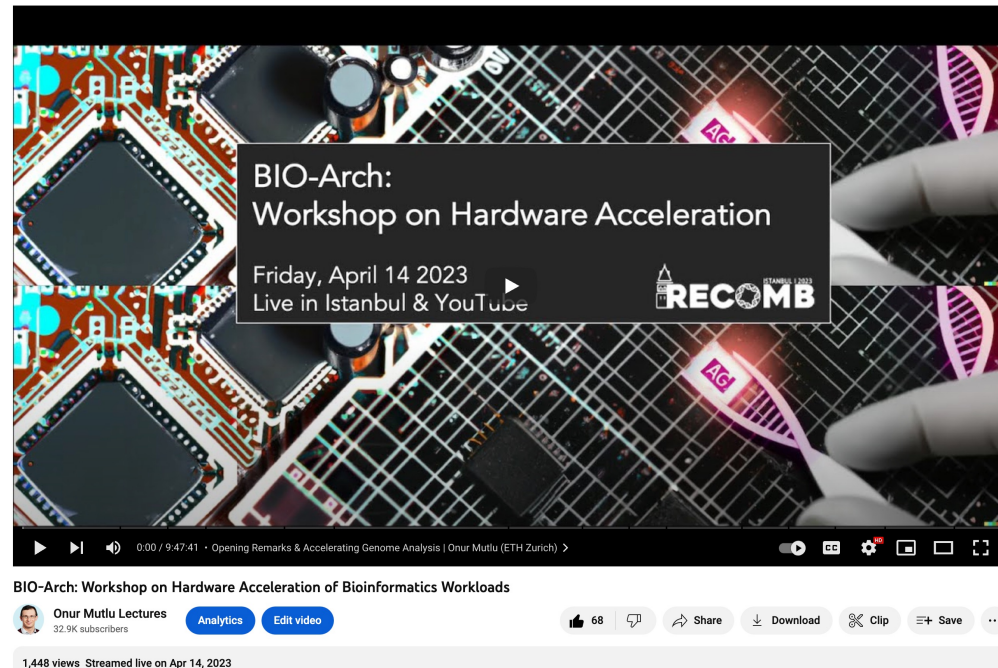
About

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>

SAFARI

<https://safari.ethz.ch/recomb23-arch-workshop/>

Genomics Course (Spring 24)

Spring 2024 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2024/doku.php?id=bioinformatics

Fall 2023 Edition:

- https://safari.ethz.ch/projects_and_seminars/fall2023/doku.php?id=bioinformatics

Youtube Livestream (Fall 2023):

- https://youtube.com/playlist?list=PL5Q2soXY2Zi_00wyOjiMShG4t2QPZoeE3

Project course

- Taken by Bachelor's students
- Genomics lectures
- Hands-on research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>

Complete Lecture Playlist (Spring 2023):

Fall 2023 Schedule

Week	Date	Livestream	Meeting
W0	05.10 Thu.		L0: Project Introductions and Q&A
W1	11.10 Wed.	YouTube Live	L1: P&S Course Introduction & Scope PDF PPT
W2	25.10 Wed.		L2: Introduction to Genome Analysis PDF PPT
W3	01.11 Wed.		L3: From Molecules to Data: An Overview of DNA Sequencing Technologies PDF PPT
W4	08.11 Wed.		L4a: Fundamentals of Sequence Alignment: Algorithms and Applications PDF PPT L4b: Optimizing Sequence Search: Hashing, Indexing, and Filtering Techniques PDF PPT

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 Sequence Analysis

from Raw Nanopore Electrical Signals
with Accurate Hash-based Search Mechanisms

Can Firtina

canfirtina@gmail.com

<https://cfirtina.com>

10 May 2024

The Broad Institute of MIT and Harvard

SAFARI

ETH zürich

A Tsunami of Sequencing Data

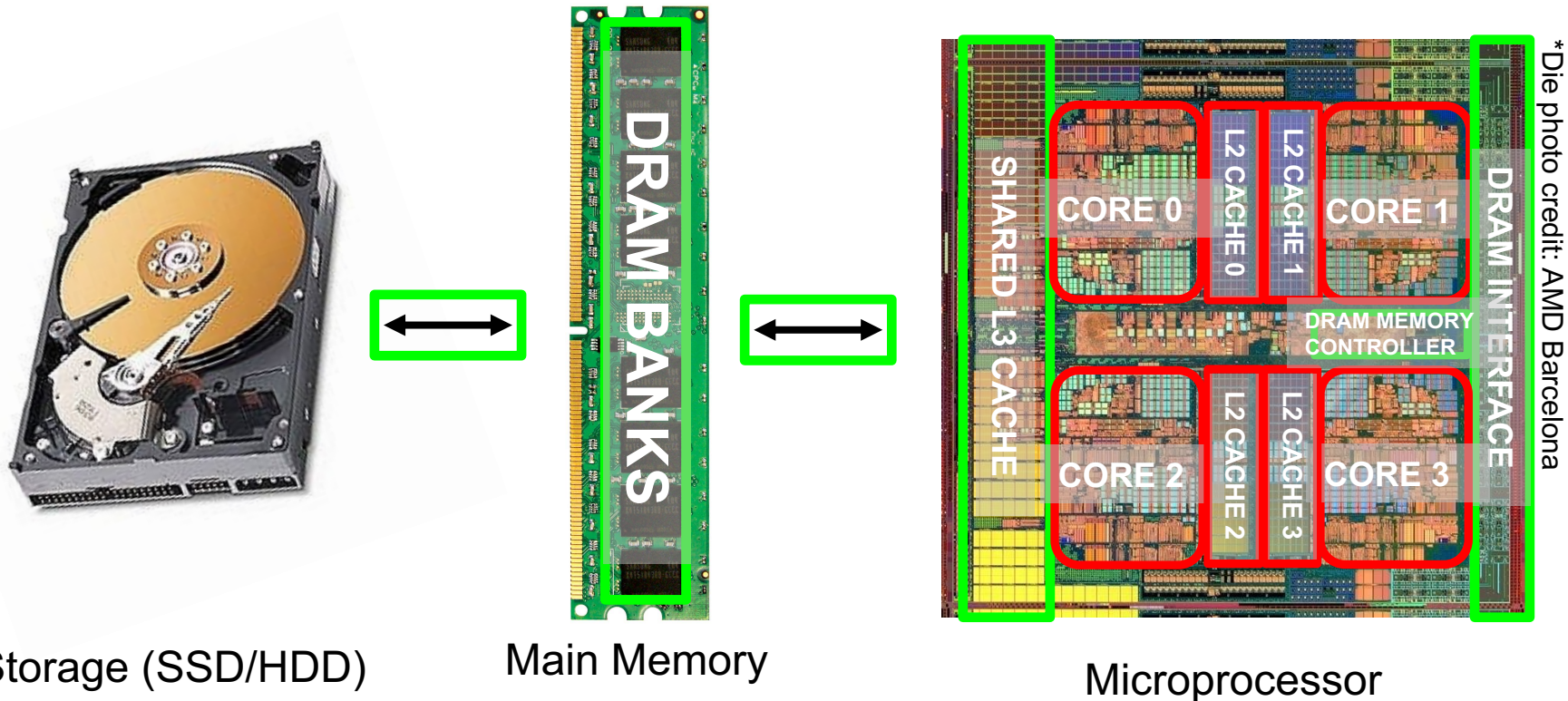
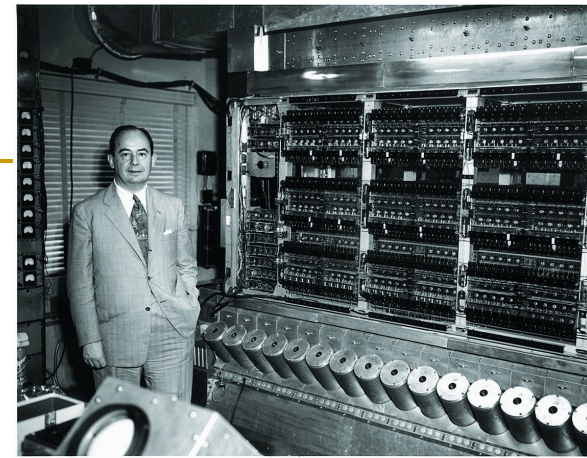
A Tera-scale increase in sequencing production in the past 25 years		
Genes & Operons	1990	Kilo = 1,000
Bacterial genomes	1995	Mega = 1,000,000
Human genome	2000	Giga = 1,000,000,000
Human microbiome	2005	Tera = 1,000,000,000,000
50K Microbiomes	2015	Peta = 1,000,000,000,000,000
what is expected for the next 15 years ? (a Giga?)		
200K Microbiomes	2020	Exa = 1,000,000,000,000,000,000
1M Microbiomes	2025	Zetta = 1,000,000,000,000,000,000,000
Earth Microbiome	2030	Yotta = 1,000,000,000,000,000,000,000,000

Source:
[@kyrpides](#)

