Enabling Fast, Accurate, and Efficient Real-Time Genome Analysis via New Algorithms and Techniques

Can Firtina

Doctoral Examination
11.11.2024

Advisor:
Onur Mutlu (ETH Zurich)

Co-Examiners:
Reetuparna Das (University of Michigan)
Hasindu Gamaarachchi (UNSW Sydney)
Benjamin Langmead (Johns Hopkins University)
Heng Li (Harvard Medical School)

ETH zürich

SAFARI

Key Applications of Genome Analysis

Uncovering and treating diseases linked to genomic variations

Altering genomes to solve fundamental challenges of life

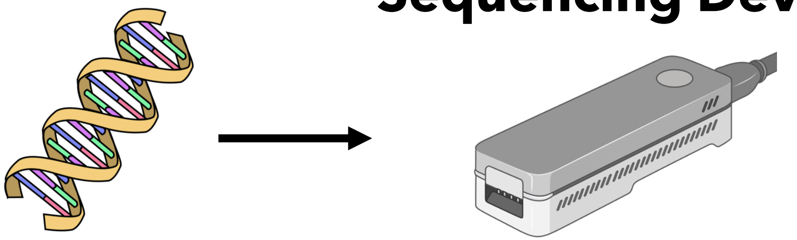
Detecting **pathogens** in the environment

Rapid surveillance of **disease outbreaks**

Genome Sequencing Data Generation

Sequencing process **converts biological molecules** into **digital nucleotide sequences called reads**

Sequencing Device

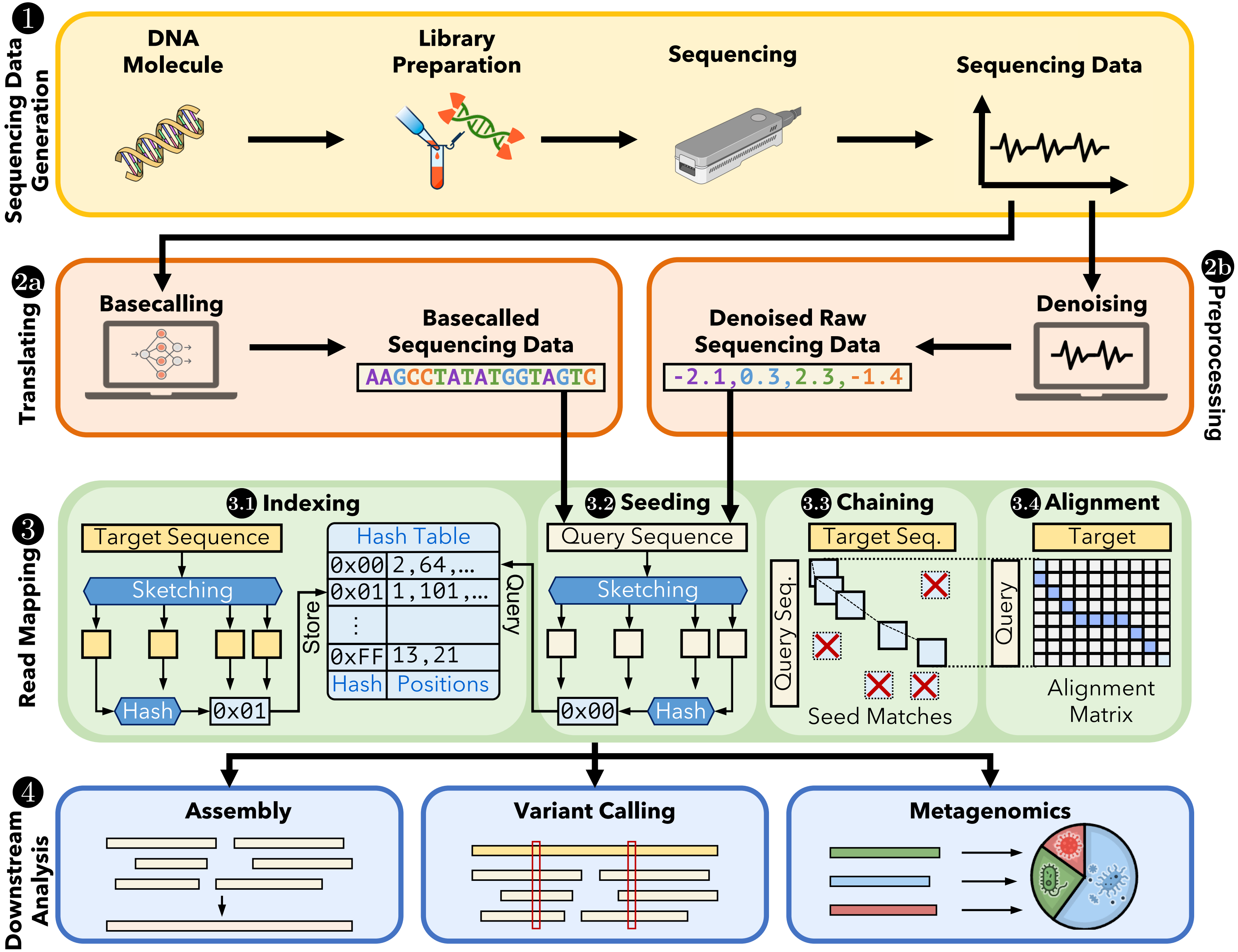


Biological Molecule (e.g., DNA)

Challenge: Unknown origins

Challenge: Large volume of data to analyze

A Common Genome Analysis Pipeline



1 Sequencing Data Generation

DNA Molecule → Library Preparation → Sequencing → Sequencing Data

2a Translating

Basecalling → Basecalled Sequencing Data

AAGCCTATATGGTAGTC

2b Preprocessing

Denoising → Denoised Raw Sequencing Data

-2.1,0.3,2.3,-1.4

3 Read Mapping

3.1 Indexing

Target Sequence

Sketching

Hash

0x01

Store

Hash Table	
0x00	2,64,…
0x01	1,101,…
⋮	
0xFF	13,21
Hash	Positions

Query

3.2 Seeding

Query Sequence

Sketching

0x00

Hash

3.3 Chaining

Target Seq.

Query Seq.

Seed Matches

3.4 Alignment

Target

Query

Alignment Matrix

4 Downstream Analysis

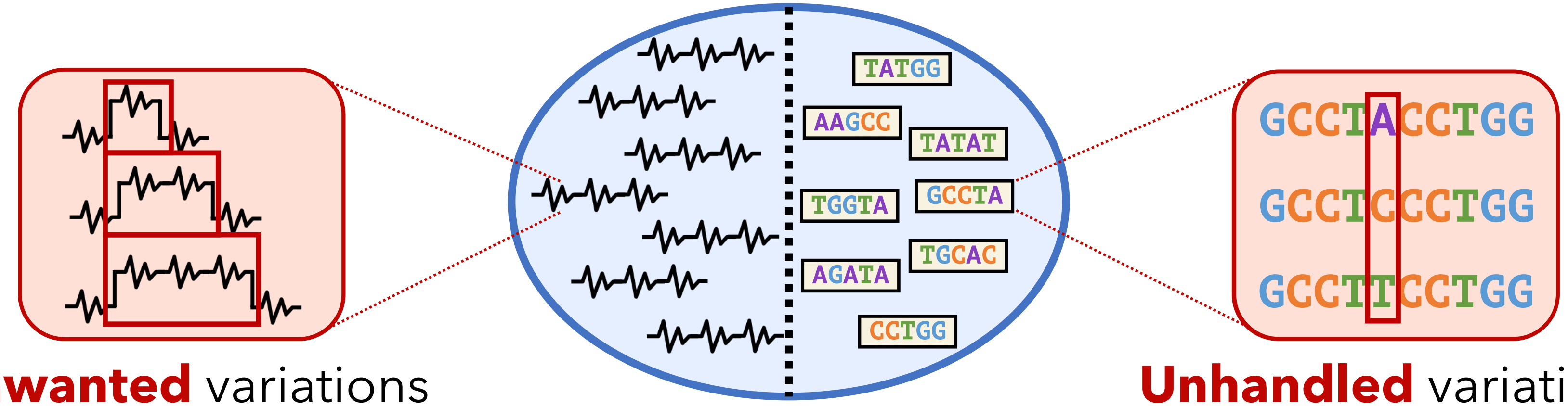
Assembly

Variant Calling

Metagenomics

Problem: Noise in Genome Analysis

Imperfections in sequencing data and its analysis negatively impact genome analysis



Unwanted variations

Unhandled variations

Significant computation overhead

Limited accuracy and application scope

Thesis Statement

We can mitigate **noise** in sequencing data and analysis by

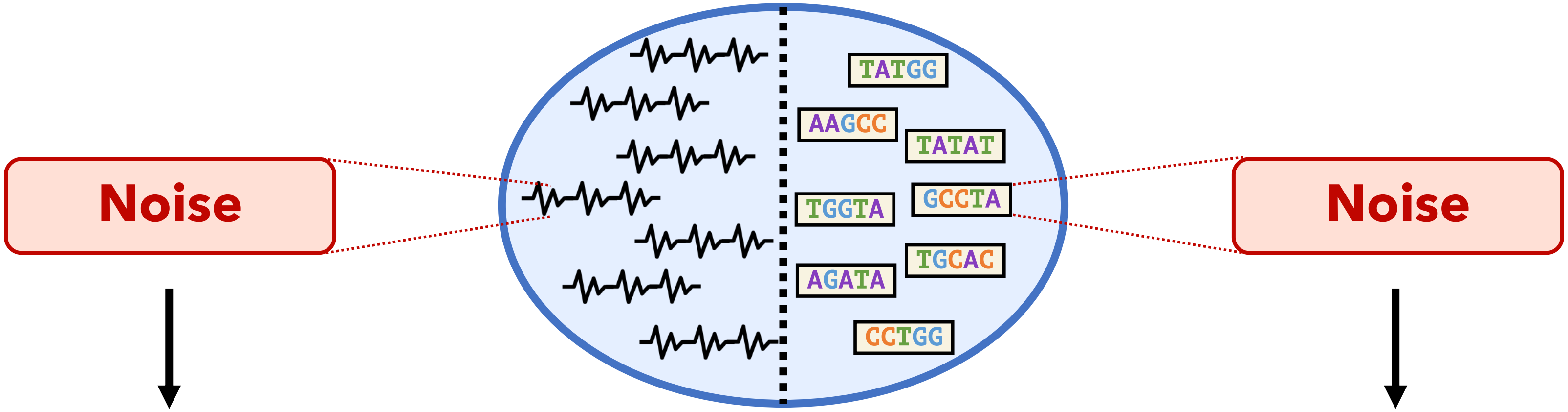
1 Building a **better understanding** of the types of noise, and

2 Developing new algorithms and techniques that can **tolerate and reduce noise**

Thereby providing

Accurate, scalable, and real-time analysis of sequencing data and enabling **new applications** in genome analysis

Core Contributions



1 **BLEND (NARGAB'23 & RECOMB'23)**

Tolerate arbitrary noise with a hash-based search

2 **RawHash (ISMB/ECCB'23)**

Enable hash-based search by reducing noise in raw nanopore signals

3 **RawHash2 (Bioinformatics'24)**

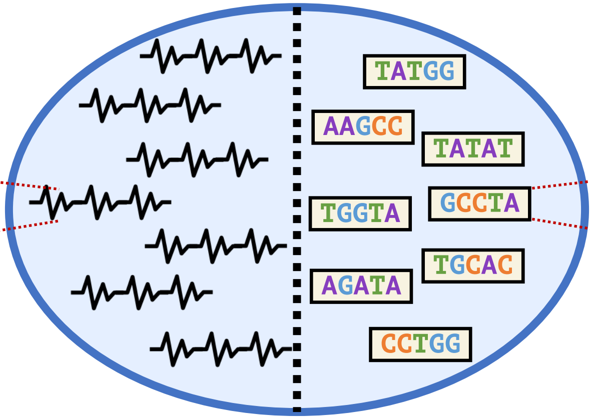
Better noise reduction by leveraging novel insights

4 **Rawsamble (arXiv'24 & HitSeq'24)**

Enabling new applications by overlapping and assembling raw nanopore signals

Core Contributions – BLEND

Noise



Noise

2 RawHash (ISMB/ECCB'23)

Enable hash-based search by reducing noise in raw nanopore signals

1 **BLEND (NARGAB'23 & RECOMB'23)**

Tolerate arbitrary noise with a hash-based search

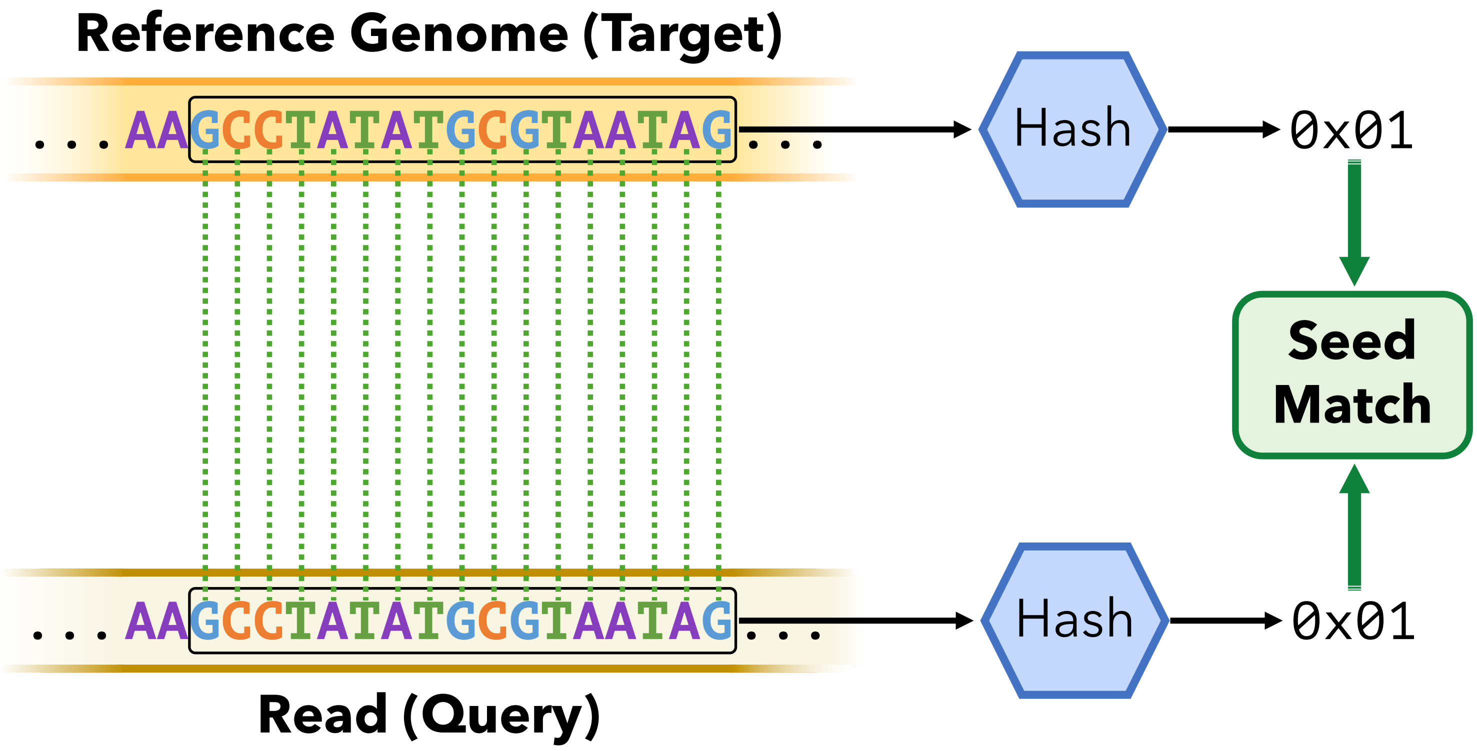
3 RawHash2 (Bioinformatics'24)

Better noise reduction by leveraging novel insights

4 Rawsamble (arXiv'24 & HitSeq'24)

Enabling new applications by overlapping and assembling raw nanopore signals

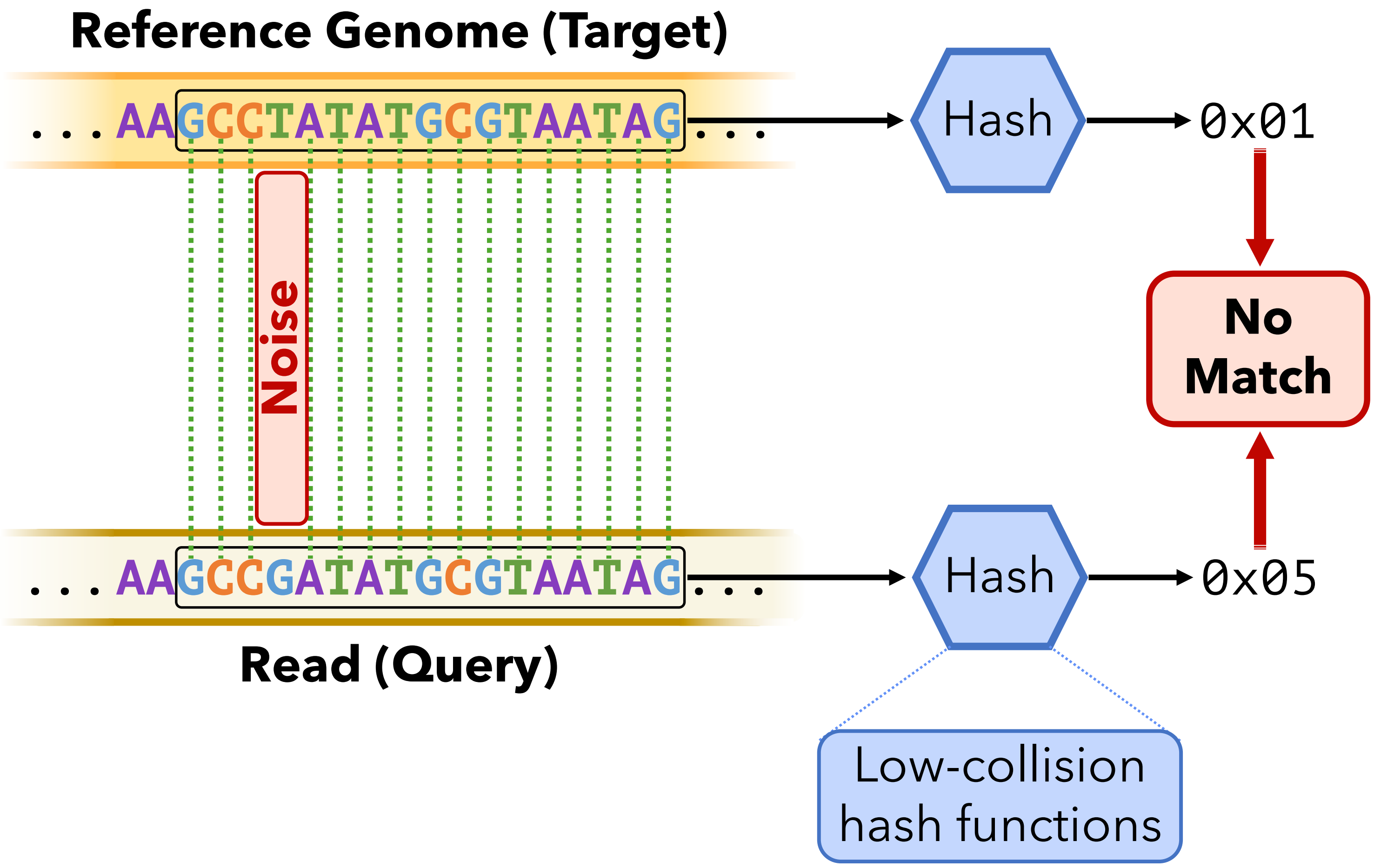
Traditional Hash-Based Seed Matching


Reference Genome (Target)
...AAGCCTATATGCGTAATAG...
Hash
0x01
Seed Match
...AAGCCTATATGCGTAATAG...
Hash
0x01
Read (Query)

✓ Fast and memory–efficient **exact seed matching**

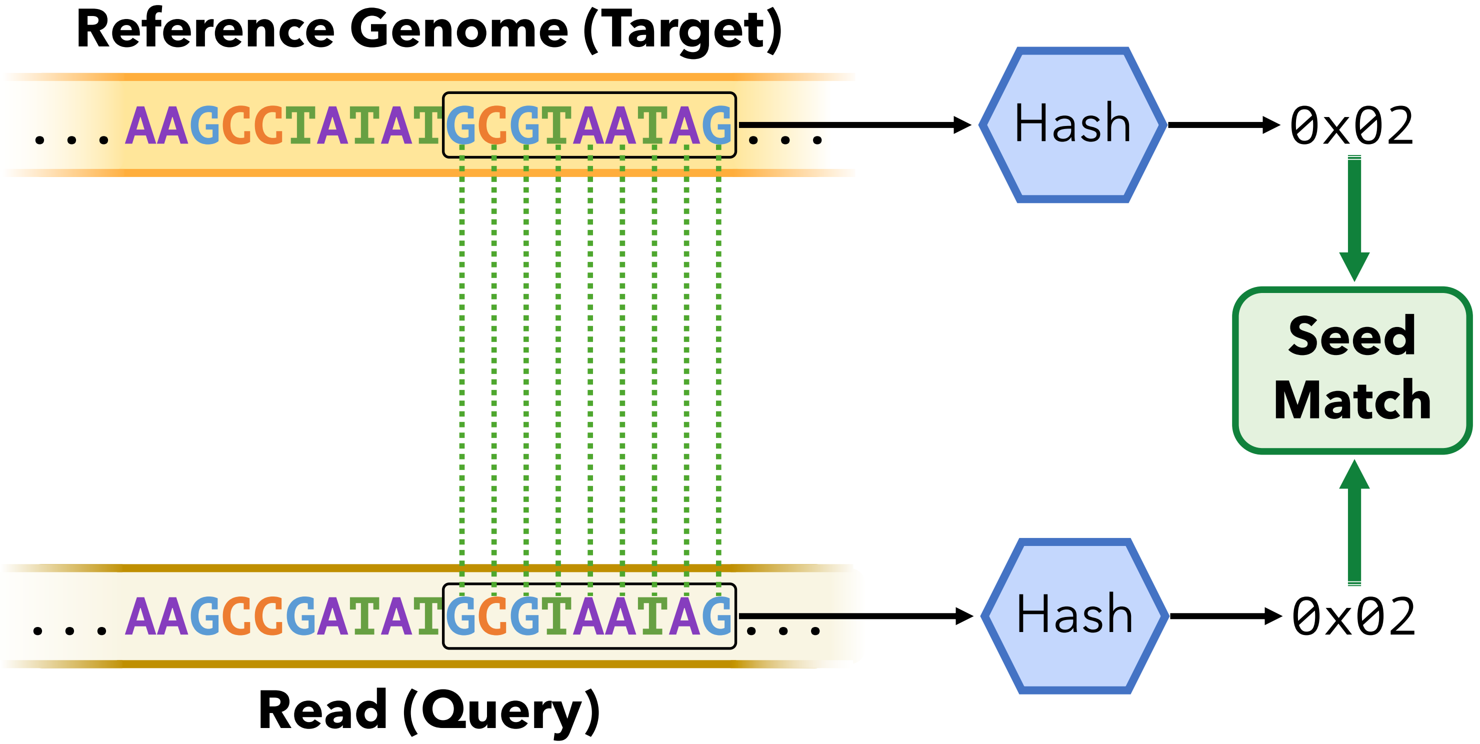
✓ **Dissimilar** seeds are unlikely to match

Limitations of Traditional Hashing


Reference Genome (Target)
...AAGCCTATATGCGTAATAG...
Noise
...AAGCCGATATGCGTAATAG...
Read (Query)
Hash
0x01
Hash
0x05
No Match
Low-collision hash functions

Highly similar seeds are unlikely to match

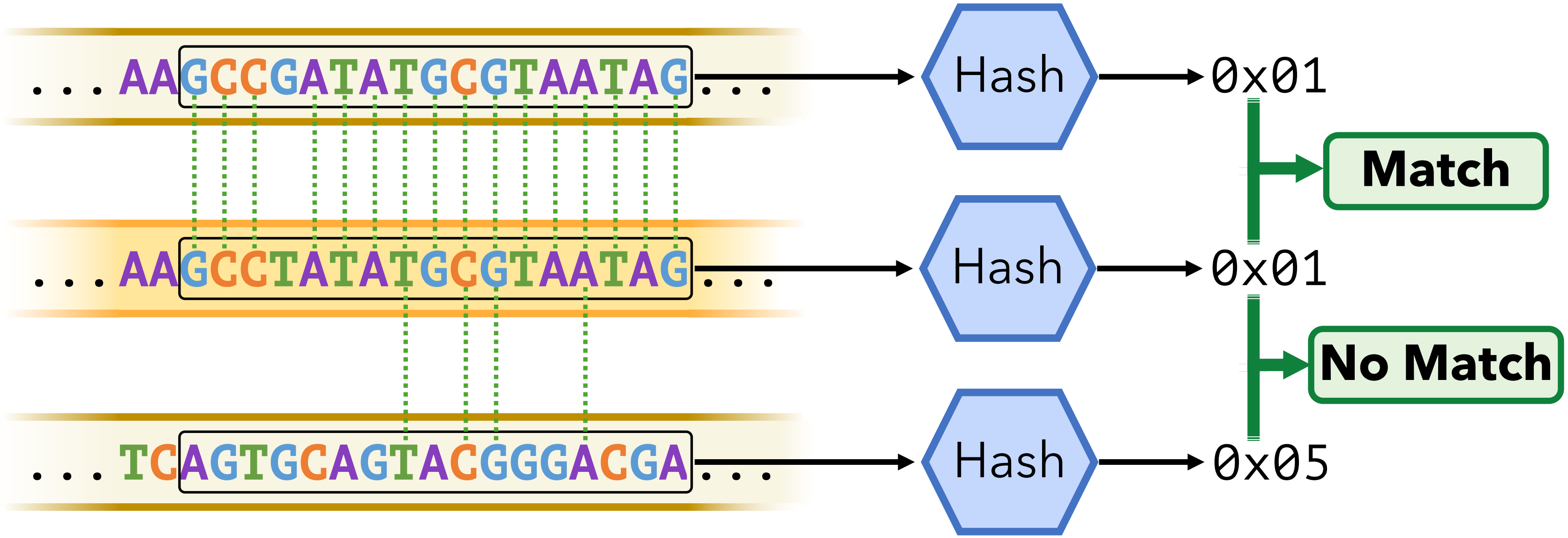
Problems with Low-Collision Hashing


Reference Genome (Target)
...AAGCCTATATGCGTAATAG...
Hash
0x02
Seed Match
...AAGCCGATATGCGTAATAG...
Hash
0x02
Read (Query)

❌ **Larger number** of **shorter exact seed matches** to analyze

❌ Leading to **computational overhead and inaccuracy**

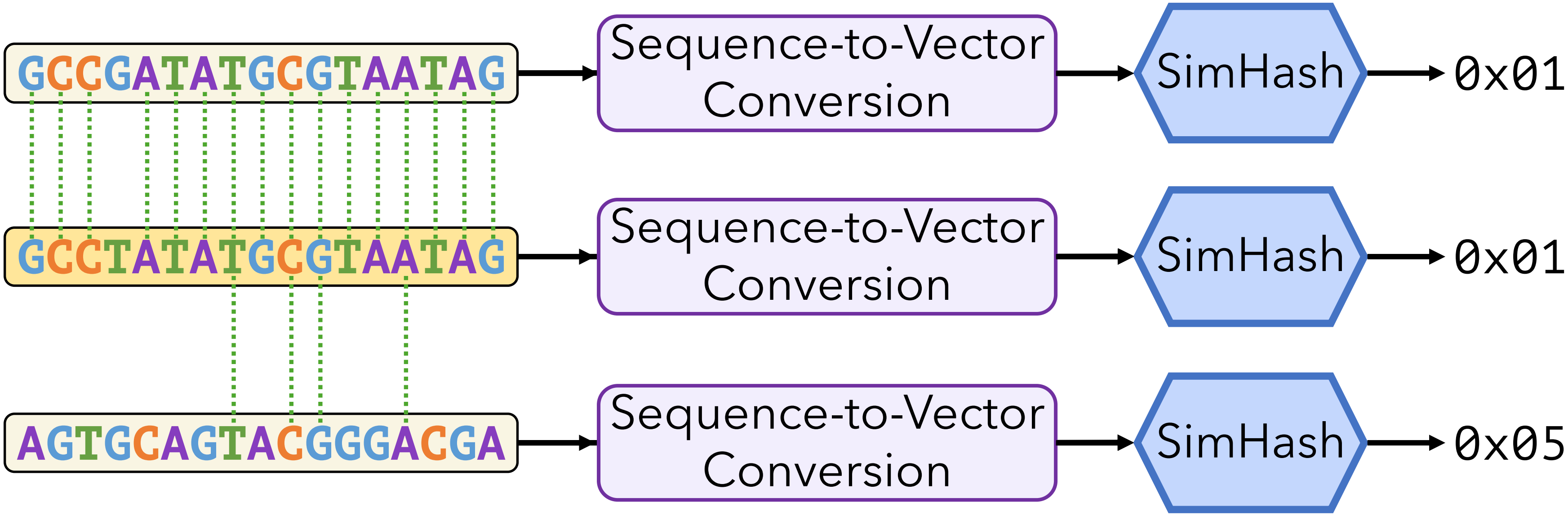
Goal – Fuzzy Seed Matching


...AAGCCGATATGCGTAATAG...
Hash
0x01
Match
...AAGCCTATATGCGTAATAG...
Hash
0x01
No Match
...TCAGTGCAGTACGGGACGA...
Hash
0x05

- ✓ Fast **highly similar** seed matching with **mismatches at any arbitrary positions**
- ✓ **Highly dissimilar** seeds are unlikely to match

Fuzzy seed matching

BLEND Key Idea – Integrate SimHash


GCCGATATGCGTAATAG
GCCTATATGCGTAATAG
AGTGCAGTACGGGACGA
Sequence-to-Vector Conversion
Sequence-to-Vector Conversion
Sequence-to-Vector Conversion
SimHash
SimHash
SimHash
0x01
0x01
0x05

- **Key Idea:** Replace the existing low-collision hash functions with **SimHash to enable fuzzy seed matching**
  - SimHash can generate **the same hash value for similar vectors**
  - **Challenge:** Accurately encoding a seed as a vector (of items)

- BLEND provides **sequence-to-vector conversion strategies** to effectively integrate SimHash in seed matching

Evaluation Methodology

- **Integrated into minimap2** [Li, Bioinformatics'18] to perform end-to-end mapping

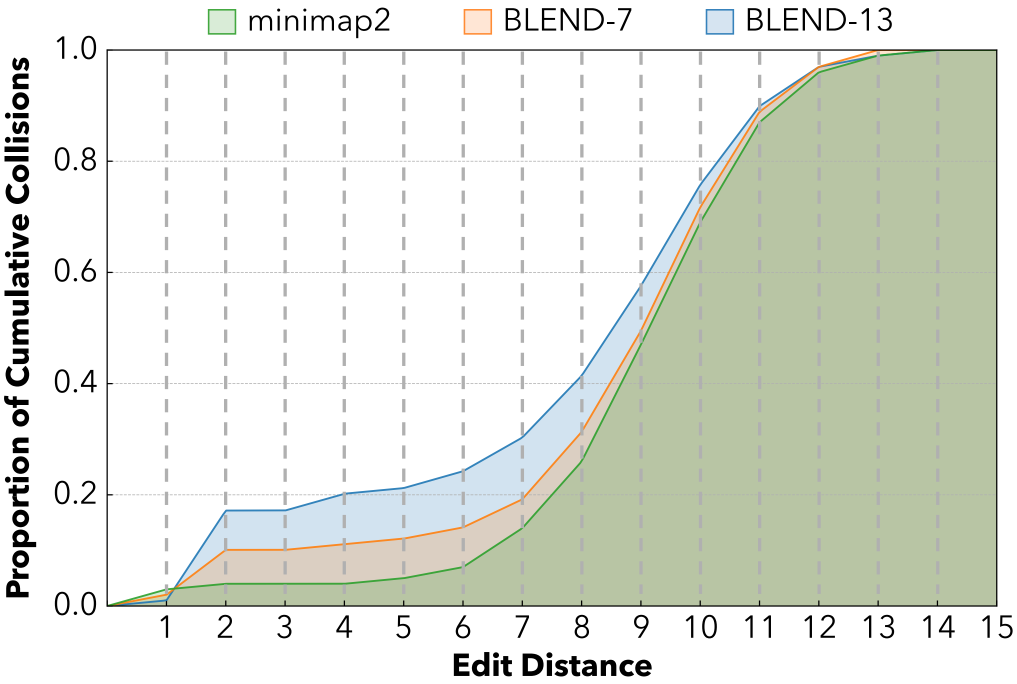
- **Real and simulated datasets** from
  - PacBio (HiFi and CLR), ONT, and Illumina reads
  - Human CHM13 and HG002, Fruit fly, Yeast, and bacterial genomes

- 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**

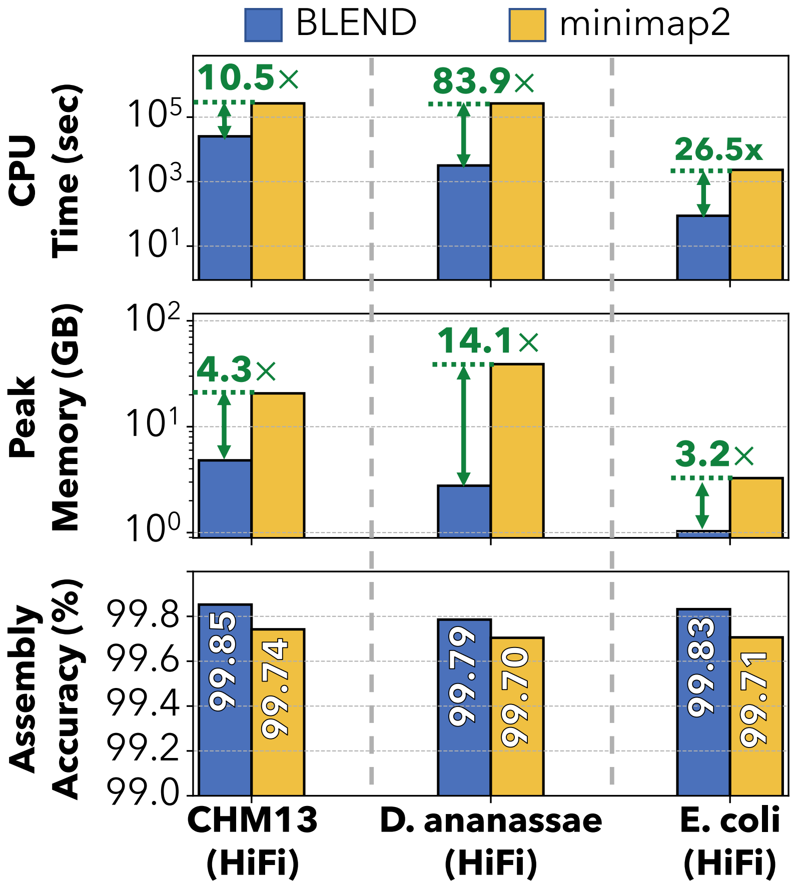
Empirical Analysis on Fuzzy Seed Matching

- We calculate **cumulative proportion of hash collisions** based on the edit distance between seeds (16-mers) with hash collisions
  - **Goal:** Increasing the proportion of hash collisions **at lower edit distances**



BLEND enables fuzzy seed matching by systematically increasing **the collision rate for highly similar seeds**

Key Results – Overlapping and Assembly



Speedup of up to **83.9×**

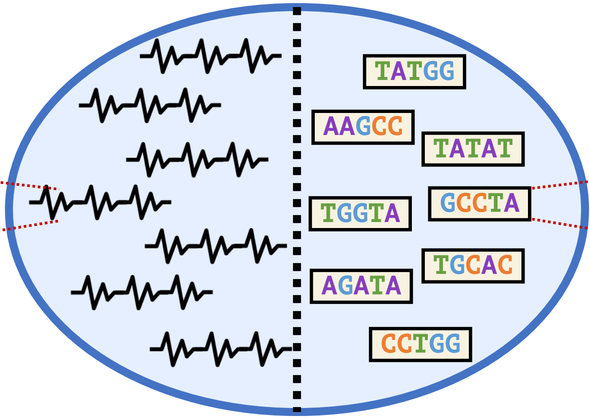
Reduced **peak memory** by up to **14.1×**

Most accurate *de novo* assemblies

Effectively tolerating noise

improves both **performance** **and** **accuracy**

Core Contributions – BLEND



Noise

Noise

1 **BLEND (NARGAB'23 & RECOMB'23)**

Tolerate arbitrary noise with a hash-based search

2 **RawHash (ISMB/ECCB'23)**

Enable hash-based search by reducing noise in raw nanopore signals

3 **RawHash2 (Bioinformatics'24)**

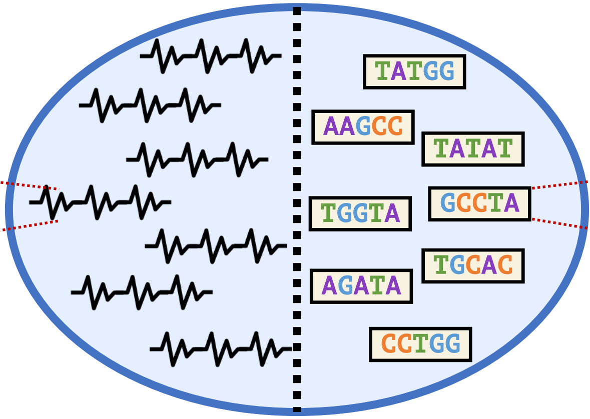
Better noise reduction by leveraging novel insights

