Accelerating Genome Analysis
A Primer on an Ongoing Journey

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
14 April 2023
BIO-Arch Workshop @ RECOMB
Overview

- **System design for bioinformatics** is a critical problem
  - It has large scientific, medical, societal, personal implications

- This talk is about accelerating **a key step in bioinformatics**: genome sequence analysis
  - In particular, **read mapping**

- Many bottlenecks exist in accessing and manipulating huge amounts of genomic data during analysis

- Many **recent ideas to accelerate read mapping**
  - My personal journey since September 2006
An embedded device that can perform comprehensive genome analysis in real time (within a minute)

- Which of these DNAs does this DNA segment match with?
- What is the likely genetic disposition of this patient to this drug?
- What disease/condition might this particular DNA/RNA piece associated with?
- . . .
We Need Faster & Scalable Genome Analysis

Understanding **genetic variations, species, evolution, ...**

Predicting the **presence and relative abundance of microbes** in a sample

**SAFARI**

**SAFARI**

Rapid surveillance of **disease outbreaks**

Developing **personalized medicine**

And, many, many other applications ...
MinION from ONT

SmidgION from ONT

A Bright Future for Intelligent Genome Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396
Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu

"Accelerating Genome Analysis: A Primer on an Ongoing Journey"

Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gomez-Luna, Henk Corporaal, Onur Mutlu,

"FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications"

[Source Code]
A Few Overview Readings (III)

Mohammed Alser, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu

“From Molecules to Genomic Variations: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”
Computational and Structural Biotechnology Journal, 2022
[Source code]

Review

From molecules to genomic variations: Accelerating genome analysis via intelligent algorithms and architectures

Mohammed Alser *, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu *

ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland

A Few Overview Readings (IV)

- To appear at DAC 2023

Invited: Accelerating Genome Analysis via Algorithm-Architecture Co-Design

Onur Mutlu  Can Firtina

ETH Zürich
Algorithm-Arch-Device Co-Design is Critical

Computer Architecture (expanded view)

- Problem
- Algorithm
- Program/Language
- System Software
- SW/HW Interface
- Micro-architecture
- Logic
- Devices
- Electrons
Agenda

- The Problem: DNA Read Mapping
  - State-of-the-art Read Mapper Design

- Algorithmic Acceleration
  - Exploiting Structure of the Genome
  - Exploiting SIMD Instructions

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory & Storage

- Future Opportunities: New Technologies & Applications
What Is a Genome Made Of?

The chromosome is made up of genes

The genes consist of DNA

The discovery of DNA's double-helical structure (Watson+, 1953)
The Central Dogma of Molecular Biology

DNA Genotypes → Transcription → RNA → Translation → Protein Phenotypes

- Replication

COCOOKBOOK

13
DNA Under Electron Microscope

human chromosome #12 from HeLa’s cell
How Large is a Genome?

- Phi X174 virus: 5.386 Kilo bp
- E. coli O157:H7: 5.44 Million bp
- Homo Sapiens: 3.2 Billion bp
- Onion, Allium Cepa: 16 Billion bp
- Paris Japonica: 149 Billion bp
DNA Sequencing

Goal:
- Find the complete sequence of A, C, G, T’s in an organism’s DNA

Challenge:
- There is no machine that takes long DNA as an input, and gives the complete sequence as output
- All sequencing machines chop DNA into pieces and identify relatively small pieces (but not how they fit together)
Genome Sequencing

Large DNA molecule

Small DNA fragments

Reads:
- ACGTACCCCGT
- TTTTTTTAATT
- ACGAGCGGGGT
- GATACACTGTG
- AAAAAAAAAAA
- CTAGGGACCTT
- ACGACGTAGCT

SAFARI
Current sequencing machines provide small randomized fragments of the original DNA sequence.

Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021
Untangling Yarn Balls & DNA Sequencing
Genome Sequencers

... and more! All produce data with different properties.

- Roche/454
- AB SOLiD
- Illumina MiSeq
- Complete Genomics
- Oxford Nanopore MinION
- Oxford Nanopore GridION
- Illumina NovaSeq 6000
- Illumina HiSeq2000
- Pacific Biosciences RS
- Illumina MiSeq
- SAFARI
- Ion Torrent PGM
- Ion Torrent Proton
High-Throughput Sequencers

... and more! All produce data with different properties.
The Genomic Era

The development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

Genome Sequencing Cost Is Reducing

Cost per Raw Megabase of DNA Sequence

Moore’s Law

*From NIH (https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data)
Genome Sequencing Cost Is Reducing

*From NIH ([https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data](https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data))
Solving the Puzzle

Reference genome

Sequenced Reads

.FASTA file

Reference genome

Sequenced Reads

.FASTQ file

Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

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

*Briefings in Bioinformatics*, bby017, https://doi.org/10.1093/bib/bby017

Published: 02 April 2018  Article history ▼


[Open arxiv.org version] [Slides (pptx) (pdf)] [Talk Video at AACBB 2019]
Types of Genomic Reads

Wenger+, "Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome", Nature Biotechnology, 2019

https://labs.wsu.edu/genomicscore/illumina-sequencing/
https://pacbio.gs.washington.edu/
1 Sequencing

2 Read Mapping

3 Variant Calling

4 Scientific Discovery

Genome Analysis

reference: TTTATCGCTTCCATGACGCAG
read1: ATCGCATCC
read2: TATCGCATC
read3: CATCCATGA
read4: CGCTTCCAT
read5: CCATGACGC
read6: TTCCATGAC
Read Mapping Techniques in 111 Pages

In-depth analysis of 107 read mappers (1988-2020)

Mohammed Alser, Jeremy Rotman, Dhrithi Deshpande, Kodi Taraszka, Huwenbo Shi, Pelin Icer Baykal, Harry Taegyun Yang, Victor Xue, Sergey Knyazev, Benjamin D. Singer, Brunilda Balliu, David Koslicki, Pavel Skums, Alex Zelikovsky, Can Alkan, Onur Mutlu, Serghei Mangul

"Technology dictates algorithms: Recent developments in read alignment"
Genome Biology, 2021
[Source code]
One Problem
We Are Bottlenecked in Read Mapping

Illumina HiSeq4000

300 M bases/min

on average

2 M bases/min (0.6%)
The Read Mapping Bottleneck

300 Million bases/minute
Read Sequencing**

↓↓

150x slower

2 Million bases/minute
Read Mapping*

* BWA-MEM
** HiSeqX10, MinION
The Read Mapping Bottleneck

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

Illumina NovaSeq 6000

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

Problems with (Genome) Analysis Today

Special-Purpose Machine for Data Generation

General-Purpose Machine for Data Analysis

FAST

SLOW

Slow and inefficient processing capability
Large amounts of data movement

SAFARI This picture is similar for many “data generators & analyzers” today
One Problem

Need to construct the entire genome from many sequenced reads
Read Mapping

- Map many short DNA fragments (reads) to a known reference genome with some differences allowed.

Mapping short reads to reference genome is challenging (billions of 50-300 base pair reads)
Read Mapping for Metagenomic Analysis

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

Genetic material recovered directly from environmental samples

Reads in "text format"

Reference Database
Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false)

Arturs Backurs, Piotr Indyk

The edit distance (a.k.a. the Levenshtein distance) between two strings is defined as the minimum number of insertions, deletions or substitutions of symbols needed to transform one string into another. The problem of computing the edit distance between two strings is a classical computational task, with a well-known algorithm based on dynamic programming. Unfortunately, all known algorithms for this problem run in nearly quadratic time.