Today's Computing Systems

von Neumann model, 1945

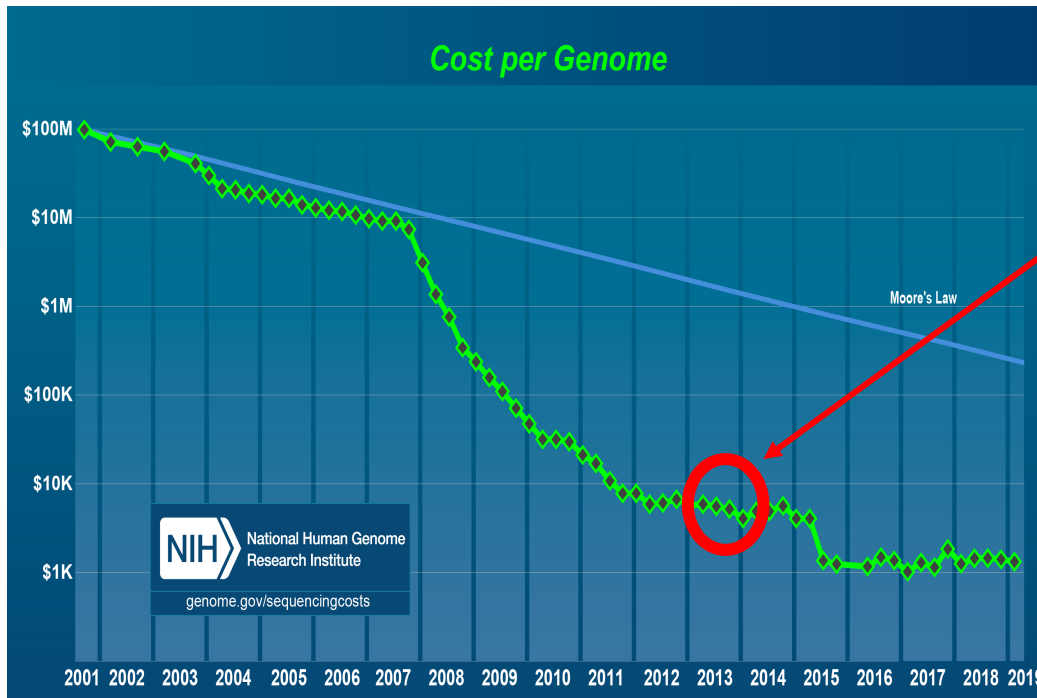
where the **CPU** can **access data** stored in an off-chip main memory only through **power-hungry bus**



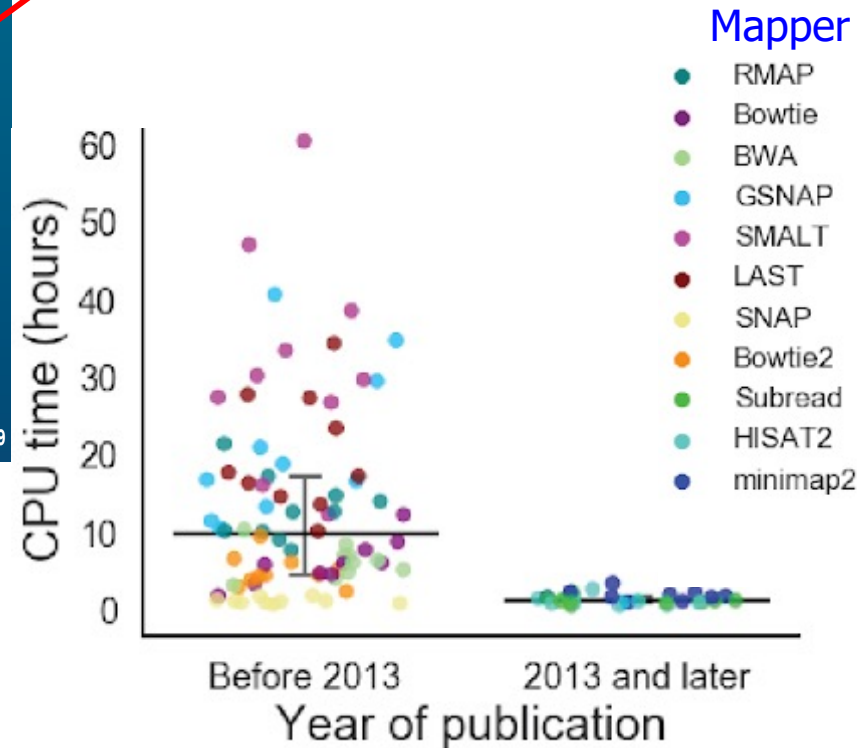
The Problem

Data analysis
is performed
far away from the data

The Need for Speed



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

Enumerating all possible prefixes

- **Data dependencies** limit the computation parallelism

Processing row (or column) after another

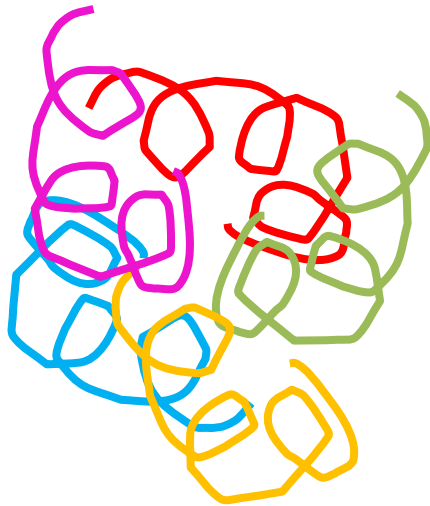
- **Entire matrix** is computed even though strings can be dissimilar.

Number of differences is computed only at the backtraking step.

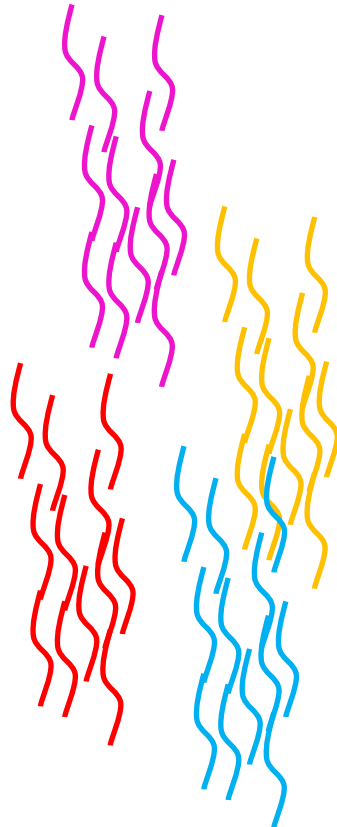
		N	E	T	H	E	R	L	A	N	D	S
	0	1	2	3	4	5	6	7	8	9	10	11
S	1	1	2	3	4	5	6	7	8	9	10	10
W	2	2	2	3	4	5	6	7	8	9	10	11
I	3	3	3	3	4	5	6	7	8	9	10	11
T	4	4	4	3	4	5	6	7	8	9	10	11
Z	5	5	5	4	4	5	6	7	8	9	10	11
E	6	6	5	5	5	4	5	6	7	8	9	10
R	7	7	6	6	6	5	4	5	6	7	8	9
L	8	8	7	7	7	6	5	4	5	6	7	8
A	9	9	8	8	8	7	6	5	4	5	6	7
N	10	9	9	9	9	8	7	6	5	4	5	6
D	11	10	10	10	10	9	8	7	6	5	4	5

Metagenomics Analysis

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

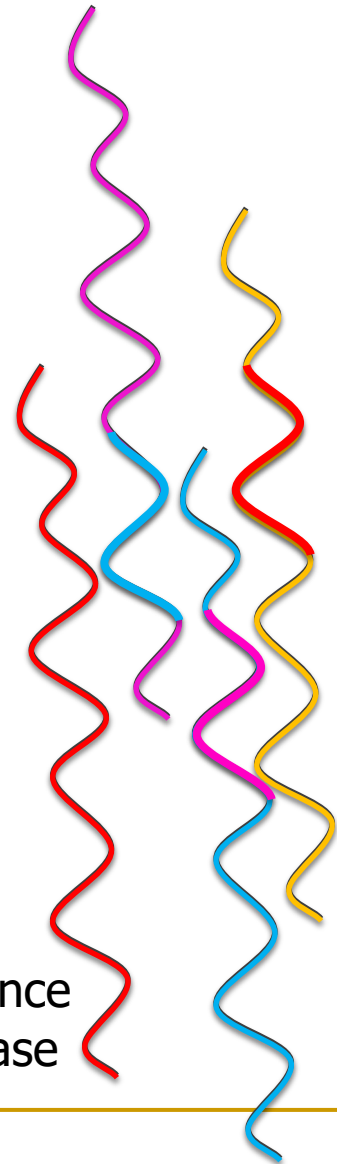


genetic material recovered directly from environmental samples

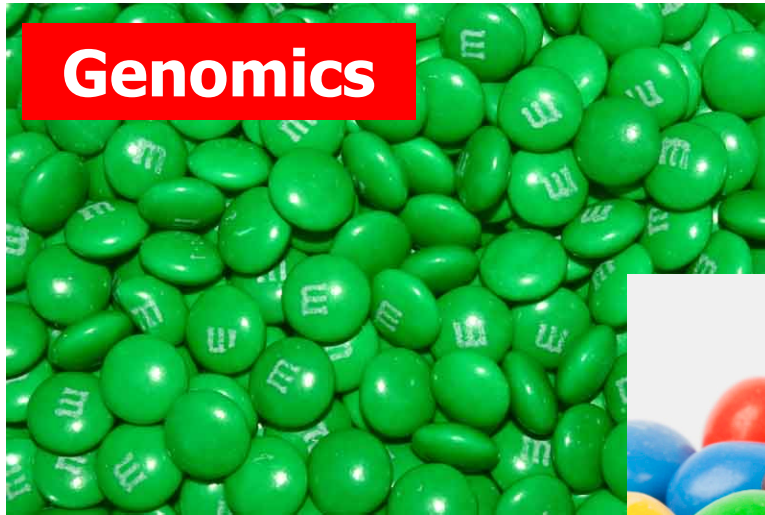


Reads "text format"