4 **Rawsamble (arXiv'24 & HitSeq'24)**

Enabling new applications by overlapping and assembling raw nanopore signals

Core Contributions – RawHash

Noise



Noise

1 **BLEND (NARGAB'23 & RECOMB'23)**

Tolerate arbitrary noise with a hash-based search

2 **RawHash (ISMB/ECCB'23)**

Enable hash-based search by reducing noise in raw nanopore signals

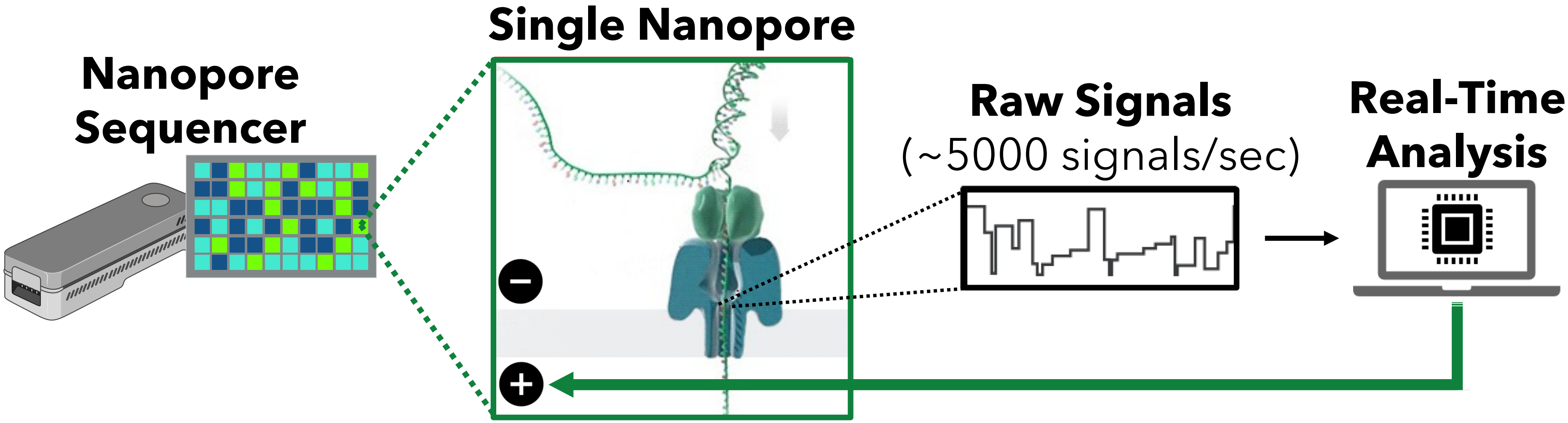
3 **RawHash2 (Bioinformatics'24)**

Better noise reduction by leveraging novel insights

4 **Rawsamble (arXiv'24 & HitSeq'24)**

Enabling new applications by overlapping and assembling raw nanopore signals

Nanopore Sequencing & Real-Time Analysis



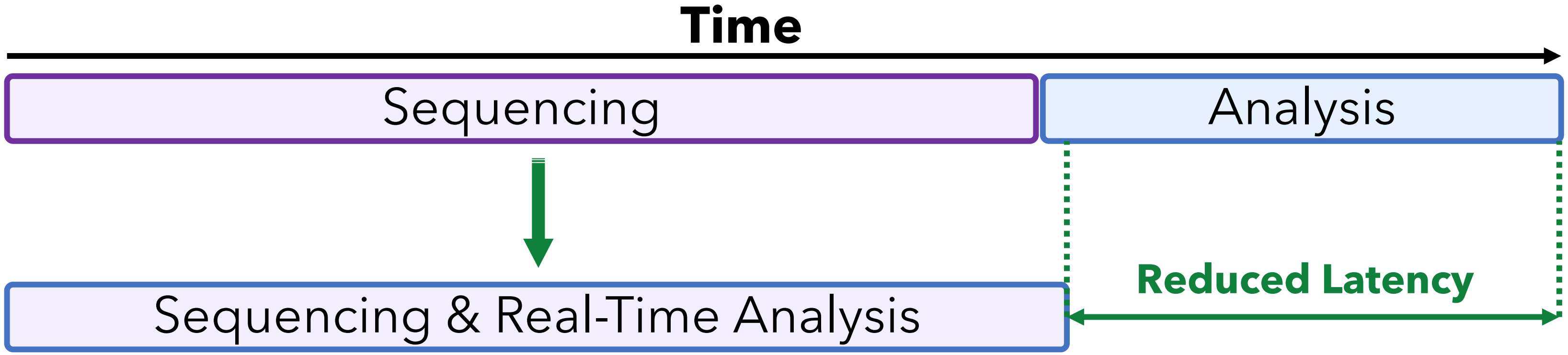
Raw Signals: Ionic current measurements generated at a certain **throughput**

Real-Time Analysis: Analyzing raw signals **instantly as they are generated**

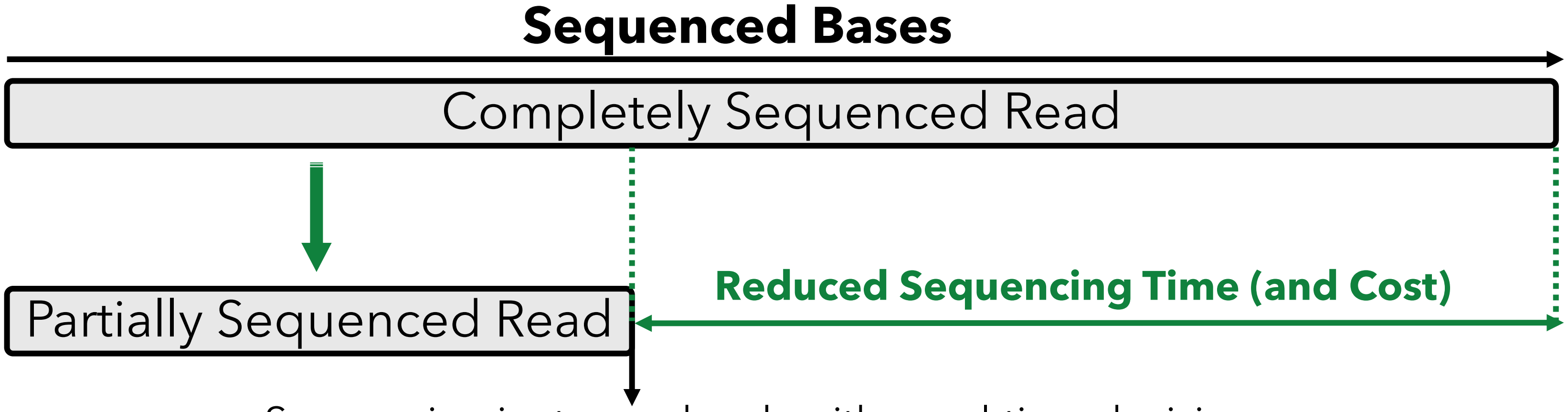
Real-Time Decisions: Stopping sequencing **early** based on real-time analysis

Benefits of Real-Time Analysis

✓ **Reducing latency** by overlapping analysis with sequencing

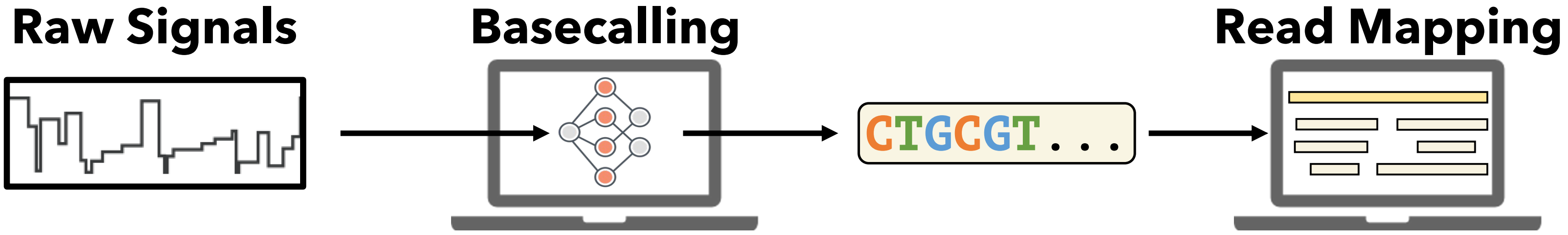


✓ **Reducing sequencing time and cost** by stopping sequencing early



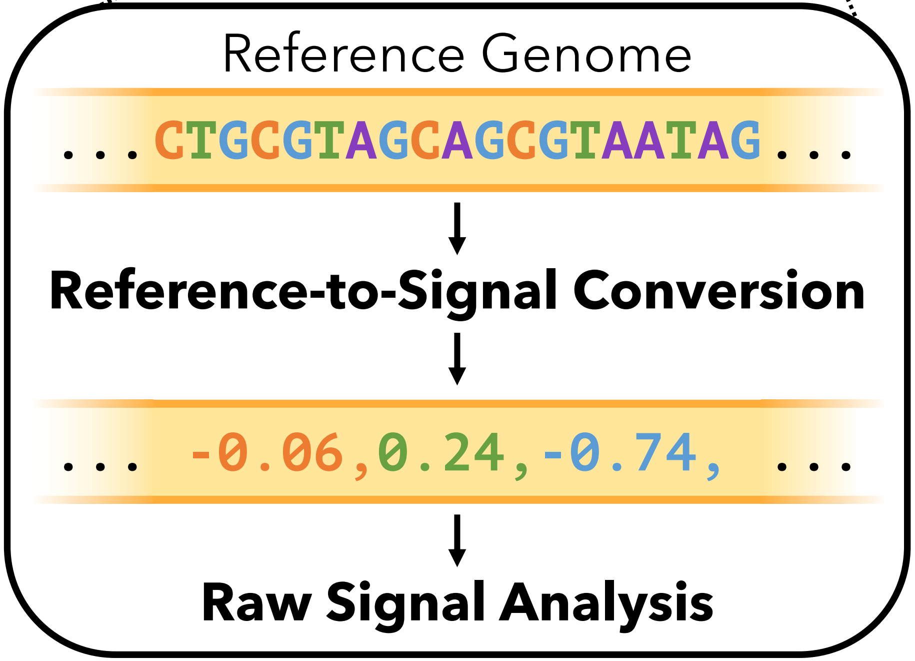
Sequencing is stopped early with a real-time decision

Real-Time Raw Signal Analysis



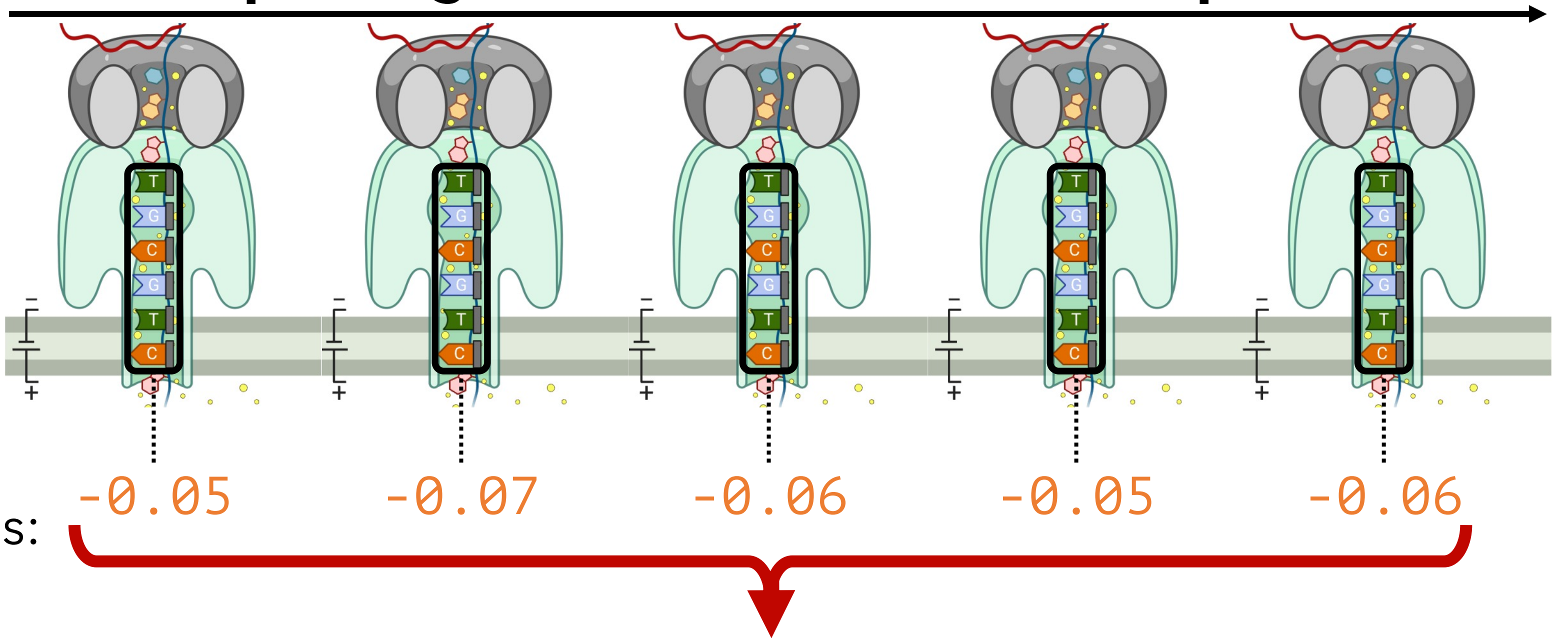
✓ We can **avoid** the **costly basecalling** step by directly analyzing raw signals

- Converting the **reference genome** to **its expected signal values** using a pre-constructed sequence-to-signal conversion model



Noise in Raw Signal Analysis

Sequencing CTGCGT with Different Nanopores



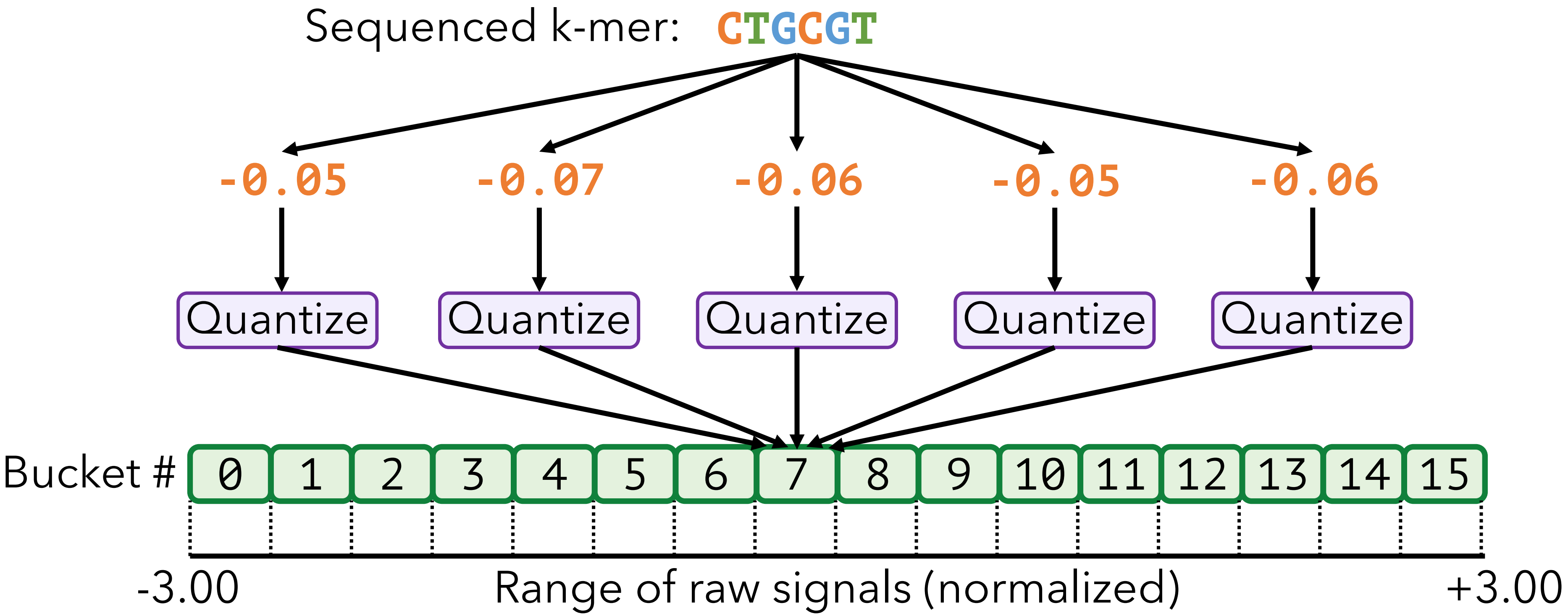
Raw Signals: -0.05 -0.07 -0.06 -0.05 -0.06

Noise causes slight differences in raw signals from **the same k-mer**

Challenge: Directly matching raw signals is not feasible

Challenge: A single k-mer is too short for accurate matching

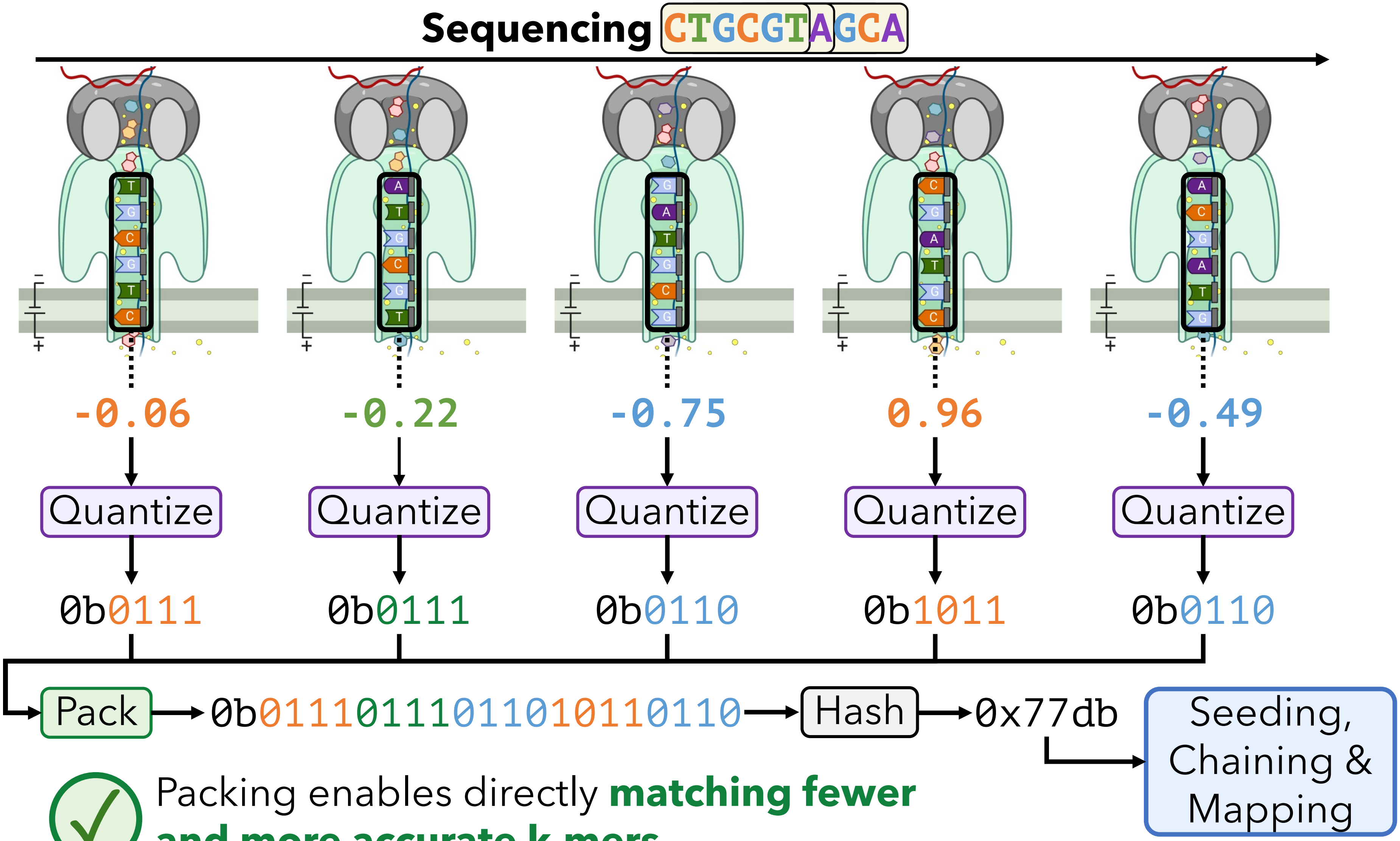
RawHash Key Idea – Quantization



✓ **Reducing noise** by **quantizing** raw signals into equal-width buckets

✓ **Enables matching raw signals** by eliminating slight differences

RawHash Key Idea – Hash-based Seeding



Sequencing CTGCGTAGCA

-0.06 → Quantize → 0b0111

-0.22 → Quantize → 0b0111

-0.75 → Quantize → 0b0110

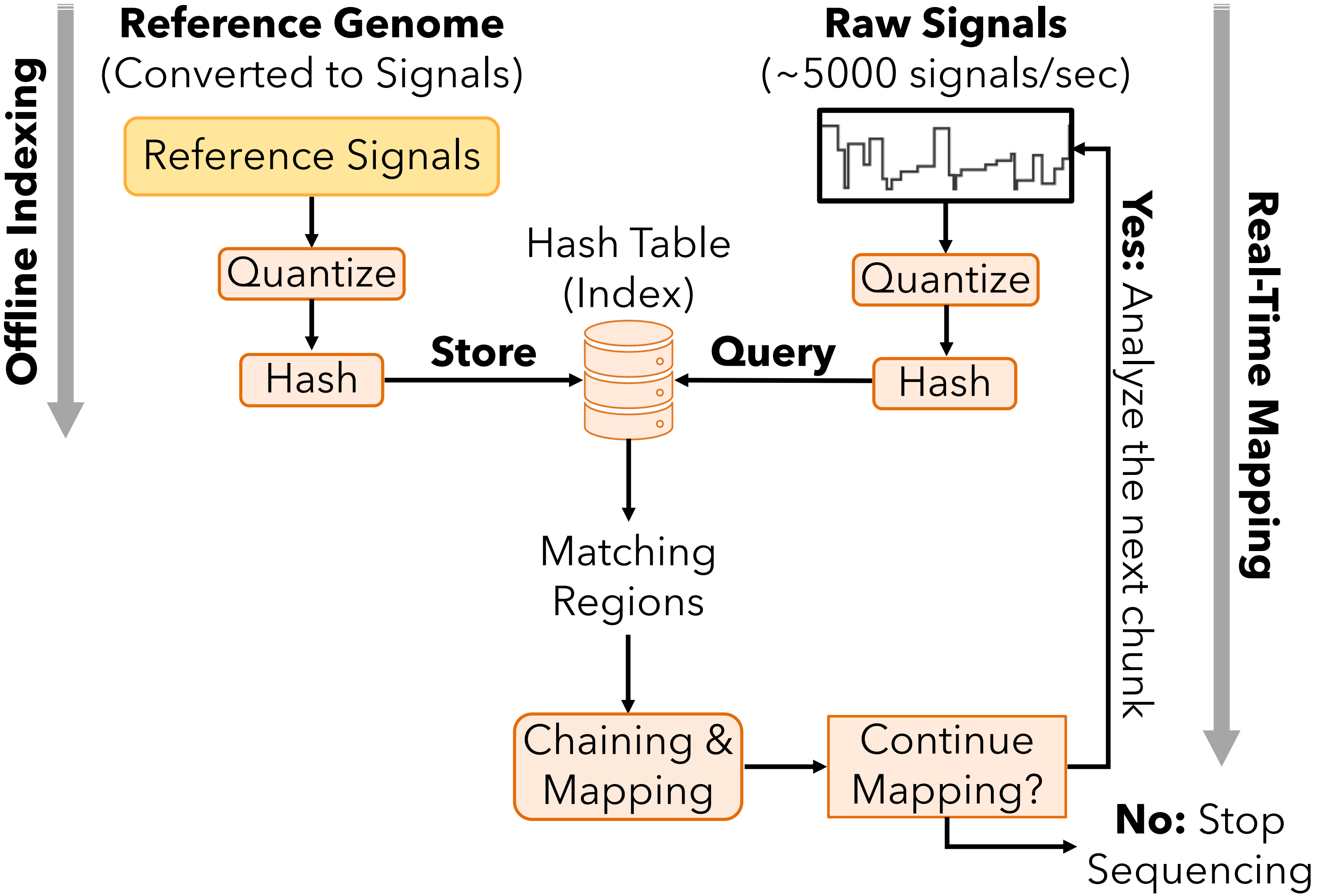
0.96 → Quantize → 0b1011

-0.49 → Quantize → 0b0110

Pack → 0b01110111011010110110 → Hash → 0x77db → Seeding, Chaining & Mapping

✓ Packing enables directly **matching fewer and more accurate k-mers**

Real-Time Mapping with RawHash



Key Results

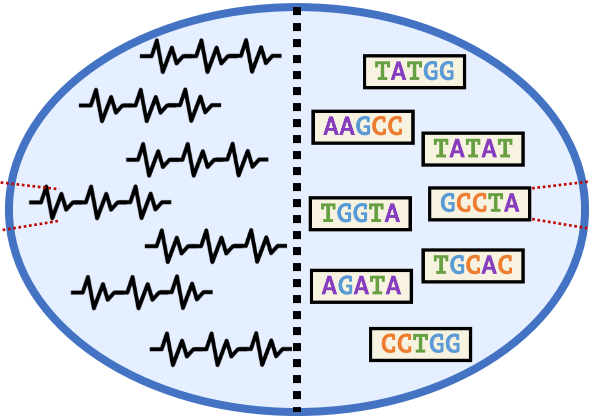
Compared to the state-of-the-art raw signal analysis tools
UNCALLED [Kovaka+'21] and **Sigmap** [Zhang+'21]:

Average speedup of **25.8**× (UNCALLED)
and **3.4**× (Sigmap)

Effective noise reduction
improves both
performance and accuracy

Up to ~1.75× better accuracy
for large (human) genomes

Core Contributions – RawHash



Noise

Noise

2 **RawHash (ISMB/ECCB'23)**

Enable hash-based search by reducing noise in raw nanopore signals

1 **BLEND (NARGAB'23 & RECOMB'23)**

Tolerate arbitrary noise with a hash-based search

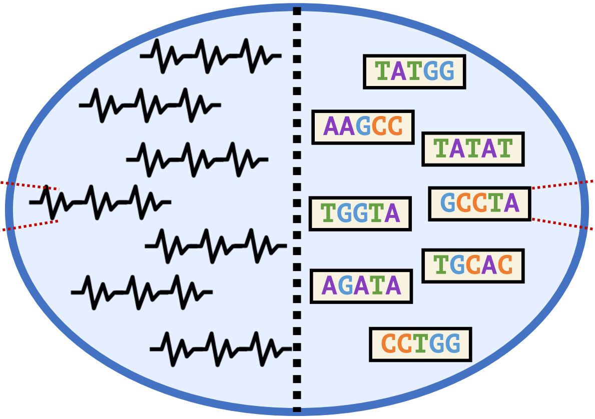
3 **RawHash2 (Bioinformatics'24)**

Better noise reduction by leveraging novel insights

4 **Rawsamble (arXiv'24 & HitSeq'24)**

Enabling new applications by overlapping and assembling raw nanopore signals

Core Contributions – RawHash2



Noise

Noise

2 **RawHash (ISMB/ECCB'23)**

Enable hash-based search by reducing noise in raw nanopore signals

1 **BLEND (NARGAB'23 & RECOMB'23)**

Tolerate arbitrary noise with a hash-based search

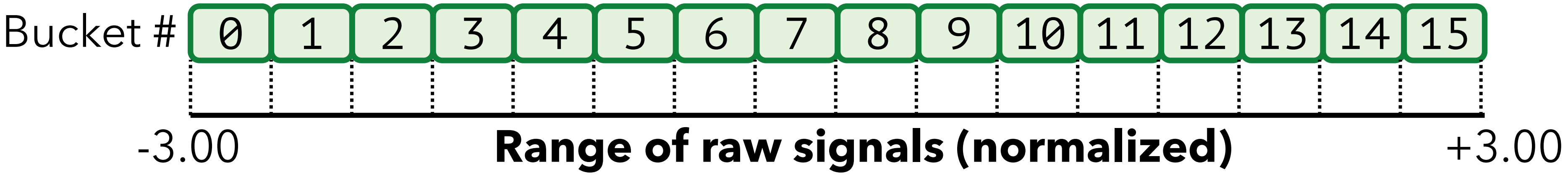
3 **RawHash2 (Bioinformatics'24)**

Better noise reduction by leveraging novel insights

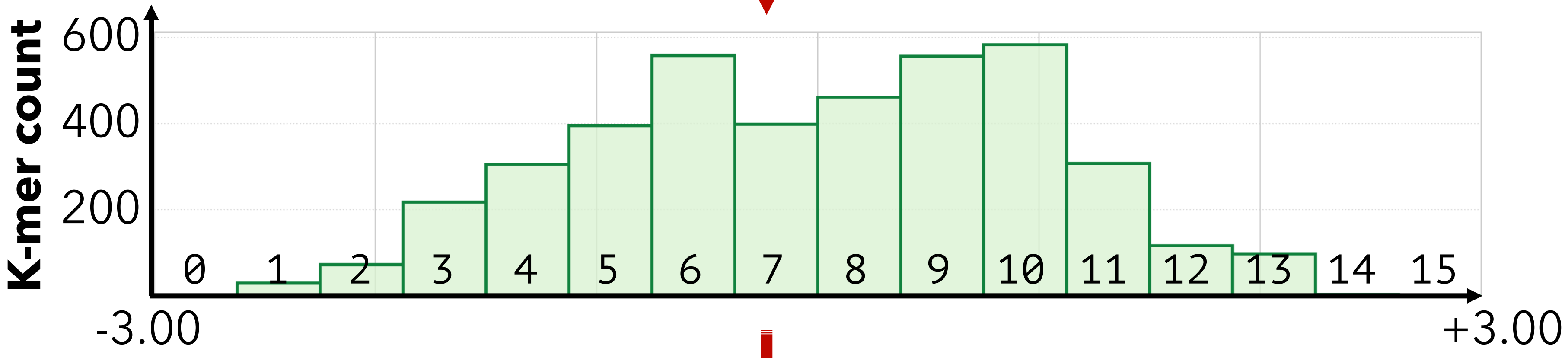
4 **Rawsamble (arXiv'24 & HitSeq'24)**

Enabling new applications by overlapping and assembling raw nanopore signals

Better Understanding of Noise



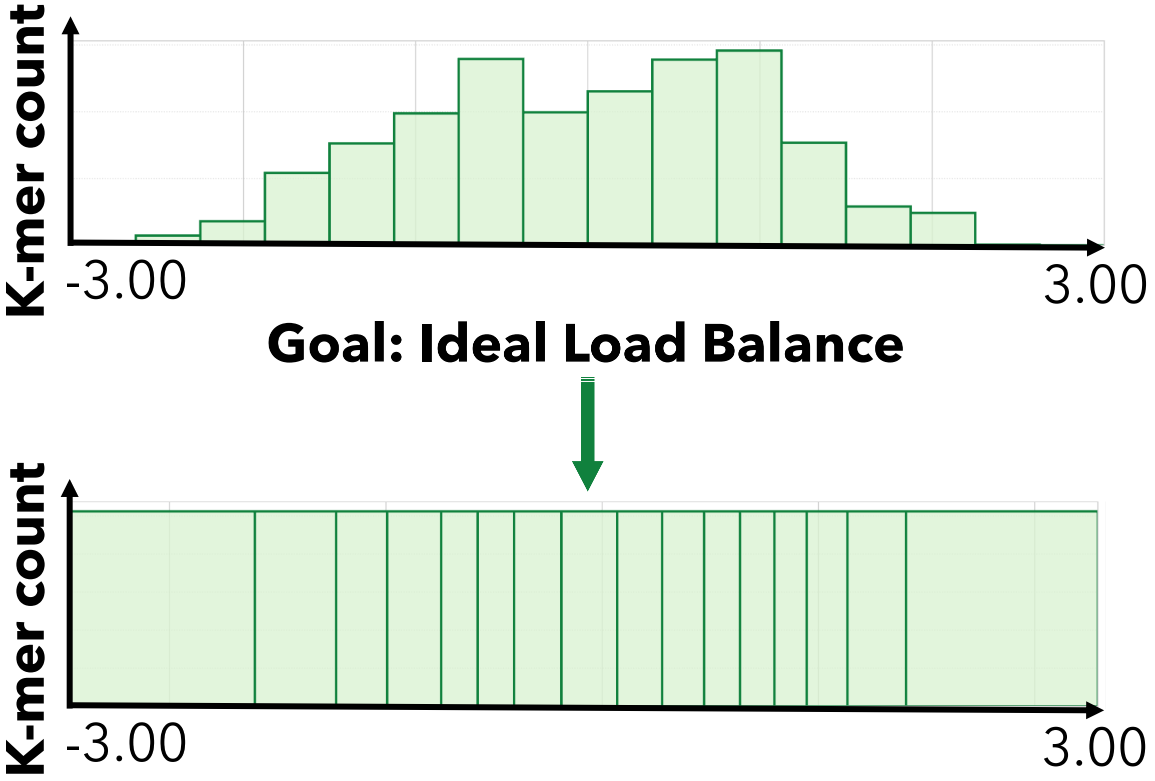
Equal-width buckets leads to **unbalanced loading**



More **false hash matches** due to reduced uniqueness

Adaptive Quantization

- **Key Idea:** Quantizing raw signals with **non-equal bucket widths to maximize load balancing**

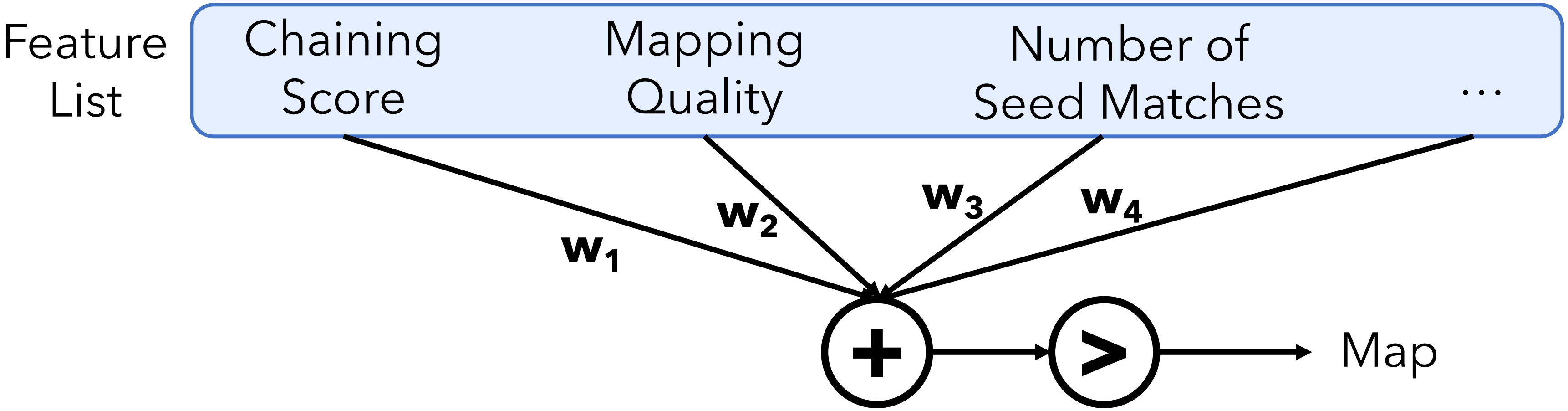


Goal: Ideal Load Balance

✓ **Adaptive quantization reduces collisions** caused due to skewed raw signal distributions

Other Key Improvements in RawHash2

- **Weighted mapping decisions**



Feature List

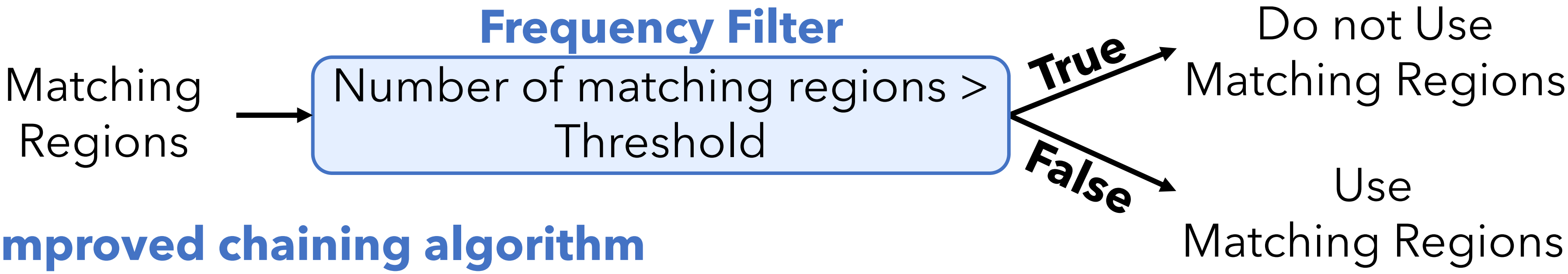
Chaining Score

Mapping Quality

Number of Seed Matches

…

- **Sampling strategies** for reduced storage and computation overheads
  - Frequency filter and minimizer sketching



Frequency Filter

Matching Regions

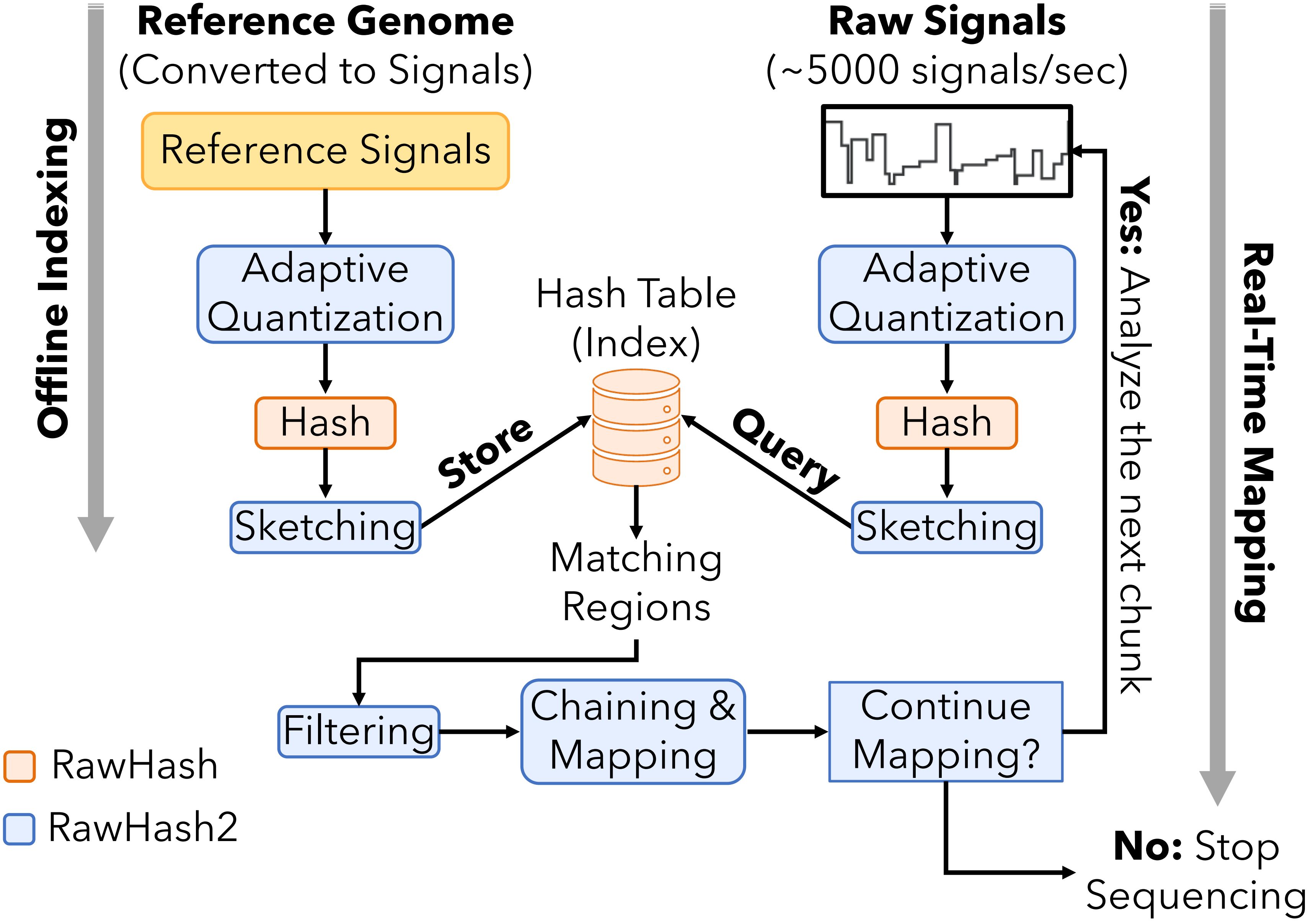
Number of matching regions > Threshold

Do not Use Matching Regions

Use Matching Regions

- **Improved chaining algorithm**
  - More sensitive scoring functions

Real-Time Mapping with RawHash2



Offline Indexing

Reference Genome
(Converted to Signals)

Reference Signals

Adaptive Quantization

Hash

Sketching

Store

Hash Table (Index)

Matching Regions

Query

Raw Signals
(~5000 signals/sec)

Adaptive Quantization

Hash

Sketching

Filtering

Chaining & Mapping

Continue Mapping?