In this paper we provide evidence that the near-quadratic running time bounds known for the problem of computing edit distance might be tight. Specifically, we show that, if the edit distance can be computed in time $O(n^{2-\delta})$ for some constant $\delta > 0$, then the satisfiability of conjunctive normal form formulas with $N$ variables and $M$ clauses can be solved in time $M^{O(1)2^{(1-\epsilon)N}}$ for a constant $\epsilon > 0$. The latter result would violate the Strong Exponential Time Hypothesis, which postulates that such algorithms do not exist.

https://arxiv.org/abs/1412.0348
Read Mapping Techniques in 111 Pages

In-depth analysis of 107 read mappers (1988-2020)

Mohammed Alser, Jeremy Rotman, Dhrithi Deshpande, Kodi Taraszka, Huwenbo Shi, Pelin Icer Baykal, Harry Taegyun Yang, Victor Xue, Sergey Knyazev, Benjamin D. Singer, Brunilda Balliu, David Koslicki, Pavel Skums, Alex Zelikovsky, Can Alkan, Onur Mutlu, Serghei Mangul

"Technology dictates algorithms: Recent developments in read alignment"

Genome Biology, 2021

[Source code]
Read Mapping Execution Time (Modern)

>60% of the read mapper’s execution time is spent in sequence alignment

ONT FASTQ size: 103MB (151 reads), Mean length: 356,403 bp, std: 173,168 bp, longest length: 817,917 bp
Accelerating Read Mapping

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu

“Accelerating Genome Analysis: A Primer on an Ongoing Journey”
Genomics Course (Fall 2022)

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=bioinformatics

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics

- **Youtube Livestream (Fall 2022):**
  - https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDqOR1yRU_Cxxjw-u18

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

https://www.youtube.com/onurmutlulectures
Agenda

- The Problem: DNA Read Mapping
  - State-of-the-art Read Mapper Design

- Algorithmic Acceleration
  - Exploiting Structure of the Genome
  - Exploiting SIMD Instructions

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory & Storage

- Future Opportunities: New Technologies & Applications
GateKeeper: FPGA-Based Alignment Filtering

Mohammed Alser, Hasan Hassan, Hongyi Xin, Oguz Ergin, Onur Mutlu, and Can Alkan

"GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping"

Bioinformatics, [published online, May 31], 2017.

Source Code

Online link at Bioinformatics Journal

GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping

Mohammed Alser, Hasan Hassan, Hongyi Xin, Oğuz Ergin, Onur Mutlu, Can Alkan

Bioinformatics, Volume 33, Issue 21, 1 November 2017, Pages 3355–3363,
https://doi.org/10.1093/bioinformatics/btx342

Published: 31 May 2017  Article history ▼
GateKeeper Accelerator Architecture

- **Maximum data throughput** \(\approx 13.3 \text{ billion bases/sec}\)
- Can examine **8 (300 bp) or 16 (100 bp) mappings concurrently** at 250 MHz
- **Occupies 50\%** (100 bp) to **91\%** (300 bp) of the FPGA slice LUTs and registers

![GateKeeper Architecture Diagram](image-url)
FPGA Chip Layout

GateKeeper: 17.6%, PCIe Controller, RIFFA, and IO: 5%

Read length: 300 bp
Error threshold: E = 15
GateKeeper: Speed & Accuracy Results

90x-130x faster filter
than SHD (Xin et al., 2015) and the Adjacency Filter (Xin et al., 2013)

4x lower false accept rate
than the Adjacency Filter (Xin et al., 2013)

10x speedup in read mapping
with the addition of GateKeeper to the mrFAST mapper (Alkan et al., 2009)

Freely available online
github.com/BilkentCompGen/GateKeeper
GateKeeper Conclusions

- FPGA-based pre-alignment greatly speeds up read mapping
  - 10x speedup of a state-of-the-art mapper (mrFAST)

- FPGA-based pre-alignment can be integrated with the sequencer
  - It can help to hide the complexity and details of the FPGA
  - Enables real-time filtering while sequencing
  - Paves the way to on-device genome analysis
More on GateKeeper

- Mohammed Alser, Hasan Hassan, Hongyi Xin, Oguz Ergin, Onur Mutlu, and Can Alkan

"GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping"

*Bioinformatics*, [published online, May 31], 2017.

[Source Code]

[Online link at Bioinformatics Journal]

---

GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping

Mohammed Alser, Hasan Hassan, Hongyi Xin, Oğuz Ergin, Onur Mutlu, and Can Alkan

*Bioinformatics*, Volume 33, Issue 21, 1 November 2017, Pages 3355–3363,

https://doi.org/10.1093/bioinformatics/btx342

**Published:** 31 May 2017

**Article history** ▼
Algorithm-Arch-Device Co-Design is Critical

Computer Architecture (expanded view)

- Problem
- Algorithm
- Program/Language
- System Software
- SW/HW Interface
- Micro-architecture
- Logic
- Devices
- Electrons
Mohammed Alser, Hasan Hassan, Akash Kumar, Onur Mutlu, and Can Alkan,
"Shouji: A Fast and Efficient Pre-Alignment Filter for Sequence Alignment"

*Source Code*

[Online link at Bioinformatics Journal]
Counting is performed concurrently for all bit-vectors and all sliding windows in a single clock cycle using multiple 4-input LUTs.
Mohammed Alser, Taha Shahroodi, Juan-Gomez Luna, Can Alkan, and Onur Mutlu, "SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs" *Bioinformatics*, to appear in 2020.

[Source Code]
[Online link at Bioinformatics Journal]

Subject Section

**SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs**

Mohammed Alser¹,²,*, Taha Shahroodi¹, Juan Gómez-Luna¹,², Can Alkan⁴,*, and Onur Mutlu¹,²,³,⁴,*

¹ Department of Computer Science, ETH Zurich, Zurich 8006, Switzerland
² Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland
³ Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh 15213, PA, USA
⁴ Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey
Key observation:
- Correct alignment is a sequence of non-overlapping long matches

Dot plot, dot matrix (Lipman and Pearson, 1985)
SneakySnake

- **Key observation:**
  - Correct alignment is a sequence of non-overlapping long matches

- **Key idea:**
  - Reduce the approximate string matching problem to the Single Net Routing problem in VLSI chip layout
SneakySnake

- **Key observation:**
  - Correct alignment is a sequence of non-overlapping long matches

- **Key idea:**
  - Reduce the approximate string matching problem to the Single Net Routing problem in VLSI chip layout

- **Key result:**
  - SneakySnake is up to four orders of magnitude more accurate than Shouji (Bioinformatics’19) and GateKeeper (Bioinformatics’17)
  - SneakySnake greatly accelerates state-of-the-art CPU sequence aligners, Edlib (Bioinformatics’17) and Parasail (BMC Bioinformatics’16)
    - by up to 37.7× and 43.9× (>12× on average), on CPUs
    - by up to 413× and 689× (>400× on average) with FPGAs/GPUs
More on SneakySnake [Alser+, Bioinformatics 2020]

Mohammed Alser, Taha Shahroodi, Juan-Gomez Luna, Can Alkan, and Onur Mutlu,
"SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs"


[Source Code]
[Online link at Bioinformatics Journal]

Subject Section

SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs

Mohammed Alser¹,²,*, Taha Shahroodi¹, Juan Gómez-Luna¹,², Can Alkan⁴,*, and Onur Mutlu¹,²,³,⁴,*

¹Department of Computer Science, ETH Zurich, Zurich 8006, Switzerland
²Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland
³Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh 15213, PA, USA
⁴Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey
GenASM Framework [MICRO 2020]


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

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

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

†Carnegie Mellon University  ‡Processor Architecture Research Lab, Intel Labs  †Bilkent University  †‡ETH Zürich  †Facebook  ‡King Mongkut’s University of Technology North Bangkok  *University of Illinois at Urbana–Champaign
Our Goal:

Accelerate approximate string matching by designing a fast and flexible framework, which can accelerate *multiple steps of genome sequence analysis*.

