

# GenPIP

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

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

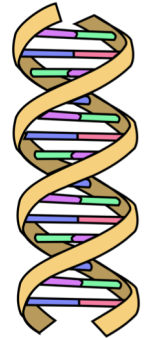
**SAFARI**

**ETH** zürich

# Overview: Genome Analysis

❑ **Genome analysis:** Enables us to determine the order of the DNA sequence in an organism's genome

- Plays an **important role** in
  - Personalized medicine
  - Outbreak tracing
  - Understanding of evolution
  - ...



❑ Modern genome sequencing machines extract smaller randomized fragments of the original DNA sequence, known as **reads**

- **Oxford Nanopore Technologies (ONT):**

A widely-used sequencing technology

- Portable sequencing devices
- High-throughput
- Cheap



**ONT sequencing device** [forbes.com]

# Overview: Two Limitations

---

Multiple steps in genome analysis



**Large data movement**  
between multiple steps



A lot of  
**wasted computation**  
done on data that is  
later discovered to be  
**useless**

# Overview: GenPIP

---

❑ **GenPIP:** A fast and energy-efficient **in-memory** acceleration system for the Genome analysis PIpline via **tight integration of genome analysis steps**

❑ **GenPIP** has two key techniques

- **Chunk-based pipeline (CP)**

- **Provides fine-grained collaboration** of genome analysis steps

- **Early rejection (ER)**

- **Timely stops the execution on useless data** by predicting which reads will not be useful

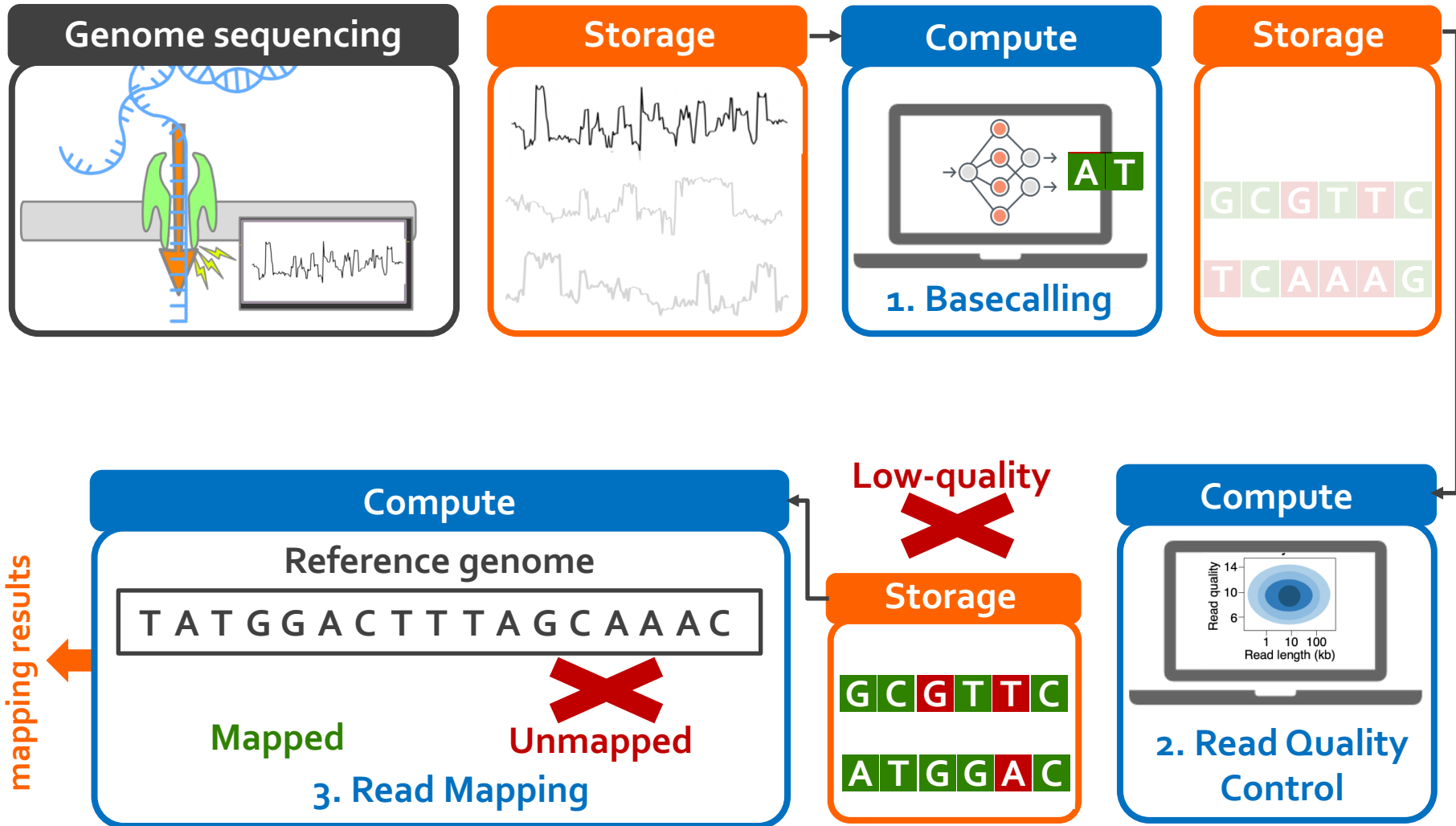
❑ **GenPIP** outperforms state-of-the-art software & hardware solutions using **CPU**, **GPU**, and **optimistic PIM** by **41.6x**, **8.4x**, and **1.4x**, respectively.