Yes: Analyze the next chunk

No: Stop Sequencing

Real-Time Mapping

🟧 RawHash

🟦 RawHash2

Evaluation Methodology

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

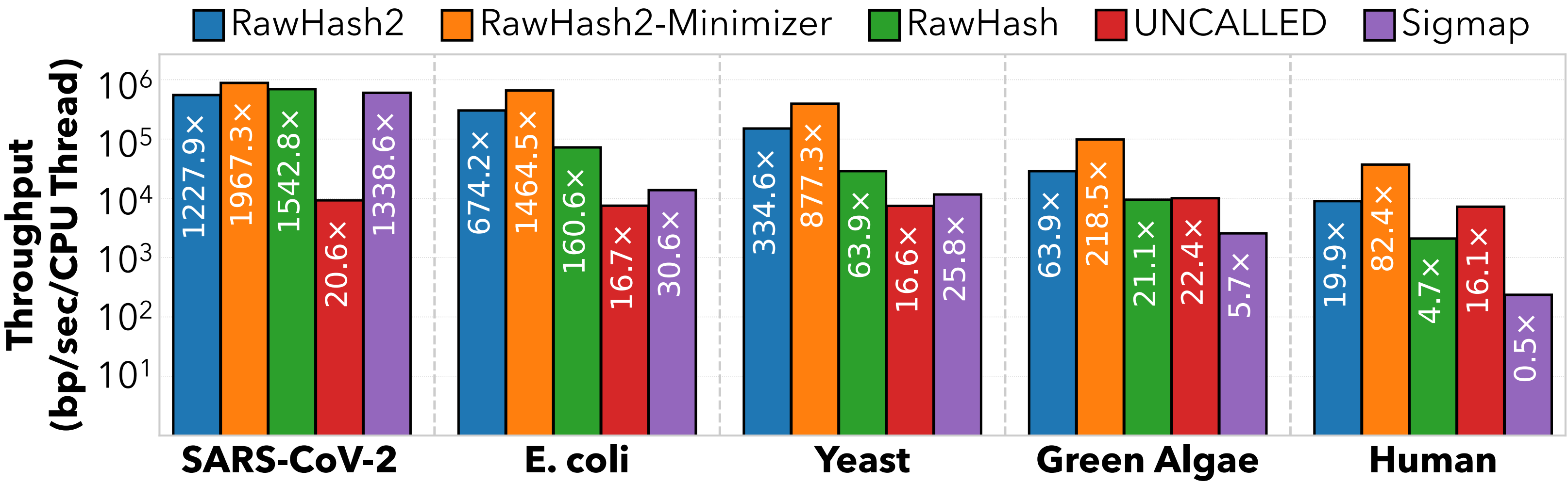
- Compared to **UNCALLED** [Kovaka+, Nat. Biotech.'21], **Sigmap** [Zhang+, ISMB/ECCB'21], and **RawHash** [Firtina+, ISMB/ECCB'23]

- **Use cases** for real-time genome analysis:

1. Read mapping
2. Relative abundance estimation
3. Contamination analysis

Key Results – Throughput

- Data generation throughput of **a single nanopore**: **~450 bp/sec**
  - **A single nanopore device** contains roughly 512 to 2500 nanopores
- Computation throughput of a **single CPU thread**: **bases processed/sec**
  - **Scalability:** The number of nanopores that a single CPU thread can process

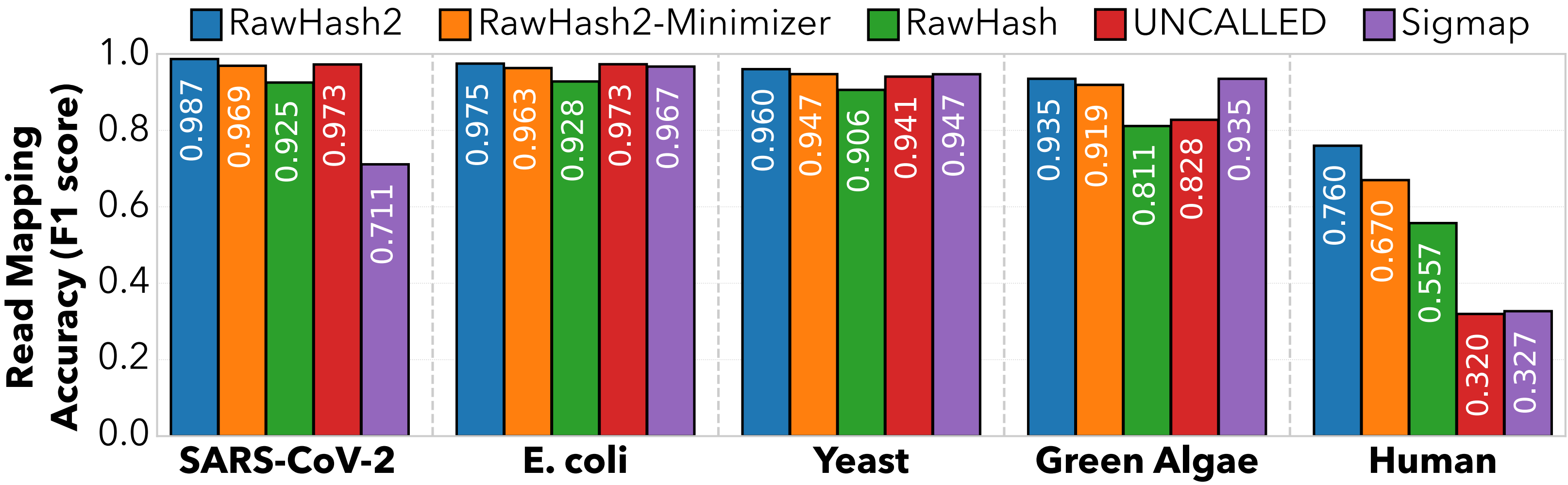


RawHash2 average speedup:
26.5× (UNCALLED), **19.2**× (Sigmap), and **4**× (RawHash)

RawHash2-Minimizer average speedup: 2.5× (RawHash2)

Key Results – Mapping Accuracy

- Accuracy of **mapping positions** (F1 score)
  - Ground truth: Mapping positions of **basecalled sequences** using minimap2



RawHash2 provides the best accuracy in all datasets
(up to ~2.4× for large genomes)

RawHash2-Minimizer provides mapping accuracy **comparable to RawHash2**

Conclusion – RawHash2

RawHash2 average speedup:
26.5× (UNCALLED), **19.2**× (Sigmap), and **4**× (RawHash)

RawHash2-Minimizer average speedup: 2.5× (RawHash2)

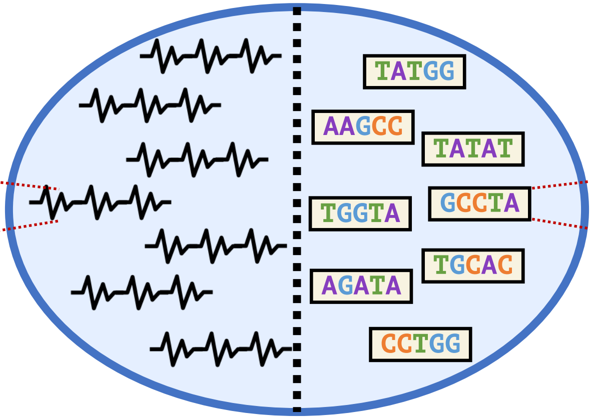
Better understanding of noise can be utilized to further improve **performance and accuracy**

RawHash2 provides the best accuracy in all datasets
(up to ~2.4× for large genomes)

RawHash2-Minimizer provides mapping accuracy **comparable to RawHash2**

Core Contributions – RawHash2

Noise



Noise

1 **BLEND (NARGAB'23 & RECOMB'23)**

Tolerate arbitrary noise with a hash-based search

2 **RawHash (ISMB/ECCB'23)**

Enable hash-based search by reducing noise in raw nanopore signals

3 **RawHash2 (Bioinformatics'24)**

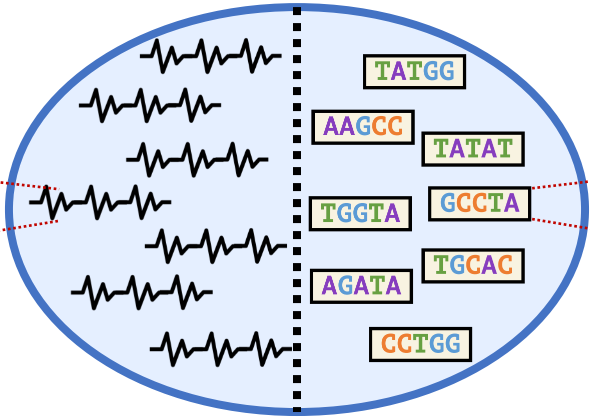
Better noise reduction by leveraging novel insights

4 **Rawsamble (arXiv'24 & HitSeq'24)**

Enabling new applications by overlapping and assembling raw nanopore signals

Core Contributions – Rawsamble

Noise



Noise

1

BLEND (NARGAB'23 & RECOMB'23)

Tolerate arbitrary noise with a hash-based search

2

RawHash (ISMB/ECCB'23)

Enable hash-based search by reducing noise in raw nanopore signals

3

RawHash2 (Bioinformatics'24)

Better noise reduction by leveraging novel insights

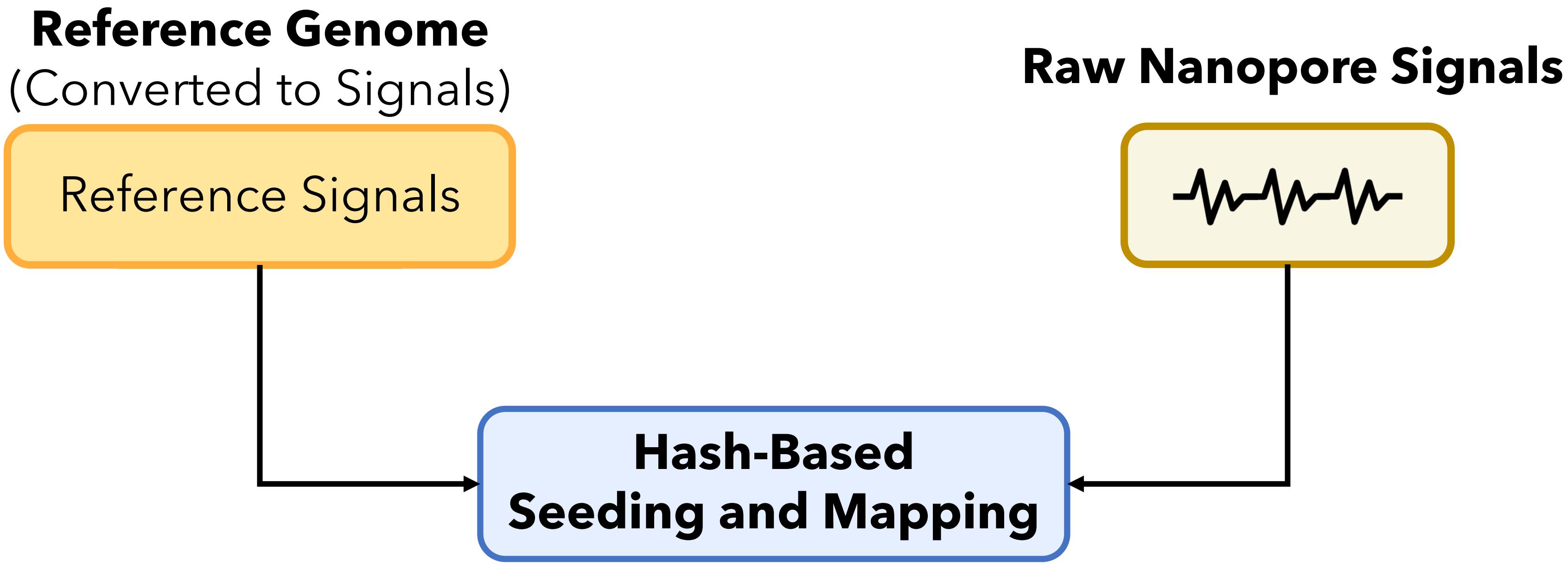
4

Rawsamble (arXiv'24 & HitSeq'24)

Enabling new applications by overlapping and assembling raw nanopore signals

BLEND → RawHash → RawHash2 → **Rawsamble**

Beyond Reference Mapping: Overlapping

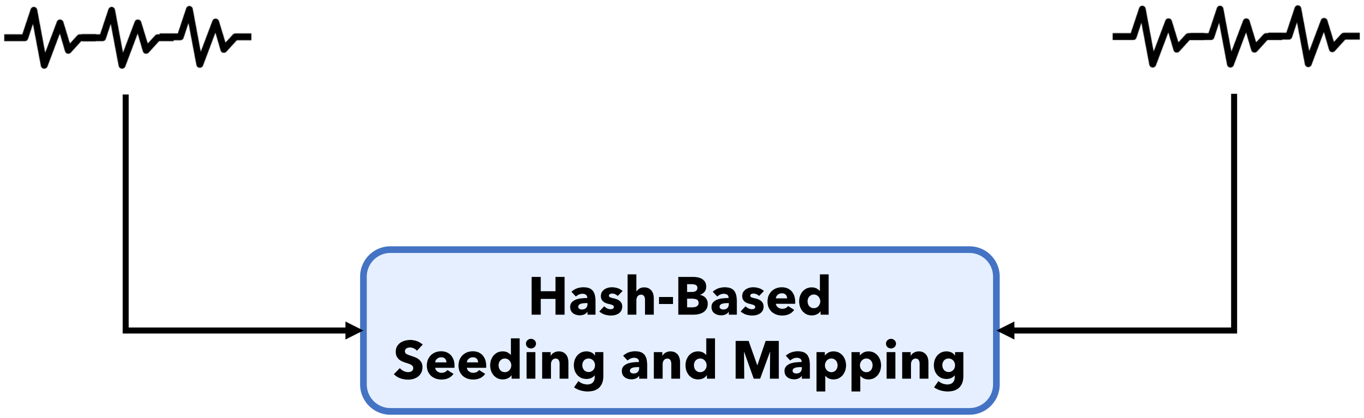


Challenge: Reference genome is not always available

Assembly: Constructing genome from **overlapping reads**

Existing solutions cannot find overlapping reads **without basecalling**

Challenges with Overlapping Raw Signals



Hash-Based Seeding and Mapping

Challenge: Identifying hash matches **when both signals are noisy**

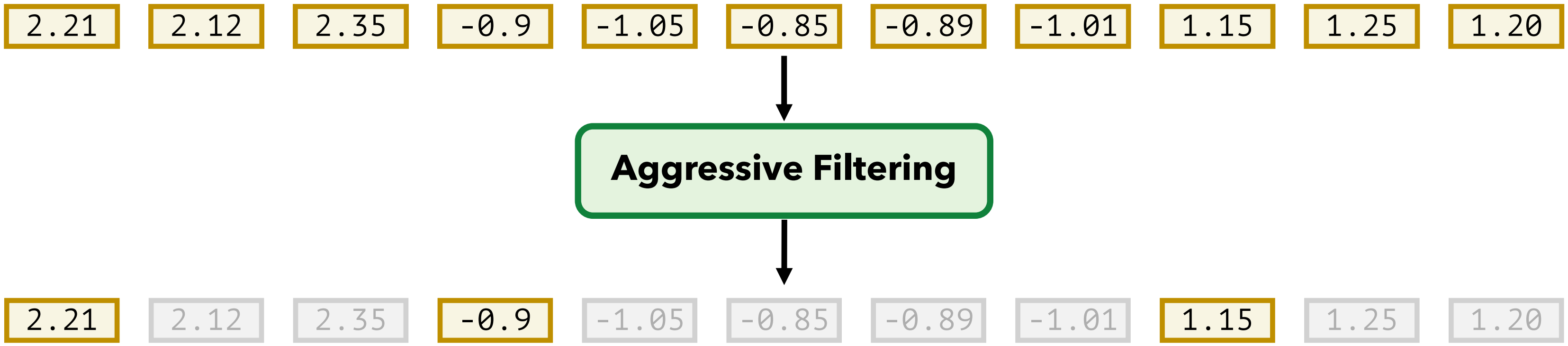
Challenge: Finding **many** useful overlapping pairs (all-vs-all overlapping)

Challenge: Generating **long paths** from useful overlaps

Key Improvements in Rawsamble

- **Aggressively filtering** consecutive and similar signals to **substantially reduce noise** at the cost of data loss

Raw Nanopore Signals

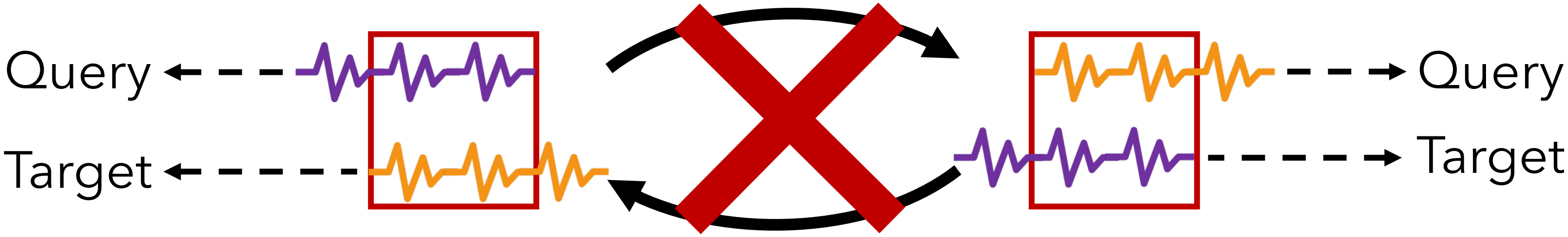


2.21 | 2.12 | 2.35 | -0.9 | -1.05 | -0.85 | -0.89 | -1.01 | 1.15 | 1.25 | 1.20

Aggressive Filtering

2.21 | 2.12 | 2.35 | -0.9 | -1.05 | -0.85 | -0.89 | -1.01 | 1.15 | 1.25 | 1.20

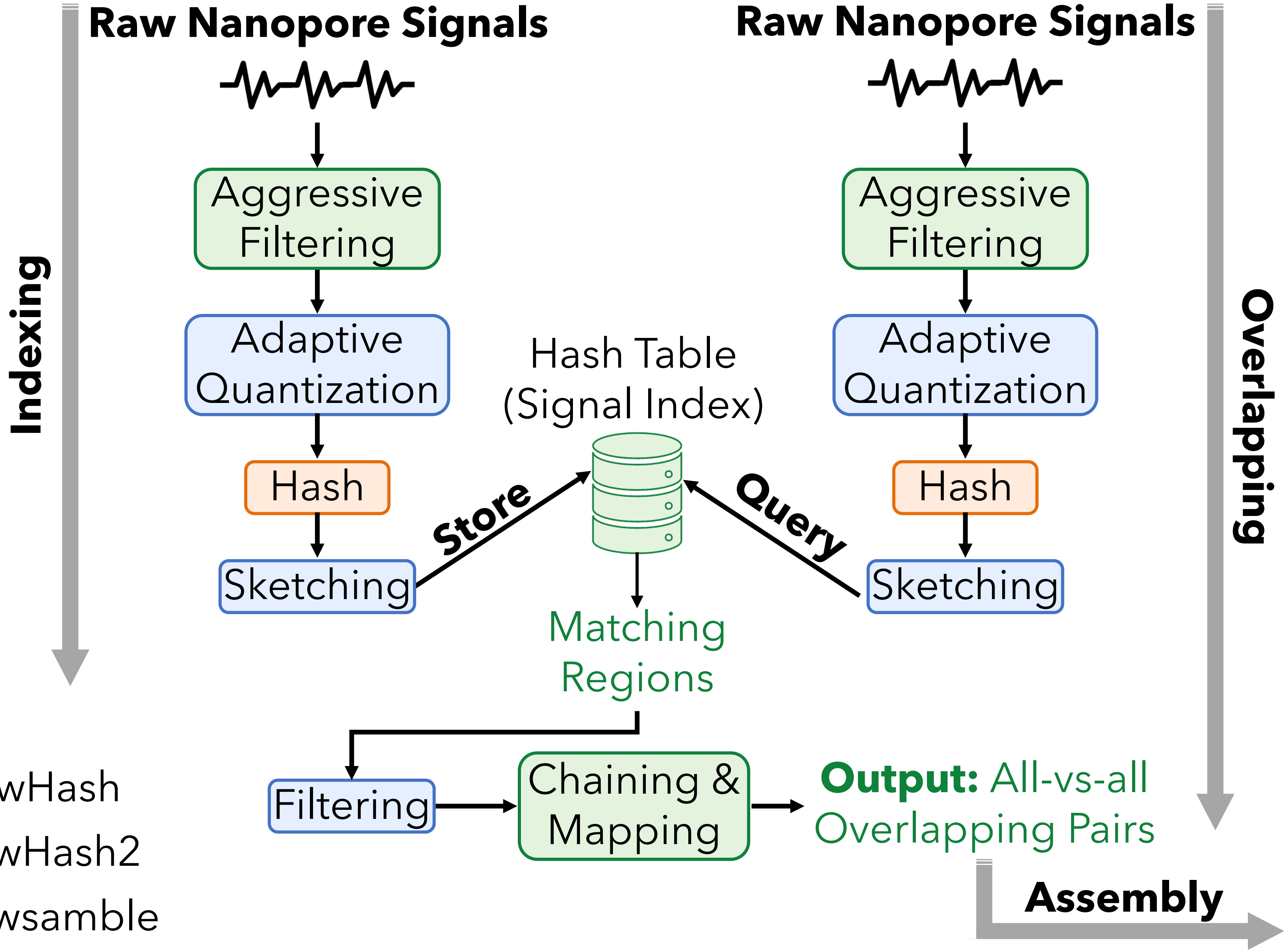
- **Avoid cyclic overlaps** with deterministic comparisons



- Identifying and reporting all **highly accurate chains** to generate **all-vs-all overlapping pairs**

- Generating the hash table index **from raw nanopore signals**

Integrating Rawsamble into RawHash2



- 🟧 RawHash
- 🟦 RawHash2
- 🟩 Rawsamble

Evaluation Methodology

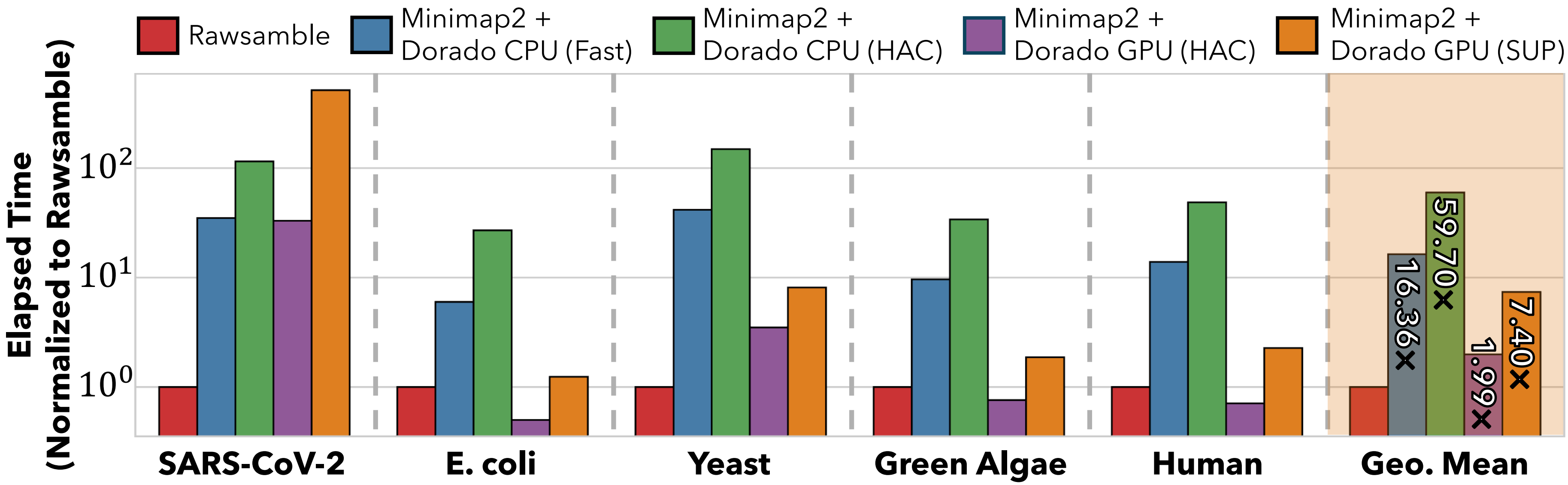
- Rawsamble is integrated into **RawHash2** [Firtina+, Bioinformatics'24]

- Compared to the **minimap2** [Li, Bioinformatics'18] overlaps (forward strand)
  - Basecalling with **Dorado**'s various models (using CPUs & GPUs)

- Use case for raw signal overlapping:
  - De novo assembly construction using **miniasm** [Li, Bioinformatics'16]

- **Real datasets** with
  - Various **coverage** (0.6× – 445×) and
  - **Genome lengths** (viral to human genomes)

Key Results – Performance

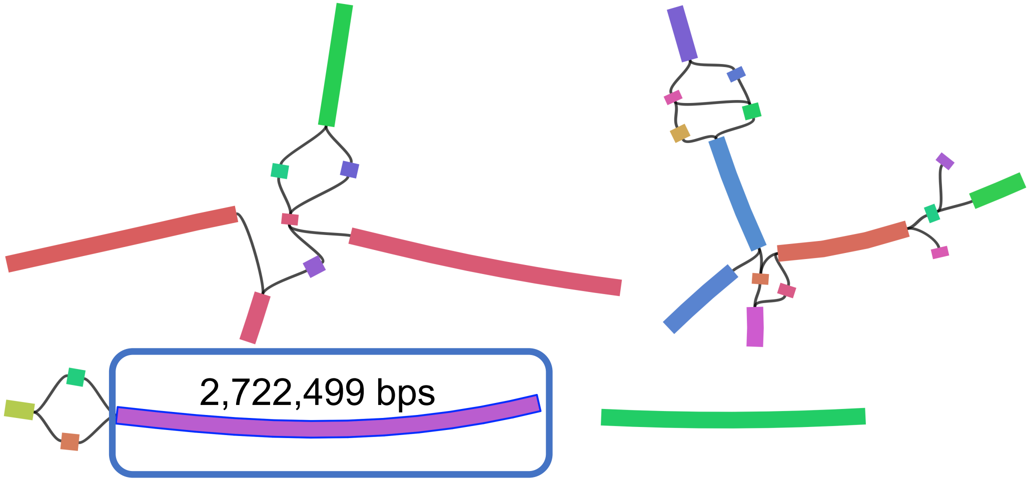


Compared to the fastest CPU model (Fast):
Average speedup of **16.36**×

Compared to the conventional GPU model (HAC):
Average speedup of **1.99**×

Key Results – *de novo* Assemblies

E. coli Assembly
(From the Rawsamble Overlaps)

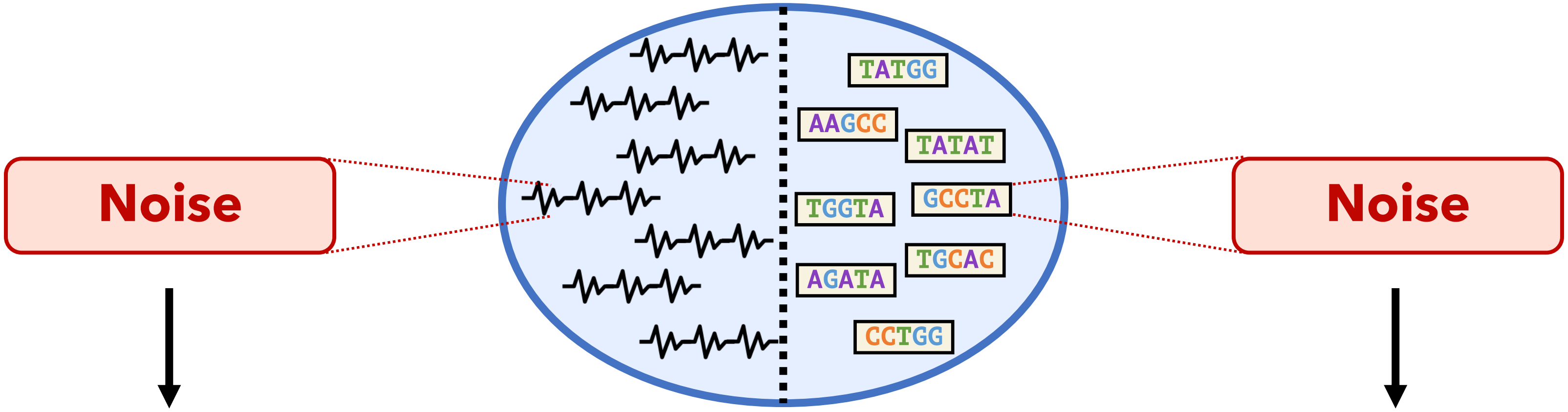


First *de novo* assemblies ever constructed from raw signal overlaps **without basecalling**

Contigs of **half the E. coli genome length** (~2.7 Mbases): **~400× longer than the average read length**

New directions in genome analysis can be enabled without basecalling

Core Contributions



1 **BLEND (NARGAB'23 & RECOMB'23)**

Tolerate arbitrary noise with a hash-based search

2 **RawHash (ISMB/ECCB'23)**

Enable hash-based search by reducing noise in raw nanopore signals

3 **RawHash2 (Bioinformatics'24)**

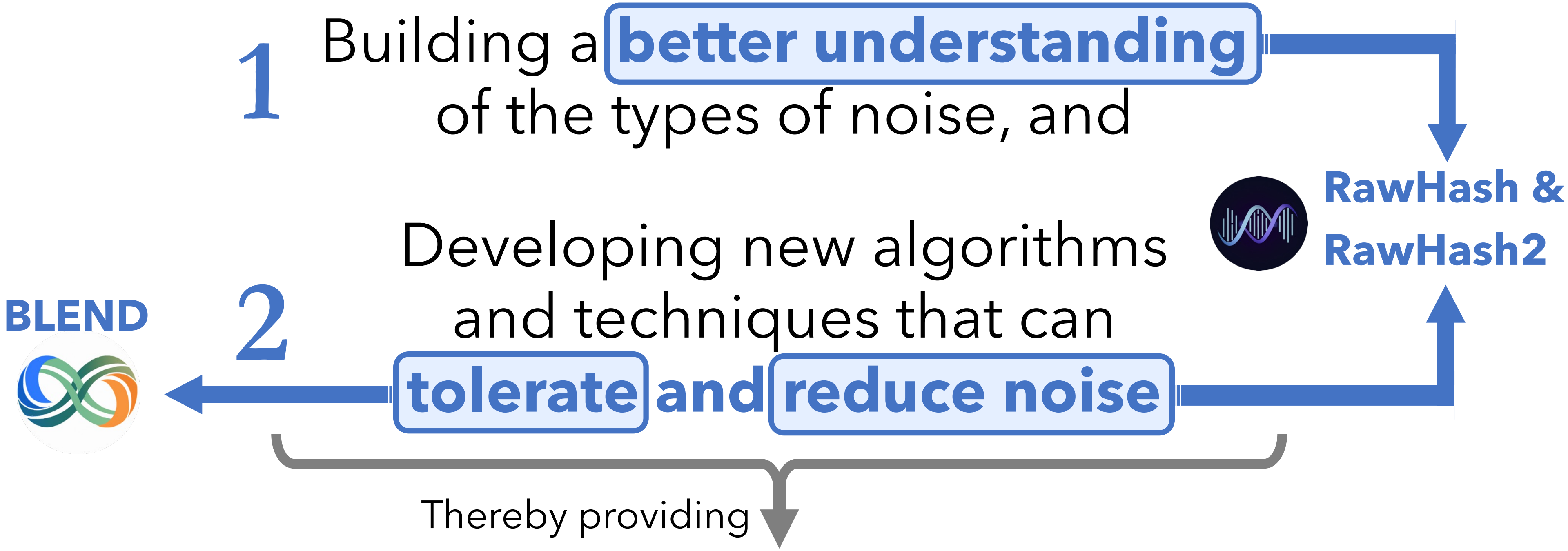
Better noise reduction by leveraging novel insights

4 **Rawsamble (arXiv'24 & HitSeq'24)**

Enabling new applications by overlapping and assembling raw nanopore signals

Conclusion

We can mitigate **noise** in sequencing data and analysis by



1 Building a **better understanding** of the types of noise, and

2 Developing new algorithms and techniques that can **tolerate** **and** **reduce noise**

RawHash &
RawHash2

BLEND

Thereby providing

Accurate, scalable, and real-time analysis of sequencing data and enabling **new applications** in genome analysis

Rawsamble

Future Research Directions

Guiding Basecalling with Raw Signal Analysis

- Pre-basecalling filtering
- Utilizing raw signal overlaps with basecallers

Full Genome Analysis without Basecalling

- Specialized assemblers for raw signals
- Full downstream analysis

Rethinking Heuristic Techniques for Highly Accurate Reads

- Very long seeds with fuzzy seed matching
- Reference-free comparisons

My Involvements During my Ph.D.