- **GenASM**: First ASM acceleration framework for GSA
  - Based on the *Bitap* algorithm
    - Uses *fast and simple bitwise operations* to perform ASM
  - Modified and extended ASM algorithm
    - *Highly-parallel Bitap* with long read support
    - *Bitvector-based* novel algorithm to perform *traceback*
  - Co-design of our modified scalable and memory-efficient algorithms with *low-power and area-efficient hardware accelerators*
GenASM: Hardware Design

GenASM-DC: generates bitvectors and performs edit distance calculation.

GenASM-TB: performs TraceBack and assembles the optimal alignment.
Linear cyclic systolic array based accelerator

- Designed to maximize parallelism and minimize memory bandwidth and memory footprint
GenASM-TB: Hardware Design

Very simple logic:

1. Reads the bitvectors from one of the TB-SRAMs using the computed address
2. Performs the required bitwise comparisons to find the traceback output for the current position
3. Computes the next TB-SRAM address to read the new set of bitvectors
Key Results – Area and Power

- Based on our *synthesis* of GenASM-DC and GenASM-TB accelerator datapaths using the Synopsys Design Compiler with a **28nm** LP process:
  - Both GenASM-DC and GenASM-TB operate **@ 1GHz**

![Area and Power Pie Charts](image.png)

<table>
<thead>
<tr>
<th>Component</th>
<th>Area (mm²)</th>
<th>Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenASM-DC (64 PEs)</td>
<td>0.049</td>
<td>0.033</td>
</tr>
<tr>
<td>GenASM-TB</td>
<td>0.016</td>
<td>0.004</td>
</tr>
<tr>
<td>DC-SRAM (8 KB)</td>
<td>0.013</td>
<td>0.009</td>
</tr>
<tr>
<td>TB-SRAMs (64 x 1.5 KB)</td>
<td></td>
<td>0.055</td>
</tr>
</tbody>
</table>

| Total (1 vault)                  | 0.334 mm²  | 0.101 W   |
| Total (32 vaults)                | 10.69 mm²  | 3.23 W    |
| % of a Xeon CPU core:            | 1%         | 1%        |
Key Results – Area and Power

- Based on our synthesis of GenASM-DC and GenASM-TB accelerator datapaths using the Synopsys Design Compiler with a 28nm LP process:
  - Both GenASM-DC and GenASM-TB operate @ 1GHz

<table>
<thead>
<tr>
<th></th>
<th>Area (mm²^2)</th>
<th>Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenASM-DC (64 PEs)</td>
<td>0.049</td>
<td>0.033</td>
</tr>
<tr>
<td>GenASM-TB</td>
<td>0.016</td>
<td>0.004</td>
</tr>
<tr>
<td>DC-SRAM (8 KB)</td>
<td>0.013</td>
<td>0.009</td>
</tr>
<tr>
<td>TB-SRAMs (64 x 1.5 KB)</td>
<td>0.256</td>
<td>0.055</td>
</tr>
</tbody>
</table>

GenASM has low area and power overheads
Use Cases of GenASM

Reference genome

Indexing

Hash table based index

Seeding

Candidate mapping locations

Pre-Alignment Filtering

Remaining candidate mapping locations

Read Alignment

Optimal alignment

Reads from sequenced genome
Use Cases of GenASM (cont’d.)

(1) Read Alignment Step of Read Mapping
   - Find the optimal alignment of how reads map to candidate reference regions

(2) Pre-Alignment Filtering for Short Reads
   - Quickly identify and filter out the unlikely candidate reference regions for each read

(3) Edit Distance Calculation
   - Measure the similarity or distance between two sequences

We also discuss other possible use cases of GenASM in our paper:
   - Read-to-read overlap finding, hash-table based indexing, whole genome alignment, generic text search
### Key Results

#### (1) Read Alignment
- 116× speedup, 37× less power than **Minimap2** (state-of-the-art SW)
- 111× speedup, 33× less power than **BWA-MEM** (state-of-the-art SW)
- 3.9× better throughput, 2.7× less power than **Darwin** (state-of-the-art HW)
- 1.9× better throughput, 82% less logic power than **GenAx** (state-of-the-art HW)

#### (2) Pre-Alignment Filtering
- 3.7× speedup, 1.7× less power than **Shouji** (state-of-the-art HW)

#### (3) Edit Distance Calculation
- 22–12501× speedup, 548–582× less power than **Edlib** (state-of-the-art SW)
- 9.3–400× speedup, 67× less power than **ASAP** (state-of-the-art HW)
More on GenASM Framework [MICRO 2020]

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

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

Damla Senol Cali†‡ Carnegie Mellon University  ‡ Processor Architecture Research Lab, Intel Labs  ▽ Bilkent University  ◇ ETH Zürich
Gurpreet S. Kalsi‡ Processor Architecture Research Lab, Intel Labs  Zülal Bingöl▼ Bilkent University  Can Firtina◊ King Mongkut’s University of Technology North Bangkok
Lavanya Subramanian‡ Carnegie Mellon University  Jeremie S. Kim†‡ Carnegie Mellon University
Rachata Ausavarungnirun○ Bilkent University  Mohammed Alser◊ King Mongkut’s University of Technology North Bangkok  Juan Gomez-Luna◊ University of Illinois at Urbana–Champaign
Amirali Boroumand† Bilkent University  Anant Nori‡ Bilkent University  Allison Scibisz†′ King Mongkut’s University of Technology North Bangkok  Sreenivas Subramoney◊ King Mongkut’s University of Technology North Bangkok  Can Alkan▼ Bilkent University  Saugata Ghose*†* University of Illinois at Urbana–Champaign  Onur Mutlu†▽

Safari
Scrooge: Overcoming GenASM Limitations

Joël Lindegger, Damla Senol Cali, Mohammed Alser, Juan Gómez-Luna, Nika Mansouri Ghiasi, and Onur Mutlu,
"Scrooge: A Fast and Memory-Frugal Genomic Sequence Aligner for CPUs, GPUs, and ASICs"
[Online link at Bioinformatics Journal]
[arXiv preprint]
[Scrooge Source Code]

Scrooge: A Fast and Memory-Frugal Genomic Sequence Aligner for CPUs, GPUs, and ASICs

Joël Lindegger§
Juan Gómez-Luna§

Damla Senol Cali†
Nika Mansouri Ghiasi§

Mohammed Alser§
Onur Mutlu§

§ETH Zürich  †Bionano Genomics

Accelerating Sequence-to-Graph Mapping

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika Mansouri-Ghiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,

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

[arXiv version]