# Outline

---

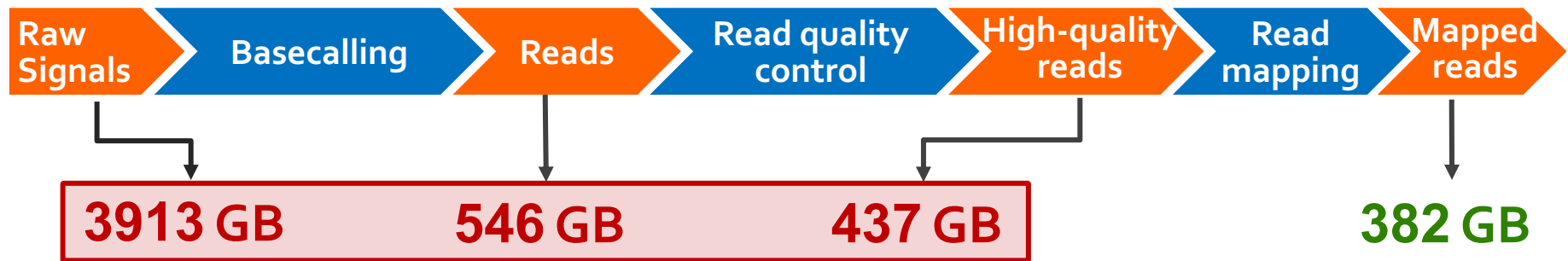
- ❑ Background and Motivation
- ❑ GenPIP: Tight Integration of Genome Analysis Steps
  - Chunk-based Pipeline (CP)
  - Early Rejection (ER)
- ❑ GenPIP Implementation
- ❑ Evaluation
- ❑ Conclusion

# Genome Analysis Pipeline



# Limitation 1: Large Data Movement

- Using a human dataset in [NC'19] as an example:

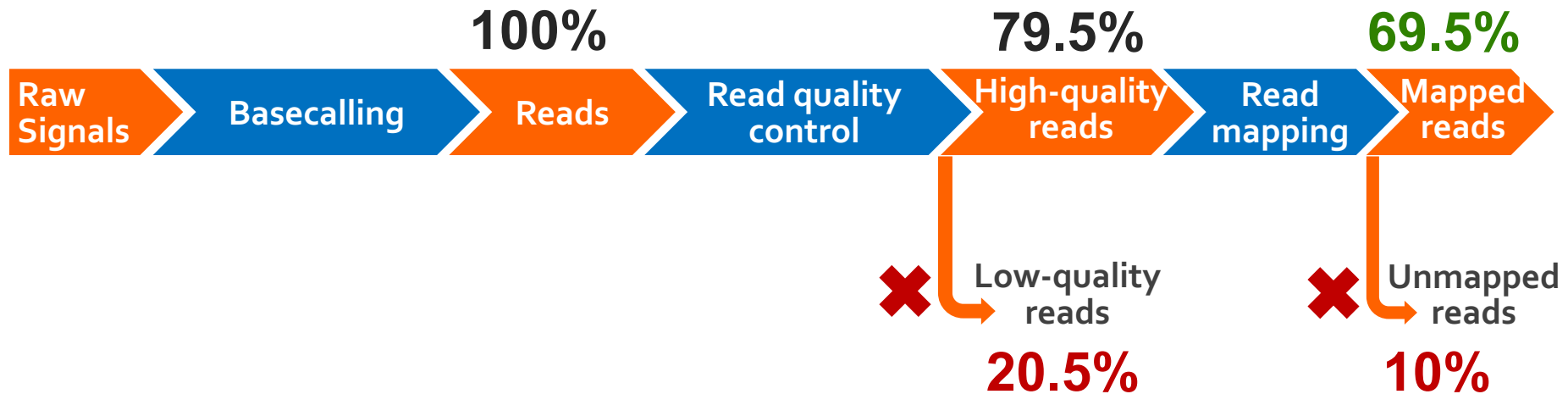


**Large data movement** between genome analysis steps

[NC'19] Rory Bowden, Robert W Davies, Andreas Heger, Alistair T Pagnamenta, Mariateresa de Cesare, Laura E Oikkinen, Duncan Parkes, Colin Freeman, Fatima Dhalla, Smita Y Patel, et al. Sequencing of human genomes with nanopore technology. Nature Communications, 2019.