🟨 First/Co-First Author Publications

Thesis Publications				🟨 Firtina+ **NARGAB** 🟨 Firtina+ **ISMB/ECCB**	🟨 Firtina+ **Bioinformatics** 🟨 Firtina+ **arXiv**
Nanopore Signal Analysis & Basecalling					Singh+ **Gen. Bio.** Cavlak+ **Frnt. Gn.** Lindegger+ **IEEE Access**
Error Correction	🟨 Firtina+ **Bioinformatics**				
Hardware Accelerators	Cali+ **MICRO**		Ghiasi+ **ASPLOS** Cali+ **ISCA** Mao+ **MICRO**	Kanellopoulos+ **MICRO** Shahroodi+ **MICRO**	🟨 Firtina+ **ACM TACO** Ghiasi+ **ISCA**
Metagenomics			Shahroodi+ **IEEE Access**	Gollwitzer+ **arXiv**	
Filters & Liftover			Kim+ **Bioinformatics**	Rumpf+ **arXiv**	🟨 Kim & Firtina+ **IEEE/ACM TCBB**
Surveys			Alser+ **CSBJ**	🟨 Mutlu & Firtina **DAC**	Alser+ **Nature Protocols**

2020 **2021** **2022** **2023** **2024**

SAFARI

Acknowledgements

- **Advisor:** Onur Mutlu
- **Committee Members:** Reetuparna Das, Hasindu Gamaarachchi, Benjamin Langmead, and Heng Li
- **Chair:** Janos Vörös

- **Funding agencies and industry sponsors:**
  - BioPIM, SNSF, Intel, Google, Huawei, Microsoft, VMware, and SRC

- **Colleagues, mentors, collaborators, and friends worldwide:**
  - SAFARI Research Group members
  - Can Alkan & Alkan Lab

- **Family:**
  - **My parents,** Emine and Turan
  - **My wife,** Çiçek

Enabling Fast, Accurate, and Efficient Real-Time Genome Analysis via New Algorithms and Techniques

Can Firtina

Doctoral Examination
11.11.2024

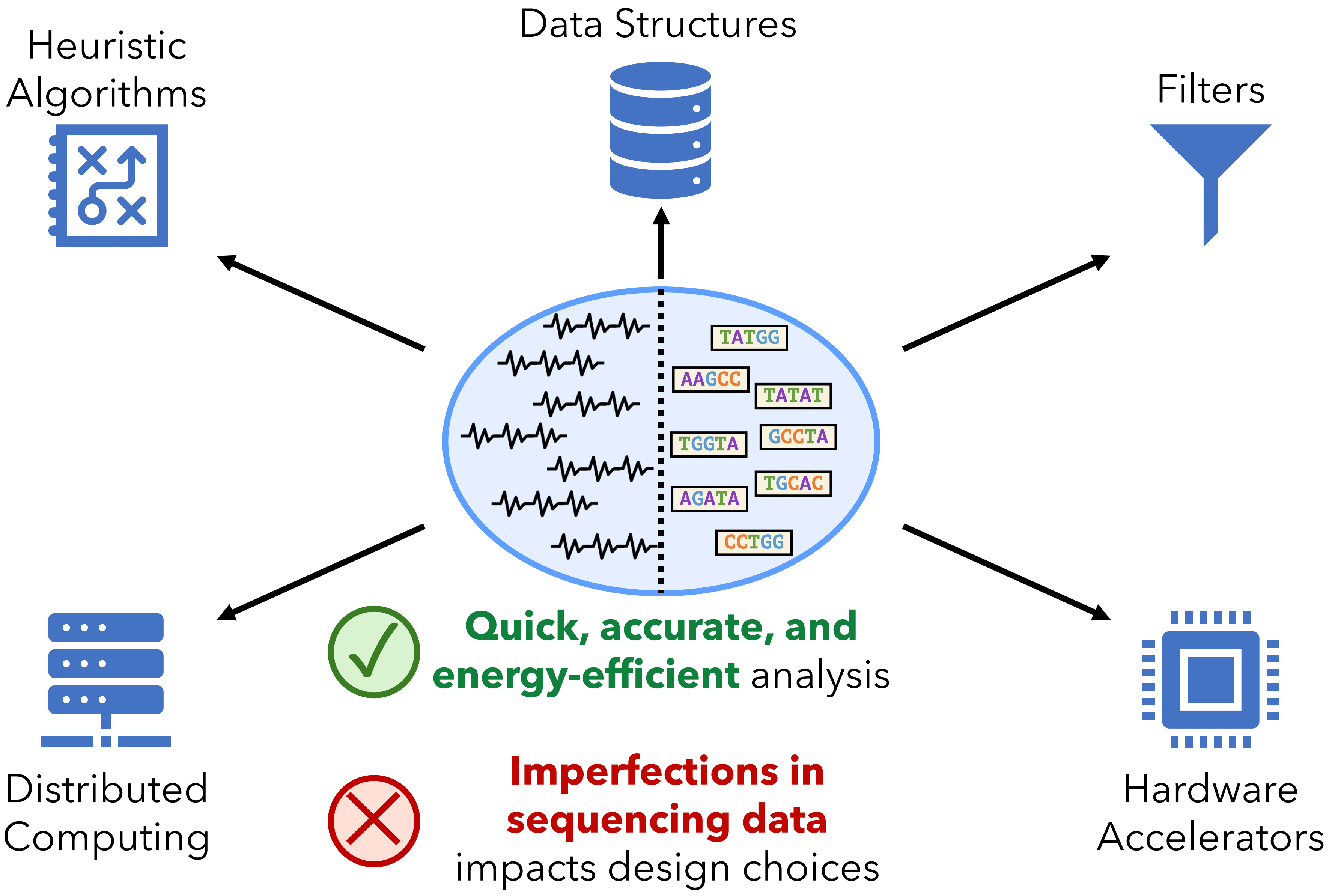
Advisor:
Onur Mutlu (ETH Zurich)

Co-Examiners:
Reetuparna Das (University of Michigan)
Hasindu Gamaarachchi (UNSW Sydney)
Benjamin Langmead (Johns Hopkins University)
Heng Li (Harvard Medical School)

ETH zürich

SAFARI

Sequencing Data Analysis



Heuristic Algorithms

Data Structures

Filters

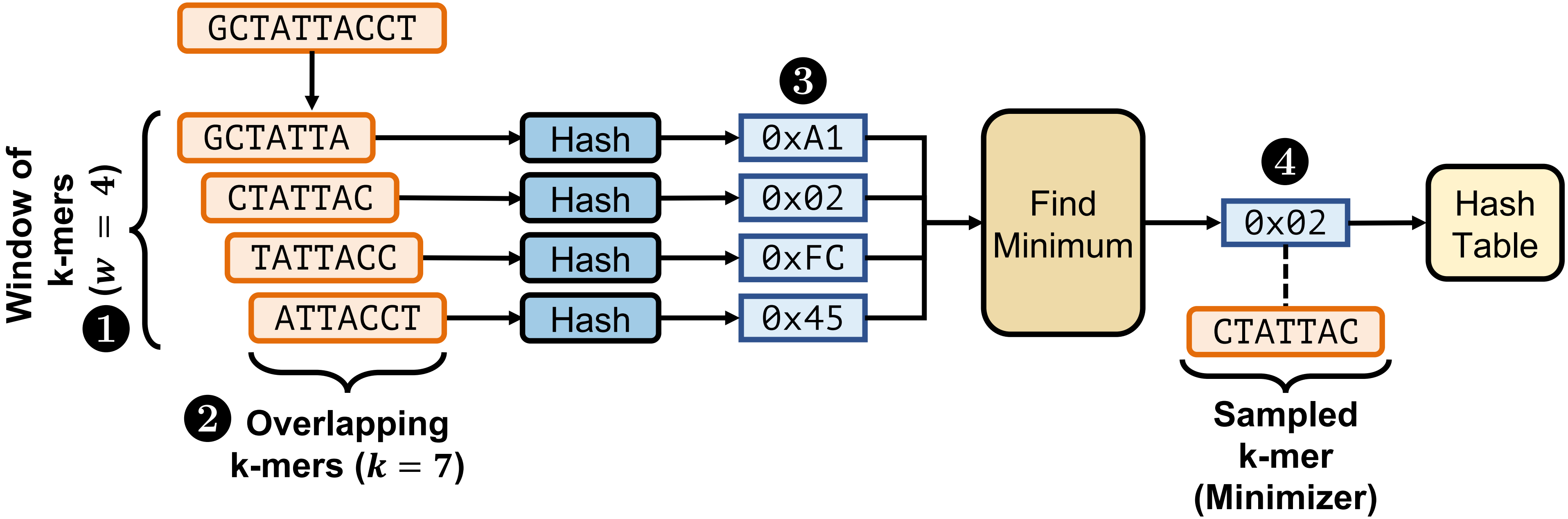
Distributed Computing

Hardware Accelerators

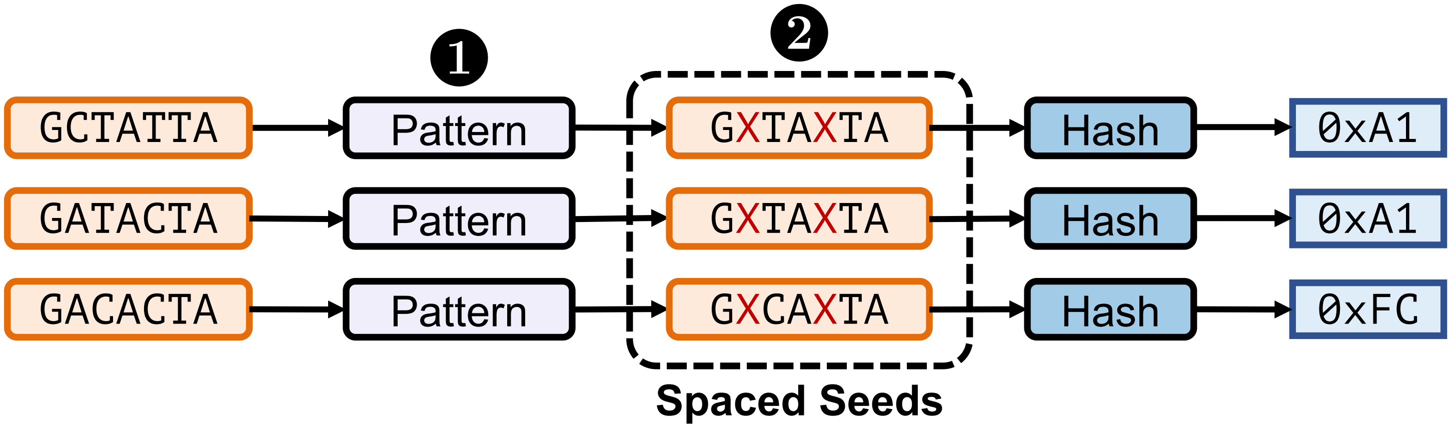
✓ **Quick, accurate, and energy-efficient** analysis

✗ **Imperfections in sequencing data** impacts design choices

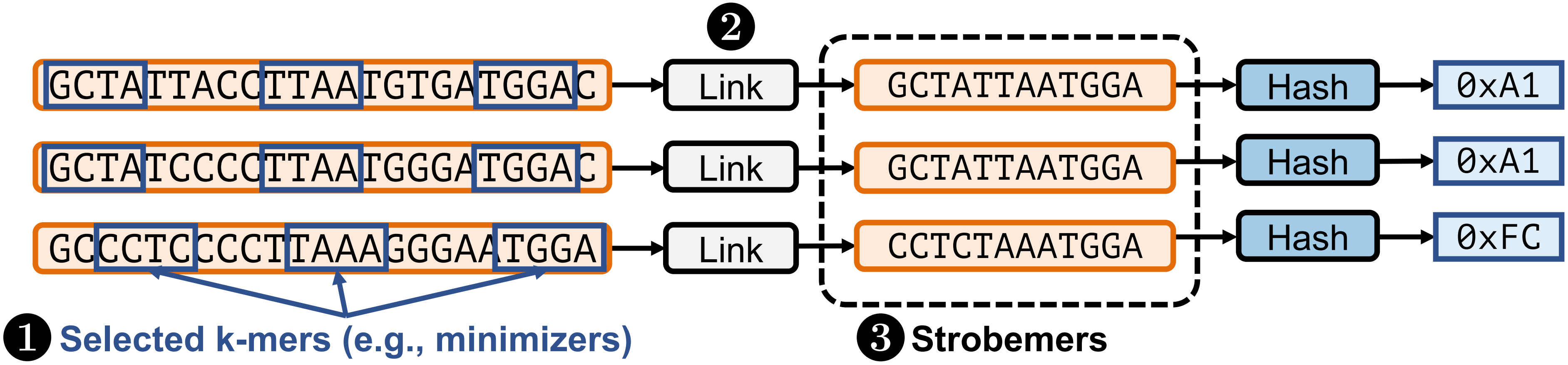
Minimizer Sketching



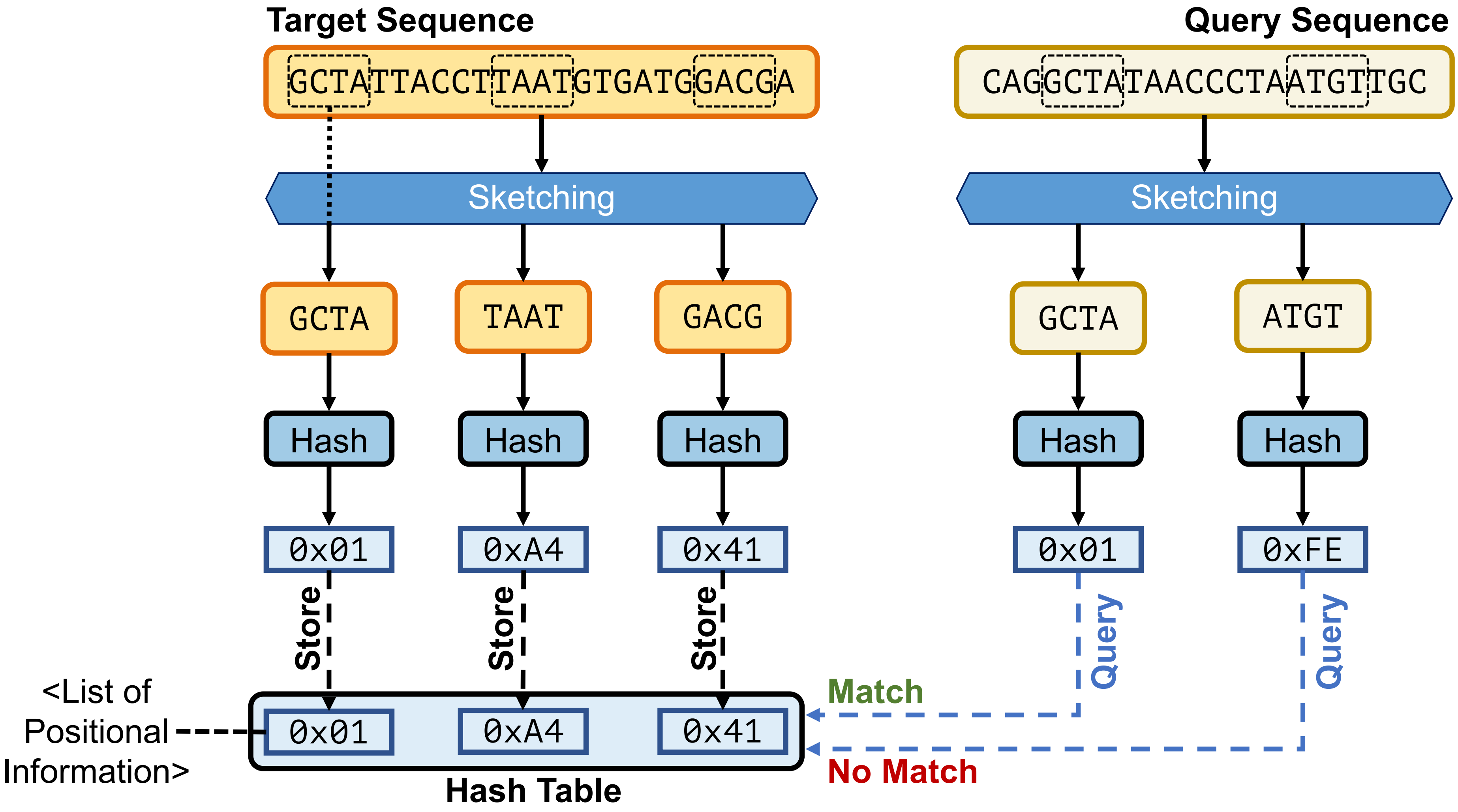
Spaced Seeding



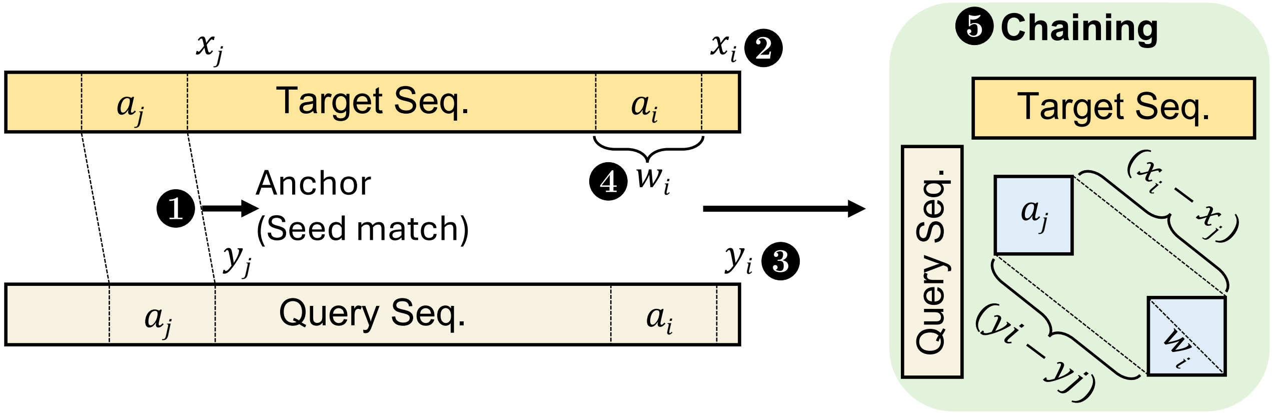
Strobemer Sketches



Hash-Based Sketching and Seed Matching



Chaining (Two Points)

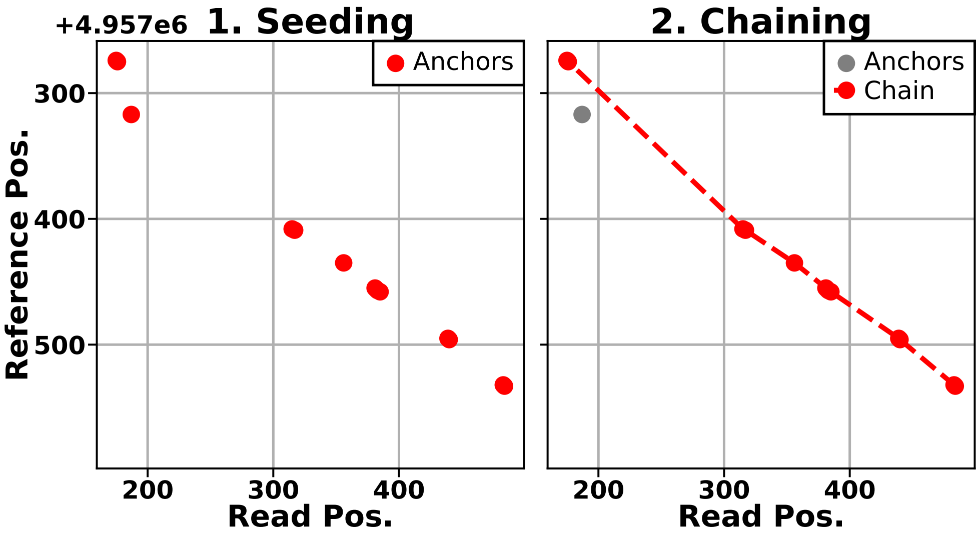


Chaining (Multiple Points)

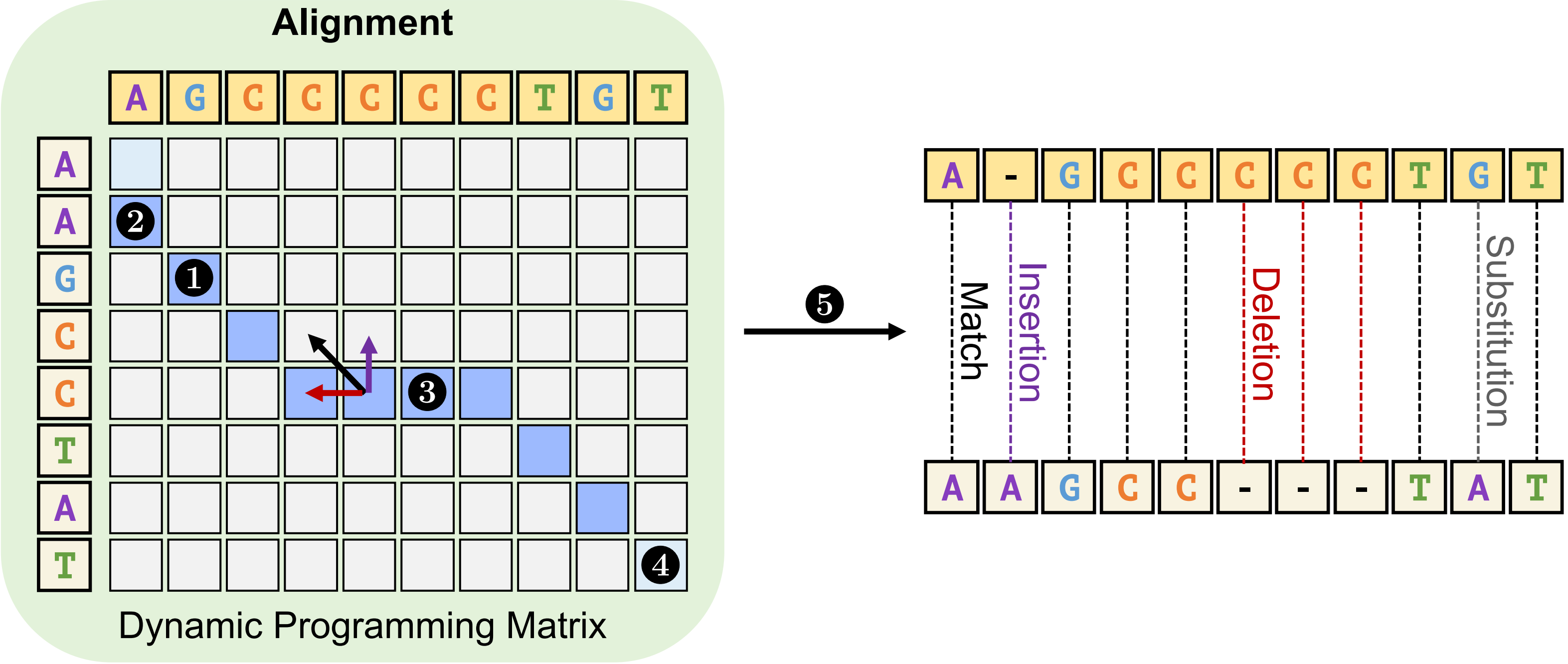
- **Exact hash value matches:** Needed for finding matching regions between a reference genome and a read

- What if there are mutations or errors?
  - **No hash (seed) match** will occur in such positions

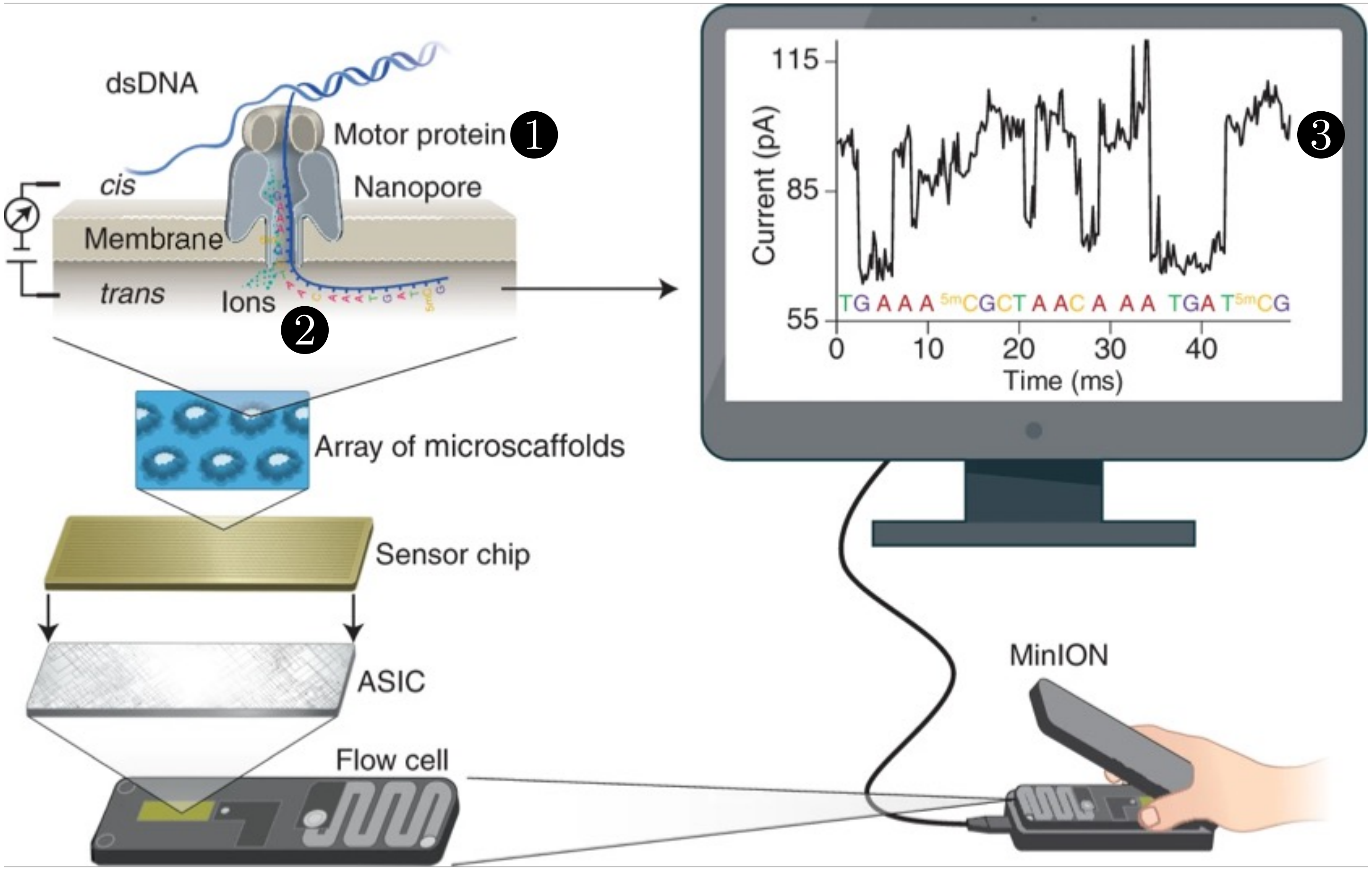
- The chaining algorithm links **exact matches in a proximity** even though there are gaps (no seed matches) between them



Sequence Alignment



Nanopore Sequencing



Source of Noise in Nanopore Sequencing

- **Stochastic thermal fluctuations in the ionic current**
  - Random ionic movement due to inherent thermal energy (Brownian motion)

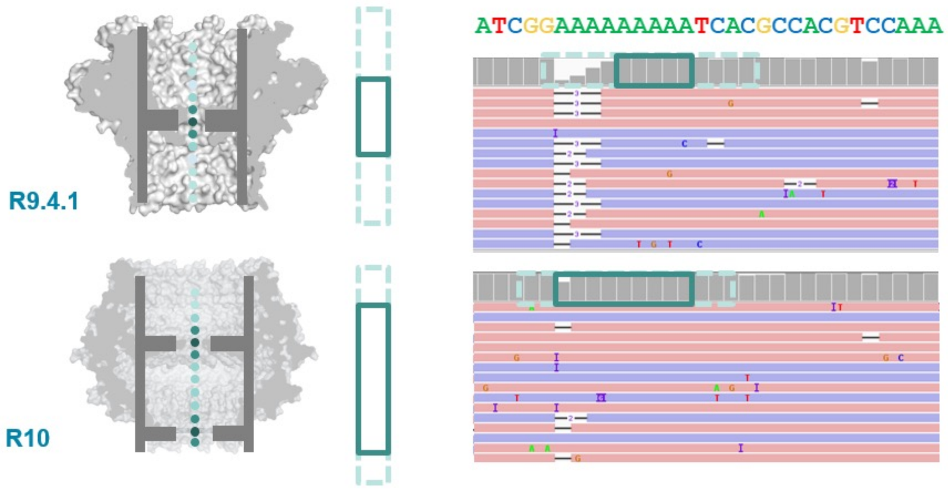
- **Variations in the translocation speed**
  - Mainly due to the motor protein

- **Environmental factors**
  - **Temperature:** Affecting enzymes including the motor protein
  - **pH levels:** Affecting charge and the shape of molecules

- **Maybe: Aging & material-related noise between nanopores**
  - Their effects potentially can be minimized with normalization techniques

R9 vs. R10 Chemistries

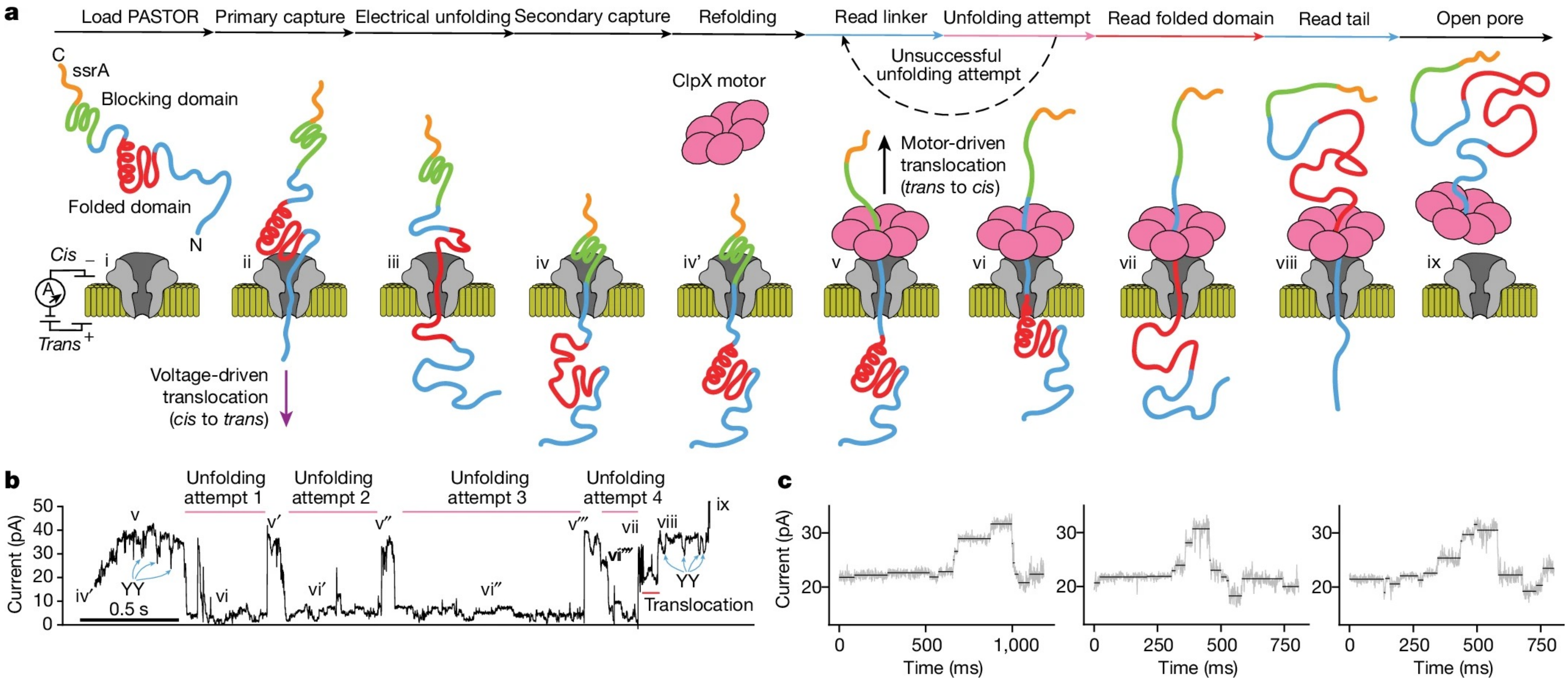
- **Dual reader head**



- **Motor protein** with more consistent translocation speed in R10

- **Duplex sequencing** in R10

Proteomics with Nanopores



Motone+, "Multi-pass, single-molecule nanopore reading of long protein strands", Nature 2024.

Related Works

- **Noise-tolerant sketching**
  - Sampling mechanisms (e.g., minimizers, syncmers, strobemers, MinHash)
  - Generated sketches must still exactly match
  - BLEND can be applied to generate their hash values to find similar sketches

- **Spaced seeds**
  - Cannot tolerate arbitrary mismatches due to fixed patterns
  - BLEND can be applied to generate hash values of spaced seeds

- **Different masking and hashing techniques**
  - LexicHash: Similar to finding maximal exact match with MinHash properties
  - Order-insensitive hashing

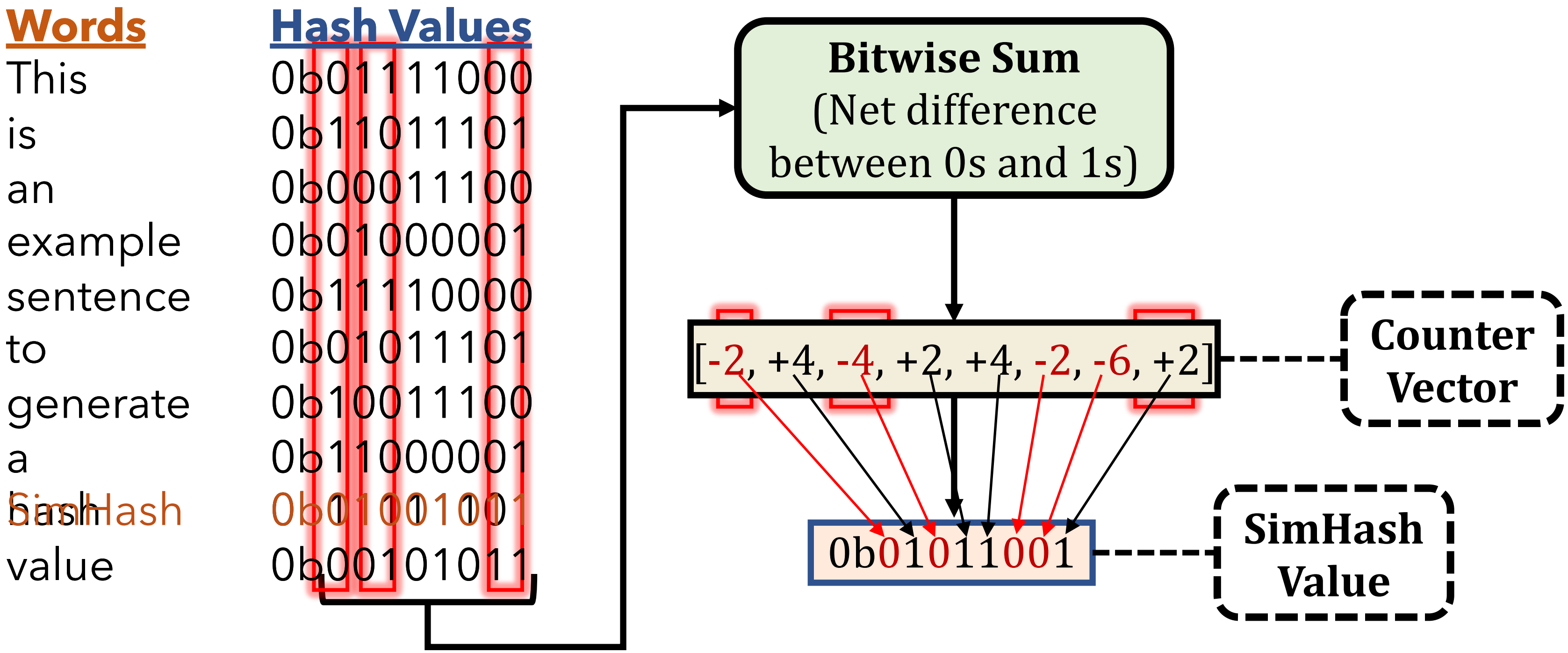
- **Other locality sensitive hashing mechanisms**
  - Identifying similarity with multiple hash matches, instead of a single one
  - Future work: DenseFly, FlyHash … (some of them use Random Matrices)

The SimHash Technique

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

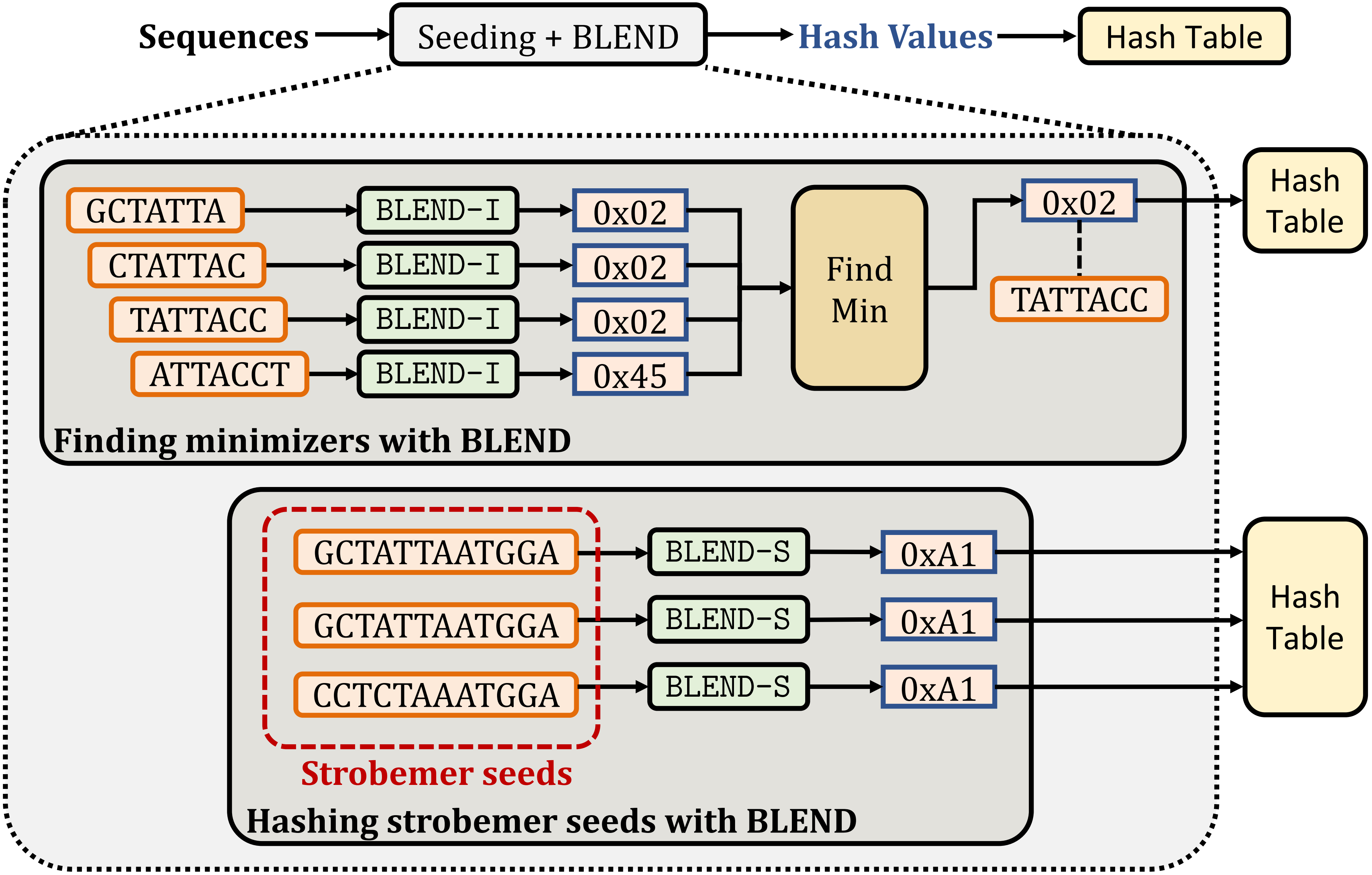
- Count the net difference between 0s and 1s at each position

This is an example sentence to generate a hash value



Words	Hash Values
This	0b01111000
is	0b11011101
an	0b00011100
example	0b01000001
sentence	0b11110000
to	0b01011101
generate	0b10011100
a	0b11000001
hash	[illegible]
SimHash	0b01001011
value	0b00101011