Reference Database

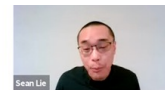


Genomics vs. Metagenomics

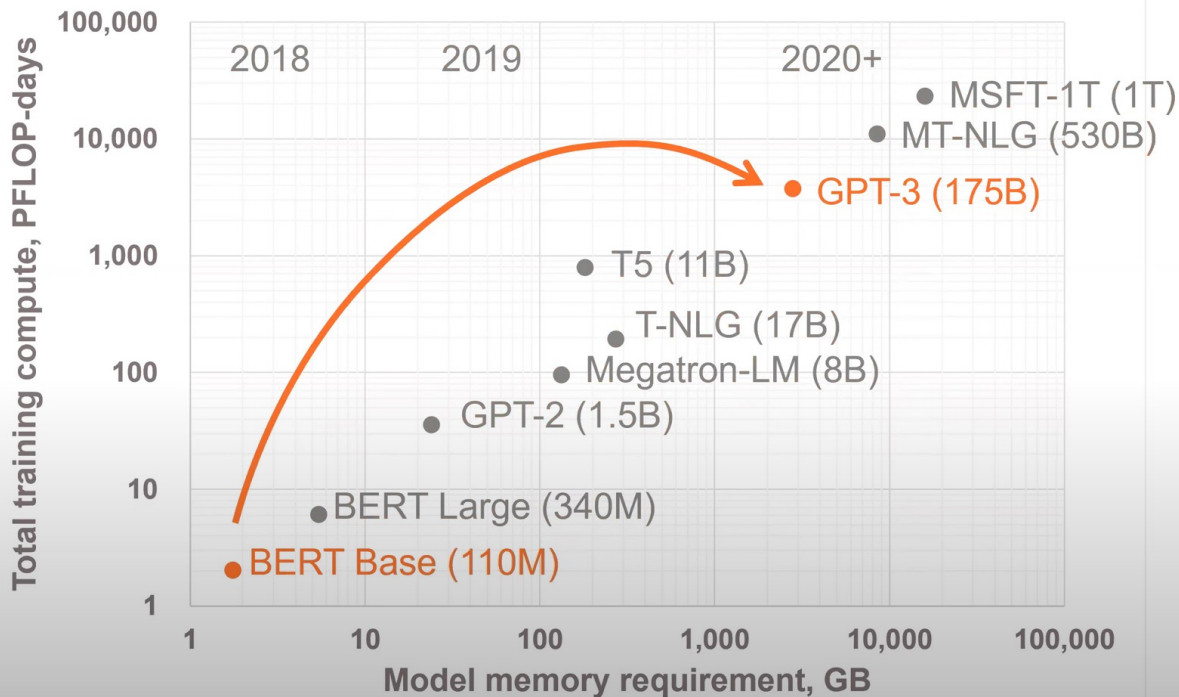


Huge Demand for Performance & Efficiency

Exponential Growth of Neural Networks



Memory and compute requirements



1800x more compute
In just 2 years

Tomorrow, **multi-trillion** parameter models

Deeper and Larger Memory Hierarchies

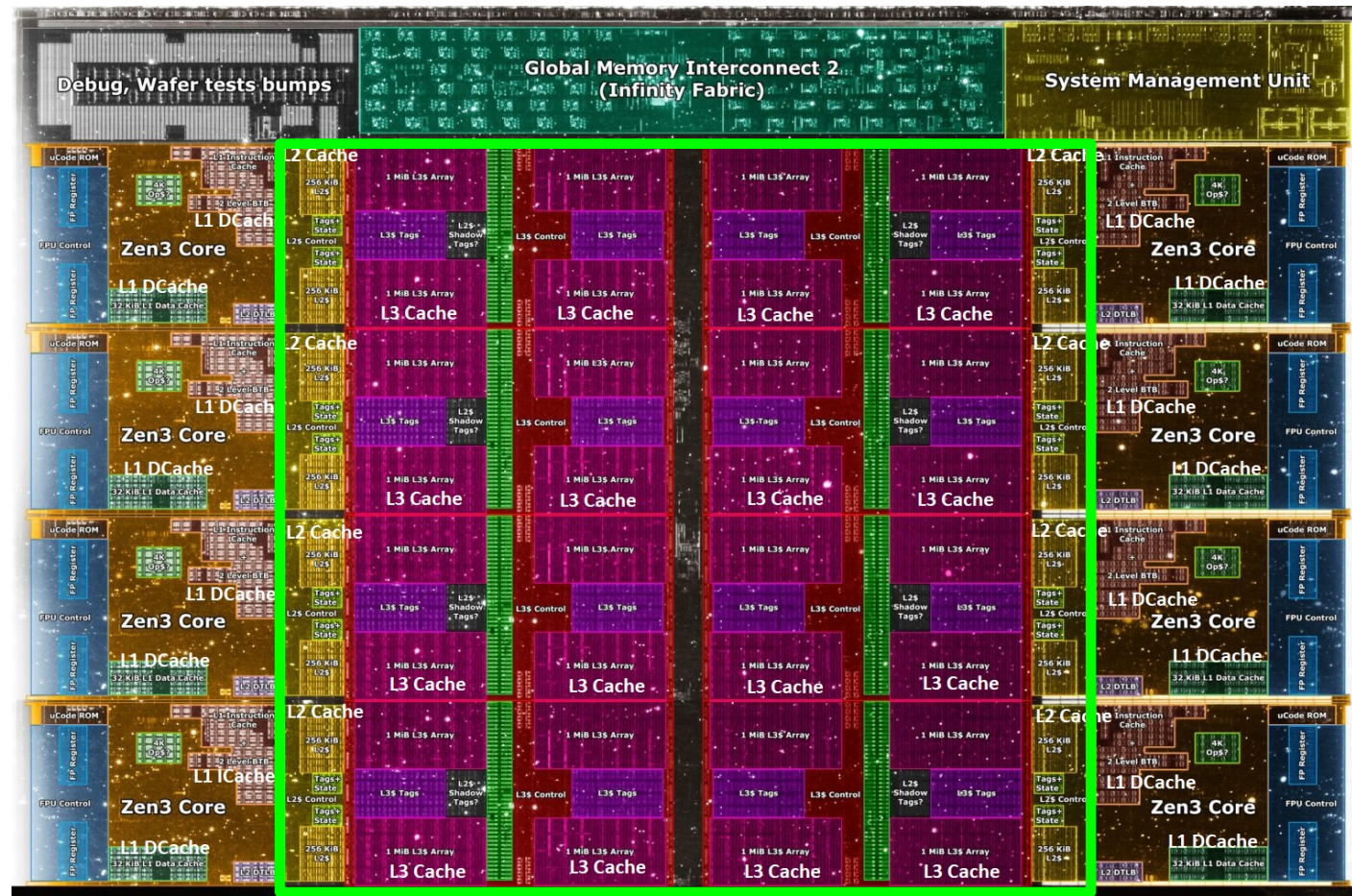
Core Count:
8 cores/16 threads

L1 Caches:
32 KB per core

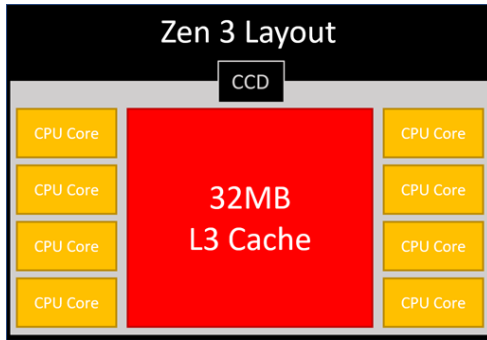
L2 Caches:
512 KB per core

L3 Cache:
32 MB shared

AMD Ryzen 5000, 2020



AMD's 3D Last Level Cache (2021)