# Limitation 2: Wasted Computation

- Using a human dataset in [NC'19] as an example:



A considerable amount of computation on **useless data** due to

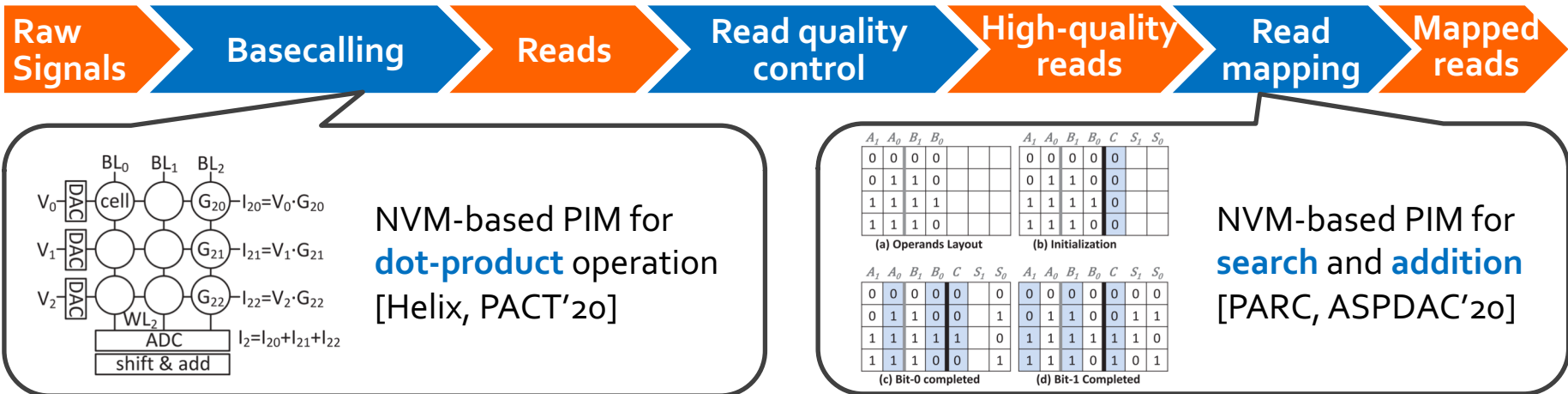
- Low-quality reads
- Unmapped reads

[NC'19] Rory Bowden, Robert W Davies, Andreas Heger, Alistair T Pagnamenta, Mariateresa de Cesare, Laura E Oikkonen, Duncan Parkes, Colin Freeman, Fatima Dhalla, Smita Y Patel, et al. Sequencing of human genomes with nanopore technology. Nature Communications, 2019.



# State-of-the-art Works

- ❑ NVM-based PIM is an efficient technique to reduce data movement by processing data using or near memory



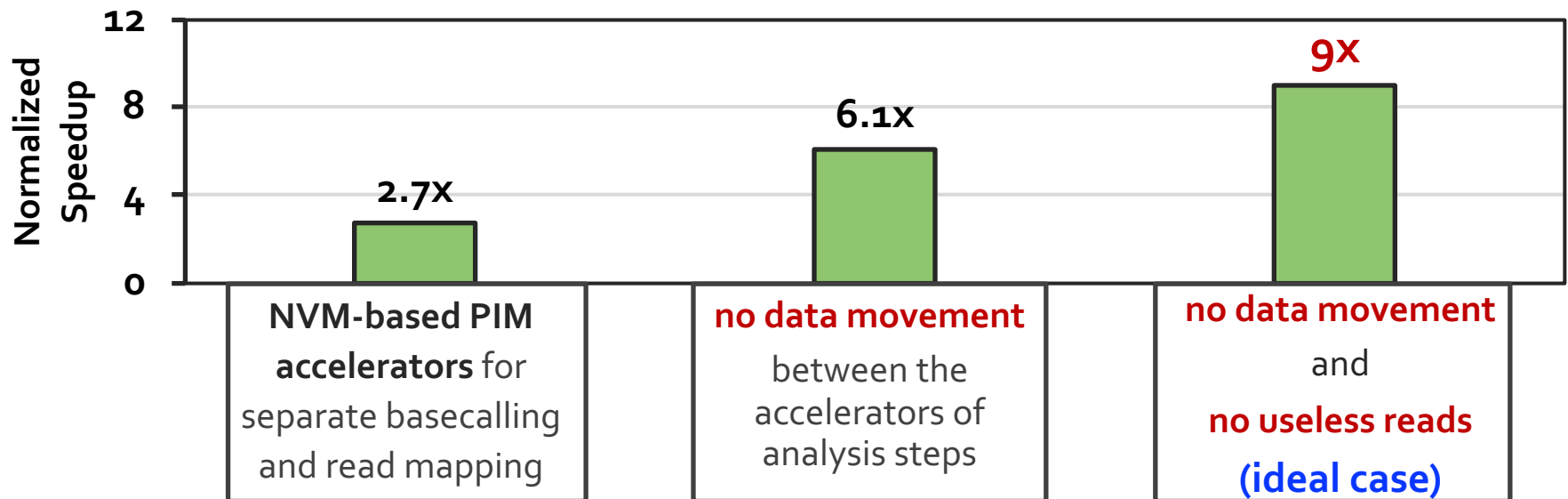
- Reduce the data movement in a single genome analysis step
- Exacerbate the data movement overhead between analysis steps

**No prior work tackles data movement between analysis steps and reduces useless computation**

# Goal and Opportunities

**Goal:** Efficiently accelerate the entire genome analysis pipeline while **minimizing data movement and useless computation**