Integrating BLEND for Seeding



Sequences → Seeding + BLEND → Hash Values → Hash Table

GCTATTA → BLEND-I → 0x02

CTATTAC → BLEND-I → 0x02

TATTACC → BLEND-I → 0x02

ATTACCT → BLEND-I → 0x45

Find Min → 0x02 → Hash Table

TATTACC

Finding minimizers with BLEND

GCTATTAATGGA → BLEND-S → 0xA1

GCTATTAATGGA → BLEND-S → 0xA1

CCTCTAAATGGA → BLEND-S → 0xA1

→ Hash Table

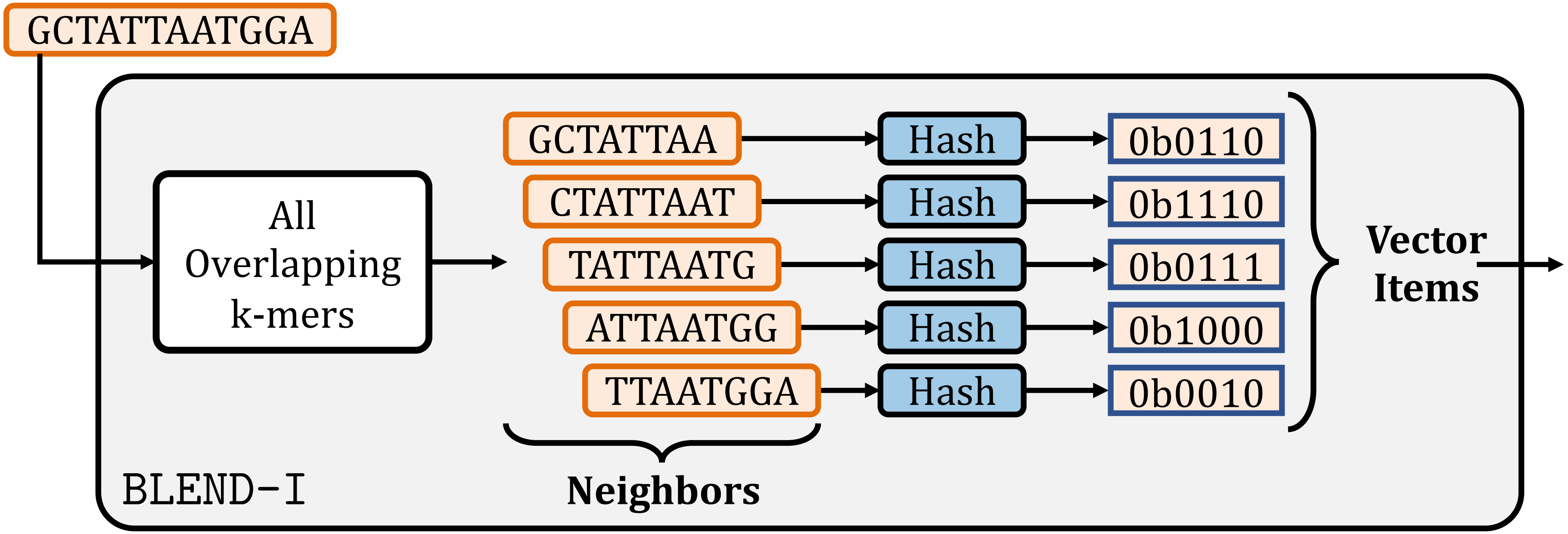
Strobemer seeds

Hashing strobemer seeds with BLEND

Sequence-to-Vector Conversion (BLEND-I)

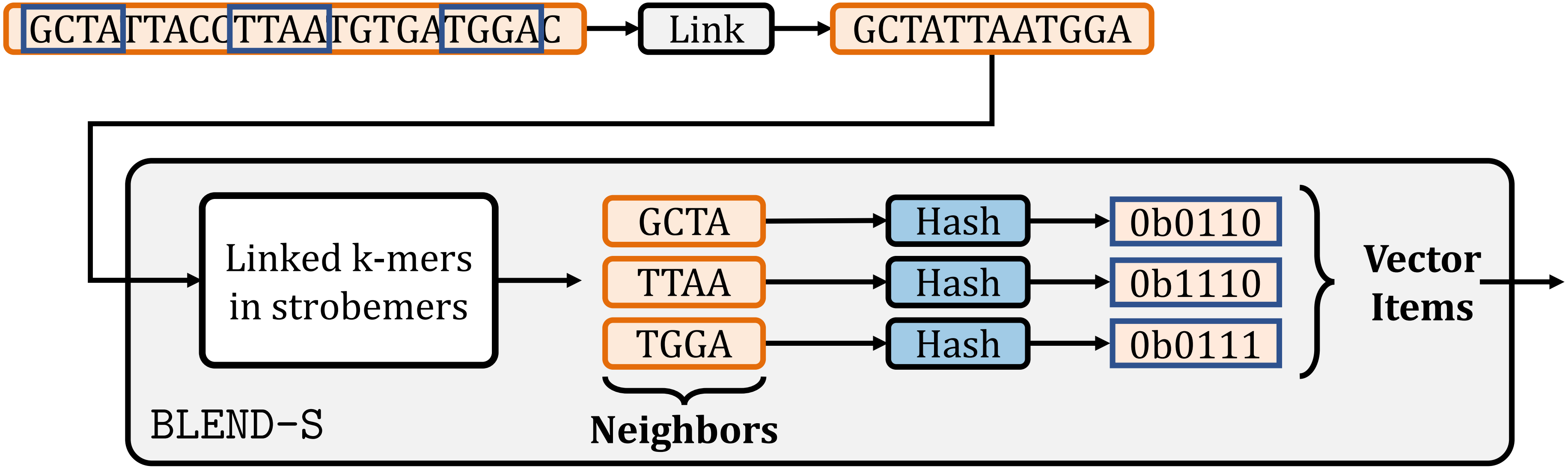
- **Goal:** Convert seed sequences into vector of items
  - **Input:** A fixed-length sequence (seed sequence)

1. Extract **all overlapping k-mers** of the seed **(neighbors)**
2. Generate the **hash values of neighbors** using any hash function
3. **Vector items:** Hash values of neighbors

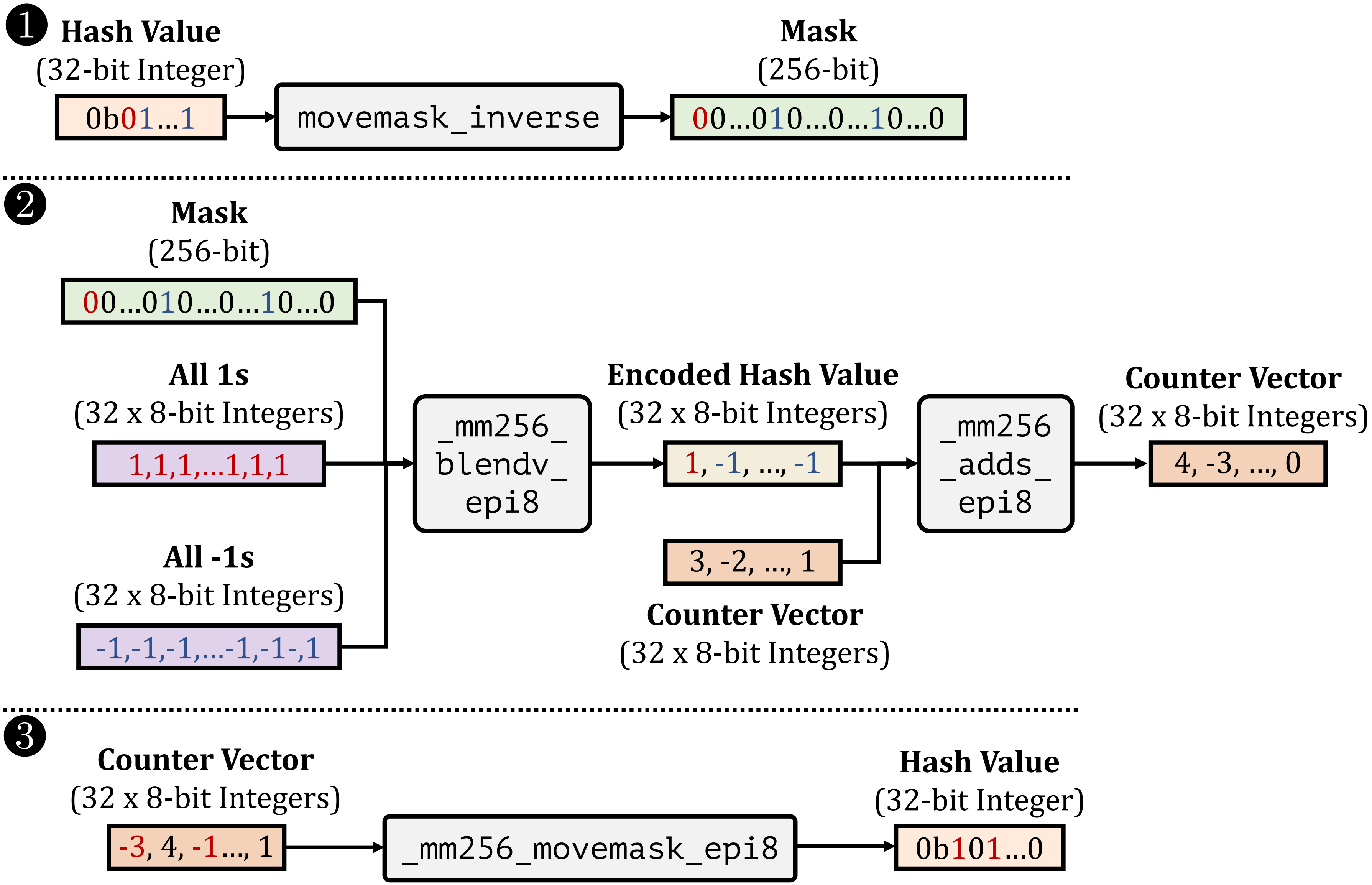


Sequence-to-Vector Conversion (BLEND-S)

- **Goal:** Convert seed sequences into vector of items
  - **Input:** A fixed-length sequence (seed sequence)



Sequence-to-Vector Conversion (SIMD)



❶ **Hash Value**
(32-bit Integer)
0b01…1
movemask_inverse
Mask
(256-bit)
00…010…0…10…0

❷ **Mask**
(256-bit)
00…010…0…10…0

All 1s
(32 x 8-bit Integers)
1,1,1,…1,1,1

All -1s
(32 x 8-bit Integers)
-1,-1,-1,…-1,-1-,1

_mm256_blendv_epi8

Encoded Hash Value
(32 x 8-bit Integers)
1, -1, …, -1

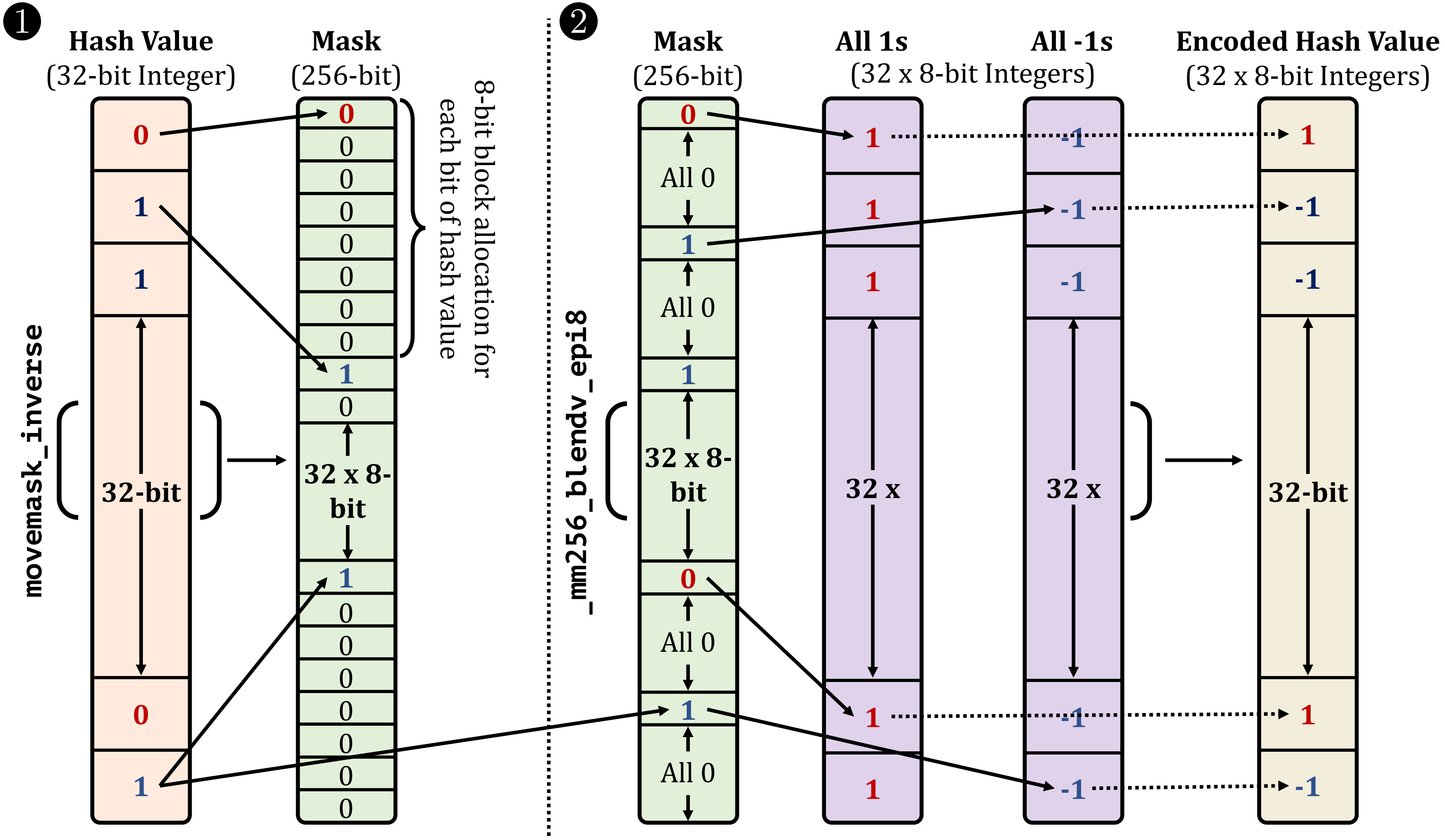
3, -2, …, 1
Counter Vector
(32 x 8-bit Integers)

_mm256_adds_epi8

Counter Vector
(32 x 8-bit Integers)
4, -3, …, 0

❸ **Counter Vector**
(32 x 8-bit Integers)
-3, 4, -1…, 1
_mm256_movemask_epi8
Hash Value
(32-bit Integer)
0b101…0

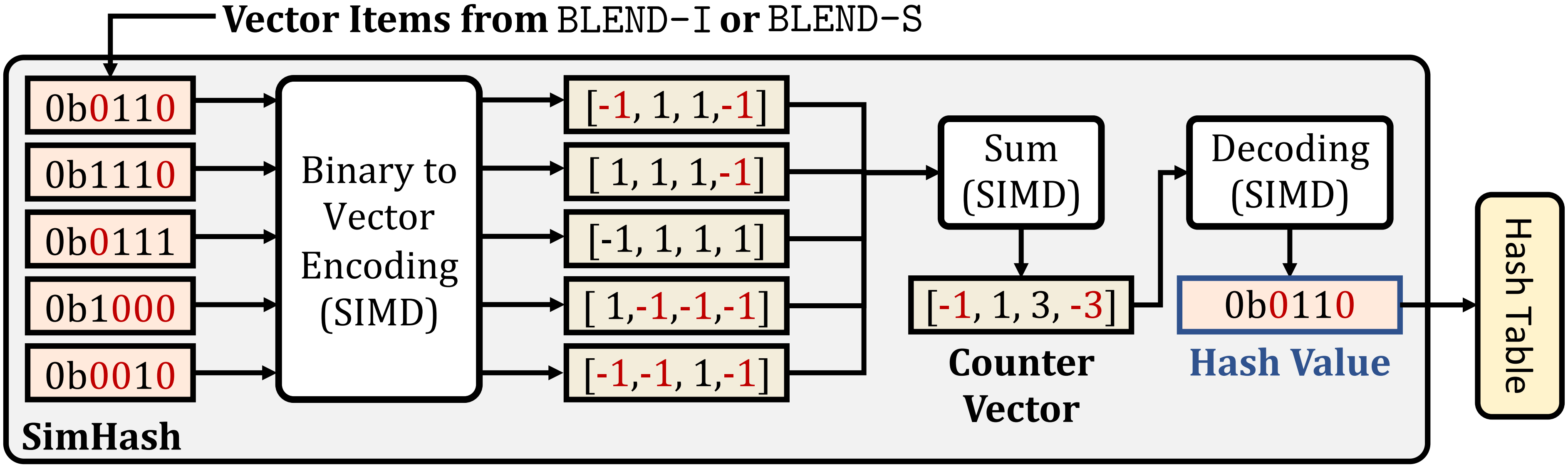
Sequence-to-Vector Conversion (SIMD)



Generating the SimHash values

- **Goal:** Generate the SimHash value of a seed
  - **Input:** Vector items from BLEND-I or BLEND-S

1. Encode hash values using vectors of **-1s** and **+1s**
2. Bitwise sum in SimHash: **Vector summation**
3. Decode the counter vector into a **SimHash value for the seed**



Datasets

Organism	Library	Reads (#)	Seq. Depth	SRA Accession	Reference Genome
Human CHM13	PacBio HiFi	3,167,477	16	SRR11292122-3	T2T-CHM13 (v1.1)
	ONT*	10,380,693	30	Simulated R9.5	T2T-CHM13 (v2.0)
Human HG002	PacBio HiFi	11,714,594	52	SRR10382244-9	GRCh37
D. ananassae	PacBio HiFi	1,195,370	50	SRR11442117	[1036]
Yeast	PacBio CLR*	270,849	200	Simulated P6-C4	GCA_000146045.2
	ONT*	135,296	100	Simulated R9.5	GCA_000146045.2
	Illumina MiSeq	3,318,467	80	ERR1938683	GCA_000146045.2
E. coli	PacBio HiFi	38,703	100	SRR11434954	[1036]
	PacBio CLR	76,279	112	SRR1509640	GCA_000732965.1

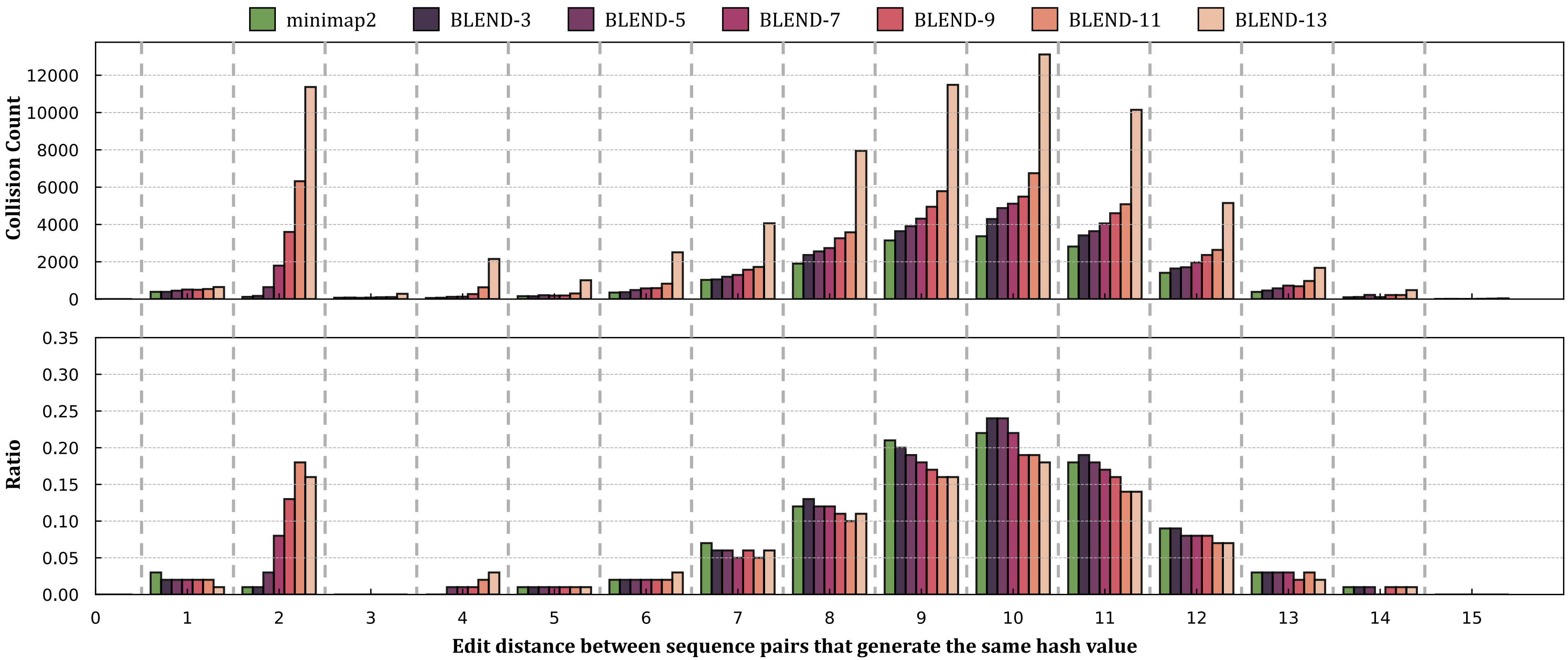
Hash Collisions

Tool	Number of Minimizers	Number of Collisions	Collision/Minimizer Ratio	Avg. Edit Distance Between Minimizers With Collision
minimap2	903,043	15,306	0.016949	9.327061
BLEND-3	1,014,173	18,224	0.017969	9.393437
BLEND-5	1,090,468	20,659	0.018945	9.213660
BLEND-7	1,140,254	23,591	0.020689	8.874698
BLEND-9	1,173,198	28,411	0.024217	8.495301
BLEND-11	1,186,687	35,500	0.029915	8.067549
BLEND-13	1,197,966	72,078	0.060167	8.075918

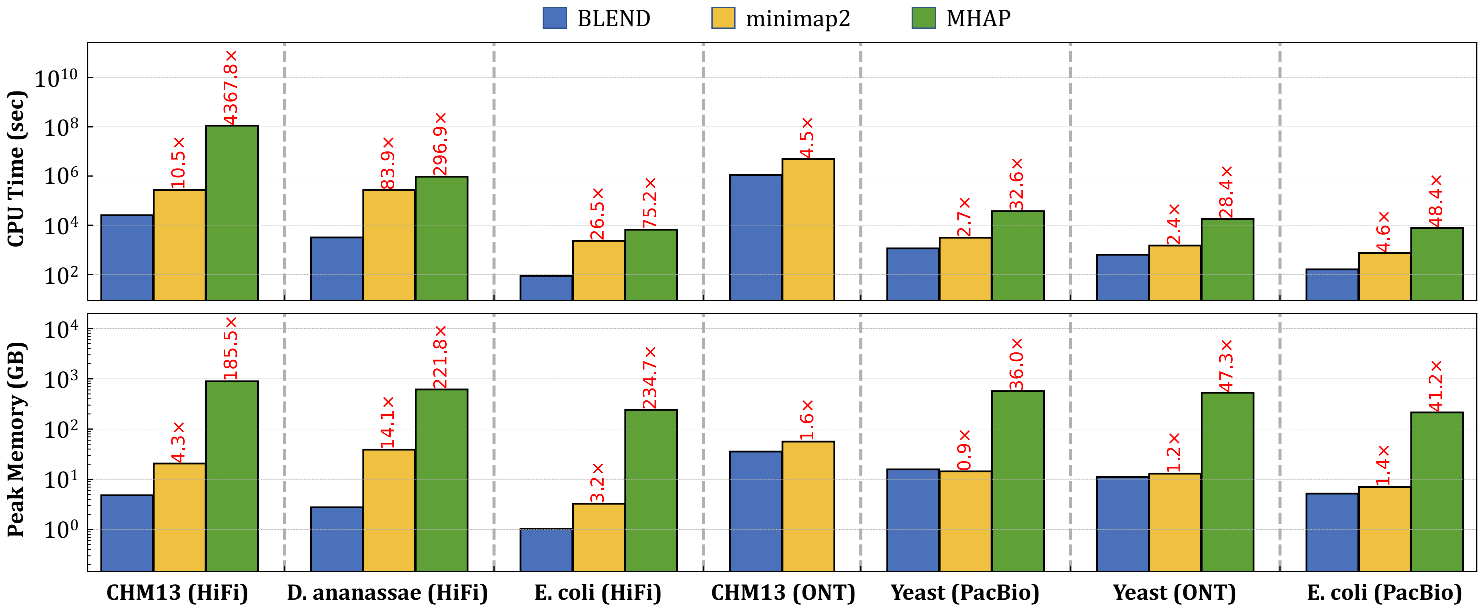
Hash Collisions between Similar Seeds

Tool	Number of Sequences	Number of Sequences with Collision	Collision/Sequence Ratio	Avg. Edit Distance Between K-mers With Collision
minimap2	4,130	0	0	N/A
BLEND-3	4,130	0	0	N/A
BLEND-5	4,130	11	0.00263663	1.45455
BLEND-7	4,130	50	0.0119847	1.5
BLEND-9	4,130	77	0.0184564	2.01299
BLEND-11	4,130	273	0.0654362	2.80952
BLEND-13	4,130	329	0.0788591	2.20669

Empirical Analysis on Fuzzy Seed Matching



Read Overlapping – Performance & Memory



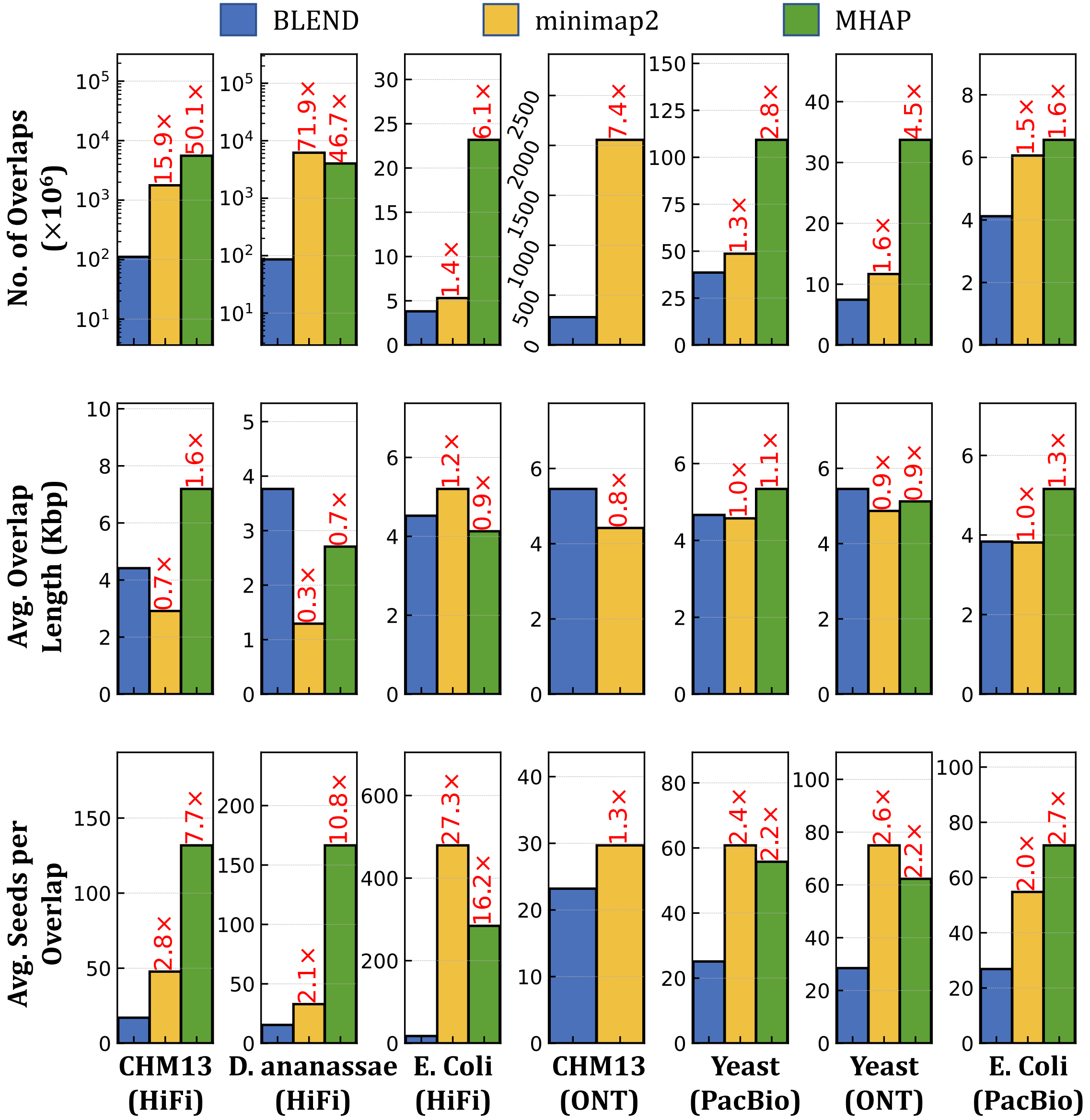
For HiFi: Average **speedup** of **40.3x (minimap2)**

Reducing the **memory** footprint **by 7.2x**

Improving critical parameters without hurting the accuracy:

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

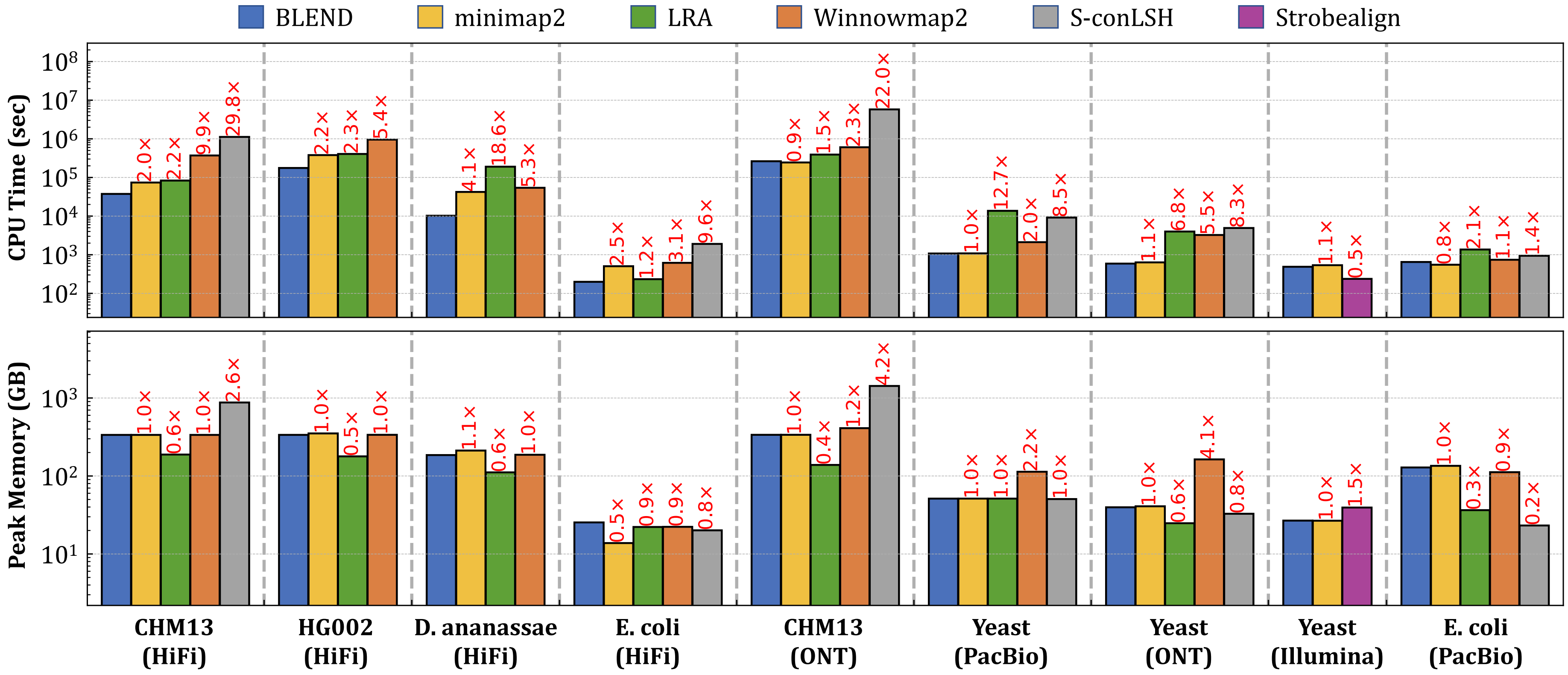
Overlapping Statistics



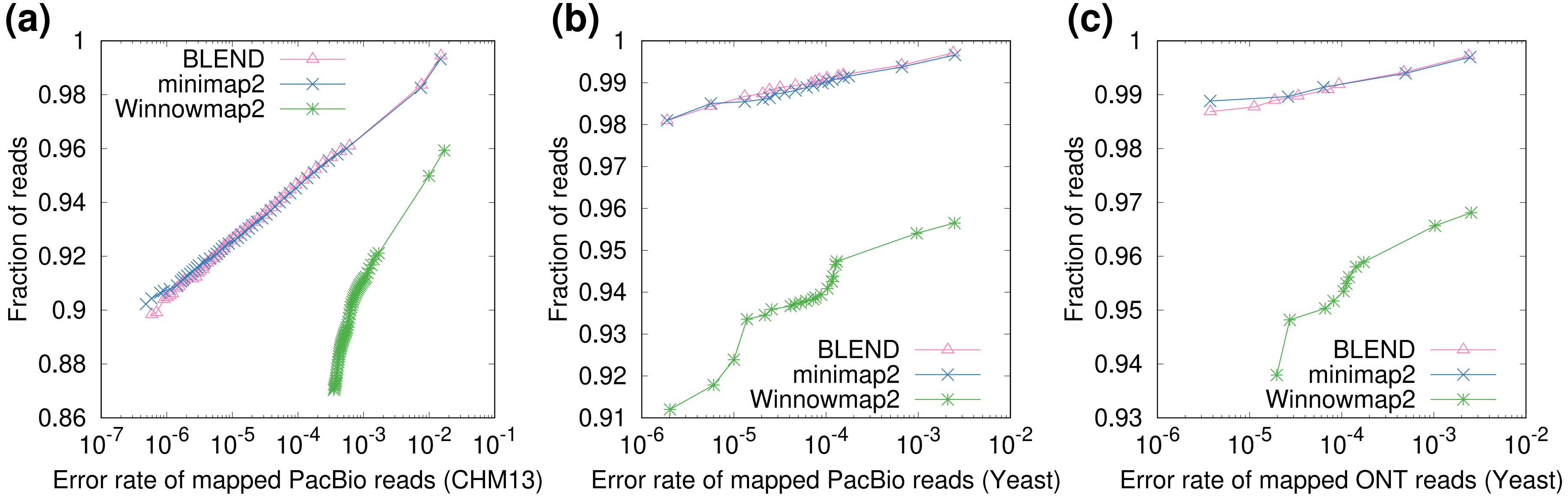
Assembly Results

Dataset	Tool	Average Identity (%)	Genome Fraction (%)	K-mer Compl. (%)	Aligned Length (Mbp)	Mismatch per 100Kbp (#)	Average GC (%)	Assembly Length (Mbp)	Largest Contig (Mbp)	NGA50 (Kbp)	NG50 (Kbp)
CHM13 (HiFi)	BLEND	**99.8526**	**98.4847**	**90.15**	3,092.54	**22.02**	**40.78**	**3,095.21**	22.8397	5,442.25	5,442.31
	minimap2	99.7421	97.1493	83.05	**3,094.79**	55.96	40.71	3,100.97	**47.1387**	**7,133.43**	**7,134.31**
	MHAP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Reference	100	100	100	3,054.83	0.00	40.85	3,054.83	248.387	154,260	154,260
D. ananassae (HiFi)	BLEND	**99.7856**	**97.2308**	**86.43**	240.391	**143.13**	**41.75**	**247.153**	**6.23256**	**792.407**	**798.913**
	minimap2	99.7044	96.3190	72.33	**289.453**	191.53	41.68	298.28	4.43396	273.398	278.775
	MHAP	99.5551	0.7276	0.21	2.29	239.76	42.07	2.34951	0.028586	N/A	N/A
	Reference	100	100	100	213.805	0.00	41.81	213.818	30.6728	26,427.4	26,427.4
E. coli (HiFi)	BLEND	**99.8320**	**99.8801**	**87.91**	**5.12155**	**3.77**	**50.53**	5.12155	**3.41699**	**3,416.99**	**3,416.99**
	minimap2	99.7064	99.8748	79.27	5.09249	19.71	50.47	**5.09436**	3.08849	3,087.05	3,087.05
	MHAP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Reference	100	100	100	5.04628	0.00	50.52	5.04628	4.94446	4,944.46	4,944.46
CHM13 (ONT)	BLEND	N/A	N/A	**29.26**	**2,891.28**	**4,077.53**	**41.32**	2,897.87	25.2071	5,061.52	5,178.59
	minimap2	N/A	N/A	28.32	2,860.26	4,660.73	41.36	**2,908.55**	**66.7564**	**13,189.2**	**13,820.3**
	Reference	100	100	100	3,117.29	0.00	40.75	3,117.29	248.387	150,617	150,617
Yeast (PacBio)	BLEND	89.1677	**97.0854**	**33.81**	**12.3938**	2,672.37	38.84	12.4176	1.54807	635.966	636.669
	minimap2	88.9002	96.9709	33.38	12.0128	2,684.38	38.85	**12.3325**	**1.56078**	**810.046**	**828.212**
	MHAP	**89.2182**	88.5928	32.39	10.9039	**2,552.05**	**38.81**	10.9896	1.02375	85.081	436.285
	Reference	100	100	100	12.1571	0.00	38.15	12.1571	1.53193	924.431	924.431
Yeast (ONT)	BLEND	**89.6889**	99.2974	**35.95**	**12.3222**	**2,529.47**	**38.64**	**12.3225**	1.10582	793.046	793.046
	minimap2	88.9393	**99.6878**	34.84	12.304	2,782.59	38.74	12.3725	**1.56005**	**796.718**	**941.588**
	MHAP	89.1970	89.2785	33.58	10.8302	2,647.19	38.84	10.9201	1.44328	118.886	618.908
	Reference	100	100	100	12.1571	0.00	38.15	12.1571	1.53193	924.431	924.431
E. coli (PacBio)	BLEND	**88.5806**	**96.5238**	**32.32**	**5.90024**	**1,857.56**	**49.81**	6.21598	2.40671	**769.981**	2,060.4
	minimap2	88.1365	92.7603	30.74	5.37728	2,005.72	49.66	**6.02707**	**3.77098**	367.442	**3,770.98**
	MHAP	88.4883	90.5533	31.32	5.75159	1,999.48	49.69	6.26216	1.04286	110.535	456.01
	Reference	100	100	100	5.6394	0.00	50.43	5.6394	5.54732	5,547.32	5,547.32