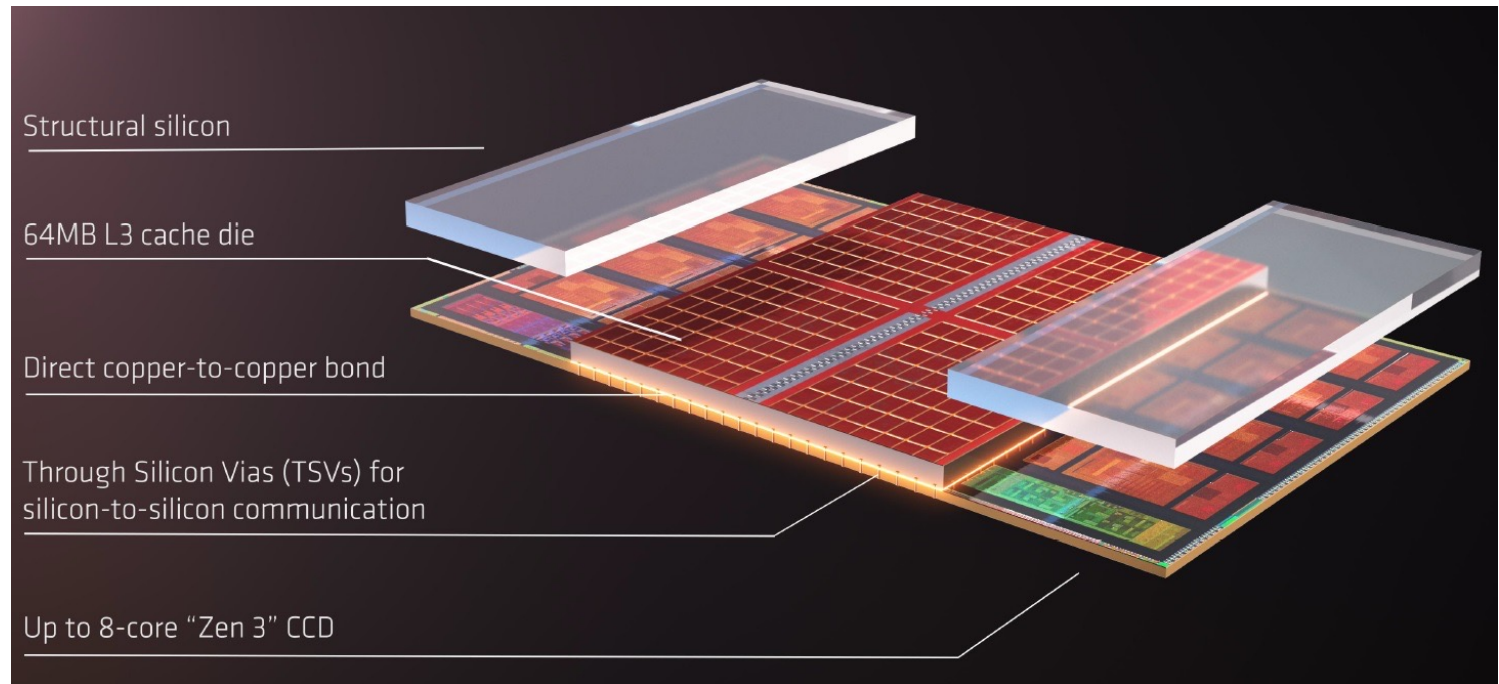


AMD increases the L3 size of their 8-core Zen 3 processors from 32 MB to 96 MB

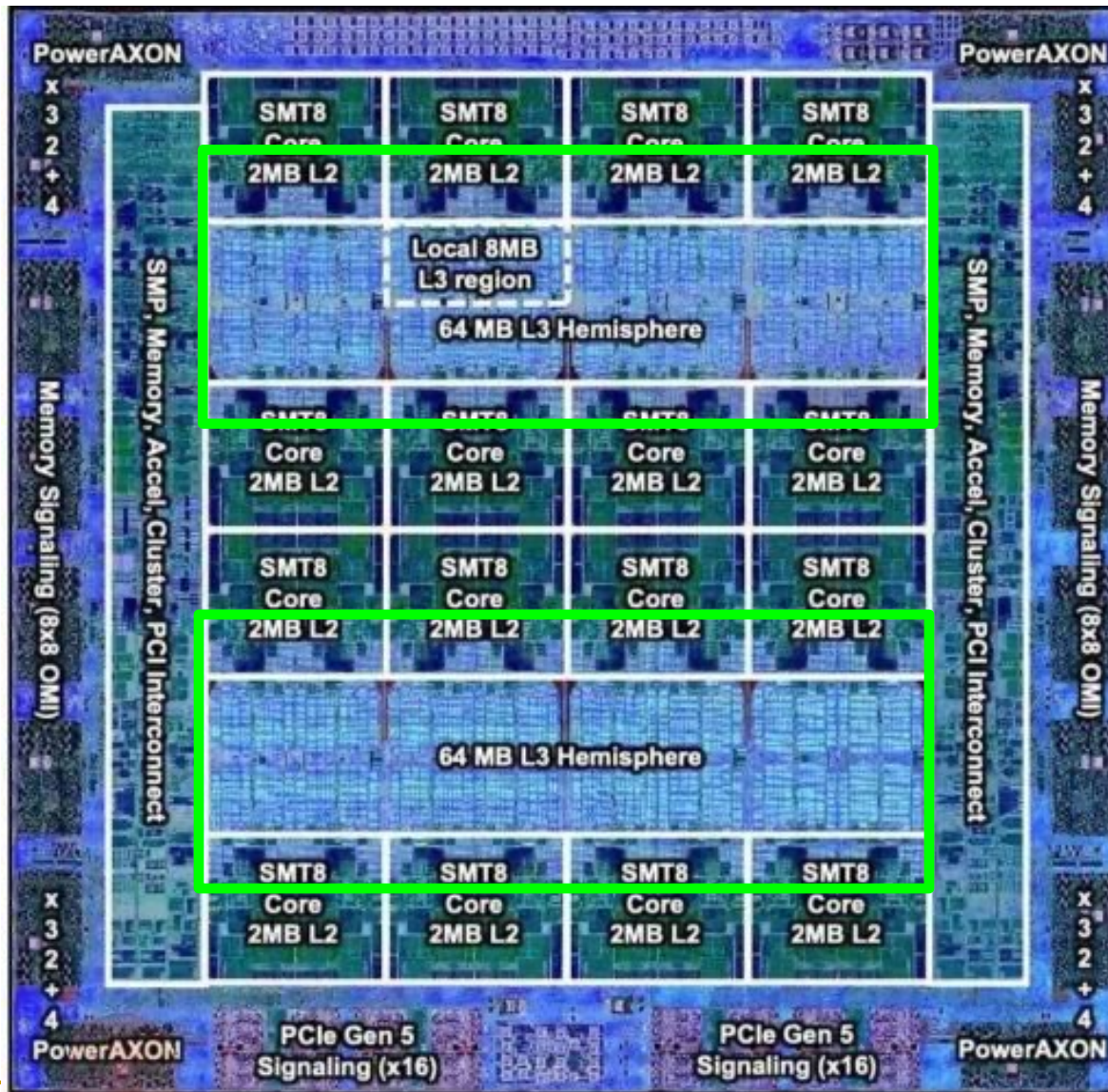
Additional 64 MB L3 cache die
stacked on top of the processor die

- Connected using Through Silicon Vias (TSVs)
- Total of 96 MB L3 cache

<https://community.microcenter.com/discussion/5134/comparing-zen-3-to-zen-2>



Deeper and Larger Memory Hierarchies



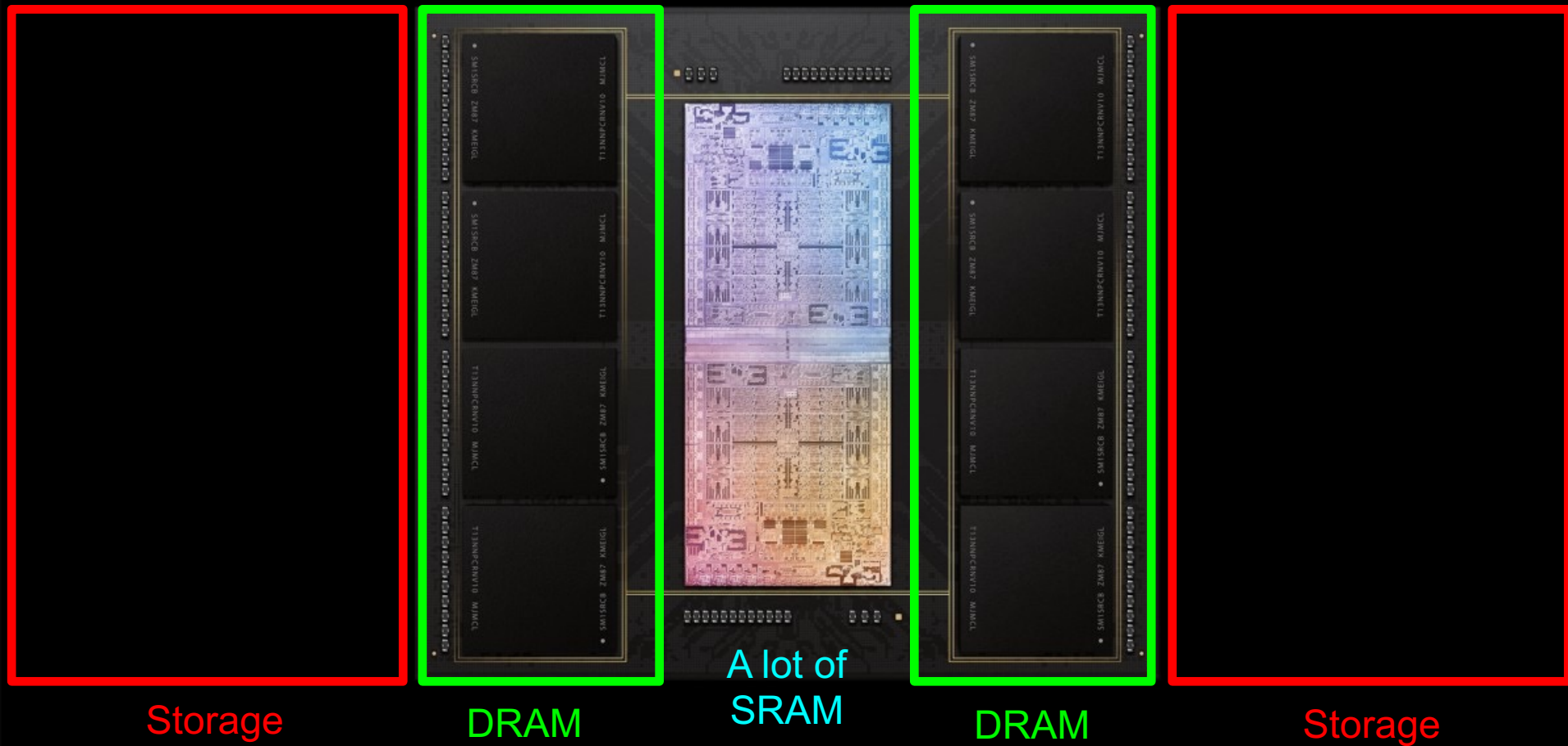
IBM POWER10,
2020

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)

Data Movement Overwhelms Modern Machines

- Amirali Boroumand, Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun, Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela, Allan Knies, Parthasarathy Ranganathan, and Onur Mutlu, "[Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks](#)" *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Williamsburg, VA, USA, March 2018.