- We perform a study to quantify potential performance benefits
  - Results are normalized to the performance of GPU



# Outline

---

- ❑ Background and Motivation
- ❑ **GenPIP: Tight Integration of Genome Analysis Steps**
  - **Chunk-based Pipeline (CP)**
  - **Early Rejection (ER)**
- ❑ GenPIP Implementation
- ❑ Evaluation
- ❑ Conclusion

# GenPIP

---

- ❑ **First holistic in-memory accelerator for the genome analysis pipeline**, including basecalling, read quality control, and read mapping steps
- ❑ **GenPIP** has two key techniques

- **Chunk-based Pipeline (CP)**

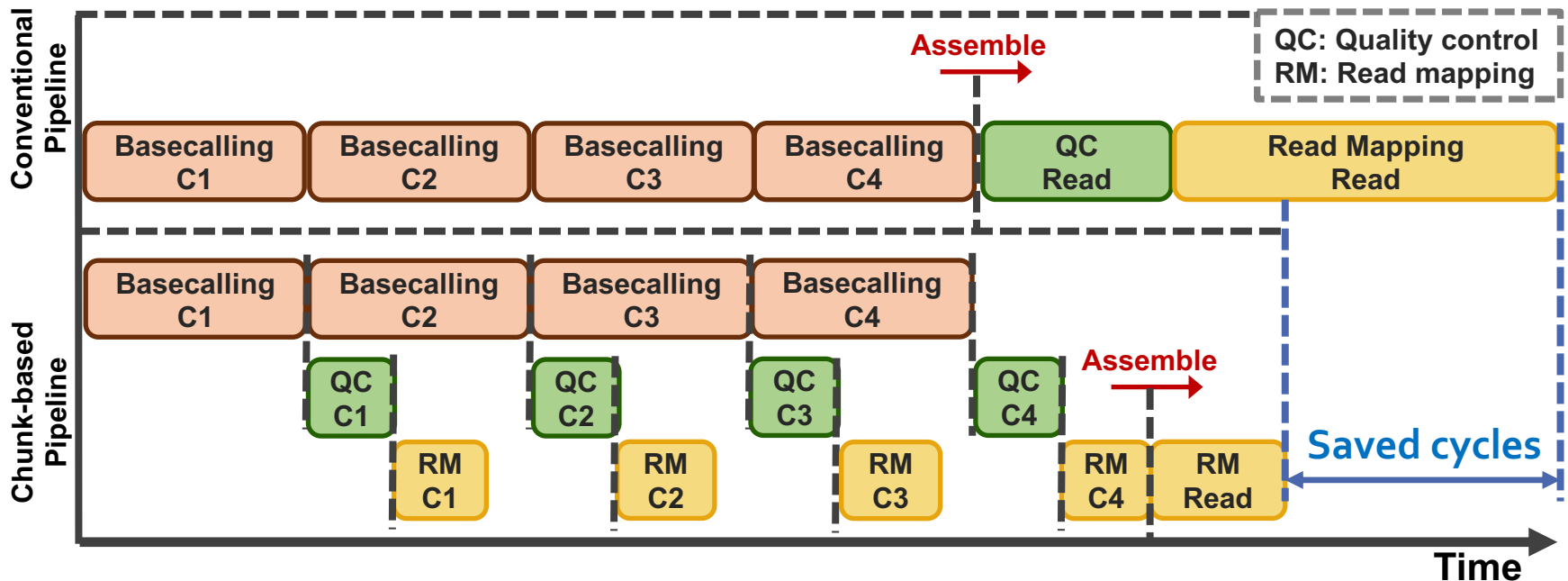
- **Enables fine-grained pipelining** of genome analysis steps
- Processes reads at **chunk** granularity (i.e., a subsequence; 300 bases)

- **Early Rejection (ER)**

# Chunk-based Pipeline (CP)

- ❑ CP **increases parallelism** by overlapping the execution of different steps at chunk granularity
- ❑ CP **reduces intermediate data** by computing on data as soon as data is generated
- ❑ CP **provides opportunities for ER** by analyzing a read at chunk granularity

A read consists of four chunks: **C1, C2, C3, C4**



# GenPIP

---

- ❑ **First holistic in-memory accelerator for the genome analysis pipeline**, including basecalling, read quality control, and read mapping steps
- ❑ **GenPIP** has two key techniques