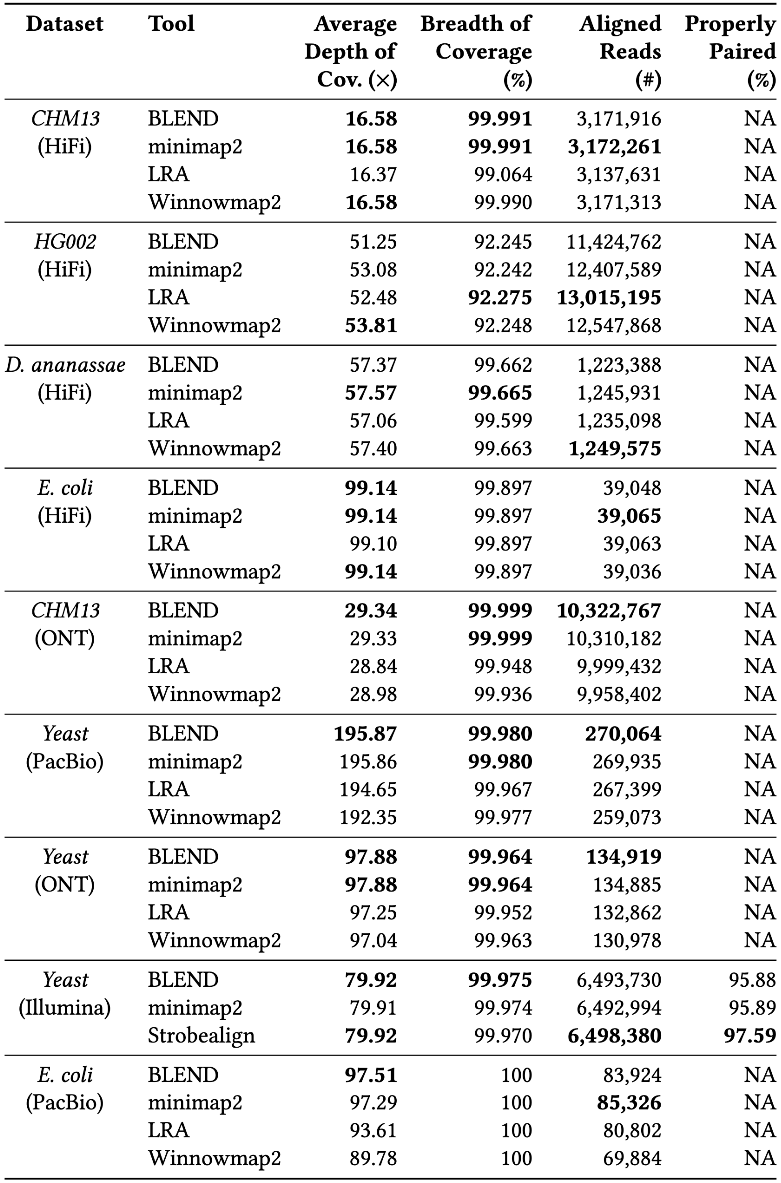
Read Mapping Results



Read Mapping Accuracy – BLEND



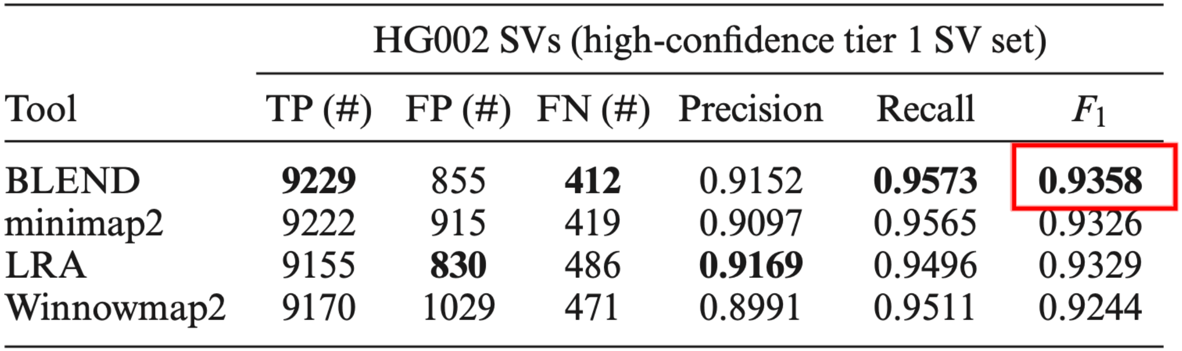
Read Mapping Quality



Dataset	Tool	Average Depth of Cov. (×)	Breadth of Coverage (%)	Aligned Reads (#)	Properly Paired (%)
CHM13 (HiFi)	BLEND	**16.58**	**99.991**	3,171,916	NA
	minimap2	**16.58**	**99.991**	**3,172,261**	NA
	LRA	16.37	99.064	3,137,631	NA
	Winnowmap2	**16.58**	99.990	3,171,313	NA
HG002 (HiFi)	BLEND	51.25	92.245	11,424,762	NA
	minimap2	53.08	92.242	12,407,589	NA
	LRA	52.48	**92.275**	**13,015,195**	NA
	Winnowmap2	**53.81**	92.248	12,547,868	NA
D. ananassae (HiFi)	BLEND	57.37	99.662	1,223,388	NA
	minimap2	**57.57**	**99.665**	1,245,931	NA
	LRA	57.06	99.599	1,235,098	NA
	Winnowmap2	57.40	99.663	**1,249,575**	NA
E. coli (HiFi)	BLEND	**99.14**	99.897	39,048	NA
	minimap2	**99.14**	99.897	**39,065**	NA
	LRA	99.10	99.897	39,063	NA
	Winnowmap2	**99.14**	99.897	39,036	NA
CHM13 (ONT)	BLEND	**29.34**	**99.999**	**10,322,767**	NA
	minimap2	29.33	**99.999**	10,310,182	NA
	LRA	28.84	99.948	9,999,432	NA
	Winnowmap2	28.98	99.936	9,958,402	NA
Yeast (PacBio)	BLEND	**195.87**	**99.980**	**270,064**	NA
	minimap2	195.86	**99.980**	269,935	NA
	LRA	194.65	99.967	267,399	NA
	Winnowmap2	192.35	99.977	259,073	NA
Yeast (ONT)	BLEND	**97.88**	**99.964**	**134,919**	NA
	minimap2	**97.88**	**99.964**	134,885	NA
	LRA	97.25	99.952	132,862	NA
	Winnowmap2	97.04	99.963	130,978	NA
Yeast (Illumina)	BLEND	**79.92**	**99.975**	6,493,730	95.88
	minimap2	79.91	99.974	6,492,994	95.89
	Strobealign	**79.92**	99.970	**6,498,380**	**97.59**
E. coli (PacBio)	BLEND	**97.51**	100	83,924	NA
	minimap2	97.29	100	**85,326**	NA
	LRA	93.61	100	80,802	NA
	Winnowmap2	89.78	100	69,884	NA

Read Mapping – SV Calling

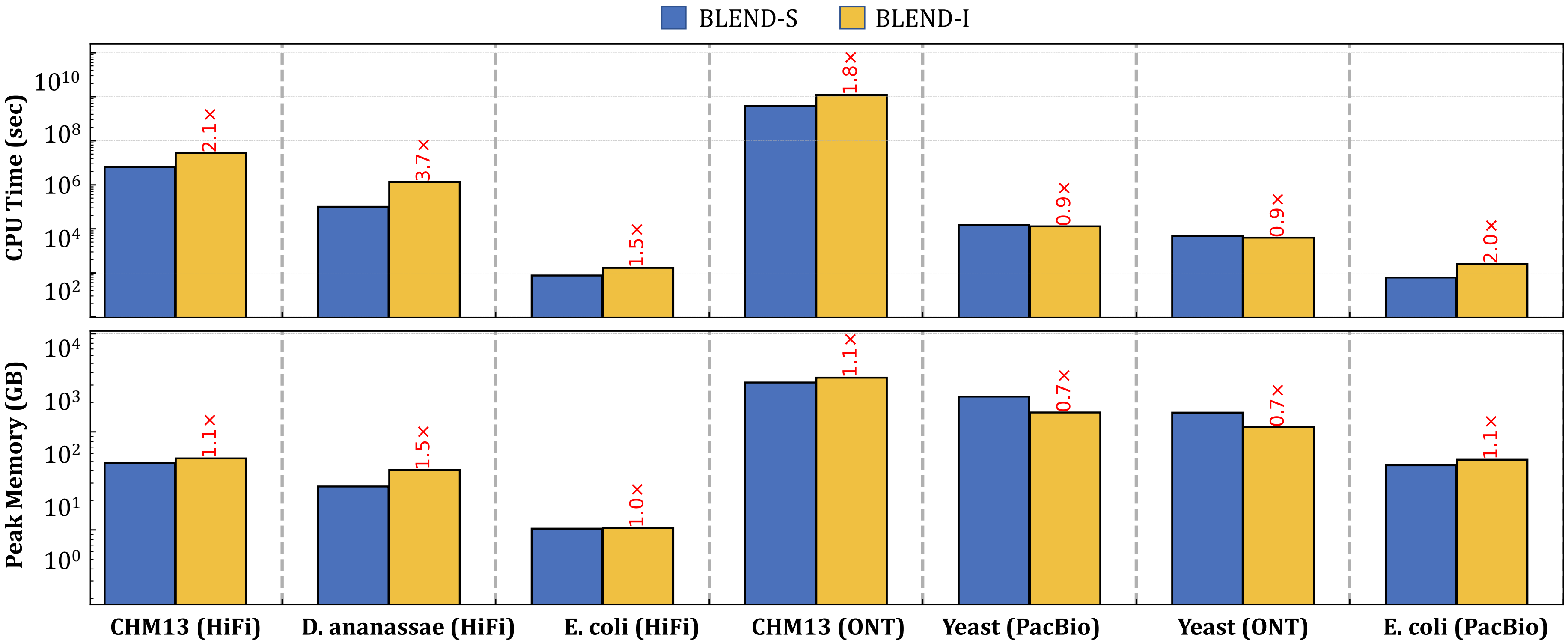
- Structural variant (SV) calling using read mappings from each tool
  - Sniffles2 to call SVs from HG002 long read mappings
  - Truvari to compare the resulting SVs with the benchmarking SV set (Tier 1 set from GIAB)



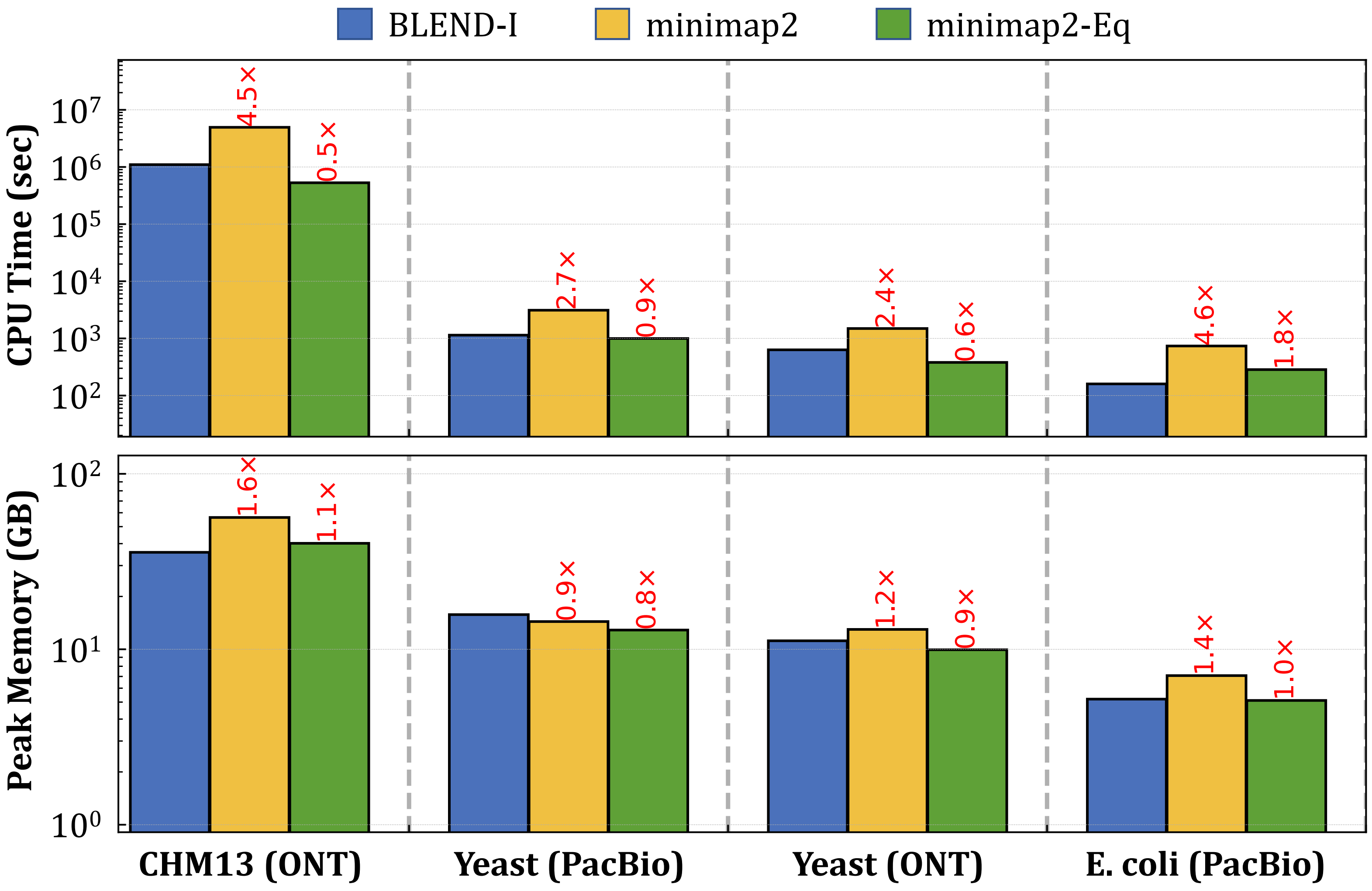
	HG002 SVs (high-confidence tier 1 SV set)					
Tool	TP (#)	FP (#)	FN (#)	Precision	Recall	F_1
BLEND	**9229**	855	**412**	0.9152	**0.9573**	**0.9358**
minimap2	9222	915	419	0.9097	0.9565	0.9326
LRA	9155	**830**	486	**0.9169**	0.9496	0.9329
Winnowmap2	9170	1029	471	0.8991	0.9511	0.9244

Best overall accuracy in downstream analysis

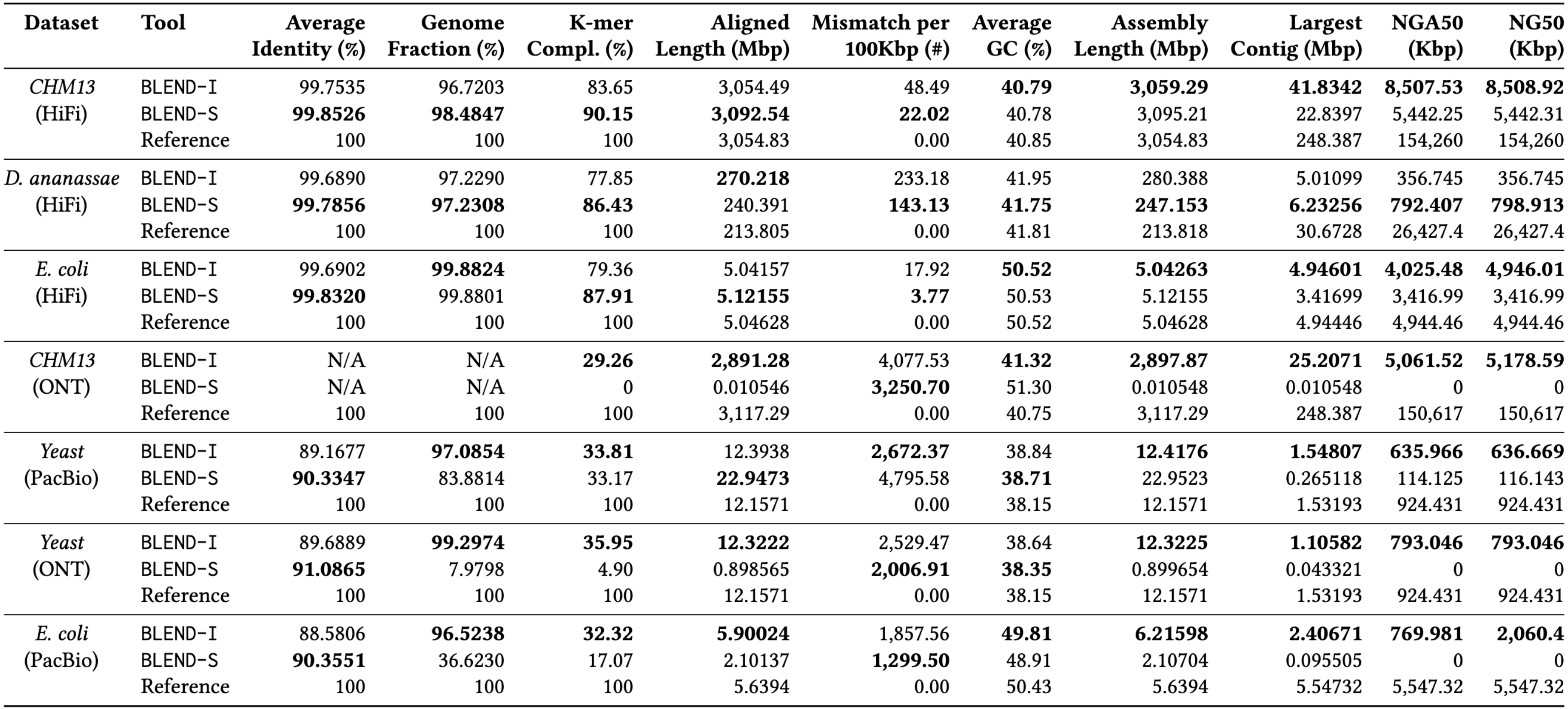
Overlapping Perf. – BLEND-I vs BLEND-S



Overlapping Perf. – BLEND-I vs. minimap2



Assembly Stats. – BLEND-I vs. BLEND-S

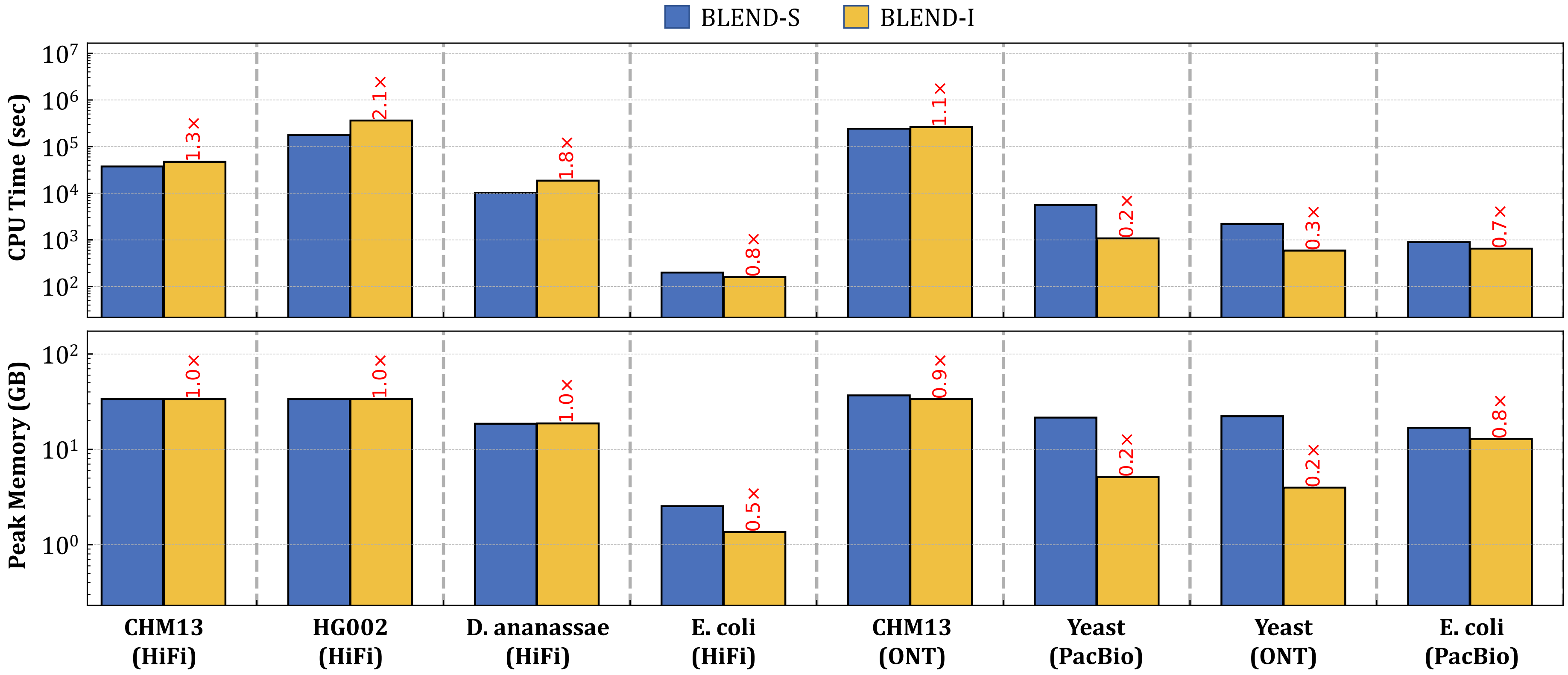


Dataset	Tool	Average Identity (%)	Genome Fraction (%)	K-mer Compl. (%)	Aligned Length (Mbp)	Mismatch per 100Kbp (#)	Average GC (%)	Assembly Length (Mbp)	Largest Contig (Mbp)	NGA50 (Kbp)	NG50 (Kbp)
CHM13 (HiFi)	BLEND-I	99.7535	96.7203	83.65	3,054.49	48.49	**40.79**	**3,059.29**	**41.8342**	**8,507.53**	**8,508.92**
	BLEND-S	**99.8526**	**98.4847**	**90.15**	**3,092.54**	**22.02**	40.78	3,095.21	22.8397	5,442.25	5,442.31
	Reference	100	100	100	3,054.83	0.00	40.85	3,054.83	248.387	154,260	154,260
D. ananassae (HiFi)	BLEND-I	99.6890	97.2290	77.85	**270.218**	233.18	41.95	280.388	5.01099	356.745	356.745
	BLEND-S	**99.7856**	**97.2308**	**86.43**	240.391	**143.13**	**41.75**	**247.153**	**6.23256**	**792.407**	**798.913**
	Reference	100	100	100	213.805	0.00	41.81	213.818	30.6728	26,427.4	26,427.4
E. coli (HiFi)	BLEND-I	99.6902	**99.8824**	79.36	5.04157	17.92	**50.52**	**5.04263**	**4.94601**	**4,025.48**	**4,946.01**
	BLEND-S	**99.8320**	99.8801	**87.91**	**5.12155**	**3.77**	50.53	5.12155	3.41699	3,416.99	3,416.99
	Reference	100	100	100	5.04628	0.00	50.52	5.04628	4.94446	4,944.46	4,944.46
CHM13 (ONT)	BLEND-I	N/A	N/A	**29.26**	**2,891.28**	4,077.53	**41.32**	**2,897.87**	**25.2071**	**5,061.52**	**5,178.59**
	BLEND-S	N/A	N/A	0	0.010546	**3,250.70**	51.30	0.010548	0.010548	0	0
	Reference	100	100	100	3,117.29	0.00	40.75	3,117.29	248.387	150,617	150,617
Yeast (PacBio)	BLEND-I	89.1677	**97.0854**	**33.81**	12.3938	**2,672.37**	38.84	**12.4176**	**1.54807**	**635.966**	**636.669**
	BLEND-S	**90.3347**	83.8814	33.17	**22.9473**	4,795.58	**38.71**	22.9523	0.265118	114.125	116.143
	Reference	100	100	100	12.1571	0.00	38.15	12.1571	1.53193	924.431	924.431
Yeast (ONT)	BLEND-I	89.6889	**99.2974**	**35.95**	**12.3222**	2,529.47	38.64	**12.3225**	**1.10582**	**793.046**	**793.046**
	BLEND-S	**91.0865**	7.9798	4.90	0.898565	**2,006.91**	**38.35**	0.899654	0.043321	0	0
	Reference	100	100	100	12.1571	0.00	38.15	12.1571	1.53193	924.431	924.431
E. coli (PacBio)	BLEND-I	88.5806	**96.5238**	**32.32**	**5.90024**	1,857.56	**49.81**	**6.21598**	**2.40671**	**769.981**	**2,060.4**
	BLEND-S	**90.3551**	36.6230	17.07	2.10137	**1,299.50**	48.91	2.10704	0.095505	0	0
	Reference	100	100	100	5.6394	0.00	50.43	5.6394	5.54732	5,547.32	5,547.32

Assembly Stats. – BLEND-I vs. minimap2

Dataset	Tool	Average Identity (%)	Genome Fraction (%)	K-mer Compl. (%)	Aligned Length (Mbp)	Mismatch per 100Kbp (#)	Average GC (%)	Assembly Length (Mbp)	Largest Contig (Mbp)	NGA50 (Kbp)	NG50 (Kbp)
CHM13 (ONT)	BLEND-I	N/A	N/A	29.26	2,891.28	4,077.53	**41.32**	2,897.87	25.2071	5,061.52	5,178.59
	minimap2	N/A	N/A	28.32	2,860.26	4,660.73	41.36	**2,908.55**	**66.7564**	**13,189.2**	**13,820.3**
	minimap2-Eq	N/A	N/A	**29.32**	**3,117.29**	**4,025.22**	**41.32**	2,882.94	24.6651	3,634.05	3,653.47
	Reference	100	100	100	3,117.29	0.00	40.75	3,117.29	248.387	150,617	150,617
Yeast (PacBio)	BLEND-I	89.1677	97.0854	33.81	**12.3938**	2,672.37	38.84	12.4176	1.54807	635.966	636.669
	minimap2	88.9002	96.9709	33.38	12.0128	2,684.38	38.85	**12.3325**	**1.56078**	**810.046**	**828.212**
	minimap2-Eq	**89.2166**	**97.2674**	**33.93**	12.3886	**2,653.08**	**38.82**	12.4241	1.53435	643.136	781.136
	Reference	100	100	100	12.1571	0.00	38.15	12.1571	1.53193	924.431	924.431
Yeast (ONT)	BLEND-I	**89.6889**	99.2974	**35.95**	**12.3222**	2,529.47	**38.64**	**12.3225**	1.10582	793.046	793.046
	minimap2	88.9393	**99.6878**	34.84	12.304	2,782.59	38.74	12.3725	**1.56005**	**796.718**	**941.588**
	minimap2-Eq	89.6653	97.3273	35.62	11.826	**2,465.87**	**38.64**	11.8282	1.07367	605.201	677.415
	Reference	100	100	100	12.1571	0.00	38.15	12.1571	1.53193	924.431	924.431
E. coli (PacBio)	BLEND-I	88.5806	96.5238	32.32	**5.90024**	1,857.56	**49.81**	6.21598	2.40671	769.981	2,060.4
	minimap2	88.1365	92.7603	30.74	5.37728	2,005.72	49.66	**6.02707**	3.77098	367.442	3,770.98
	minimap2-Eq	**88.6371**	**96.8540**	**32.33**	5.82218	**1,816.29**	49.76	6.05821	**3.77318**	**1,119.04**	**3,773.18**
	Reference	100	100	100	5.6394	0.00	50.43	5.6394	5.54732	5,547.32	5,547.32

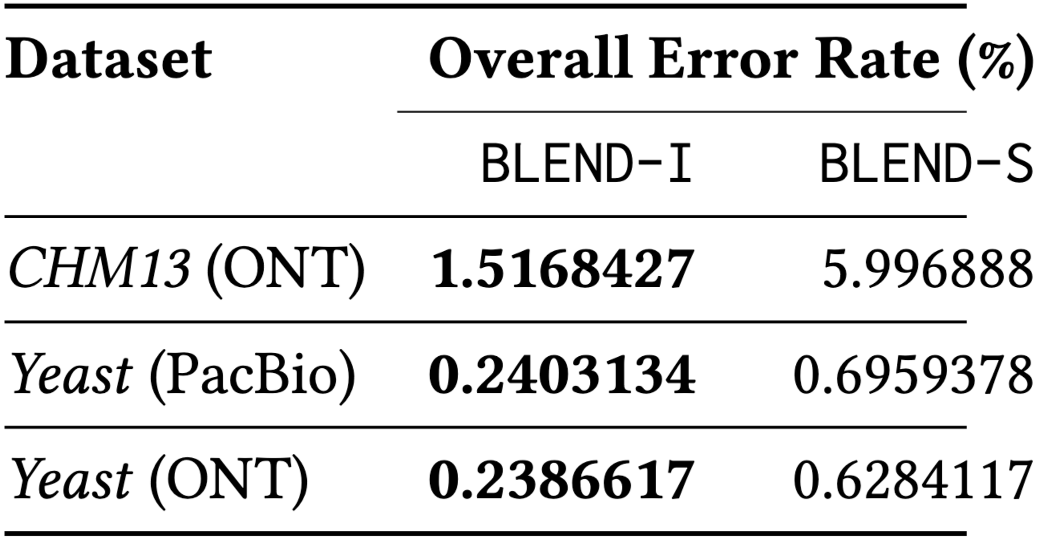
Mapping Perf. – BLEND-I vs. BLEND-S



Mapping Quality – BLEND-I vs. BLEND-S

Dataset	Tool	Average Depth of Cov. (×)	Breadth of Coverage (%)	Aligned Reads (#)	Properly Paired (%)
CHM13	BLEND-I	16.58	99.991	**3,172,305**	NA
(HiFi)	BLEND-S	16.58	99.991	3,171,916	NA
HG002	BLEND-I	**51.25**	**92.245**	6,813,886	NA
(HiFi)	BLEND-S	11.24	13.860	**11,424,762**	NA
D. ananassae	BLEND-I	**57.51**	99.650	**1,249,666**	NA
(HiFi)	BLEND-S	57.37	**99.662**	1,223,388	NA
E. coli	BLEND-I	99.14	99.897	**39,064**	NA
(HiFi)	BLEND-S	99.14	99.897	39,048	NA
CHM13	BLEND-I	**29.34**	**99.999**	**10,322,767**	NA
(ONT)	BLEND-S	17.51	99.700	5,760,401	NA
Yeast	BLEND-I	**195.87**	**99.980**	**270,064**	NA
(PacBio)	BLEND-S	142.31	99.975	179,039	NA
Yeast	BLEND-I	**97.88**	**99.964**	**134,919**	NA
(ONT)	BLEND-S	59.57	99.906	75,110	NA
E. coli	BLEND-I	**97.51**	100	**83,924**	NA
(PacBio)	BLEND-S	56.87	100	40,694	NA

Mapping Acc. – BLEND-I vs. BLEND-S



Dataset	Overall Error Rate (%)	
	BLEND-I	BLEND-S
CHM13 (ONT)	**1.5168427**	5.996888
Yeast (PacBio)	**0.2403134**	0.6959378
Yeast (ONT)	**0.2386617**	0.6284117

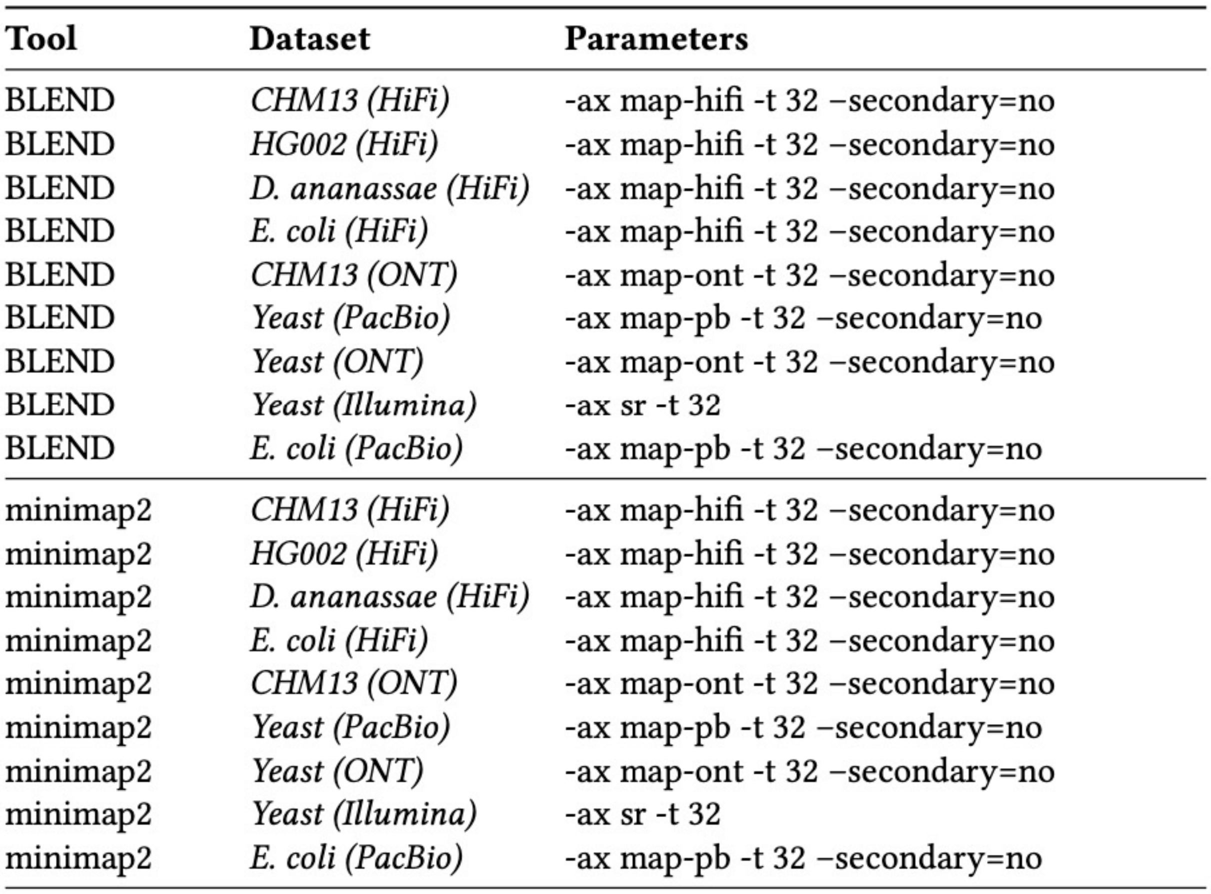
BLEND Parameter Definitions

Parameter	Definition
–strobemers	Use the BLEND-S mechanism when generating the list of k-mers of a seed
–immediate	Use the BLEND-I mechanism when generating the list of k-mers of a seed
-H	Use homopolymer-compressed k-mers
-w INT	Window size used when finding minimizers.
-k INT	k-mer size used when generating the list of k-mers of a seed
–neighbors INT	Number of k-mers included in the list of seeds. Combination of both -k (k) and –neighbors (n) determines the seed length. Seed length in BLEND-S is calculated as: $k \times n$ Seed length in BLEND-I is calculated as: $k + (n - 1)$
–fixed-bits INT	Bit length of hash values that BLEND generates for each seed. Setting it to $2 \times k$ is the default behavior.
-t INT	Number of CPU threads to use.
-x STR	Preset for setting the default parameters given the use case (STR)
-x map-ont	Preset for mapping ONT reads. It uses the following parameters: –immediate -w 10 -k 9 –neighbors 7 –fixed-bits 30
-x map-pb	Preset for mapping erroneous PacBio reads. It uses the following parameters: –immediate -H -w 10 -k 13 –neighbors 7 –fixed-bits 32
-x map-hifi	Preset for mapping accurate long (HiFi) reads. It uses the following parameters: –strobemers -w 50 -k 19 –neighbors 5 –fixed-bits 38
-x sr	Preset for mapping short reads. It uses the following parameters: –immediate -w 11 -k 21 –neighbors 5 –fixed-bits 32
-x ava-ont	Preset for overlapping ONT reads. It uses the following parameters: –immediate -w 10 -k 15 –neighbors 5 –fixed-bits 30
-x ava-pb	Preset for overlapping erroneous PacBio reads. It uses the following parameters: –immediate -H -w 10 -k 19 –neighbors 5 –fixed-bits 38
-x ava-hifi	Preset for overlapping accurate long (HiFi) reads. It uses the following parameters: –strobemers -w 200 -k 25 –neighbors 7 –fixed-bits 50