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

Damla Senol Cali\textsuperscript{1} Konstantinos Kanellopoulos\textsuperscript{2} Joël Lindegger\textsuperscript{2} Zülal Bingöl\textsuperscript{3}
Gurpreet S. Kalsi\textsuperscript{4} Ziyi Zuo\textsuperscript{5} Can Firtina\textsuperscript{2} Meryem Banu Cavlak\textsuperscript{2} Jeremie Kim\textsuperscript{2}
Nika Mansouri Ghiasi\textsuperscript{2} Gagandeep Singh\textsuperscript{2} Juan Gómez-Luna\textsuperscript{2} Nour Almadhoun Alserr\textsuperscript{2}
Mohammed Alser\textsuperscript{2} Sreenivas Subramoney\textsuperscript{4} Can Alkan\textsuperscript{3} Saugata Ghose\textsuperscript{6} Onur Mutlu\textsuperscript{2}

\textsuperscript{1}Bionano Genomics \hspace{1em} \textsuperscript{2}ETH Zürich \hspace{1em} \textsuperscript{3}Bilkent University \hspace{1em} \textsuperscript{4}Intel Labs
\textsuperscript{5}Carnegie Mellon University \hspace{1em} \textsuperscript{6}University of Illinois Urbana-Champaign

SAFARI \hspace{1em} \url{https://arxiv.org/pdf/2205.05883.pdf}
Mapping the reads to a reference genome (i.e., *read mapping*) is a critical step in genome sequence analysis.

**Linear Reference:** ACGTACGT

**Read:** ACGG

**Alternative Sequence:** ACGGACGT
**Alternative Sequence:** ACGTTACGT
**Alternative Sequence:** ACG–ACGT

**Sequence-to-Sequence (S2S) Mapping**

**Graph-based Reference:**

**Read:** ACGG

**Sequence-to-Graph (S2G) Mapping**

*Sequence-to-graph mapping* results in notable quality improvements. However, it is a more difficult computational problem, with no prior hardware design.
SeGraM: First Graph Mapping Accelerator

Our Goal:

Specialized, high-performance, scalable, and low-cost algorithm/hardware co-design that alleviates bottlenecks in multiple steps of sequence-to-graph mapping

SeGraM: First universal algorithm/hardware co-designed genomic mapping accelerator that can effectively and efficiently support:

- Sequence-to-graph mapping
- Sequence-to-sequence mapping
- Both short and long reads
Sequence-to-Graph Mapping Pipeline

**Pre-Processing Steps (Offline)**

1. **Genome Graph Construction**
   - (construct the graph using a linear reference genome and variations)

2. **Indexing**
   - (index the nodes of the graph)

**Seed-and-Extend Steps (Online)**

1. **Seeding**
   - (query the index & find the seed matches)

2. **Filtering/Chaining/Clustering**
   - (filter out dissimilar query read and subgraph pairs)

3. **S2G Alignment**
   - (perform distance/score calculation & traceback)

**Hash-table-based index (of graph nodes)**

**Genome Graph Construction**

- Linear reference genome
- Known genetic variations

**Indexing**

**Seeding**

**Filtering/Chaining/Clustering**

**S2G Alignment**

**Optimal alignment between read & subgraph**

Reads from sequenced genome

Genome graph

Candidate mapping locations (subgraphs)

Remaining candidate mapping locations (subgraphs)

Genome graph

Hash-table-based index (of graph nodes)
SeGraM Hardware Design

MinSeed: first hardware accelerator for Minimizer-based Seeding

BitAlign: first hardware accelerator for (Bit)vector-based sequence-to-graph Alignment

**MinSeed Scratchpad**
- Find Minimizers
- Filter Minimizers by Frequency
- Find Candidate Seed Regions

**Seed Scratchpad**

**Input Scratchpad**
- Generate Bitvectors
- Hop Queues

**Bitvector Scratchpad**
- Perform Traceback

**Main Memory (graph-based reference & index)**
- Frequencies
- Seed locations
- Graph nodes

**Graph Nodes**
- 3
- 5
- 7

**Query K-mers**
- 1

**Optimal Alignment Information**
- 12
Use Cases & Key Results

(1) Sequence-to-Graph (S2G) Mapping
- 5.9×/106× speedup, 4.1×/3.0× less power than GraphAligner for long and short reads, respectively (state-of-the-art SW)
- 3.9×/742× speedup, 4.4×/3.2× less power than vg for long and short reads, respectively (state-of-the-art SW)

(2) Sequence-to-Graph (S2G) Alignment
- 41×–539× speedup over PaSGAL with AVX-512 support (state-of-the-art SW)

(3) Sequence-to-Sequence (S2S) Alignment
- 1.2×/4.8× higher throughput than GenASM and GACT of Darwin for long reads (state-of-the-art HW)
- 1.3×/2.4× higher throughput than GenASM and SillaX of GenAX for short reads (state-of-the-art HW)
SeGraM: A Universal HW Accelerator for Genomic Sequence-to-Graph Mapping - Damla Senol Cali (ISCA)
Accelerating Sequence-to-Graph Mapping

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindeger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika MansouriGhiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,

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


[arXiv version]

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

Damla Senol Cali\textsuperscript{1} Konstantinos Kanellopoulos\textsuperscript{2} Joël Lindeger\textsuperscript{2} Züal Bingöl\textsuperscript{3} Gurpreet S. Kalsi\textsuperscript{4} Ziyi Zuo\textsuperscript{5} Can Firtina\textsuperscript{2} Meryem Banu Cavlak\textsuperscript{2} Jeremie Kim\textsuperscript{2} Nika Mansouri Ghiasi\textsuperscript{2} Gagandeep Singh\textsuperscript{2} Juan Gómez-Luna\textsuperscript{2} Nour Almadhoun Alserr\textsuperscript{2} Mohammed Alser\textsuperscript{2} Sreenivas Subramoney\textsuperscript{4} Can Alkan\textsuperscript{3} Saugata Ghose\textsuperscript{6} Onur Mutlu\textsuperscript{2}

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

A Framework for Designing Efficient Deep Learning-Based Genomic Basecallers

Gagandeep Singh\textsuperscript{a}  Mohammed Alser\textsuperscript{*a}  Alireza Khodamoradi\textsuperscript{*b}
Kristof Denolf\textsuperscript{b}  Can Firtina\textsuperscript{a}  Meryem Banu Cavlak\textsuperscript{a}
Henk Corporaal\textsuperscript{c}  Onur Mutlu\textsuperscript{a}
\textsuperscript{a}ETH Zürich  \textsuperscript{b}AMD  \textsuperscript{c}Eindhoven University of Technology

Nanopore sequencing is a widely-used high-throughput genome sequencing technology that can sequence long fragments of a genome. Nanopore sequencing generates noisy electrical signals that need to be converted into a standard string of DNA nucleotide bases (i.e., A, C, G, T) using a computational step called basecalling. The accuracy and speed of basecalling have critical implications for every subsequent step in genome analysis. Currently, basecallers are developed mainly based on deep learning techniques to provide high sequencing accuracy without considering the compute demands of such tools. We observe that state-of-the-art basecallers (i.e., Guppy, Bonito, Fast-Bonito) are slow, inefficient, and memory-hungry
Agenda

- The Problem: DNA Read Mapping
  - State-of-the-art Read Mapper Design

- Algorithmic Acceleration
  - Exploiting Structure of the Genome
  - Exploiting SIMD Instructions

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory & Storage

- Future Opportunities: New Technologies & Applications
Process Data Where It Makes Sense