- **Chunk-based Pipeline (CP)**

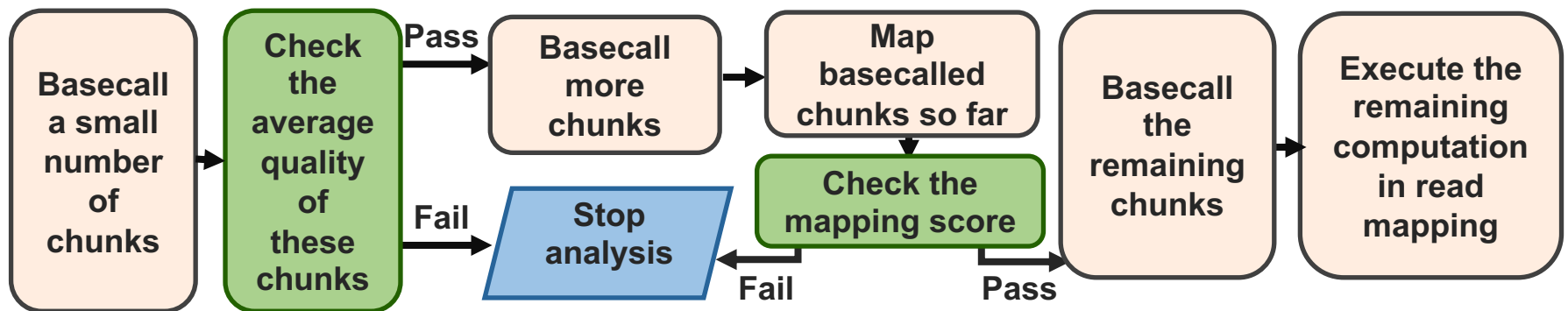
- **Enables fine-grained collaboration** of genome analysis steps by processing reads at chunk granularity (i.e., a subsequence of a read, e.g., 300 bases)

- **Early Rejection (ER)**

- **Stops the execution on useless reads as early as possible** by using a small number of chunks to predict the usefulness of a read

# Early Rejection (ER)

- ❑ **Predict and eliminate** low-quality and unmapped reads from the genome analysis pipeline **as early as possible**



- ❑ **Early-Rejection based on chunk quality scores (ER-QSR)**

- Predict **low-quality** reads using chunk quality scores

- ❑ **Early-Rejection based on chunk mapping scores (ER-CMR)**

- Predict **unmapped reads** using chunk mapping scores

# Implementation of CP and ER

---

CP and ER can be applied on different systems, e.g., CPU, GPU, and PIM

We implement CP and ER using PIM since PIM is more efficient to reduce the data movement between genome analysis steps

We also apply CP and ER on CPU and GPU baselines and observe speedup and energy savings