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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

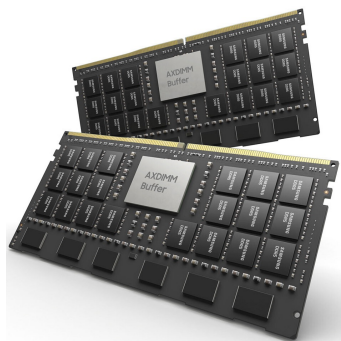
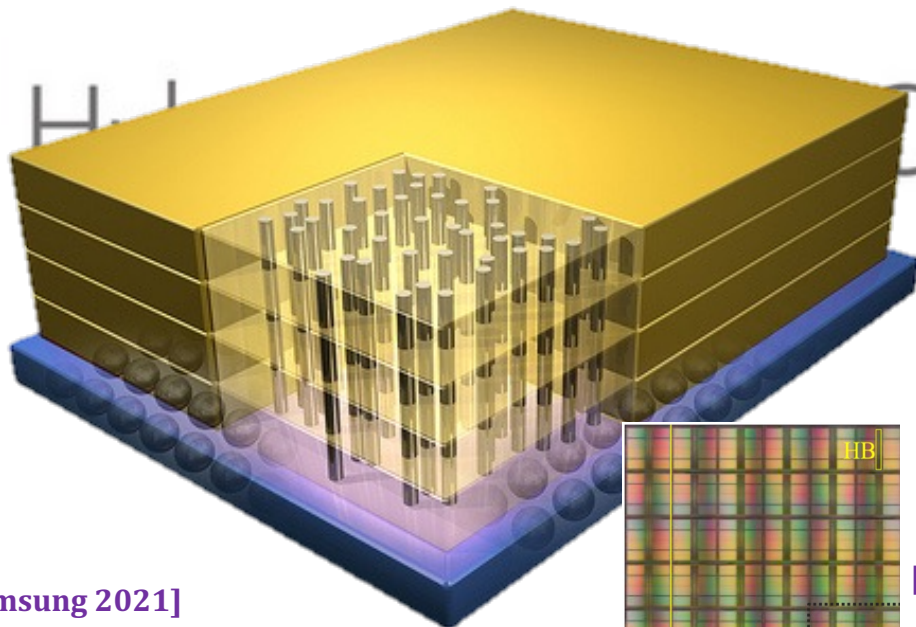
Aki Kuusela³

Allan Knies³

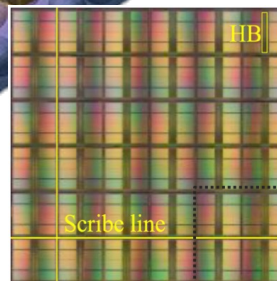
Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

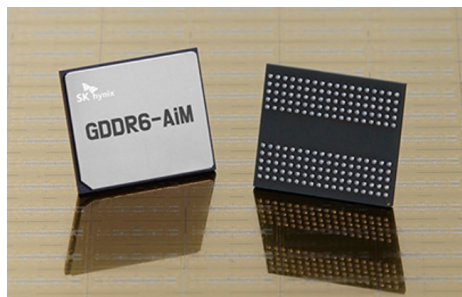
Processing-in-Memory Landscape Today



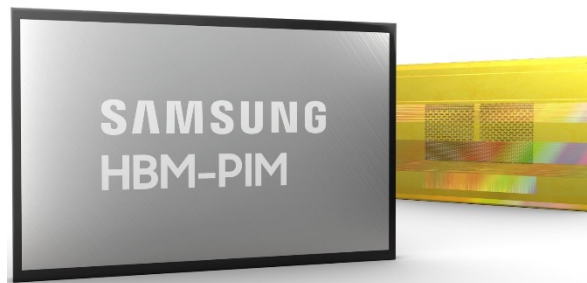
[Samsung 2021]



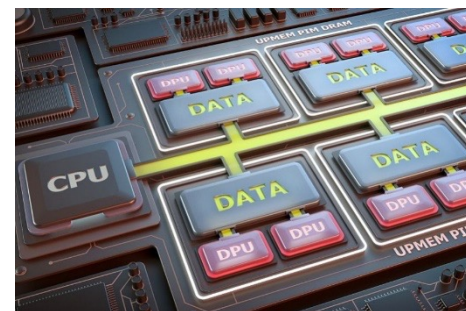
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]

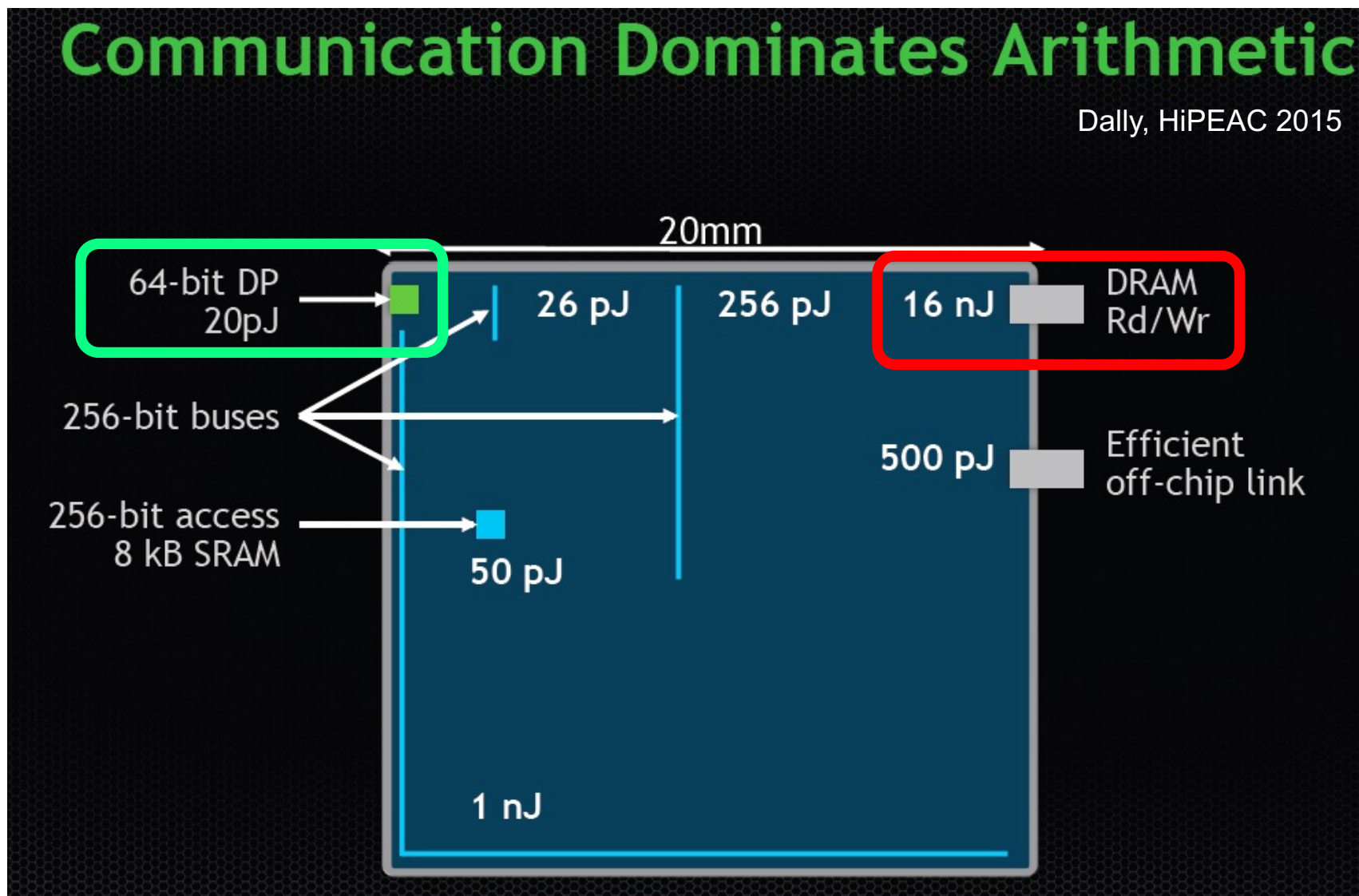


[UPMEM 2019]