Apple M1 Ultra System (2022)

https://www.gsmarena.com/apple_announces_m1.Ultra_with_20core_cpu_and_64core_gpu-news-53481.php
Goal: Processing Inside Memory

Many questions ... How do we design the:
- compute-capable memory & controllers?
- processors & communication units?
- software & hardware interfaces?
- system software, compilers, languages?
- algorithms & theoretical foundations?
We need to design mapping & filtering algorithms that fit processing-in-memory
Near-Memory Pre-Alignment Filtering

Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gomez-Luna, Henk Corporaal, Onur Mutlu,

"FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications"
[Source Code]
Near-Memory SneakySnake

- Problem: Read mapping is heavily bottlenecked by data movement from main memory

- Solution: Perform read mapping near where data resides using specialized logic

- We carefully redesign the accelerator logic of SneakySnake to exploit near-memory computation capability on real FPGA boards that use HBM (high-bandwidth memory)

- Near-memory SneakySnake improves performance and energy efficiency by 27.4× and 133×, respectively, over a 16-core (64-thread) IBM POWER9 CPU
Near-Memory Acceleration using FPGAs

IBM POWER9 CPU

HBM-based FPGA board

Near-HBM FPGA-based accelerator

Two communication technologies: CAPI2 and OCAPI
Two memory technologies: DDR4 and HBM
Two workloads: Weather Modeling and Genome Analysis
Performance & Energy Greatly Improve

5-27× performance vs. a 16-core (64-thread) IBM POWER9 CPU

12-133× energy efficiency vs. a 16-core (64-thread) IBM POWER9 CPU

HBM alleviates memory bandwidth contention vs. DDR4
More On Near-Memory SneakySnake

Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gomez-Luna, Henk Corporaal, Onur Mutlu,

"FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications"
[Source Code]
Location Filtering in 3D-Stacked PIM

- Jeremie S. Kim, Damla Senol Cali, Hongyi Xin, Donghyuk Lee, Saugata Ghose, Mohammed Alser, Hasan Hassan, Oguz Ergin, Can Alkan, and Onur Mutlu,

"GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies"

*BMC Genomics*, 2018.


[Slides (pptx) (pdf)]
[Source Code]
[arxiv.org Version (pdf)]
[Talk Video at AACBB 2019]
Opportunity: 3D-Stacked Logic+Memory

Other “True 3D” technologies under development
In-Storage Genome Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,
  "GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"
  [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video (90 seconds)]

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

Nika Mansouri Ghiasi\textsuperscript{1} Jisung Park\textsuperscript{1} Harun Mustafa\textsuperscript{1} Jeremie Kim\textsuperscript{1} Ataberk Olgun\textsuperscript{1} Arvid Gollwitzer\textsuperscript{1} Damla Senol Cali\textsuperscript{2} Can Firtina\textsuperscript{1} Haiyu Mao\textsuperscript{1} Nour Almadhoun Alserr\textsuperscript{1} Rachata Ausavarungnirun\textsuperscript{3} Nandita Vijaykumar\textsuperscript{4} Mohammed Alser\textsuperscript{1} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \textsuperscript{2}Bionano Genomics \textsuperscript{3}KMUTNB \textsuperscript{4}University of Toronto
Genome Sequence Analysis

Data Movement from Storage

Storage System

Main Memory

Cache

Alignment

Computation Unit (CPU or Accelerator)

Computation overhead

Data movement overhead
Accelerating Genome Sequence Analysis

Storage System

Heuristics

Accelerators

Filters

Main Memory

Cache

Computation Unit (CPU or Accelerator)

✓ Computation overhead

✗ Data movement overhead
Key Idea

Filter reads that do not require alignment inside the storage system

Filtered Reads

Exactly-matching reads
Do not need expensive approximate string matching during alignment

Non-matching reads
Do not have potential matching locations and can skip alignment
Filtering Opportunities

- Sequencing machines produce one of two kinds of reads:
  - **Short reads**: highly accurate and short
  - **Long reads**: less accurate and long

**Reads that do not require the expensive alignment step:**

**Exactly-matching reads**
- Do not need expensive approximate string matching during alignment
  - Low sequencing error rates (**short reads**) combined with
  - Low genetic variation

**Non-matching reads**
- Do not have potential matching locations, so they skip alignment
  - High sequencing error rates (**long reads**) or
  - High genetic variation (**short or long reads**)
Challenges

Filter reads that do not require alignment inside the storage system

Read mapping workloads can exhibit different behavior

There are limited hardware resources in the storage system
GenStore

Filter reads that do not require alignment inside the storage system

GenStore-Enabled Storage System

Main Memory

Cache

Computation Unit (CPU or Accelerator)

✓ Computation overhead

✓ Data movement overhead

GenStore provides significant speedup (1.4x - 33.6x) and energy reduction (3.9x – 29.2x) at low cost
In-Storage Genome Filtering [ASPLOS 2022]

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

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


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

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

Nika Mansouri Ghiasi¹ Jisung Park¹ Harun Mustafa¹ Jeremie Kim¹ Ataberk Olgun¹ Arvid Gollwitzer¹ Damla Senol Cali² Can Firtina¹ Haiyu Mao¹ Nour Almadhoun Alserr¹ Rachata Ausavarungnirun³ Nandita Vijaykumar⁴ Mohammed Alser¹ Onur Mutlu¹

¹ETH Zürich ²Bionano Genomics ³KMUTNB ⁴University of Toronto
Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun, "A Modern Primer on Processing in Memory"

A Workload and Programming Ease Driven Perspective of Processing-in-Memory
Saugata Ghose† Amirali Boroumand† Jeremie S. Kim†§ Juan Gómez-Luna§ Onur Mutlu§†
†Carnegie Mellon University
§ETH Zürich

Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu, "Processing-in-Memory: A Workload-Driven Perspective"
[Preliminary arXiv version]

More on Processing-in-Memory

- Onur Mutlu,
  "Memory-Centric Computing Systems"
  [Slides (pptx) (pdf)]
  [Executive Summary Slides (pptx) (pdf)]
  [Tutorial Video (1 hour 51 minutes)]
  [Executive Summary Video (2 minutes)]
  [Abstract and Bio]
  [Related Keynote Paper from VLSI-DAT 2020]
  [Related Review Paper on Processing in Memory]

https://www.youtube.com/watch?v=H3sEaINPBOE

https://www.youtube.com/onurmutlulectures

102
Memory-Centric Computing Systems

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
12 December 2020
IEDM Tutorial

https://www.youtube.com/watch?v=H3sEaINPBOE
https://www.youtube.com/onurmutlulectures
PIM Course (Fall 2022)

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory

- **Youtube Livestream (Fall 2022):**
  - https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXYZi8KzG2CQYRNQOVD0GOBrnKy

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Processing-in-Memory lectures
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures

SAFARI
Processing-in-Memory Landscape Today

This does not include many experimental chips and startups.
UPMEM Processing-in-DRAM Engine (2019)

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

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

---

UPMEM Memory Modules

- E19: 8 chips DIMM (1 rank). DPUs @ 267 MHz
- P21: 16 chips DIMM (2 ranks). DPUs @ 350 MHz
2,560-DPU Processing-in-Memory System

Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
IZZAT EL HAJI, American University of Beirut, Lebanon
IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
CHRISTINA GIANNOPULOS, ETH Zürich, Switzerland and NTUA, Greece
GERALDO F. OLIVEIRA, ETH Zürich, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland

Many modern workloads, such as neural networks, databases, and graph processing, are fundamentally memory-bound. For such workloads, the data movement between main memory and CPU cores imposes a significant overhead in terms of both latency and energy. A major reason is that this communication happens through a narrow bus with high latency and limited bandwidth, and the low data reuse in memory-bound workloads is insufficient to amortize the cost of main memory access. Fundamentally addressing this data movement bottleneck requires a paradigm where the memory system assumes an active role in computing by integrating processing capabilities. This paradigm is known as processing-in-memory (PIM).