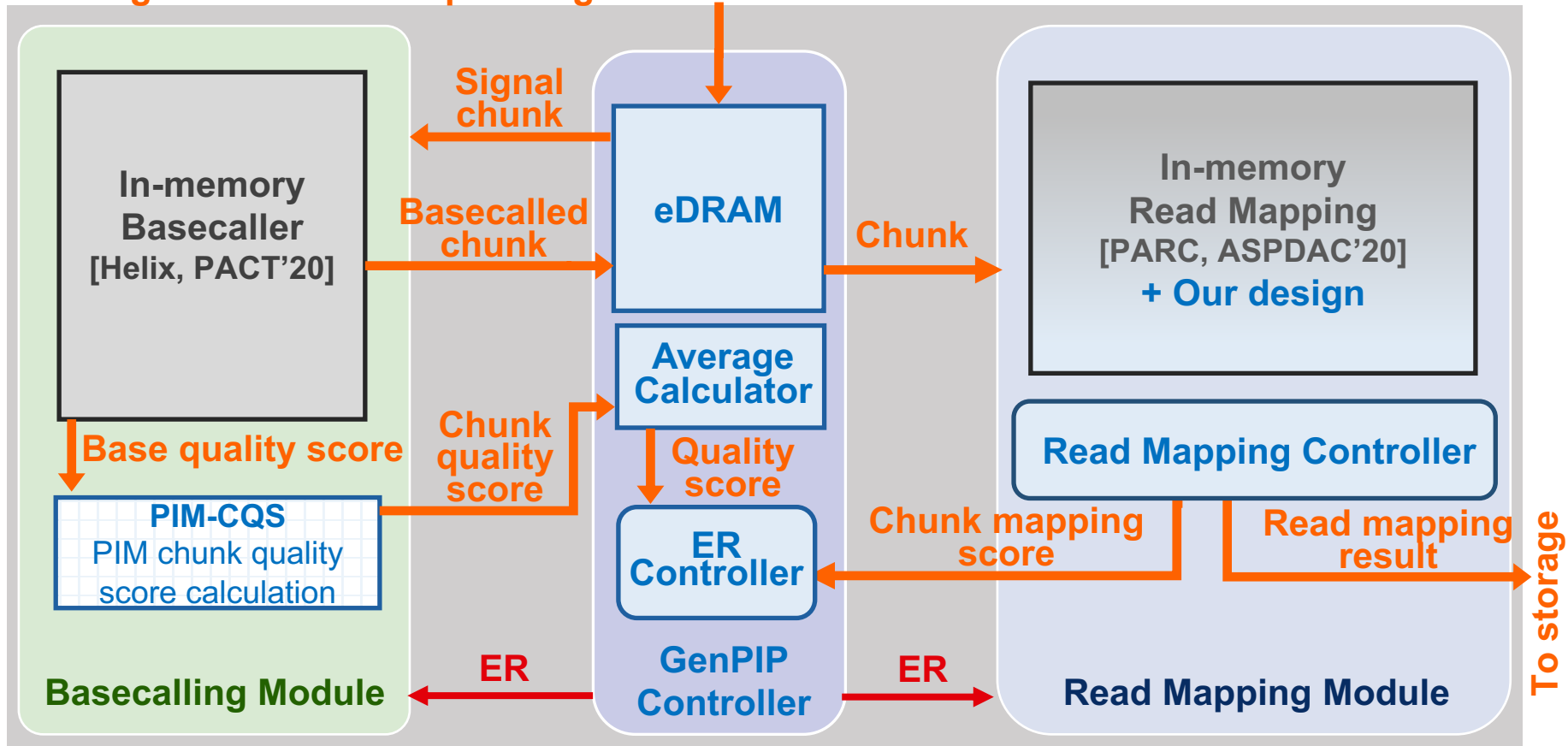
# Outline

---

- ❑ Background and Motivation
- ❑ GenPIP: Tight Integration of Genome Analysis Steps
  - Chunk-based Pipeline (CP)
  - Early Rejection (ER)
- ❑ **GenPIP Implementation**
- ❑ Evaluation
- ❑ Conclusion

# GenPIP Implementation

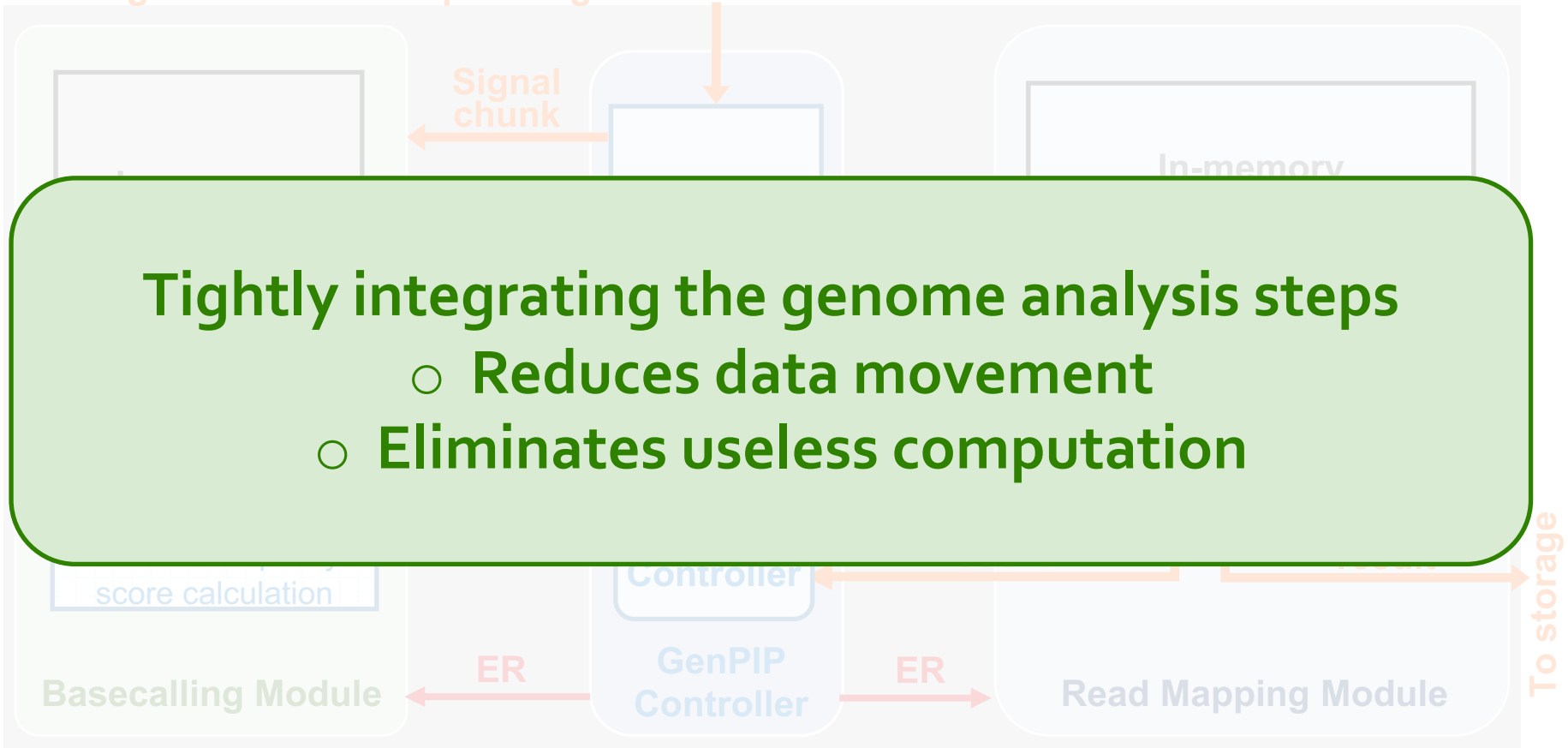
Raw signals from the sequencing machine



<https://arxiv.org/pdf/2209.08600.pdf>

# GenPIP Implementation

Raw signals from the sequencing machine



# Outline

---

- ❑ Background and Motivation
- ❑ GenPIP: Tight Integration of Genome Analysis Steps
  - Chunk-based Pipeline (CP)
  - Early Rejection (ER)
- ❑ GenPIP Implementation
- ❑ **Evaluation**
- ❑ Conclusion