The Energy Perspective

Communication Dominates Arithmetic

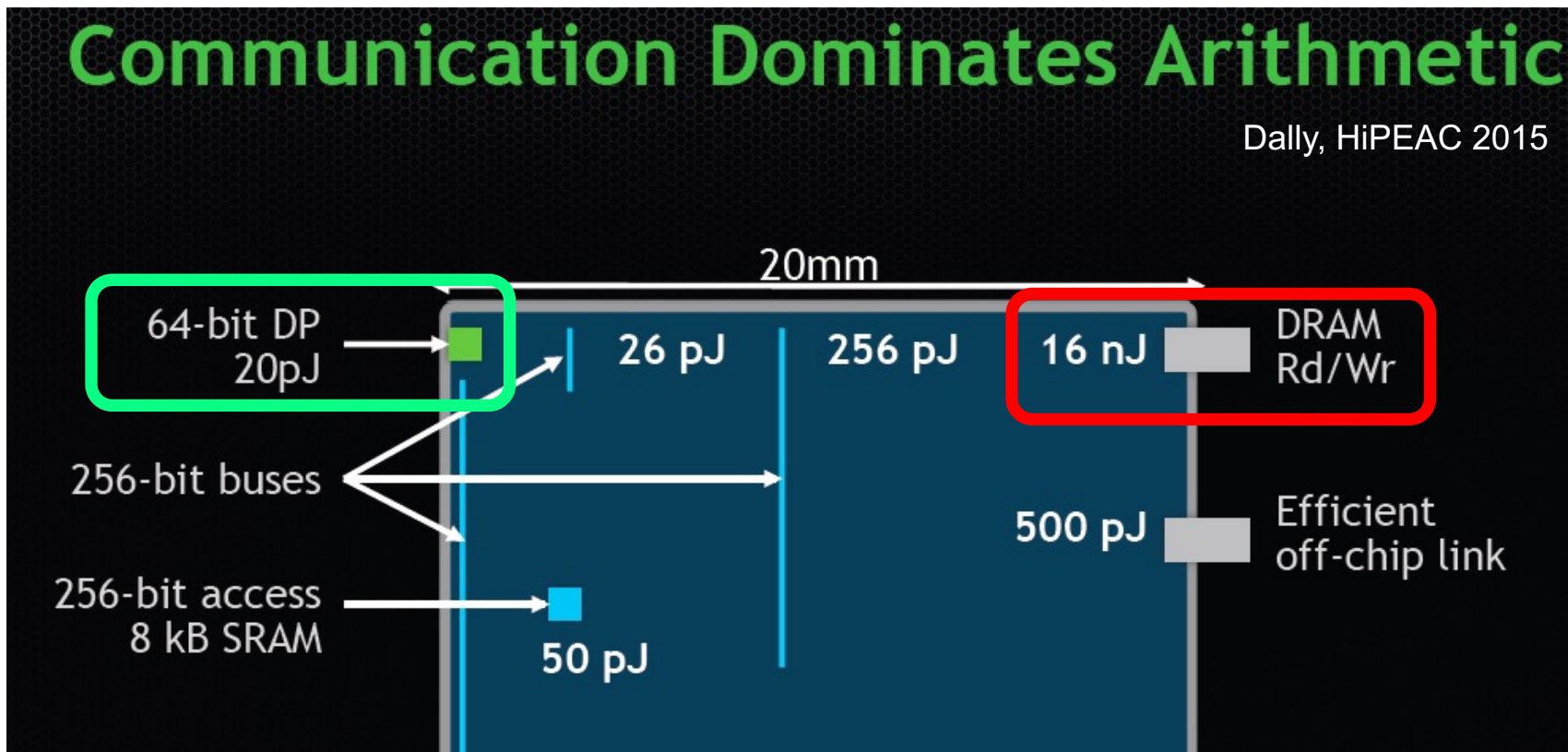
Dally, HiPEAC 2015



Data Movement vs. Computation Energy

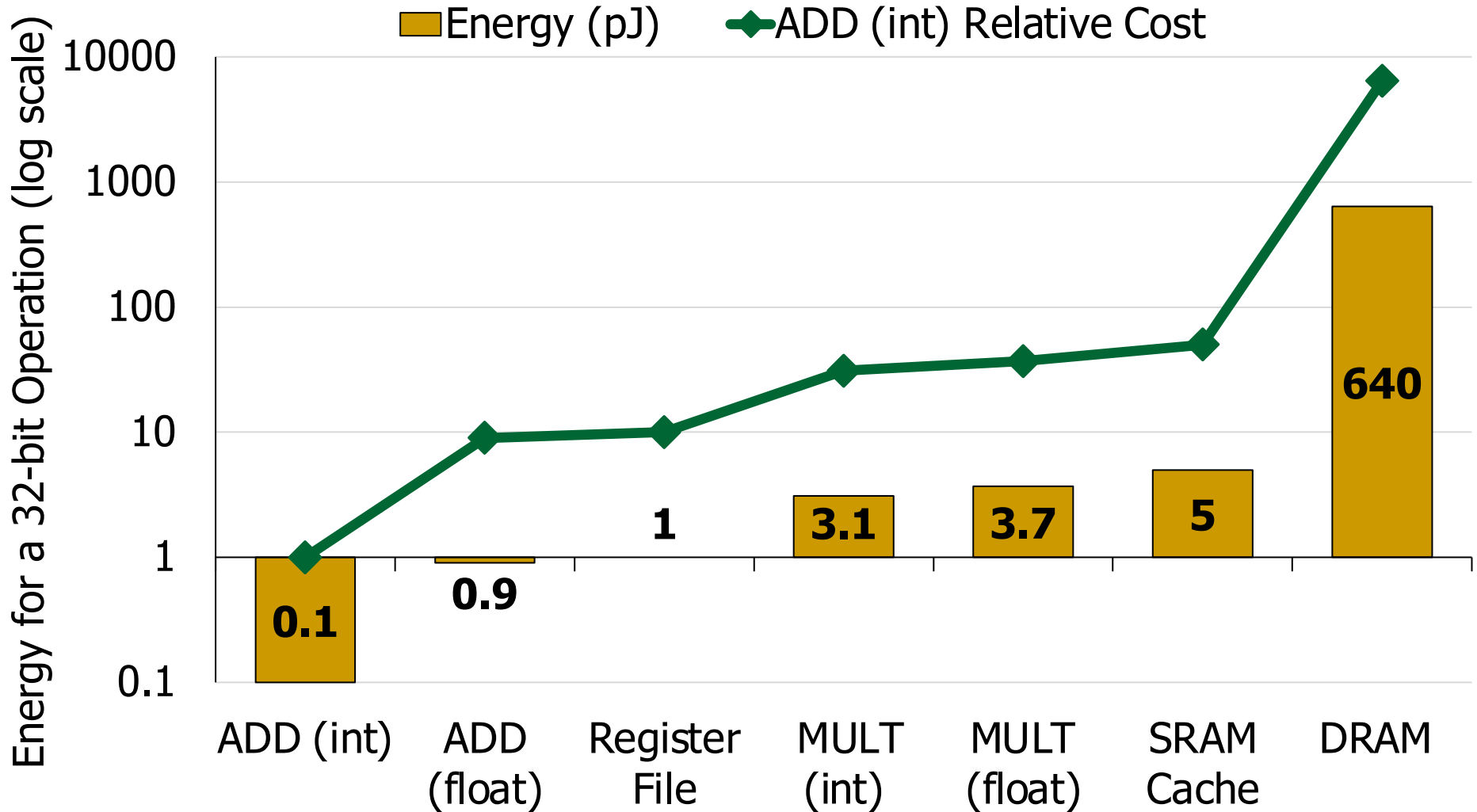
Communication Dominates Arithmetic

Dally, HiPEAC 2015

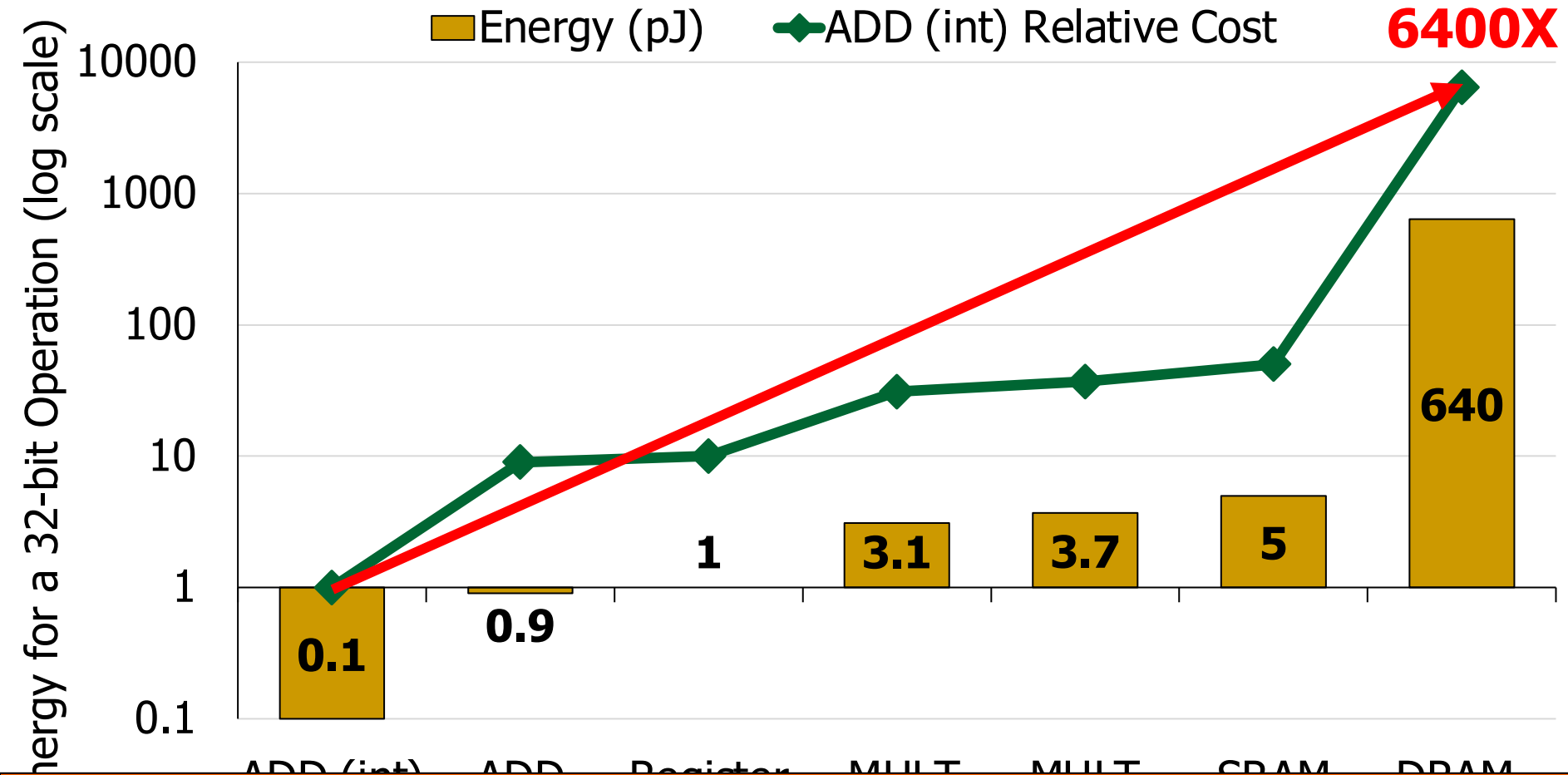


A memory access consumes $\sim 100-1000X$ the energy of a complex addition

Data Movement vs. Computation Energy

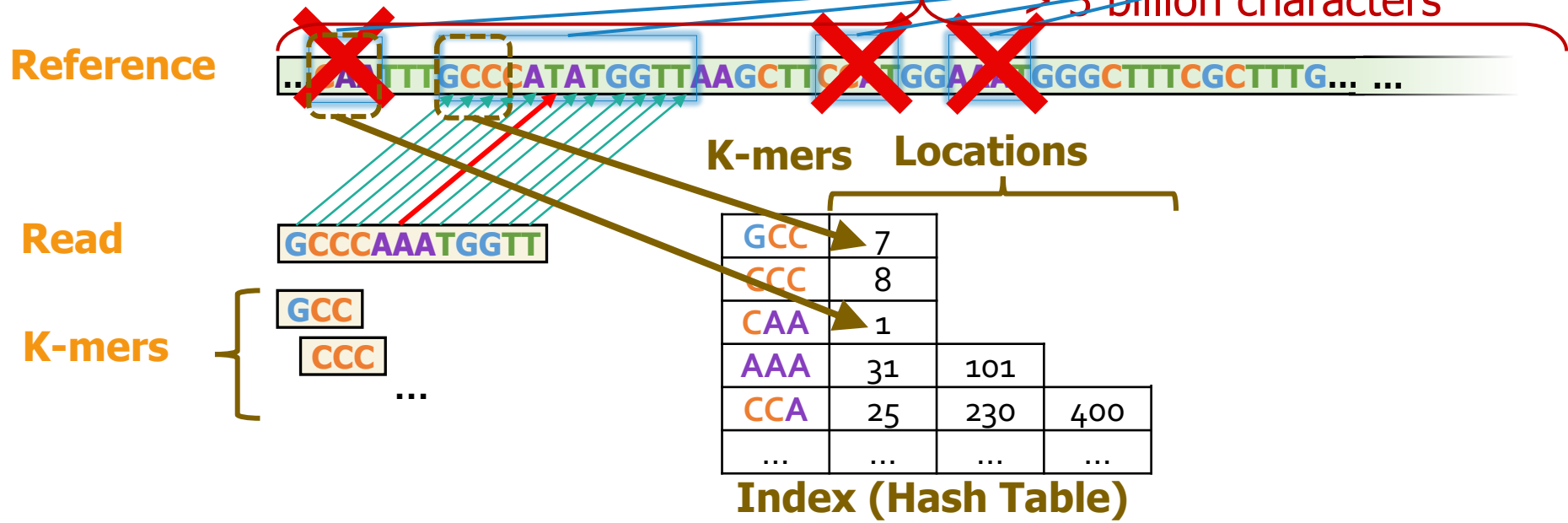


Data Movement vs. Computation Energy



A memory access consumes 6400X
the energy of a simple integer addition

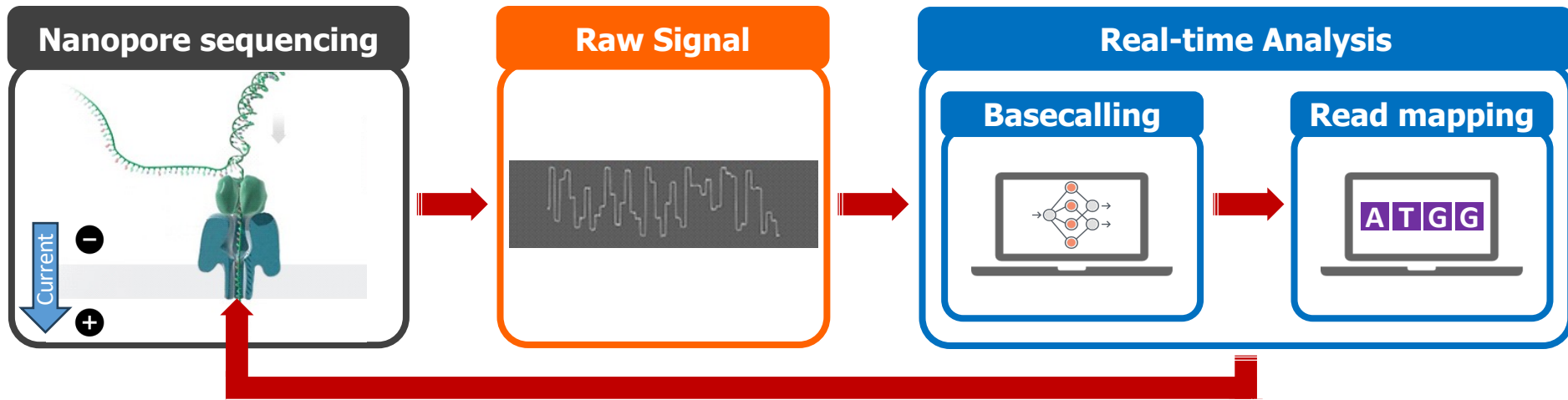
Practical Similarity Identification Seeds



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**

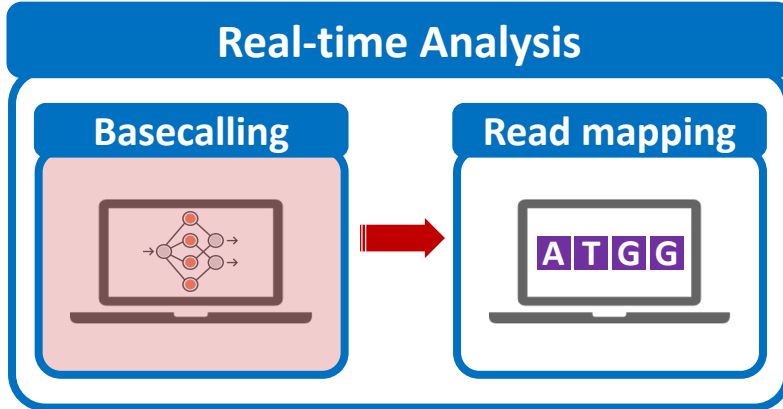