Recent research explores different forms of PIM architecture, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulations or, at best, with simplified hardware prototypes. In contrast, the UPHENM company has designed and manufactured the first publicly available real-world PIM architecture. The UPHENM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called DRAM Processing Units (DPU), integrated into the same chip.

This paper provides the first comprehensive analysis of the first publicly available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPHENM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present PIM (Processing-in-Memory benchmark), a benchmark suite of 16 workloads from different application domains (e.g., dense-sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PIM benchmarks on the UPHENM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPHENM-based PIM systems with 160 and 2,560 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.

More on the UPMEM PIM System

DRAM Processing Unit (II)
SRC TECHCON Presentation

- Dr. Juan Gomez-Luna
  - Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware
  - Based on two major works

Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-In-Memory Hardware

Year: 2021, Pages: 1-7
DOI Bookmark: 10.1109/IGSC54211.2021.9651614

Authors
Juan Gómez-Luna, ETH Zürich
Izzat El Hajji, American University of Beirut
Ivan Fernandez, University of Malaga
Christina Giannoula, National Technical University of Athens
Geraldo F. Oliveira, ETH Zürich
Onur Mutlu, ETH Zürich

[https://www.youtube.com/watch?v=nphV36SrysA](https://www.youtube.com/watch?v=nphV36SrysA)
UPMEM PIM System Summary & Analysis

- Juan Gomez-Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu,

*Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware*

*Invited Paper at Workshop on Computing with Unconventional Technologies (CUT), Virtual, October 2021.*

[arXiv version]
[PrIM Benchmarks Source Code]
[Slides (pptx) (pdf)]
[Talk Video (37 minutes)]
[Lightning Talk Video (3 minutes)]

Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware

Juan Gómez-Luna
ETH Zürich

Izzat El Hajj
American University of Beirut

Ivan Fernandez
University of Malaga

Christina Giannoula
National Technical University of Athens

Geraldo F. Oliveira
ETH Zürich

Onur Mutlu
ETH Zürich
A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems

Safaa Diab\textsuperscript{1} Amir Nassereldine\textsuperscript{1} Mohammed Alser\textsuperscript{2} Juan Gómez Luna\textsuperscript{2} Onur Mutlu\textsuperscript{2} Izzat El Hajj\textsuperscript{1}

\textsuperscript{1}American University of Beirut\textsuperscript{2}ETH Zürich

Accelerating Basecalling + Read Mapping via PIM

- Haiyu Mao, Mohammed Alser, Mohammad Sadrosadati, Can Firtina, Akanksha Baranwal, Damla Senol Cali, Aditya Manglik, Nour Almadhoun Alserr, and Onur Mutlu, "GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping". Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.

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

---

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

Haiyu Mao\(^1\)  Mohammed Alser\(^1\)  Mohammad Sadrosadati\(^1\)  Can Firtina\(^1\)  Akanksha Baranwal\(^1\)
Damla Senol Cali\(^2\)  Aditya Manglik\(^1\)  Nour Almadhoun Alserr\(^1\)  Onur Mutlu\(^1\)

\(^1\)ETH Zürich  \(^2\)Bionano Genomics

Agenda

- The Problem: DNA Read Mapping
  - State-of-the-art Read Mapper Design

- Algorithmic Acceleration
  - Exploiting Structure of the Genome
  - Exploiting SIMD Instructions

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory & Storage

- Future Opportunities: New Technologies & Applications
Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions


[Open arxiv.org version] [Slides (pptx) (pdf)] [Talk Video at AACBB 2019]
New Applications: Graph Genomes

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika Mansouri Ghiasii, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu, "SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
  [arXiv version]

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

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

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

New Applications: Frequent Reference Updates

  [bioRxiv preprint]
  [arXiv preprint]
  [AirLift Source Code and Data]

METHOD

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

Jeremie S. Kim\textsuperscript{1,\dagger}, Can Firtina\textsuperscript{1,\dagger}, Meryem Banu Cavlak\textsuperscript{2}, Damla Senol Cali\textsuperscript{3}, Nastaran Hajinazar\textsuperscript{1,4}, Mohammed Alser\textsuperscript{1}, Can Alkan\textsuperscript{2} and Onur Mutlu\textsuperscript{1,2,3*}

Mapping Constant Regions Between References

- Jeremie S. Kim, Can Firtina, Meryem Banu Cavlak, Damla Senol Cali, Can Alkan, and Onur Mutlu,
  "FastRemap: A Tool for Quickly Remapping Reads between Genome Assemblies"
  *Bioinformatics*, btac554.
  [FastRemap Source Code]

---

**FastRemap: A Tool for Quickly Remapping Reads between Genome Assemblies**

Jeremie S. Kim\(^1\)  Can Firtina\(^1\)  Meryem Banu Cavlak\(^1\)  Damla Senol Cali\(^2,3\)  
Can Alkan\(^4\)  Onur Mutlu\(^1,2,4\)  

\(^1\)ETH Zürich  \(^2\)Carnegie Mellon University  \(^3\)Bionano Genomics  \(^4\)Bilkent University

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

Can Firtina\textsuperscript{1}  Nika Mansouri Ghiasi\textsuperscript{1}  Joel Lindegger\textsuperscript{1}  Gagandeep Singh\textsuperscript{1}  Meryem Banu Cavlak\textsuperscript{1}  Haiyu Mao\textsuperscript{1}  Onur Mutlu\textsuperscript{1}  
\textsuperscript{1}ETH Zurich
New Frontiers: Raw Signal Analysis

- To appear at APBC 2023

TargetCall: Eliminating the Wasted Computation in Basecalling via Pre-Basecalling Filtering

Meryem Banu Cavlak\textsuperscript{1} Gagandeep Singh\textsuperscript{1} Mohammed Alser\textsuperscript{1} Can Firtina\textsuperscript{1} Joël Lindegger\textsuperscript{1} Mohammad Sadrosadati\textsuperscript{1} Nika Mansouri Ghiasi\textsuperscript{1} Can Alkan\textsuperscript{2} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \textsuperscript{2}Bilkent University

A Bright Future for Intelligent Genome Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396

MinION from ONT

SmidgION from ONT
Conclusion
Things Are Happening In Industry
Illumina DRAGEN Bio-IT Platform (2018)

- Processes whole genome at 30x coverage in ~25 minutes with hardware support for data compression

[Links]
emea.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html
NVIDIA Clara Parabricks (2020)

A University of Michigan startup in 2018 joined NVIDIA in 2020

PERFORMANCE COMPARISON
Germline End-to-End Secondary Analysis

- CPU/GATK: 1,200 minutes
- 8X T4: 52 minutes
- 8X V100: 35 minutes
- 8X A100: 23 minutes

https://developer.nvidia.com/clara-parabricks
NVIDIA Hopper GPU Architecture Accelerates Dynamic Programming Up to 40x Using New DPX Instructions

Dynamic programming algorithms are used in healthcare, robotics, quantum computing, data science and more.