Parameter Settings – Overlapping

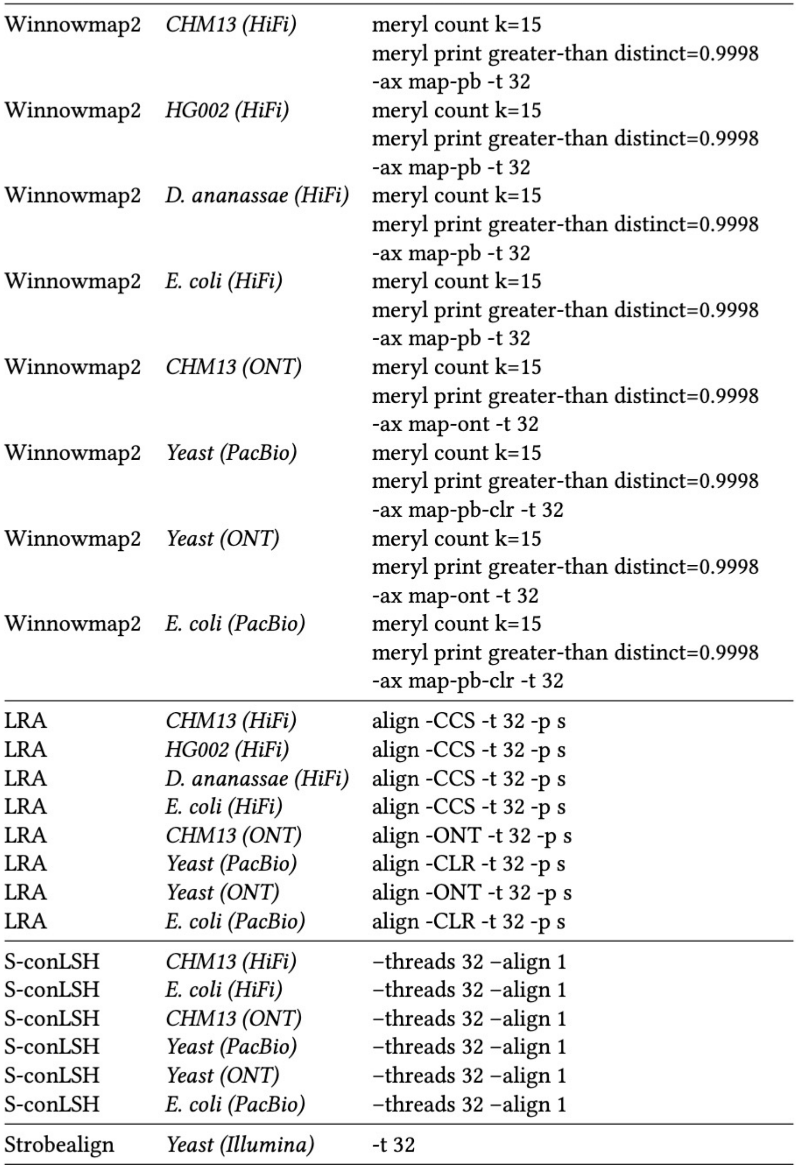
Tool	Dataset	Parameters
BLEND	*CHM13 (HiFi)*	-x ava-hifi -t 32
BLEND	*D. ananassae (HiFi)*	-x ava-hifi -t 32
BLEND	*E. coli (HiFi)*	-x ava-hifi -t 32
BLEND	*CHM13 (ONT)*	-x ava-ont -t 32
BLEND	*Yeast (PacBio)*	-x ava-pb -t 32
BLEND	*Yeast (ONT)*	-x ava-ont -t 32
BLEND	*E. coli (PacBio)*	-x ava-pb -t 32
minimap2	*CHM13 (HiFi)*	-x ava-pb -Hk21 -w14 -t 32
minimap2	*D. ananassae (HiFi)*	-x ava-pb -Hk21 -w14 -t 32
minimap2	*E. coli (HiFi)*	-x ava-pb -Hk21 -w14 -t 32
minimap2	*CHM13 (ONT)*	-x ava-ont -t 32
minimap2	*Yeast (PacBio)*	-x ava-pb -t 32
minimap2	*Yeast (ONT)*	-x ava-ont -t 32
minimap2	*E. coli (PacBio)*	-x ava-pb -t 32
minimap2-Eq	*CHM13 (ONT)*	-x ava-ont -k19 -w10 -t 32
minimap2-Eq	*Yeast (PacBio)*	-x ava-pb -k23 -w10 -t 32
minimap2-Eq	*Yeast (ONT)*	-x ava-ont -k19 -w10 -t 32
minimap2-Eq	*E. coli (PacBio)*	-x ava-pb -k23 -w10 -t 32
MHAP	*CHM13 (HiFi)*	–store-full-id –ordered-kmer-size 18 –num-hashes 128 –num-min-matches 5 –ordered-sketch-size 1000 –threshold 0.95 –num-threads 32
MHAP	*D. ananassae (HiFi)*	–store-full-id –ordered-kmer-size 18 –num-hashes 128 –num-min-matches 5 –ordered-sketch-size 1000 –threshold 0.95 –num-threads 32
MHAP	*E. coli (HiFi)*	–store-full-id –ordered-kmer-size 18 –num-hashes 128 –num-min-matches 5 –ordered-sketch-size 1000 –threshold 0.95 –num-threads 32
MHAP	*Yeast (PacBio)*	–store-full-id –num-threads 32
MHAP	*Yeast (ONT)*	–store-full-id –num-threads 32
MHAP	*E. coli (PacBio)*	–store-full-id –num-threads 32

Parameter Settings – Read Mapping #1



Tool	Dataset	Parameters
BLEND	*CHM13 (HiFi)*	-ax map-hifi -t 32 –secondary=no
BLEND	*HG002 (HiFi)*	-ax map-hifi -t 32 –secondary=no
BLEND	*D. ananassae (HiFi)*	-ax map-hifi -t 32 –secondary=no
BLEND	*E. coli (HiFi)*	-ax map-hifi -t 32 –secondary=no
BLEND	*CHM13 (ONT)*	-ax map-ont -t 32 –secondary=no
BLEND	*Yeast (PacBio)*	-ax map-pb -t 32 –secondary=no
BLEND	*Yeast (ONT)*	-ax map-ont -t 32 –secondary=no
BLEND	*Yeast (Illumina)*	-ax sr -t 32
BLEND	*E. coli (PacBio)*	-ax map-pb -t 32 –secondary=no
minimap2	*CHM13 (HiFi)*	-ax map-hifi -t 32 –secondary=no
minimap2	*HG002 (HiFi)*	-ax map-hifi -t 32 –secondary=no
minimap2	*D. ananassae (HiFi)*	-ax map-hifi -t 32 –secondary=no
minimap2	*E. coli (HiFi)*	-ax map-hifi -t 32 –secondary=no
minimap2	*CHM13 (ONT)*	-ax map-ont -t 32 –secondary=no
minimap2	*Yeast (PacBio)*	-ax map-pb -t 32 –secondary=no
minimap2	*Yeast (ONT)*	-ax map-ont -t 32 –secondary=no
minimap2	*Yeast (Illumina)*	-ax sr -t 32
minimap2	*E. coli (PacBio)*	-ax map-pb -t 32 –secondary=no

Parameter Settings – Read Mapping #1



Winnowmap2	*CHM13 (HiFi)*	meryl count k=15 meryl print greater-than distinct=0.9998 -ax map-pb -t 32
Winnowmap2	*HG002 (HiFi)*	meryl count k=15 meryl print greater-than distinct=0.9998 -ax map-pb -t 32
Winnowmap2	*D. ananassae (HiFi)*	meryl count k=15 meryl print greater-than distinct=0.9998 -ax map-pb -t 32
Winnowmap2	*E. coli (HiFi)*	meryl count k=15 meryl print greater-than distinct=0.9998 -ax map-pb -t 32
Winnowmap2	*CHM13 (ONT)*	meryl count k=15 meryl print greater-than distinct=0.9998 -ax map-ont -t 32
Winnowmap2	*Yeast (PacBio)*	meryl count k=15 meryl print greater-than distinct=0.9998 -ax map-pb-clr -t 32
Winnowmap2	*Yeast (ONT)*	meryl count k=15 meryl print greater-than distinct=0.9998 -ax map-ont -t 32
Winnowmap2	*E. coli (PacBio)*	meryl count k=15 meryl print greater-than distinct=0.9998 -ax map-pb-clr -t 32
LRA	*CHM13 (HiFi)*	align -CCS -t 32 -p s
LRA	*HG002 (HiFi)*	align -CCS -t 32 -p s
LRA	*D. ananassae (HiFi)*	align -CCS -t 32 -p s
LRA	*E. coli (HiFi)*	align -CCS -t 32 -p s
LRA	*CHM13 (ONT)*	align -ONT -t 32 -p s
LRA	*Yeast (PacBio)*	align -CLR -t 32 -p s
LRA	*Yeast (ONT)*	align -ONT -t 32 -p s
LRA	*E. coli (PacBio)*	align -CLR -t 32 -p s
S-conLSH	*CHM13 (HiFi)*	−threads 32 −align 1
S-conLSH	*E. coli (HiFi)*	−threads 32 −align 1
S-conLSH	*CHM13 (ONT)*	−threads 32 −align 1
S-conLSH	*Yeast (PacBio)*	−threads 32 −align 1
S-conLSH	*Yeast (ONT)*	−threads 32 −align 1
S-conLSH	*E. coli (PacBio)*	−threads 32 −align 1
Strobealign	*Yeast (Illumina)*	-t 32

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

Applications of Read Until

Depletion: Reads mapping to a particular reference genome is ejected

- Microbiome studies by removing host DNA
- Eliminating known residual DNA or RNA (e.g., mitochondrial DNA)
- High abundance genome removal

Enrichment: Reads **not** mapping to a particular reference genome is ejected

- Removing contaminated organisms
- Targeted sequencing (e.g., to a particular region of interest in the genome)
- Low abundance genome enrichment

Applications of Run Until & Sequence Until

Run Until: Stopping the entire sequencing run

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

Sequence Until: Run Until with accuracy-aware decision making

- 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
- …

In Vitro (e.g., PCR) vs. In Silico

- **Polymerase Chain Reaction (PCR)** as a way of in vitro "analysis"
  - Can increase the quantity of DNA in a sample
  - **Non-dynamic** targeted sequencing (e.g., low abundance *known* targets)
  - **Requires additional resources:** Time and money for preparation and execution of PCR

- **Adaptive sampling** as a way of in silico (i.e., computational) analysis
  - **Cannot** increase the existing quantity of DNA in a sample
  - **Dynamic targeted sequencing:** Decisions can be made based on real-time analysis (e.g., Sequence Until)
  - Minimal additional resources
    - ***Almost* no additional resources** for preparation and execution
    - **Simultaneous** enrichment and depletion is possible
    - Better suited for rapid whole genome sequencing
  - *Beauty* of computational analysis (e.g., high flexibility – no need for primers)

- PCR and adaptive sampling can be combined depending on the analysis type

Finding Mapping Positions

- Useful for **any application** that requires exact genomic position
  - Variant calling in downstream analysis
  - Specifically: Identifying rare variants in cancer genomics
  - Methylation profiling

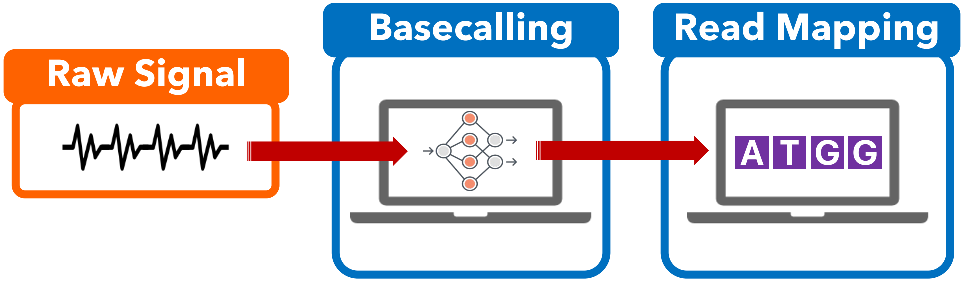
- Accurate and flexible **depth of coverage estimation**
  - **Alternative: DNA quantification** (without computational analysis)
    - DNA quantification is challenging for metagenomics analysis
  - **Computational method:** We can map to almost entire set of known reference genomes to accurately estimate the coverage of a metagenomics sample

- **Transcriptome analysis**
  - Accurately quantifying expression levels & alternative splicing

- **Better resolution** (i.e., more sensitive analysis) for any other application that does not specifically require mapping positions

Analyzing Raw Nanopore Signals

Traditional: Translating (**basecalling**) signals to bases **before** analysis

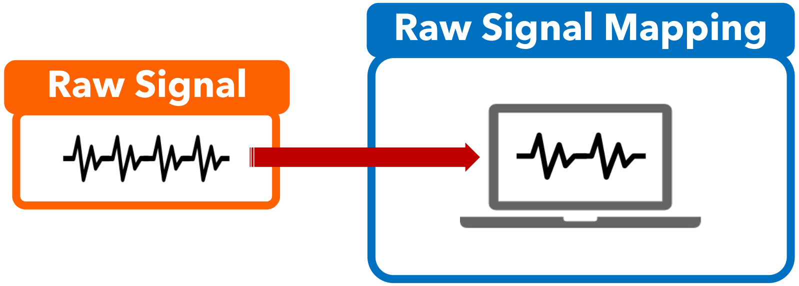


✓ Basecalled sequences are less noisy than raw signals

✓ Many analysis tools use basecalled sequences

✗ Costly and power-hungry computational requirements

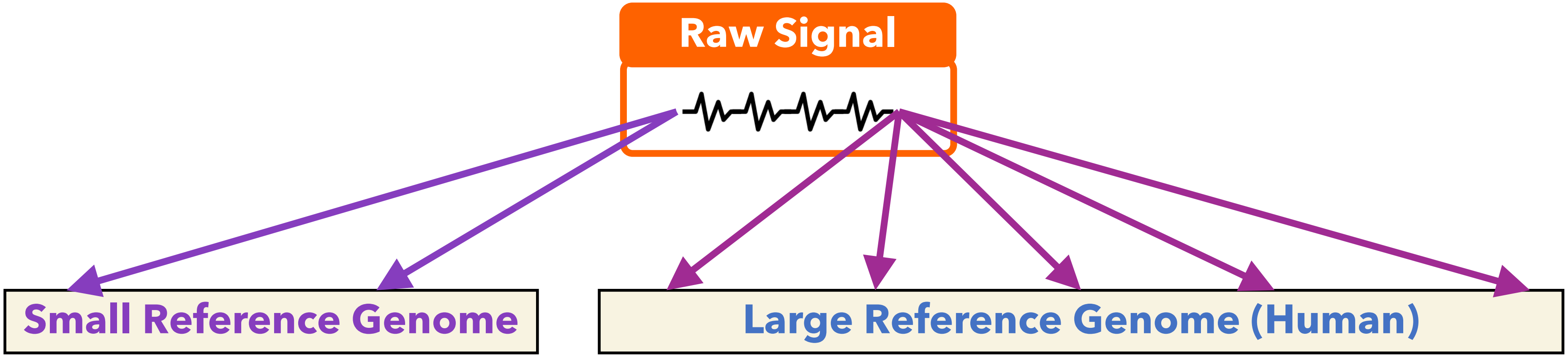
Recent Work: Directly analyzing signals **without basecalling**



✓ Efficient analysis with better scalability and portability

✓ Raw signals retain more information than just bases

The Problem – Mapping Raw Signals



Fewer candidate regions in **small genomes**

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

Accurate mapping

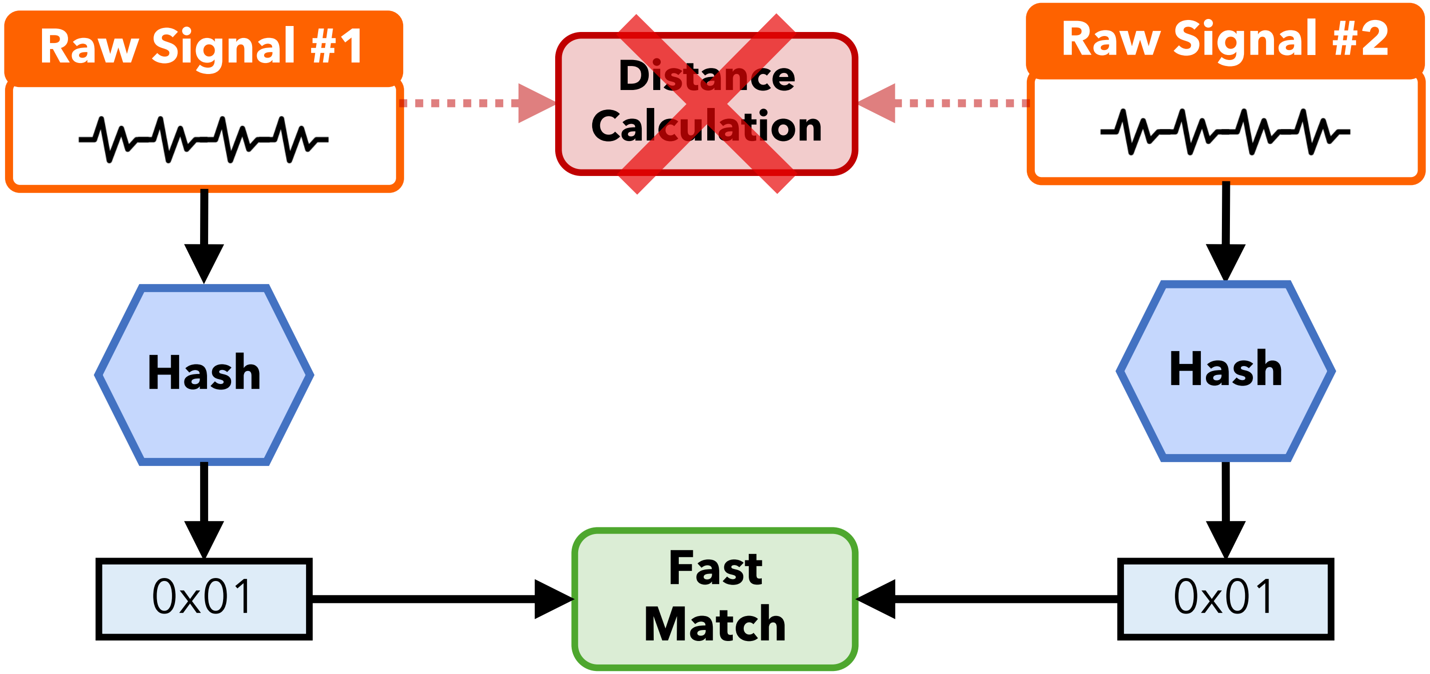
Problem: Probabilistic mechanisms on **many regions → inaccurate mapping**

High throughput

Problem: Distance calculation on **many regions → reduced throughput**

RawHash – Key Idea

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



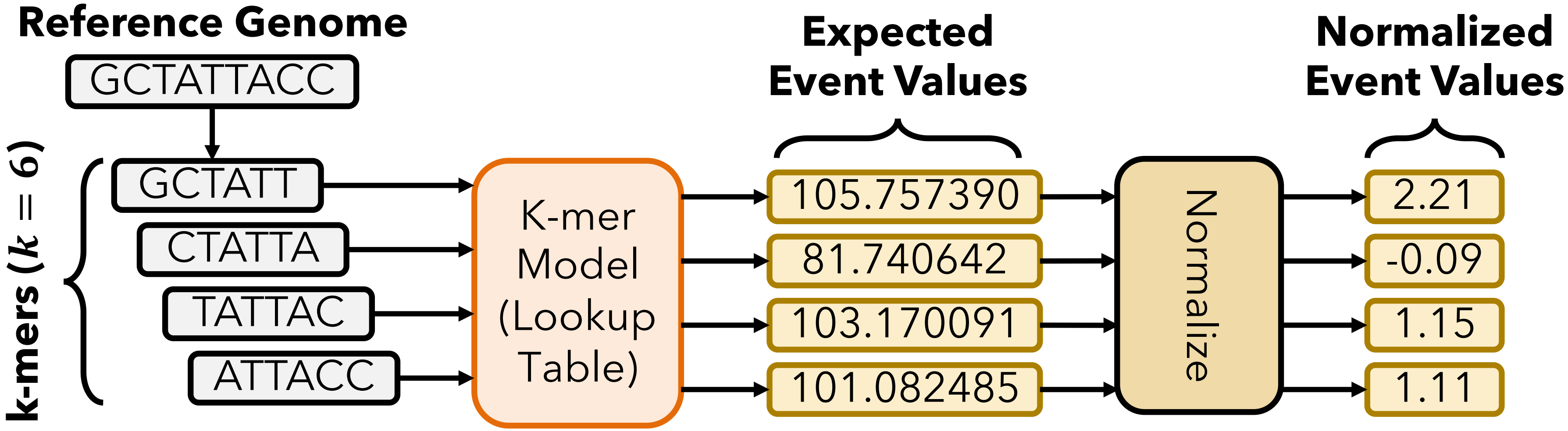
Challenge #1: Generating the **same** hash value for **similar enough** signals

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

Reference-to-Event Conversion

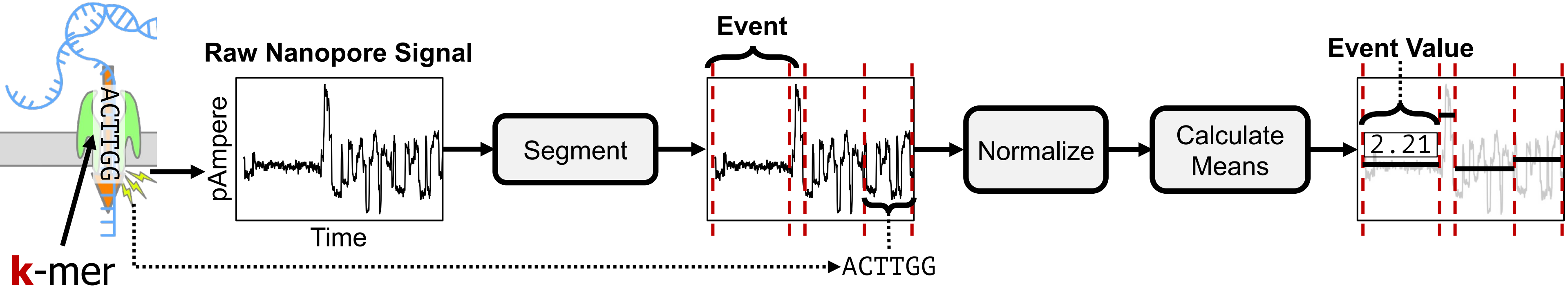
- **K-mer model:** Provides **expected** event values **for each k-mer**
  - Preconstructed based on nanopore sequencer characteristics

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



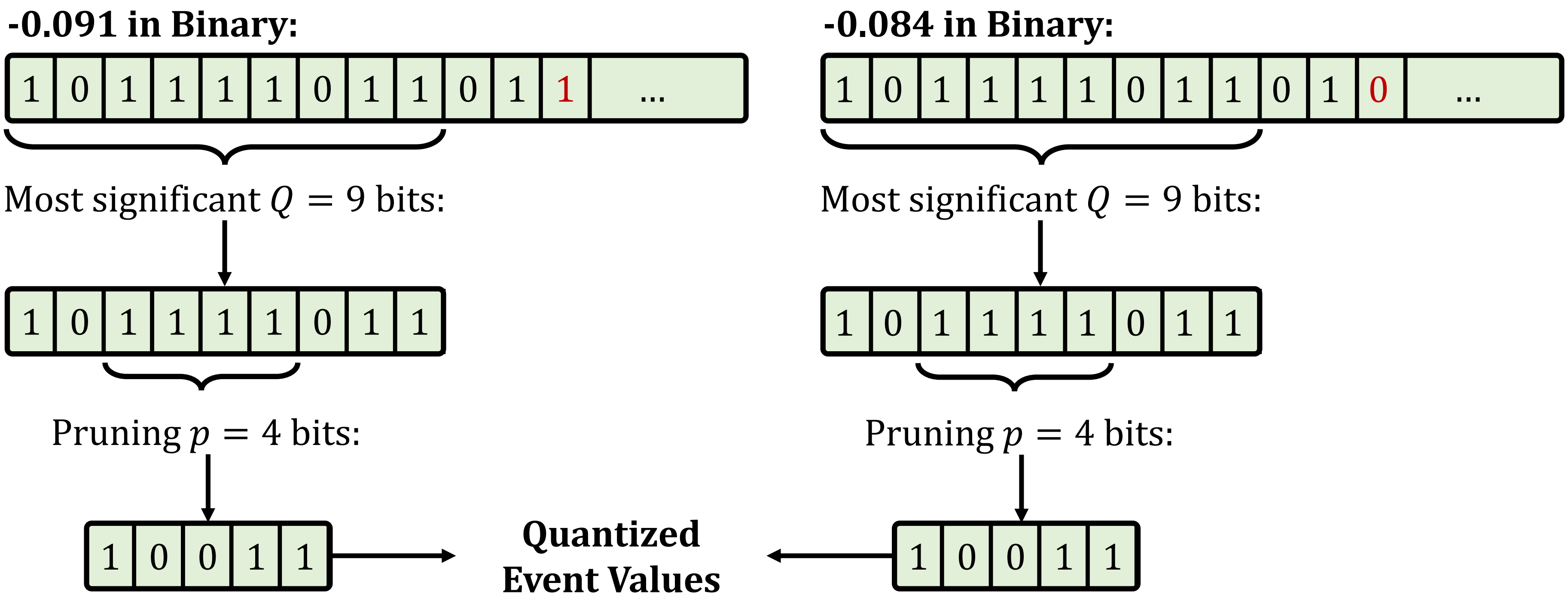
Enabling Analysis From Electrical Signals

- K many nucleotides (k-mers) sequenced at a time
- **Event:** A **segment** of the raw signal
  - Corresponds to a **particular** k-mer



- **Observation:** Event values generated after sequencing **the same k-mer** are **similar** in value (not necessarily the same)

Quantization -- RawHash



-0.091 in Binary:

1 0 1 1 1 1 0 1 1 0 1 1 ...

Most significant $Q = 9$ bits:

1 0 1 1 1 1 0 1 1

Pruning $p = 4$ bits:

1 0 0 1 1

-0.084 in Binary:

1 0 1 1 1 1 0 1 1 0 1 0 ...

Most significant $Q = 9$ bits:

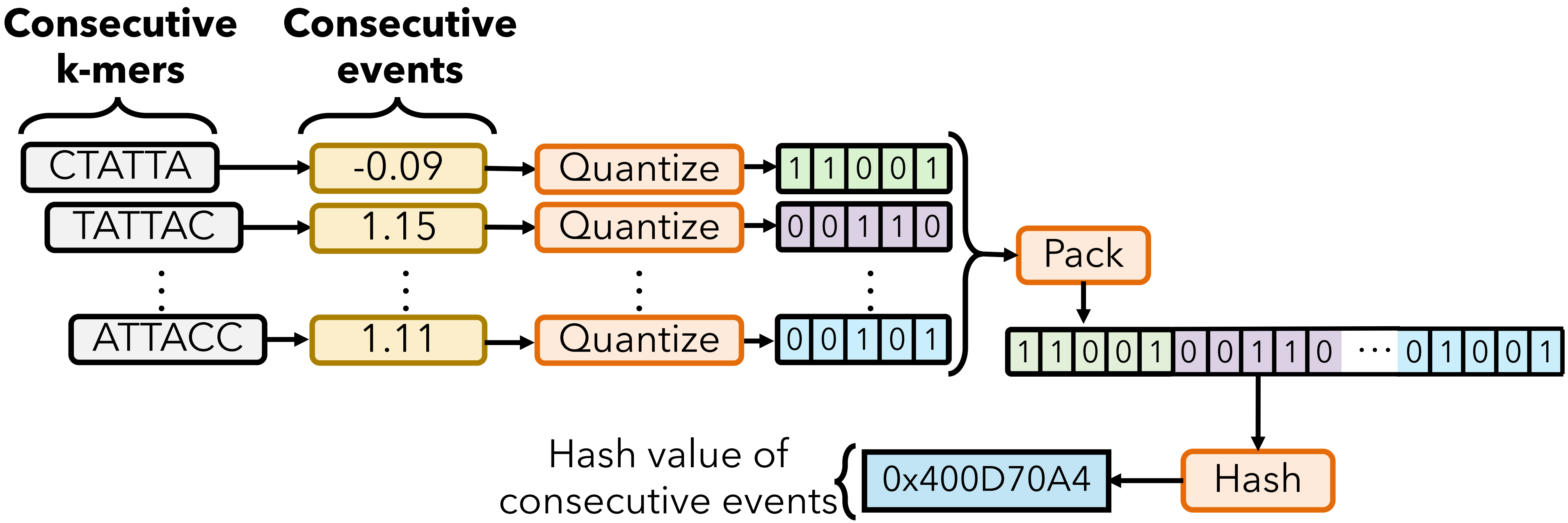
1 0 1 1 1 1 0 1 1

Pruning $p = 4$ bits:

1 0 0 1 1

Quantized Event Values

Packing and Hashing



The Sequence Until Mechanism

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

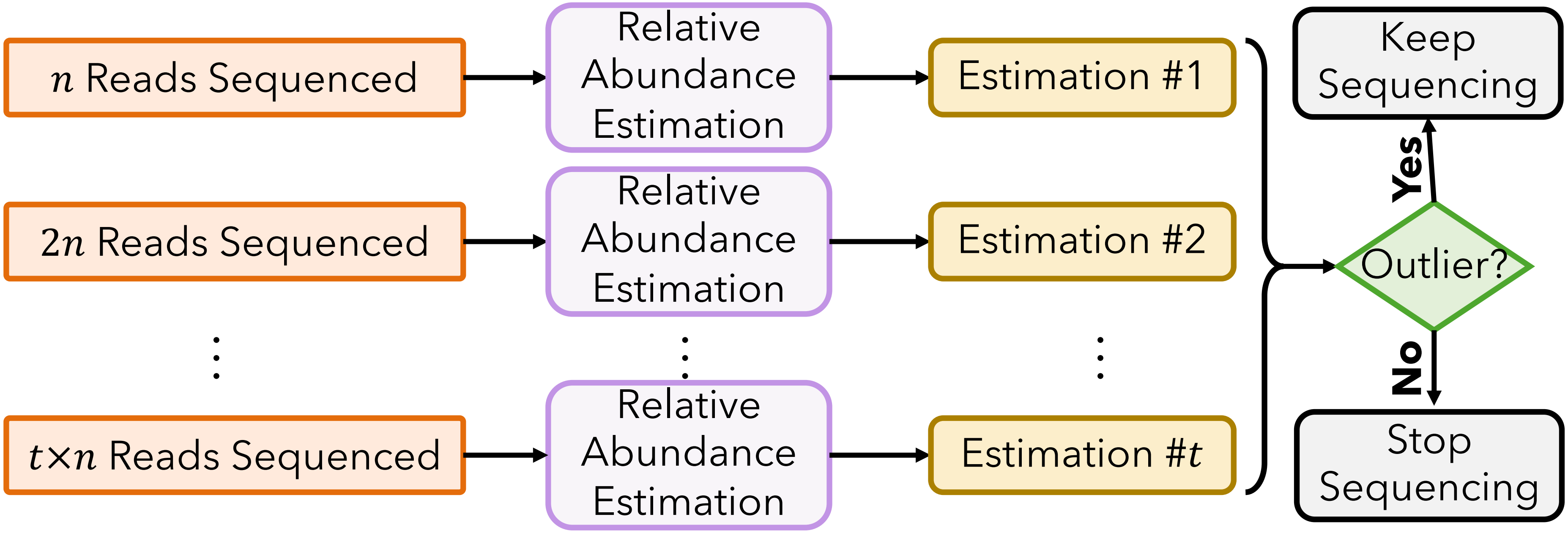
- **Key Idea:**
  - **Dynamically** decide if further sequencing of the entire sample is necessary to achieve high accuracy
  - Stop sequencing early without sacrificing accuracy

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

- Example real-time genome analysis use case:
  - **Relative abundance estimation**

The Sequence Until Mechanism

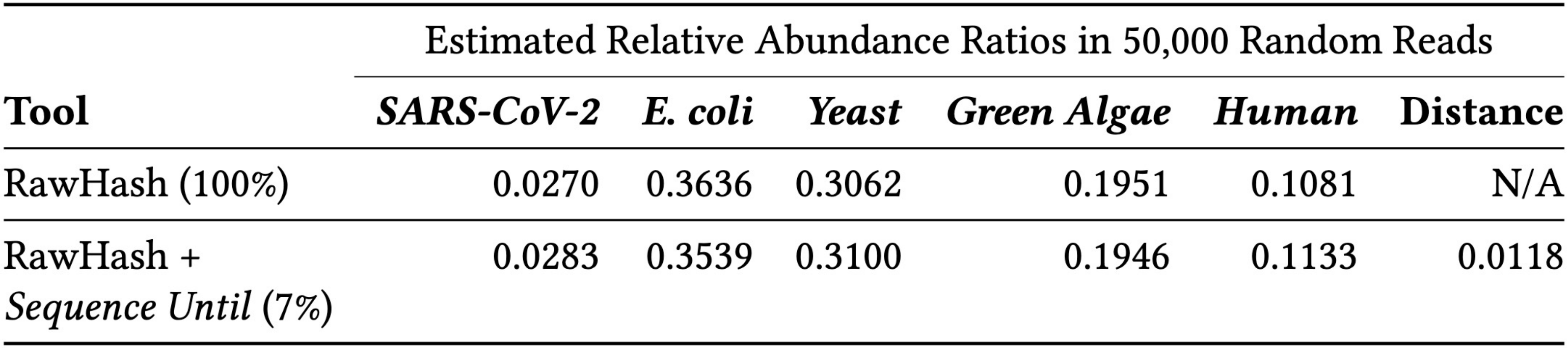
- **Key Steps:**
  1. Continuously generate relative abundance estimation after every n reads
  2. Keep the last t estimation results
  3. **Detect outliers** in the results via **cross-correlation** of the recent t results
  4. Absence of outliers indicates **consistent results**
     - Further sequencing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



Sequence Until – RawHash & UNCALLED

	Estimated Relative Abundance Ratios					
Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	**Distance**
Ground Truth	0.0929	0.4365	0.0698	0.1179	0.2828	N/A
UNCALLED (25%)	0.0026	0.5890	0.0613	0.1332	0.2139	0.1910
RawHash (25%)	0.0271	0.4853	0.0920	0.0786	0.3170	**0.0995**
UNCALLED (10%)	0.0026	0.5906	0.0611	0.1316	0.2141	0.1920
RawHash (10%)	0.0273	0.4869	0.0963	0.0772	0.3124	**0.1004**
UNCALLED (1%)	0.0026	0.5750	0.0616	0.1506	0.2103	0.1836
RawHash (1%)	0.0259	0.4783	0.0987	0.0882	0.3088	**0.0928**
UNCALLED (0.1%)	0.0040	0.4565	0.0380	0.1910	0.3105	0.1242
RawHash (0.1%)	0.0212	0.5045	0.1120	0.0810	0.2814	**0.1136**
UNCALLED (0.01%)	0.0000	0.5551	0.0000	0.0000	0.4449	0.2602
RawHash (0.01%)	0.0906	0.6122	0.0000	0.0000	0.2972	**0.2232**

Sequence Until – RawHash

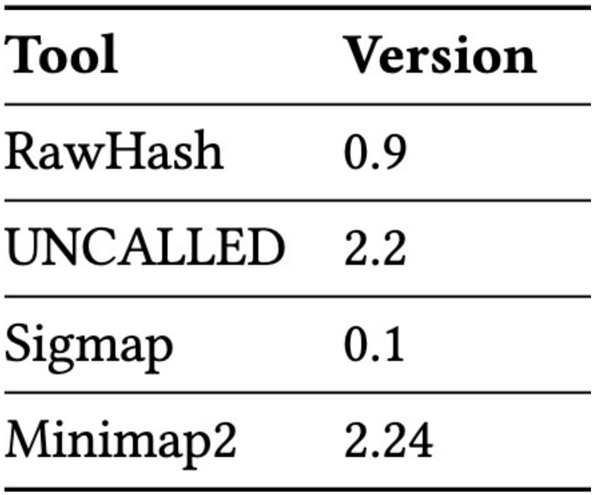


	Estimated Relative Abundance Ratios in 50,000 Random Reads					
Tool	***SARS-CoV-2***	***E. coli***	***Yeast***	***Green Algae***	***Human***	**Distance**
RawHash (100%)	0.0270	0.3636	0.3062	0.1951	0.1081	N/A
RawHash + *Sequence Until* (7%)	0.0283	0.3539	0.3100	0.1946	0.1133	0.0118

Presets

Preset (-x)	**Corresponding parameters**	Usage
viral	-e 5 -q 9 -l 3	Viral genomes
sensitive	-e 6 -q 9 -l 3	Small genomes (i.e., < 50M bases)
fast	-e 7 -q 9 -l 3	Large genomes (i.e., > 50M bases)

Versions – RawHash



Tool	Version
RawHash	0.9
UNCALLED	2.2
Sigmap	0.1
Minimap2	2.24

Related Works

- **Basecalled real-time analysis**
  - ReadFish, ReadBouncer, RUBRIC: Basecalled read mapping
  - SPUMONI, SPUMONI 2: Basecalled binary classification using r-index
  - Coriolis: Basecalled metagenomics classification
  - baseLess: k-mer calling for classification

- **Raw signal analysis without basecalling**
  - SquiggleNet, DeepSelectNet, RawMap: Target/non-target classification
  - Sigmoni: Target/non-target classification using r-index
  - UNCALLED, Sigmap, RawHash: Read mapping

Adaptive Quantization

$$q(s) = \begin{cases} \lfloor n \times (f_r \times \frac{(s-f_{min})}{f_{max}-f_{min}}) & \text{if } f_{min} \leq s \leq f_{max} \\ \lfloor n \times (f_r + c_r \times s) & \text{if } s < f_{min} \\ \lfloor n \times (f_r + c_r + c_r \times s) & \text{if } s > f_{max} \end{cases}$$

Chaining Scores – RawHash vs RawHash2

- **RawHash Chaining**

$$f(i) = \max\left\{ \max_{i>j\geq 1}\{f(j) + \alpha(j, i)\}, w_i \right\}$$

$$\alpha(j, i) = \min\left\{ \min\{y_i - y_j, x_i - x_j\}, w_i \right\}$$

- **RawHash2 Chaining**

$$f(i) = \max\left\{ \max_{i>j\geq 1}\{f(j) + \alpha(j, i) - \beta(j, i)\}, w_i \right\}$$

$$\beta(j, i) = \gamma_c\big((y_i - y_j) - (x_i - x_j)\big)$$

$$\gamma_c(l) = \begin{cases} 0.01 \cdot \bar{w} \cdot |l| + 0.5 \log_2 |l| & (l \neq 0) \\ 0 & (l = 0) \end{cases}$$