DNNs provide **less noisy analysis** from basecalled sequences

Costly and power-hungry computational requirements

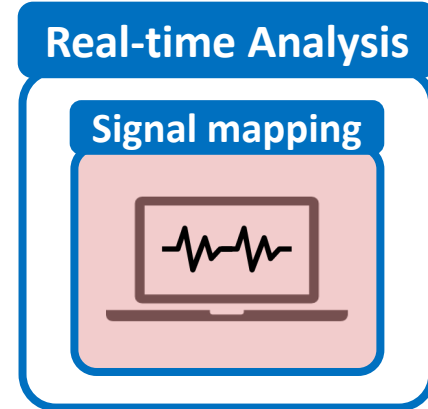
The Problem

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

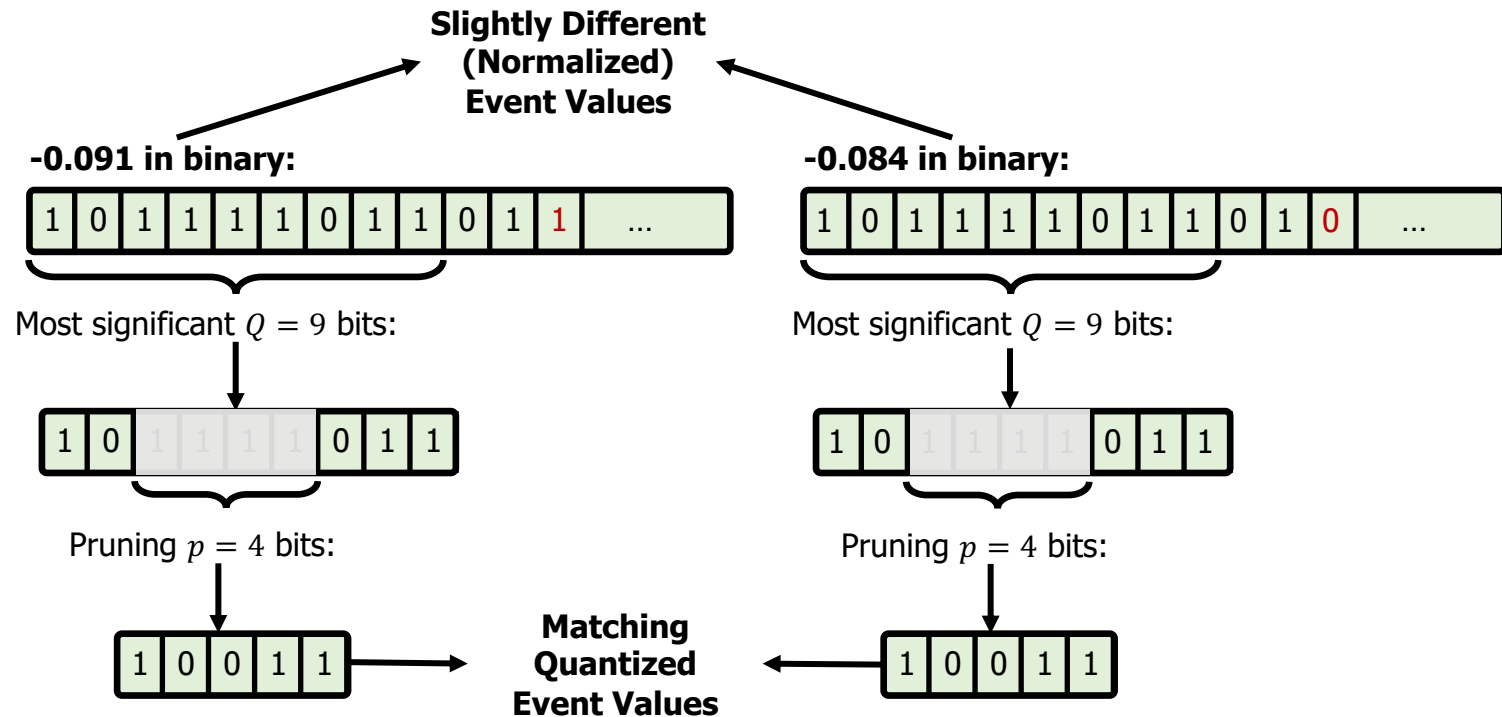
- Stopping when reads reach to a particular depth of coverage
- Stopping when the abundance of all genomes reach a particular threshold

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**



Breakdown Analysis of the RawHash Steps

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

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

Full Read Mapping Accuracy

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

Best results are **highlighted**.