March 22, 2022 by DION HARRIS

https://blogs.nvidia.com/blog/2022/03/22/nvidia-hopper-accelerates-dynamic-programming-using-dpx-instructions/
We are accelerating the transformation in how we analyze the human genome!

Bionano & NVIDIA: Accelerating Analysis for Fast Time to Results

- Technological solution to support higher throughput
- New high-performance algorithms from Bionano
- Powered by NVIDIA RTX™ 6000 Ada Generation GPUs
- Analysis of highly complex cancer whole genomes in less than 2 hours
- Workflow tailored for a small lab and IT footprint
Recall Our Dream (from 2007)

- An embedded device that can perform comprehensive genome analysis in real time (within a minute)

- Still a long ways to go
  - Energy efficiency
  - Performance (latency)
  - Security & privacy
  - Huge memory bottleneck
Conclusion

- System design for bioinformatics is a critical problem
  - It has large scientific, medical, societal, personal implications

- This talk is about accelerating a key step in bioinformatics: genome sequence analysis
  - In particular, read mapping

- We covered various recent ideas to accelerate read mapping
  - My personal journey since September 2006

- Many future opportunities exist
  - Especially with new sequencing technologies
  - Especially with new applications and use cases
A Bright Future for Intelligent Genome Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396

MinION from ONT

SmidgION from ONT
A Longer Version of This Talk (I)

Accelerating Genome Analysis
A Primer on an Ongoing Journey

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
5 April 2022
SPMA Workshop Keynote @ EuroSys

https://www.youtube.com/watch?v=NCagwf0ivT0
Read Alignment/Verification

- **Edit distance** is defined as the minimum number of edits (i.e. insertions, deletions, or substitutions) needed to make the read exactly match the reference segment.

![Diagram of NETHERLANDS and SWITZERLAND with match, deletion, insertion, and mismatch indicators]

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

[Video link: https://www.youtube.com/watch?v=tVpg0XqU_c4]
Genomics Course (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=bioinformatics](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=bioinformatics)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics)

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdx0dizD_EhVAMVQV](https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdx0dizD_EhVAMVQV)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NnPdQR1yRU_Cxxjw-u18](https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NnPdQR1yRU_Cxxjw-u18)

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

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
PIM Course (Fall 2022)

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory

- **Youtube Livestream (Fall 2022):**
  - https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQVD0GOBrnK

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Processing-in-Memory lectures
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures
SSD Course (Spring 2023)

- **Spring 2023 Edition:**

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=modern_ssds

- **Youtube Livestream (Spring 2023):**
  - https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&amp;pp=iAQB

- **Youtube Livestream (Fall 2022):**
  - https://www.youtube.com/watch?v=hqLrd-Uj0aU&amp;list=PL5Q2soXY2Zi9BJhenUq4JI5bwhAMpAp13&amp;pp=iAQB

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - SSD Basics and Advanced Topics
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures
Accelerating Genome Analysis
A Primer on an Ongoing Journey

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
14 April 2023
BIO-Arch Workshop @ RECOMB

SAFARI
ETH zürich
Carnegie Mellon
Backup Slides for Further Info
Resources & Acknowledgments
Special Research Sessions & Courses

- Special Session at ISVLSI 2022: 9 cutting-edge talks

[Video: In-Memory Processing] (https://www.youtube.com/watch?v=qeukNs5XI3g)
Detailed Lectures on Genome Analysis

- **Computer Architecture, Fall 2020, Lecture 3a**
  - *Introduction to Genome Sequence Analysis* (ETH Zürich, Fall 2020)
  - [Link](https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5)

- **Computer Architecture, Fall 2020, Lecture 8**
  - *Intelligent Genome Analysis* (ETH Zürich, Fall 2020)
  - [Link](https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14)

- **Computer Architecture, Fall 2020, Lecture 9a**
  - *GenASM: Approx. String Matching Accelerator* (ETH Zürich, Fall 2020)
  - [Link](https://www.youtube.com/watch?v=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15)

- **Accelerating Genomics Project Course, Fall 2020, Lecture 1**
  - *Accelerating Genomics* (ETH Zürich, Fall 2020)
  - [Link](https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqlgwIDRQDTyId)

[SAFARI](https://www.youtube.com/onurmutlulectures)
Comp Arch (Fall’21)

- **Fall 2021 Edition:**
  - [https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule](https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule)

- **Fall 2020 Edition:**

- **Youtube Livestream (2021):**
  - [https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF](https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF)

- **Youtube Livestream (2020):**
  - [https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN](https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN)

- Master’s level course
  - Taken by Bachelor’s/Masters/PhD students
  - Cutting-edge research topics + fundamentals in Computer Architecture
  - 5 Simulator-based Lab Assignments
  - Potential research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
DDCA (Spring 2022)

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/digitaltechnik/spring2022/doku.php?id=schedule

- **Spring 2021 Edition:**

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=cpXdE3HwyK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6

- **Youtube Livestream (Spring 2021):**
  - https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN

- Bachelor’s course
  - 2nd semester at ETH Zurich
  - Rigorous introduction into “How Computers Work”
  - Digital Design/Logic
  - Computer Architecture
  - 10 FPGA Lab Assignments

https://www.youtube.com/onurmutlulectures
Seminar in Comp Arch (Spring & Fall)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/architecture_seminar/spring2022/doku.php?id=schedule](https://safari.ethz.ch/architecture_seminar/spring2022/doku.php?id=schedule)

- **Fall 2021 Edition:**
  - [https://safari.ethz.ch/architecture_seminar/fall2021/doku.php?id=schedule](https://safari.ethz.ch/architecture_seminar/fall2021/doku.php?id=schedule)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=rS9UPk509AQ&list=PL5Q2soXY2Zi_hxizriwKmFHqoe2Q8-m0](https://www.youtube.com/watch?v=rS9UPk509AQ&list=PL5Q2soXY2Zi_hxizriwKmFHqoe2Q8-m0)

- **Youtube Livestream (Fall 2021):**
  - [https://www.youtube.com/watch?v=4TcP297mdsI&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4](https://www.youtube.com/watch?v=4TcP297mdsI&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4)

- Critical analysis course
  - Taken by Bachelor’s/Masters/PhD students
  - Cutting-edge research topics + fundamentals in Computer Architecture
  - 20+ research papers, presentations, analyses

---

**SAFARI**
Hetero. Systems (Spring’22)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=heterogeneous_systems](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=heterogeneous_systems)

- **Youtube Livestream:**
  - [https://www.youtube.com/watch?v=oFO5fTrgFIY&list=PL5Q2soXY2Zi9XrgXR38IM_FTjmY6h7Gzm](https://www.youtube.com/watch?v=oFO5fTrgFIY&list=PL5Q2soXY2Zi9XrgXR38IM_FTjmY6h7Gzm)

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - GPU and Parallelism lectures
  - Hands-on research exploration
  - Many research readings
HW/SW Co-Design (Spring 2022)

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=hw_sw_codesign

- **Youtube Livestream:**
  - https://youtube.com/playlist?list=PL5Q2soXY2Zi8nH7un3ghD2nutKWWDk-NK

- Project course
  - Taken by Bachelor’s/Master’s students
  - HW/SW co-design lectures
  - Hands-on research exploration
  - Many research readings
Funding Acknowledgments

- Alibaba, AMD, ASML, Google, Facebook, Hi-Silicon, HP Labs, Huawei, IBM, Intel, Microsoft, Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, VMware, Xilinx
- NSF
- NIH
- GSRC
- SRC
- CyLab
- EFCL
Acknowledgments

Think BIG, Aim HIGH!

https://safari.ethz.ch
Onur Mutlu’s SAFARI Research Group

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

https://safari.ethz.ch/safari-newsletter-january-2021/

Think BIG, Aim HIGH!