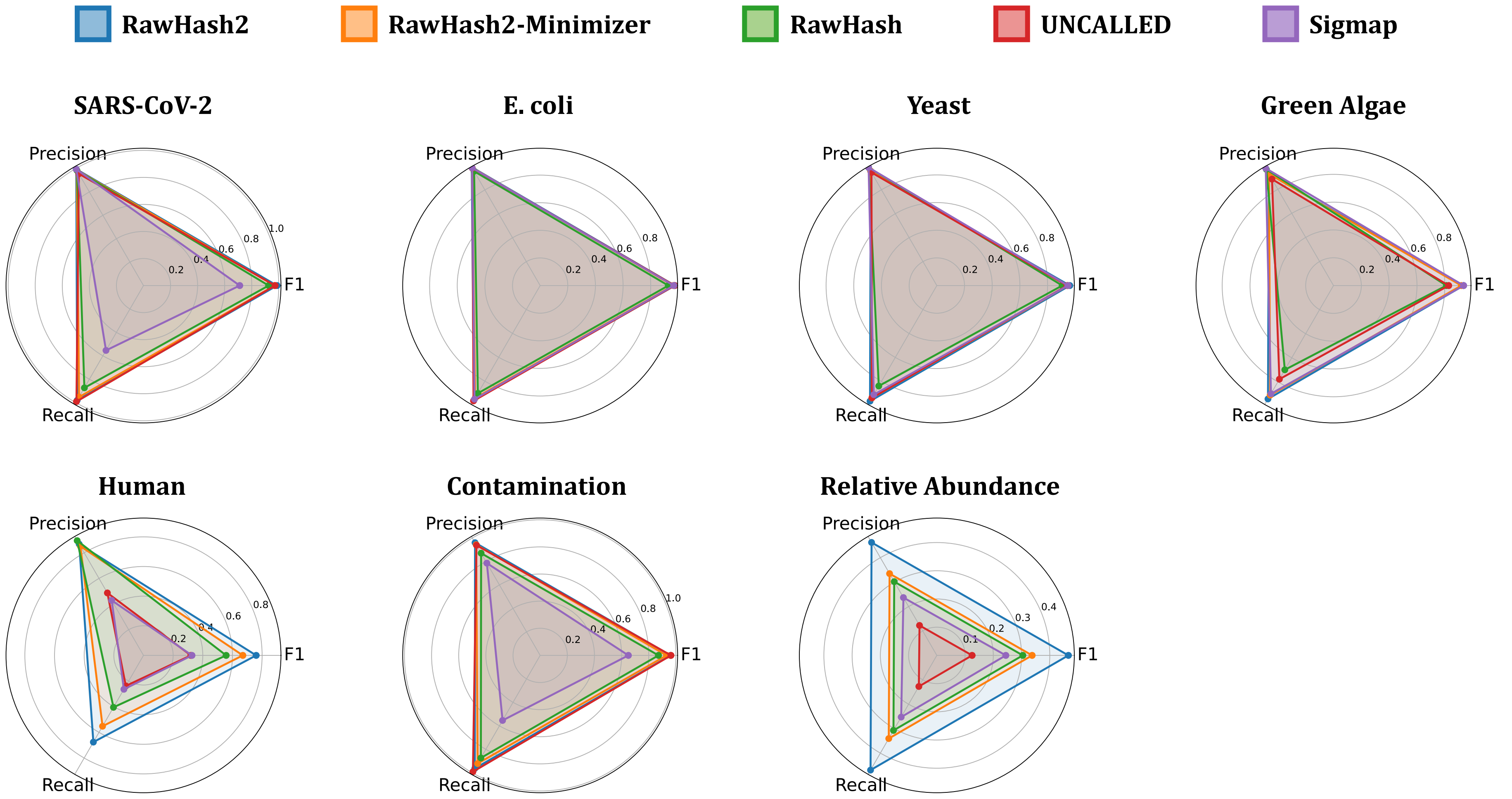
Datasets

	Organism	Device Type	Flow Cell Type	Transloc. Speed	Sampling Frequency	Basecaller Model	Reads (#)	Bases (#)	SRA Accession	Reference Genome	Genome Size
Read Mapping											
D1	*SARS-CoV-2*	MinION	R9.4.1 e8 (FLO-MIN106)	450	4000	Guppy HAC v3.2.6	1,382,016	594M	CADDE Centre	GCF_009858895.2	29,903
D2	*E. coli*	GridION	R9.4.1 e8 (FLO-MIN106)	450	4000	Guppy HAC v5.0.12	353,317	2,365M	ERR9127551	GCA_000007445.1	5M
D3	*Yeast*	MinION	R9.4.1 e8 (FLO-MIN106)	450	4000	Albacore v2.1.7	49,989	380M	SRR8648503	GCA_000146045.2	12M
D4	*Green Algae*	PromethION	R9.4.1 e8 (FLO-PRO002)	450	4000	Albacore v2.3.1	29,933	609M	ERR3237140	GCF_000002595.2	111M
D5	*Human*	MinION	R9.4.1 e8 (FLO-MIN106)	450	4000	Guppy Flip-Flop v2.3.8	269,507	1,584M	FAB42260	T2T-CHM13 (v2)	3,117M
D6	*E. coli*	GridION	R10.4 e8.1 (FLO-MIN112)	450	4000	Guppy HAC v5.0.16	1,172,775	6,123M	ERR9127552	GCA_000007445.1	5M
D7	*S. aureus*	GridION	R10.4 e8.1 (FLO-MIN112)	450	4000	Dorado SUP v0.5.3	407,727	1,281M	SRR21386013	GCF_000144955.2	2.8M
Contamination Analysis											
D1 and D5							1,651,523	2,178M	D1 and D5	D1	29,903
Relative Abundance Estimation											
D1-D5							2,084,762	5,531M	D1-D5	D1-D5	3,246M

Accuracy

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

Mapping Accuracy – Radar



SARS-CoV-2

E. coli

Yeast

Green Algae

Human

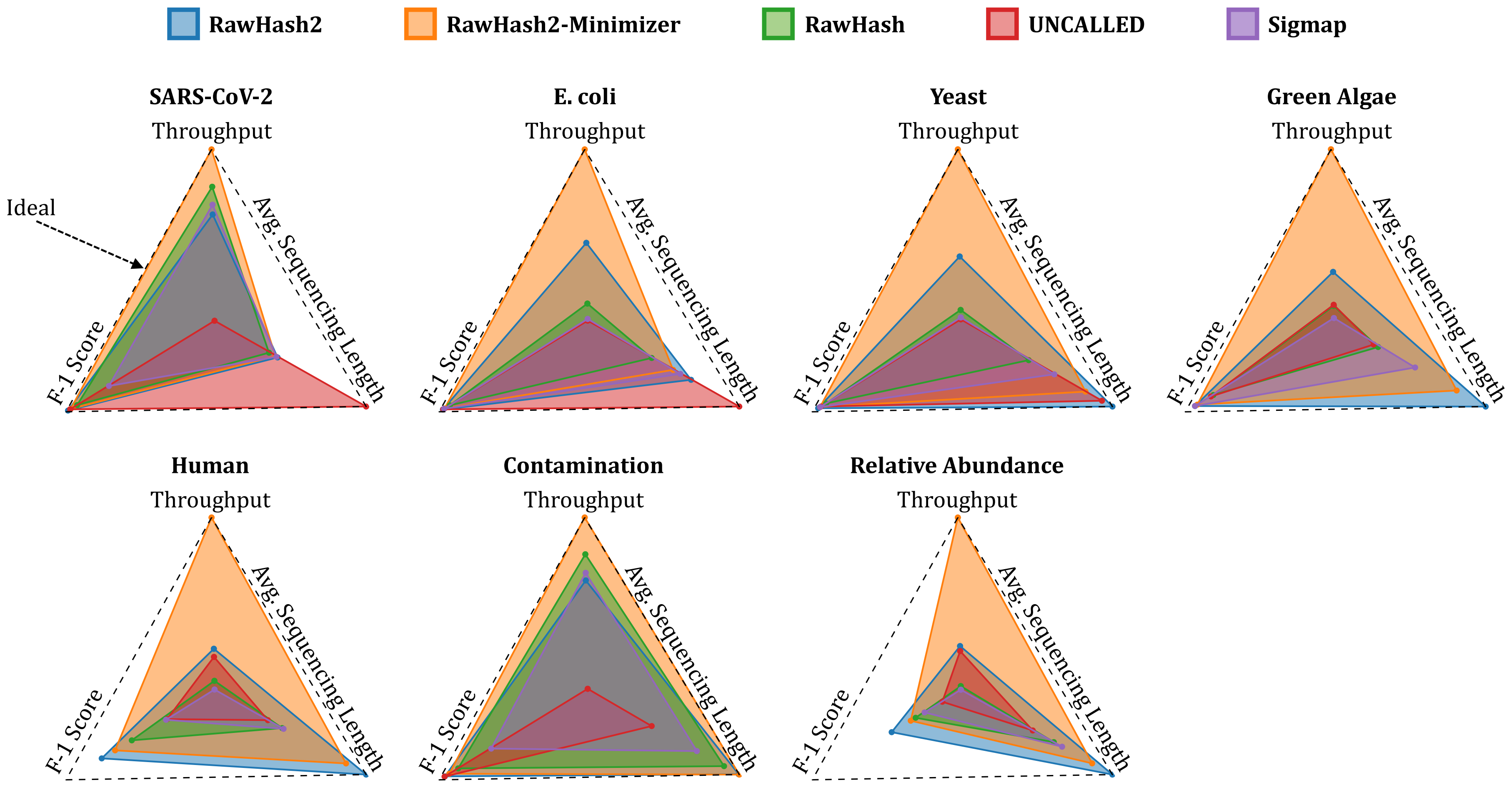
Contamination

Relative Abundance

Mapping Accuracy – All Metrics

Dataset	**Metric**	**RH2**	**RH2-Min.**	**RH**	**UNCALLED**	**Sigmap**
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

Combined Benefits – Radar



Sequenced Length

Dataset	RH2	RH2-Min.	RH	UNCALLED	Sigmap
SARS-CoV-2	443.92	460.85	513.95	**184.51**	452.38
E. coli	851.31	1,030.74	1,376.14	**580.52**	950.03
Yeast	**1,147.66**	1,395.87	2,565.09	1,233.20	1,862.69
Green Algae	**1,385.59**	1,713.46	4,760.59	5,300.15	2,591.16
Human	**2,130.59**	2,455.99	4,773.58	6,060.23	4,680.50
Contamination	670.69	**667.89**	742.56	1,582.63	927.82
Rel. Abundance	**1,024.28**	1,182.04	1,669.46	2,158.50	1,533.04

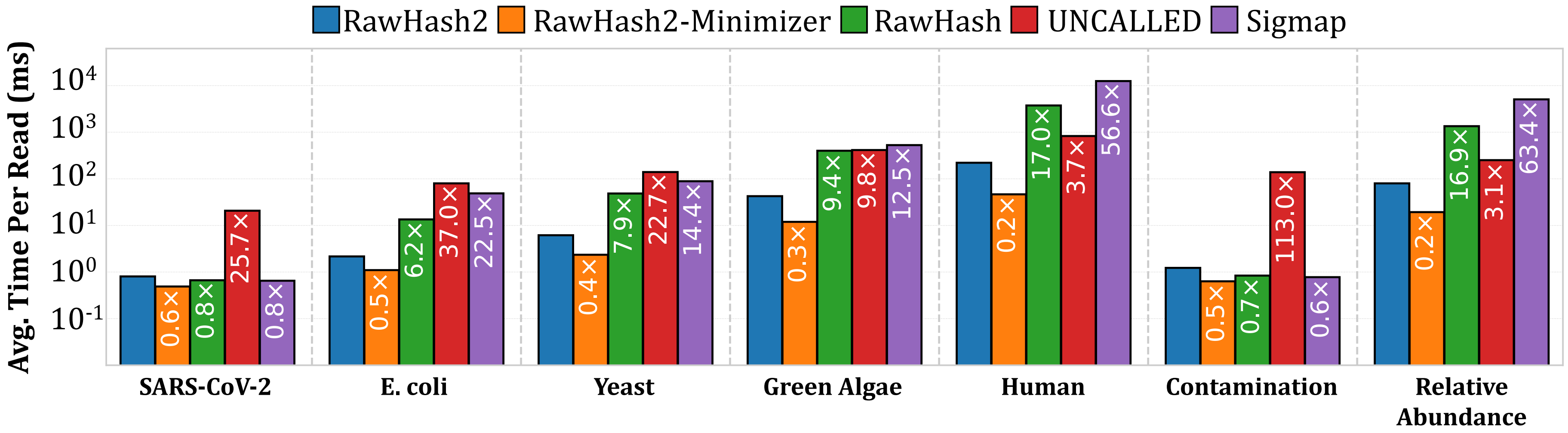
Computational Resources #1

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
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

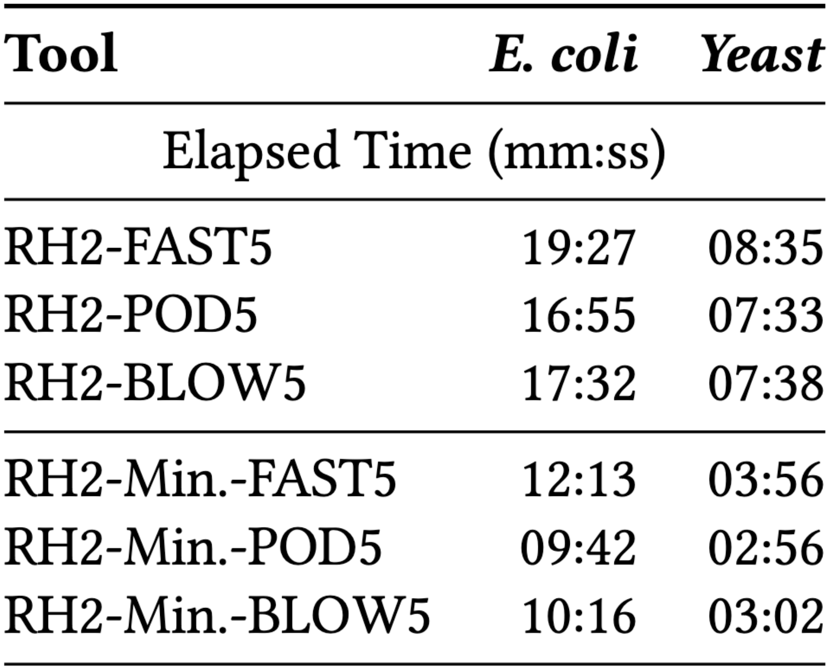
Computational Resources #2

			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
			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

Average Time Spent per Read



FAST5 vs. POD5. vs S/BLOW5

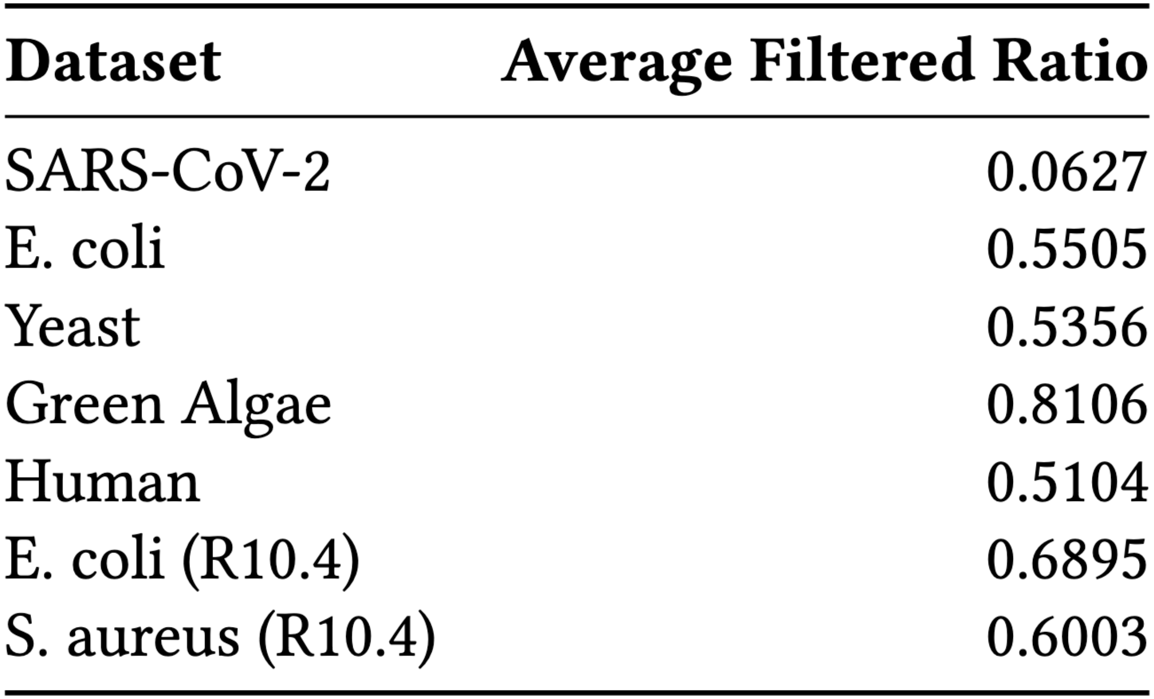


Tool	*E. coli*	*Yeast*
Elapsed Time (mm:ss)		
RH2-FAST5	19:27	08:35
RH2-POD5	16:55	07:33
RH2-BLOW5	17:32	07:38
RH2-Min.-FAST5	12:13	03:56
RH2-Min.-POD5	09:42	02:56
RH2-Min.-BLOW5	10:16	03:02

Flow Cell Types R9 vs R10.4

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

Ratio of Filtered Seed Hits



Dataset	Average Filtered Ratio
SARS-CoV-2	0.0627
E. coli	0.5505
Yeast	0.5356
Green Algae	0.8106
Human	0.5104
E. coli (R10.4)	0.6895
S. aureus (R10.4)	0.6003

Presets

Preset	Corresponding parameters	Usage
viral	-e 6 -q 4 –max-chunks 5 –bw 100 –max-target-gap 500 –max-target-gap 500 –min-score 10 –chain-gap-scale 1.2 –chain-skip-scale 0.3	Viral genomes
sensitive	-e 8 -q 4 –fine-range 0.4	Small genomes (i.e., < 500*M* bases)
fast	-e 8 -q 4 –max-chunks 20	Large genomes (i.e., > 500*M* bases)
	Other helper parameters	
depletion	–best-chains 5 –min-mapq 10 –w-threshold 0.5 –min-anchors 2 –min-score 15 –chain-skip-scale 0	Contamination analysis
r10	-k9 –seg-window-length1 3 –seg-window-length2 6 –seg-threshold1 6.5 –seg-threshold2 4 –seg-peak-height 0.2 –chain-gap-scale 1.2	For R10.4 Flow Cells

Versions

Tool	Version
RawHash2	2.1
RawHash	1.0
UNCALLED	2.3
Sigmap	0.1
Minimap2	2.24

FAST5 (HDF5)	1.10
POD5	0.2.2
S/BLOW5	1.2.0-beta

Datasets

	Organism	**Device Type**	**Reads (#)**	**Bases (#)**	**Avg. Read Length**	**Estimated Coverage (×)**	**SRA Accession**
D1	*SARS-CoV-2*	MinION	10,001	4.02M	402	135×	CADDE Centre
D2	*E. coli*	GridION	353,948	2,332M	6,588	445×	ERR9127551
D3	*Yeast*	MinION	50,023	385M	7,698	32×	SRR8648503
D4	*Green Algae*	PromethION	30,012	622M	20,731	5.6×	ERR3237140
D5	*Human*	MinION	270,006	1,773M	6,567	0.6×	FAB42260

Throughput

	D1	D2	D3	D4	D5
	SARS-CoV-2	*E. coli*	*Yeast*	*Green Algae*	*Human*
Throughput	2,065,764	2,720,702	2,128,800	1,668,065	3,579,472

Performance

Organism	Tool	Elapsed time (hh:mm:ss)	CPU time (sec)	Peak Mem. (GB)
D1 *SARS-CoV-2*	**Rawsamble**	0:00:03	33	1.07
	Minimap2	0:00:01 (0.33×)	19 (0.58×)	0.16 (0.15×)
	Minimap2 + Dorado CPU (Fast)	0:01:45 (35.00×)	3,227 (97.79×)	44.93 (41.99×)
	Minimap2 + Dorado CPU (HAC)	0:05:45 (115.00×)	5,457 (165.36×)	57.98 (54.19×)
	Minimap2 + Dorado GPU (HAC)	0:01:41 (33.67×)	NA	0.8 (0.75×)
	Minimap2 + Dorado GPU (SUP)	0:25:47 (515.67×)	NA	1.23 (1.15×)
D2 *E. coli*	**Rawsamble**	1:12:44	132,758	6.72
	Minimap2	0:14:25 (0.20×)	25,721 (0.19×)	26.73 (3.98×)
	Minimap2 + Dorado CPU (Fast)	7:17:05 (6.01×)	583,358 (4.39×)	50.43 (7.50×)
	Minimap2 + Dorado CPU (HAC)	32:26:12 (26.76×)	1,335,697 (10.06×)	38.0 (5.65×)
	Minimap2 + Dorado GPU (HAC)	0:36:14 (0.50×)	NA	26.73 (3.98×)
	Minimap2 + Dorado GPU (SUP)	1:30:30 (1.24×)	NA	26.73 (3.98×)
D3 *Yeast*	**Rawsamble**	0:01:18	2,241	6.39
	Minimap2	0:00:21 (0.27×)	290 (0.13×)	5.25 (0.82×)
	Minimap2 + Dorado CPU (Fast)	0:54:04 (41.59×)	71,796 (32.04×)	56.13 (8.78×)
	Minimap2 + Dorado CPU (HAC)	3:13:56 (149.18×)	193,640 (86.41×)	65.43 (10.24×)
	Minimap2 + Dorado GPU (HAC)	0:04:33 (3.50×)	NA	5.25 (0.82×)
	Minimap2 + Dorado GPU (SUP)	0:10:33 (8.12×)	NA	5.92 (0.93×)
D4 *Green Algae*	**Rawsamble**	0:07:57	14,064	8.67
	Minimap2	0:00:47 (0.10×)	882 (0.06×)	8.7 (1.00×)
	Minimap2 + Dorado CPU (Fast)	1:16:35 (9.63×)	79,606 (5.66×)	50.88 (5.87×)
	Minimap2 + Dorado CPU (HAC)	4:30:07 (33.98×)	286,362 (20.36×)	64.07 (7.39×)
	Minimap2 + Dorado GPU (HAC)	0:06:01 (0.76×)	NA	8.7 (1.00×)
	Minimap2 + Dorado GPU (SUP)	0:14:54 (1.87×)	NA	8.7 (1.00×)
D5 *Human*	**Rawsamble**	0:28:56	51,975	6.0
	Minimap2	0:01:52 (0.06×)	1,372 (0.03×)	20.21 (3.37×)
	Minimap2 + Dorado CPU (Fast)	6:42:24 (13.91×)	802,983 (15.45×)	81.98 (13.66×)
	Minimap2 + Dorado CPU (HAC)	23:27:18 (48.64×)	1,219,043 (23.45×)	46.12 (7.69×)
	Minimap2 + Dorado GPU (HAC)	0:20:24 (0.71×)	NA	20.31 (3.38×)
	Minimap2 + Dorado GPU (SUP)	1:05:48 (2.27×)	NA	20.21 (3.37×)

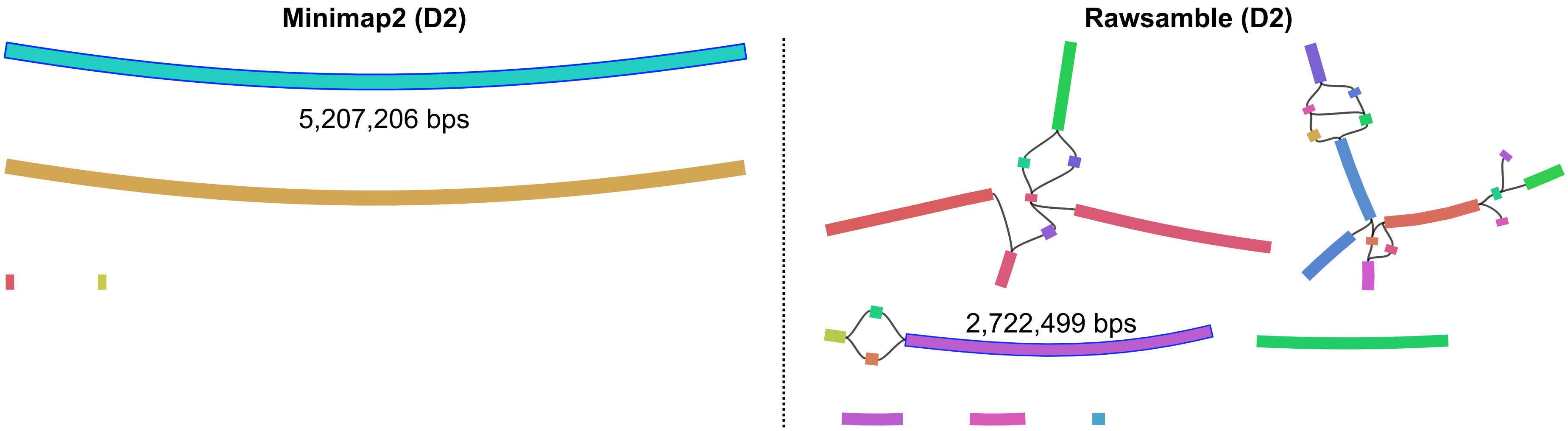
Overlapping Statistics

	Organism	Unique to Rawsamble (%)	Unique to Minimap2 (%)	Shared Overlaps (%)
D1	*SARS-CoV-2*	11.55	15.27	73.18
D2	*E. coli*	8.33	50.62	41.05
D3	*Yeast*	24.94	35.17	39.89
D4	*Green Algae*	3.76	78.64	17.61
D5	*Human*	32.69	56.18	11.13

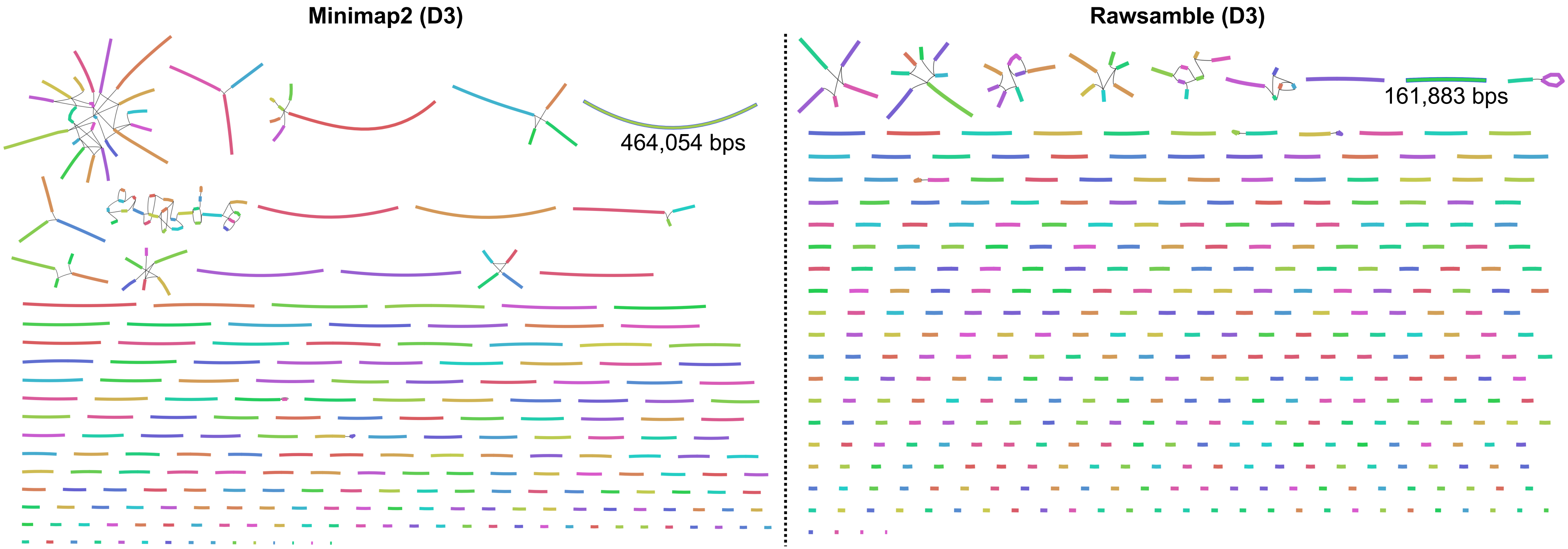
Assembly Statistics

Dataset	Tool	Total Length (bp)	Largest Comp. (bp)	N50 (bp)	auN (bp)	Longest Unitig (bp)	Unitig Count
D2	Rawsamble	14,525,505	4,841,669	1,535,079	1,309,738	2,722,499	31
E. coli	minimap2	10,434,542	5,207,206	5,204,754	5,194,738	5,207,206	4
	Gold standard	5,235,343	5,235,343	5,235,343	5,235,343	5,235,343	1
D3	Rawsamble	13,898,208	362,050	41,118	48,106	161,883	396
Yeast	minimap2	23,755,455	1,611,876	134,050	150,908	464,054	282
	Gold standard	11,963,521	11,835,059	640,934	623,210	1,073,346	68
D4	Rawsamble	3,448,899	448,422	93,111	108,818	252,038	50
Green Algae	minimap2	2,117,190	198,709	63,310	88,906	198,709	55
	Gold standard	106,479,288	2,255,807	452,774	538,136	1,667,975	420
D5	Rawsamble	1,850,419	493,004	51,300	116,049	364,113	48
Human	minimap2	747,607	65,951	19,476	22,103	48,424	61
	Gold standard	8,365,210	367,305	19,329	29,697	150,470	592

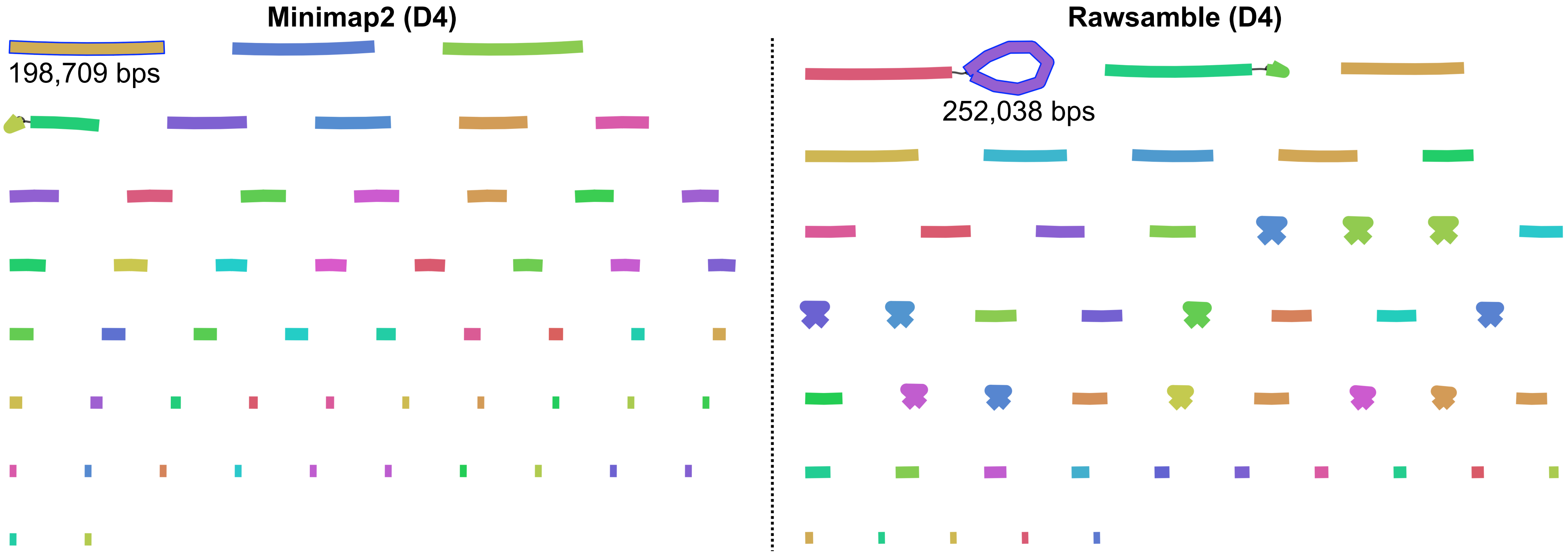
Visualizing the E. coli Assembly Graph



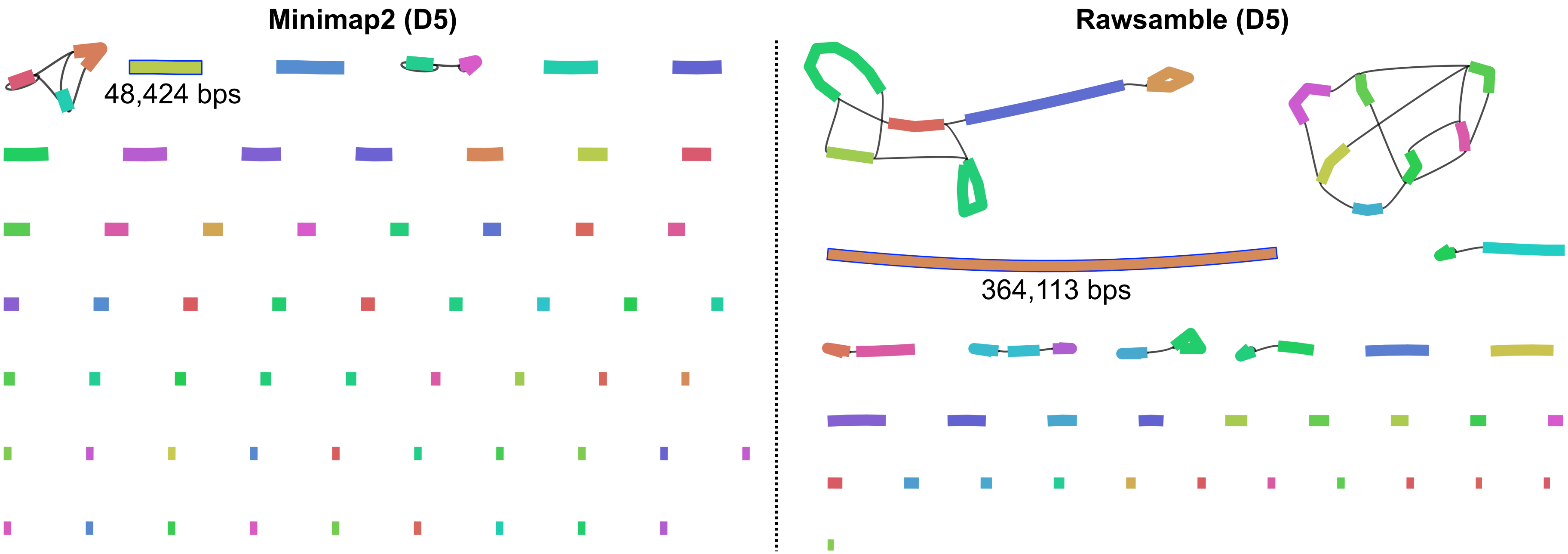
Visualizing the Yeast Assembly Graph



Visualizing the Green Algae Assembly Graph



Visualizing the Human Assembly Graph



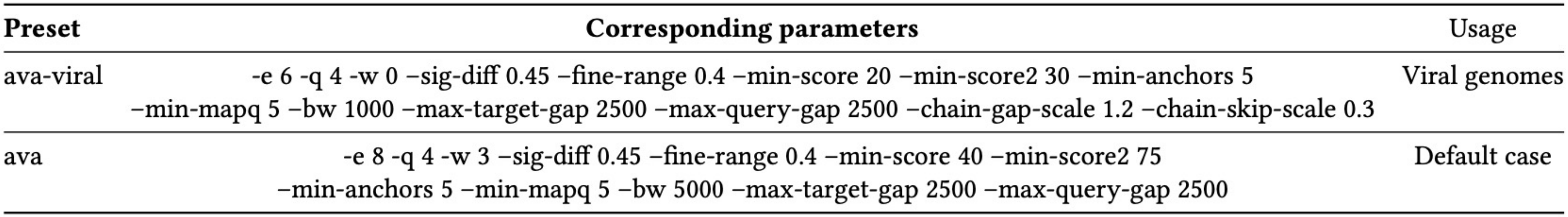
HERRO Correction Before and After

Dataset	Coverage Before Correction	Coverage After Correction
D2 *E. coli*	445×	240×
D3 *Yeast*	32×	12×
D4 *Green algae*	5.6×	3.7×
D5 *Human*	0.6×	0.002×

Parameters

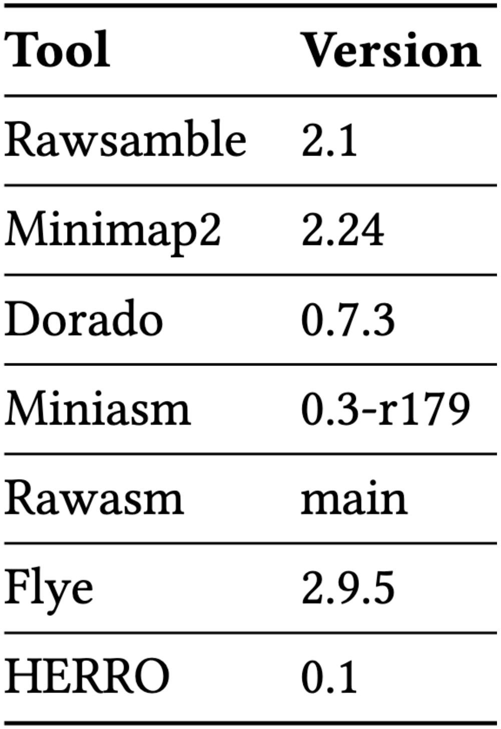
Tool	D1 *SARS-CoV-2*	D2 *E. coli*	D3 *Yeast*	D4 *Green Algae*	D5 *Human*
Rawsamble	-x ava-viral -t 32	-x ava -t 32	-x ava -t 32	-x ava -t 32	-x ava –chain-gap-scale 0.6 -t 32
Minimap2	-x ava-ont –for-only -t 32				
Dorado CPU (Fast)	basecaller -x cpu dna_r9.4.1_e8_fast@v3.4				
Dorado CPU (HAC)	basecaller -x cpu dna_r9.4.1_e8_hac@v3.3				
Dorado GPU (HAC)	basecaller dna_r9.4.1_e8_hac@v3.3				
Dorado GPU (SUP)	basecaller dna_r9.4.1_e8_sup@v3.3				
Miniasm					

Presets



Preset	Corresponding parameters	Usage
ava-viral	-e 6 -q 4 -w 0 –sig-diff 0.45 –fine-range 0.4 –min-score 20 –min-score2 30 –min-anchors 5 –min-mapq 5 –bw 1000 –max-target-gap 2500 –max-query-gap 2500 –chain-gap-scale 1.2 –chain-skip-scale 0.3	Viral genomes
ava	-e 8 -q 4 -w 3 –sig-diff 0.45 –fine-range 0.4 –min-score 40 –min-score2 75 –min-anchors 5 –min-mapq 5 –bw 5000 –max-target-gap 2500 –max-query-gap 2500	Default case

Versions



Tool	Version
Rawsamble	2.1
Minimap2	2.24
Dorado	0.7.3
Miniasm	0.3-r179
Rawasm	main
Flye	2.9.5
HERRO	0.1