R10.4 Accuracy and Performance Results

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

Full Read Mapping Accuracy – Spider Plot

RawHash2

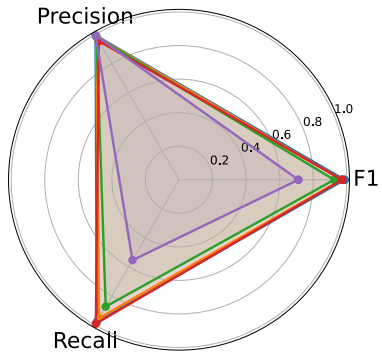
RawHash2-Minimizer

RawHash

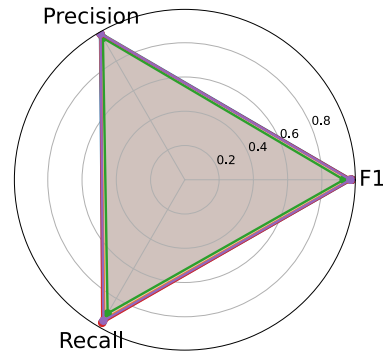
UNCALLED

Sigmap

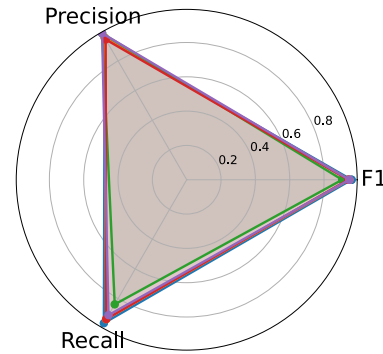
SARS-CoV-2



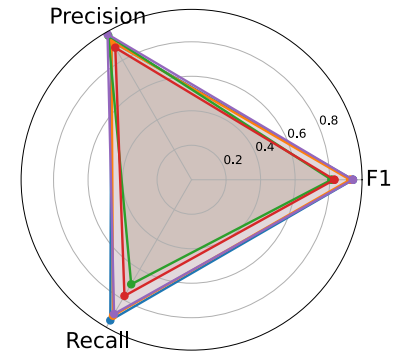
E. coli



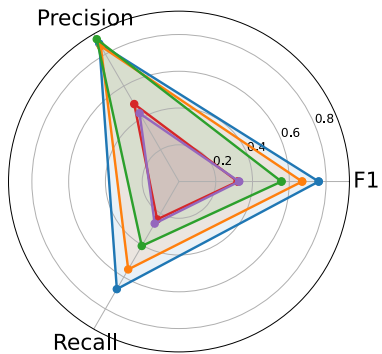
Yeast



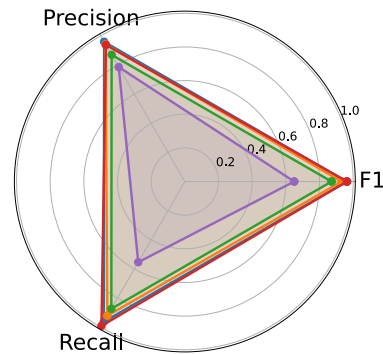
Green Algae



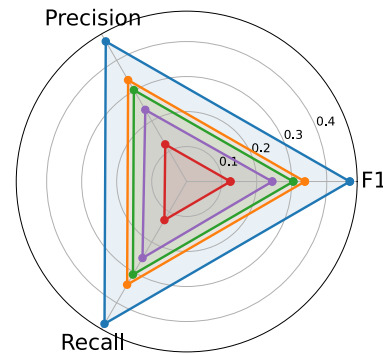
Human



Contamination



Relative Abundance



Different Metrics Combined – Spider Plot

RawHash2

RawHash2-Minimizer

RawHash

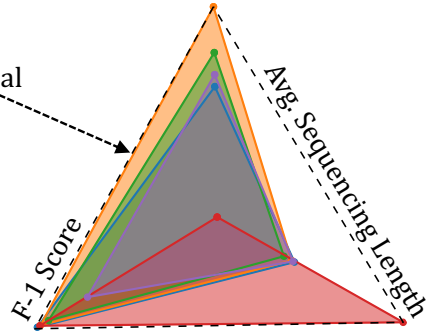
UNCALLED

Sigmap

SARS-CoV-2

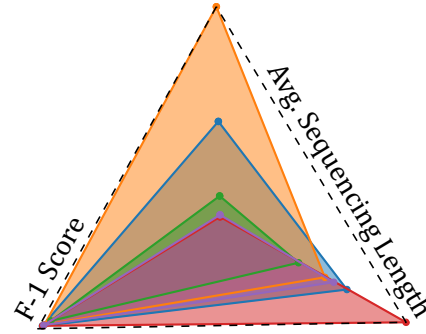
Throughput

Ideal



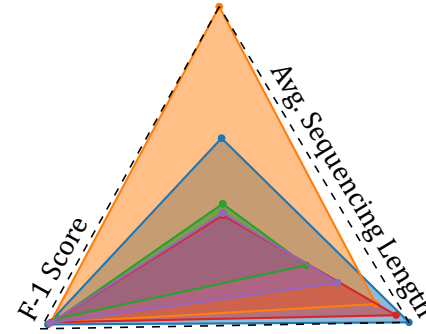
E. coli

Throughput



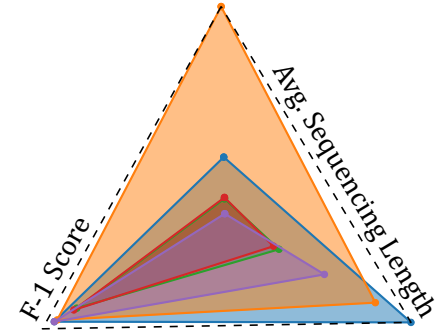
Yeast

Throughput



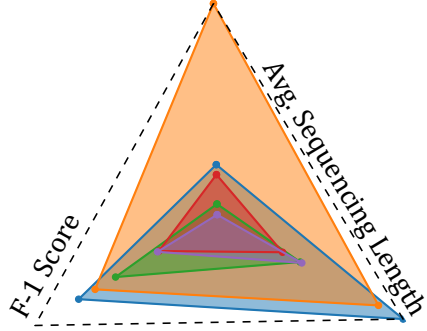
Green Algae

Throughput



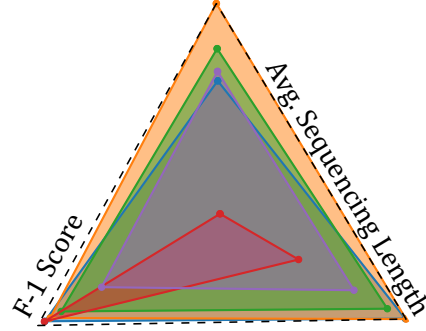
Human

Throughput



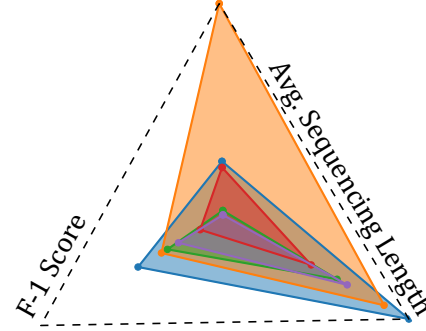
Contamination

Throughput



Relative Abundance

Throughput



Required Computation Resources in Indexing

Dataset	RH2	RH2-Min.	RH	UNCALLED	Sigmap
Indexing CPU Time (sec)					
SARS-CoV-2	0.12	0.06	0.16	8.40	0.02
E. coli	2.48	1.61	2.56	10.57	8.86
Yeast	4.56	3.02	4.44	16.40	25.29
Green Algae	27.60	17.73	24.51	213.13	420.25
Human	1,093.56	588.30	809.08	3,496.76	41,993.26
Contamination	0.13	0.06	0.15	8.38	0.03
Rel. Abundance	747.74	468.14	751.67	3,666.14	36,216.87
Indexing Peak Memory (GB)					
SARS-CoV-2	0.01	0.01	0.01	0.06	0.01
E. coli	0.35	0.19	0.35	0.11	0.40
Yeast	0.75	0.39	0.76	0.30	1.04
Green Algae	5.11	2.60	5.33	11.94	8.63
Human	80.75	40.59	83.09	48.43	227.77
Contamination	0.01	0.01	0.01	0.06	0.01
Rel. Abundance	152.59	75.62	152.84	47.80	238.32

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

Dataset	RH2	RH2-Min.	RH	UNCALLED	Sigmap
Mapping CPU Time (sec)					
SARS-CoV-2	1,705.43	1,227.05	1,539.64	29,282.90	1,413.32
E. coli	1,296.34	787.49	7,453.21	28,767.58	22,923.09
Yeast	545.77	246.37	4,145.38	7,181.44	7,146.32
Green Algae	2,135.83	657.63	22,103.03	12,593.01	26,778.44
Human	100,947.58	21,860.05	1,825,061.23	245,128.15	6,101,179.89
Contamination	3,783.69	2,332.28	3,480.43	234,199.60	3,011.78
Rel. Abundance	250,076.90	62,477.76	4,551,349.79	569,824.13	15,178,633.11
Mapping Peak Memory (GB)					
SARS-CoV-2	4.15	4.16	4.20	0.17	28.26
E. coli	4.13	4.03	4.18	0.50	111.12
Yeast	4.38	4.12	4.37	0.36	14.66
Green Algae	6.11	4.98	11.77	0.78	29.18
Human	48.75	25.04	52.43	10.62	311.94
Contamination	4.16	4.14	4.17	0.62	111.70
Rel. Abundance	49.14	25.82	54.89	8.99	486.63

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

Required CPU Threads For the Entire Flow Cell

Dataset	RH2	RH2-Min.	RH	UNCALLED	Sigmap
Mapping Throughput (bp/sec)					
SARS-CoV-2	552,561.25	885,263.48	694,274.92	9,260.31	602,380.96
E. coli	303,382.45	659,013.57	72,281.32	7,515.76	13,750.97
Yeast	150,547.61	394,766.80	28,757.15	7,471.48	11,624.82
Green Algae	28,742.46	98,323.70	9,488.79	10,069.41	2,569.89
Human	8,968.78	37,086.38	2,099.35	7,225.67	236.45
Contamination	563,129.81	884,929.30	696,873.20	9,343.95	601,936.49
Rel. Abundance	9,501.37	36,919.79	962.79	8,437.70	196.48
CPU Threads Needed for the entire MinION Flowcell (512 pores)					
SARS-CoV-2	1	1	1	25	1
E. coli	1	1	4	31	17
Yeast	2	1	9	31	20
Green Algae	9	3	25	23	90
Human	26	7	110	32	975
Contamination	1	1	1	25	1
Rel. Abundance	25	7	240	28	1173

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