SAFARI

https://safari.ethz.ch
Dear SAFARI friends,

2019 and the first three months of 2020 have been very positive eventful times for SAFARI.
Dear SAFARI friends,

Happy New Year! We are excited to share our group highlights with you in this second edition of the SAFARI newsletter (You can find the first edition from April 2020 here). 2020 has
SAFARI Newsletter December 2021 Edition

https://safari.ethz.ch/safari-newsletter-december-2021/

Think Big, Aim High

View in your browser
December 2021
Referenced Papers, Talks, Artifacts

- All are available at
  
  https://people.inf.ethz.ch/omutlu/projects.htm

  https://www.youtube.com/onurmutlulectures

  https://github.com/CMU-SAFARI/
Open Source Tools: SAFARI GitHub

SAFARI Research Group at ETH Zurich and Carnegie Mellon University
Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.

223 followers ETH Zurich and Carnegie Mellon U... https://safari.ethz.ch/ omutlu@gmail.com

Overview Repositories 83 Projects Packages Teams 1 People 46 Settings

Pinned

ramulator Public
A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, ODDRx, WIOx, HBMx, and various academic proposals. Described in the...

C++ 408 183

prim-benchmarks Public
PrIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publi...

C 80 33

MQSim Public
MQSim is a fast and accurate simulator modeling the performance of modern multi-queue (MQ) SSDs as well as traditional SATA based SSDs. MQSim faithfully models new high-bandwidth protocol implement...

C++ 183 110

rowhammer Public

C 200 40

SparseP Public
SparseP is the first open-source Sparse Matrix Vector Multiplication (SpMV) software package for real-world Processing-In-Memory (PIM) architectures. SparseP is developed to evaluate and characteri...

C 52 11

SoftMC Public
SoftMC is an experimental FPGA-based memory controller design that can be used to develop tests for DDR3 SODIMMs using a C++ based API. The design, the interface, and its capabilities and limitatio...

Verilog 96 26

https://github.com/CMU-SAFARI/
Some Other Recent Papers
Finding Approximate Seed Matches

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

"BLEND: A Fast, Memory-Efficient, and Accurate Mechanism to Find Fuzzy Seed Matches"
[arXiv preprint]
[BLEND Source Code and Data]

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

Can Firtina\(^1\)  Jisung Park\(^1\)  Mohammed Alser\(^1\)  Jeremie S. Kim\(^1\)  Damla Senol Cali\(^2\)
Taha Shahroodi\(^3\)  Nika Mansouri-Ghias\(^1\)  Gagandeep Singh\(^1\)  Konstantinos Kanellopoulos\(^1\)
Can Alkan\(^4\)  Onur Mutlu\(^1\)

\(^1\)ETH Zurich  \(^2\)Bionano Genomics  \(^3\)TU Delft  \(^4\)Bilkent University
Hardware Acceleration for pHMMs

  [Source Code]

ApHMM: A Profile Hidden Markov Model Acceleration Framework for Genome Analysis

Can Firtina¹ Kamlesh Pillai² Gurpreet S. Kalsi² Bharathwaj Suresh² Damla Senol Cali³ Taha Shahroodi⁴ Meryem Banu Cavlak¹ Joel Lindegger¹ Mohammed Alser¹ Juan Gómez Luna¹ Sreenivas Subramoney² Onur Mutlu¹

¹ETH Zurich ²Intel Labs ³Bionano Genomics ⁴TU Delft
Remapping Reads Between References

Jeremie S. Kim, Can Firtina, Meryem Banu Cavlak, Damla Senol Cali, Nastaran Hajinazar, Mohammed Alser, Can Alkan, and Onur Mutlu,
"AirLift: A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes"
[biRxiv preprint]
[arXiv preprint]
[AirLift Source Code and Data]

METHOD

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

Jeremie S. Kim\textsuperscript{1\dagger}, Can Firtina\textsuperscript{1\dagger}, Meryem Banu Cavlak\textsuperscript{2}, Damla Senol Cali\textsuperscript{3}, Nastaran Hajinazar\textsuperscript{1,4}, Mohammed Alser\textsuperscript{1}, Can Alkan\textsuperscript{2} and Onur Mutlu\textsuperscript{1,2,3\star}
Mapping Constant Regions Between References

- Jeremie S. Kim, Can Firtina, Meryem Banu Cavlak, Damla Senol Cali, Can Alkan, and Onur Mutlu,
- "FastRemap: A Tool for Quickly Remapping Reads between Genome Assemblies"
- *Bioinformatics*, btac554.
- [FastRemap Source Code]

---

**FastRemap: A Tool for Quickly Remapping Reads between Genome Assemblies**

Jeremie S. Kim\(^1\)  Can Firtina\(^1\)  Meryem Banu Cavlak\(^1\)  Damla Senol Cali\(^2,3\)
Can Alkan\(^4\)  Onur Mutlu\(^1,2,4\)

\(^1\)ETH Zürich  \(^2\)Carnegie Mellon University  \(^3\)Bionano Genomics  \(^4\)Bilkent University
COVIDHunter

Mohammed Alser, Jeremie S. Kim, Nour Almadhoun Alserr, Stefan W. Tell, Onur Mutlu

“COVIDHunter: COVID-19 Pandemic Wave Prediction and Mitigation via Seasonality Aware Modeling”
Frontiers in Public Health 2022
[Source Code]
Packaging Omics Methods

Mohammed Alser, Sharon Waymost, Ram Ayyala, Brendan Lawlor, Richard J. Abdill, Neha Rajkumar, Nathan LaPierre, Jaqueline Brito, Andre M. Ribeiro-dos-Santos, Can Firtina, Nour Almadhoun, Varuni Sarwal, Eleazar Eskin, Qiyang Hu, Derek Strong, Byoung-Do (BD) Kim, Malak S. Abedalthagafi, Onur Mutlu, Serghei Mangul

“Packaging, containerization, and virtualization of computational omics methods: Advances, challenges, and opportunities”

arrXiv 2022
Demeter (HD Food Microbiome Profiling)

Taha Shahroodi, Mahdi Zahedi, Can Firtina, Mohammed Alser, Stephan Wong, Onur Mutlu, Said Hamdioui

“Demeter: A Fast and Energy-Efficient Food Profiler using Hyperdimensional Computing in Memory”
IEEE Access, 2022

Demeter: A Fast and Energy-Efficient Food Profiler Using Hyperdimensional Computing in Memory

TAHA SHAHROODI¹, MAHDI ZAHEDI¹, CAN FIRTINA², MOHAMMED ALSER¹, STEPHAN WONG¹, (Senior Member, IEEE), ONUR MUTLU¹, (Fellow, IEEE), AND SAID HAMDIoui¹, (Senior Member, IEEE)

¹Q&CE Department, EEMCS Faculty, Delft University of Technology (TU Delft), 2628 CD Delft, The Netherlands
²SAFARI Research Group, D-ITET, ETH Zürich, 8092 Zürich, Switzerland
End of Backup Slides