# Evaluation Methodology

---

## □ Performance, Area and Power Analysis:

- Simulation via Verilog HDL, NVSim [TCAD'12], and CACTI 6.5 [MICRO'07]
- See methodology in the paper for more

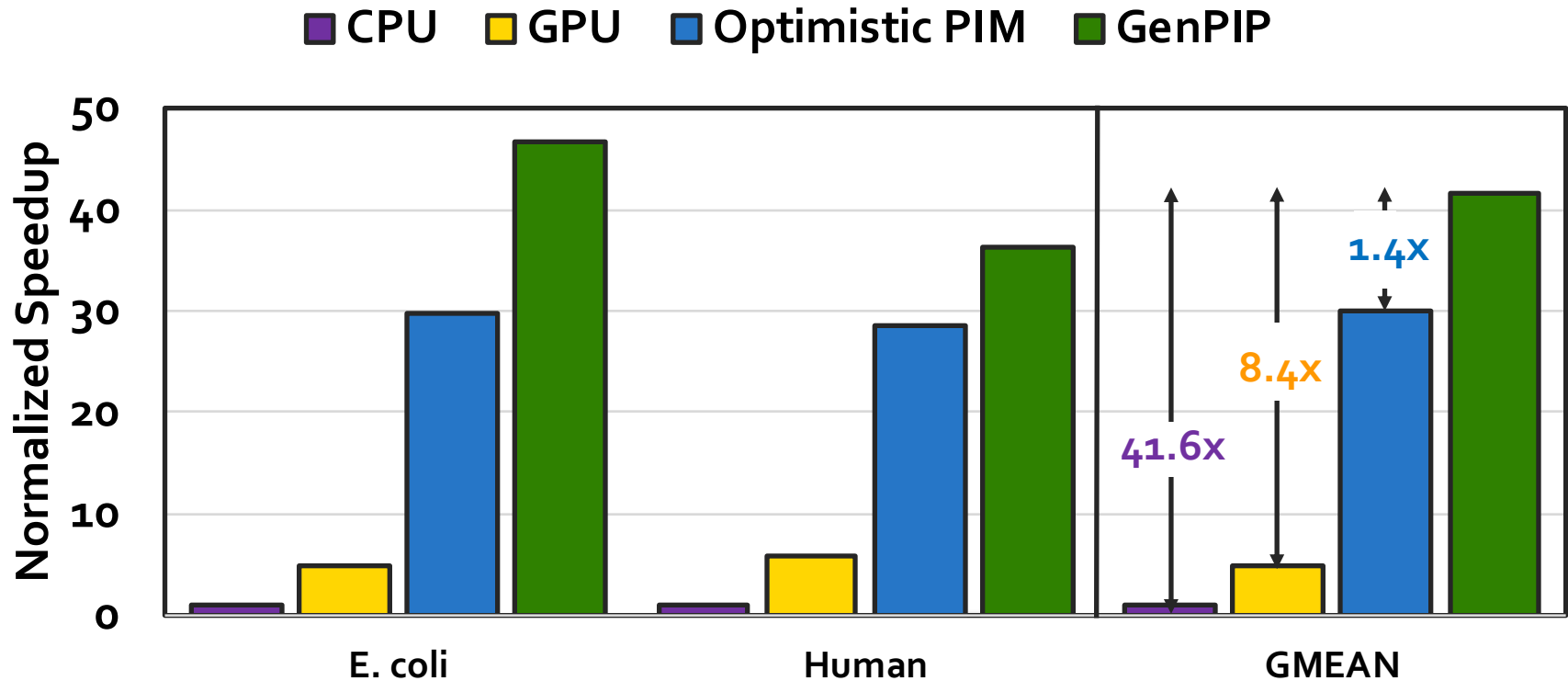
## □ Baselines:

- **CPU** (Intel Xeon Gold 5118 CPU)
- **GPU** (NVIDIA GeForce RTX 2080 Ti GPU)
- **Optimistic integration of two PIM accelerators** (Helix [PACT'20] and PARC [ASP-DAC'20])
  - Assumes **no data movement** between steps
  - Assumes intermediate data causes no overhead

## □ Datasets:

- **E. coli** (<http://lab.loman.net/2016/07/30/nano-pore-r9-data-release/>)
- **Human** (<https://www.ebi.ac.uk/ena/browser/view/PRJEB30620>)

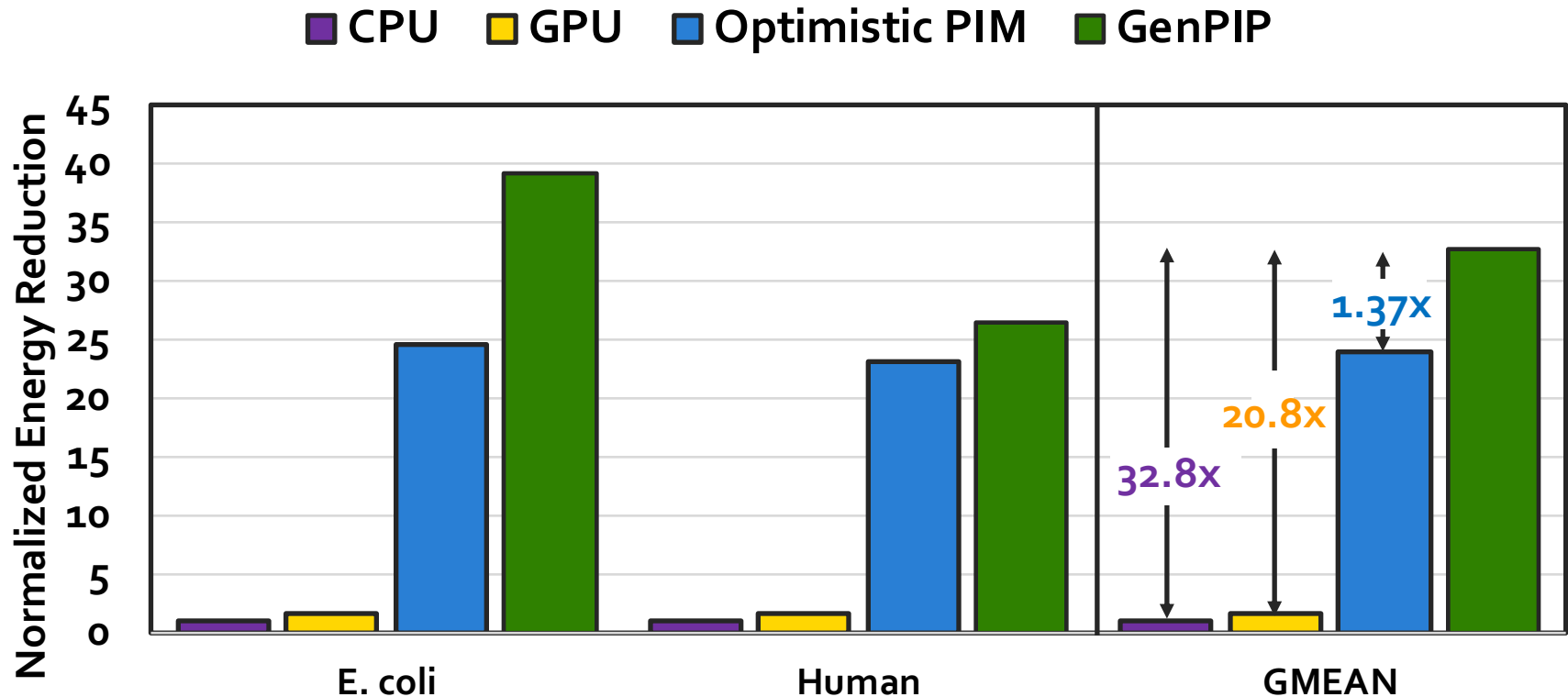
# Key Results – Performance



GenPIP provides 41.6x, 8.4x, and 1.4x speedup over CPU, GPU, and optimistic PIM

Both CP and ER are critical to the speedup

# Key Results – Energy Efficiency



GenPIP provides **32.8x**, **20.8x**, and **1.37x** energy savings over CPU, GPU, and optimistic PIM

**ER is especially critical** to the energy efficiency

# More in the Paper

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

Haiyu Mao<sup>1</sup> Mohammed Alser<sup>1</sup> Mohammad Sadrosadati<sup>1</sup> Can Firtina<sup>1</sup> Akanksha Baranwal<sup>1</sup>  
Damla Senol Cali<sup>2</sup> Aditya Manglik<sup>1</sup> Nour Almadhoun Alserr<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Bionano Genomics

<https://arxiv.org/pdf/2209.08600.pdf>





# More in the Paper

---

- ❑ Details of **CP and ER**
- ❑ Detailed **GenPIP** implementation
  - GenPIP controller
  - Early rejection implementation
  - In-memory seeding accelerator
- ❑ Results of **applying CP and ER in CPU and GPU**
- ❑ **Sensitivity analysis** on the number of sampled chunks used for ER
- ❑ **Area and power** analysis

# Outline

---

- ❑ Background and Motivation
- ❑ GenPIP: Tight Integration of Genome Analysis Steps
  - Chunk-based Pipeline (CP)
  - Early Rejection (ER)
- ❑ GenPIP Implementation
- ❑ Evaluation
- ❑ Conclusion

# Conclusion

- ❑ **Problem:** The genome analysis pipeline has **large data movement** between genome analysis steps and a significant amount of **wasted computation on useless data**
- ❑ **Goal:** **Tightly integrate genome analysis steps** to reduce the data movement between steps and eliminate computation on useless data
- ❑ **GenPIP:** The *first* in-memory genome analysis accelerator that **tightly integrates** genome analysis steps
- ❑ **GenPIP** has two key techniques
  - **A chunk-based pipeline**
  - **A new early-rejection technique**
- ❑ **GenPIP outperforms** state-of-the-art software & hardware solutions using CPU, GPU, and **optimistic PIM** by **41.6x**, **8.4x**, and **1.4x**, respectively.

# GenPIP

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

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

**SAFARI**

**ETH** zürich