Accelerating Genome Analysis
A Primer on an Ongoing Journey

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu

3 November 2022
Montenegro Academy of Sciences Conference
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

Rapid surveillance of **disease outbreaks**

Developing **personalized medicine**

SAFARI

And, many, many other applications ...
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 Few Overview Readings (I)

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

“Accelerating Genome Analysis: A Primer on an Ongoing Journey”

IEEE Micro

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

Authors
Mohammed Alser, ETH Zürich
Zülal Bingöl, Bilkent University
Damla Senol Cali, Carnegie Mellon University
Jeremie Kim, ETH Zurich and Carnegie Mellon University
Saugata Ghose, University of Illinois at Urbana–Champaign and Carnegie Mellon University
Can Alkan, Bilkent University
Onur Mutlu, ETH Zurich, Carnegie Mellon University, and Bilkent University

A Few Overview Readings (II)

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]

IEEE Micro

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

Authors

Gagandeep Singh, ETH Zürich, Zürich, Switzerland
Mohammed Alser, ETH Zürich, Zürich, Switzerland
Damla Senol Cali, Carnegie Mellon University, Pittsburgh, PA, USA
Dionysios Diamantopoulos, Zürich Lab, IBM Research Europe, Rüschlikon, Switzerland
Juan Gomez-Luna, ETH Zürich, Zürich, Switzerland
Henk Corporaal, Eindhoven University of Technology, Eindhoven, The Netherlands
Onur Mutlu, ETH Zürich, Zürich, Switzerland

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. Gloriagasse 35, 8092 Zürich, Switzerland

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

Cell

Nucleus

Chromosome - 23 pairs

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

DNA (Genotypes) → RNA → Protein (Phenotypes)

- Replication
- Transcription
- Translation
DNA Under Electron Microscope

human chromosome #12 from HeLa’s cell
CCTCCTCAGTGCCACCCAGCCCACTGGCAGCTCCCAAACA
GGCTCTTATTAAAAACACCCCTGTTCCCTGCCCCCTTGGAGTG
AGGTGTCAAGGACCTAAACTTAAAAAAGAGAAAA
AGAAAAAGAAAAAGAATTAAAAATTAAAGTAATTCTTTGAA
AAAAACTAATTTTCTAAGCTTCTTCATGTCAAGGACCTAAATG
TGCTAAACAGCACTTTTTTTTGACCATTATTTTGGATCTGAAA
GAAATCAAGAATAAATGAAAGGACTTGATACATTGGAAGA
GGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAAGAAA
AAGAAAAGAAAAAGAATTAAAAATTAAAGTAATTCTTTGA
AAAAACTAATTTTCTAAGCTTCTTCATGTCAAGGACCTAAATG
GTCTGTGTTGCAGGTCTTCTTGACATTTCCCTGTCAAAAGA
AAAAGAATTTAAAAATTAAAGTAATTCTTTGGAAAAAACTA
ATTTCTAAGCTTCTTCCATGTCAAGGACCTAAATGTCAGGCC
GGCTCTTATTAAAAACACCCCTGTTCCCTGCCCCCTTGGAGTG
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

ACGTACCCCGT
GATACACTGTG
TTTTTTTAATT
CTAGGGACCTT
ACGACGTAGCT
ACGAGCGGGT
GATAACCTGTG
AAAAAAAAAA
CTAGGGACCTT
ACGACGTAGCT

Reads
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

Roche/454

AB SOLiD

Illumina MiSeq

Complete Genomics

Oxford Nanopore MinION

Illumina NovaSeq 6000

Illumina HiSeq2000

Pacific Biosciences RS

Illumina MiSeq

Complete Genomics

Oxford Nanopore GridION

... and more! All produce data with different properties.
High-Throughput Sequencers

Illumina MiSeq

Illumina NovaSeq 6000

Pacific Biosciences RS II

Oxford Nanopore PromethION

Oxford Nanopore MinION

Oxford Nanopore SmidgION

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

development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

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))

SAFARI
Genome Sequencing Cost Is Reducing

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

SAFARI
Solving the Puzzle

Reference genome

Sequenced Reads

.FASTA file

.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

Long: 10-20 kb
Accurate: 99.8%

But still very expensive!

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

Reference: TTTATCGCTTCCATGACGCAG
read1: ATCGCATCC
read2: TATCGCATC
read3: CATCCATGA
read4: CGCTTCCAT
read5: CCATGACGC
read6: TTCCATGAC

2 Read Mapping

3 Variant Calling

4 Scientific Discovery
Technology dictates algorithms: recent developments in read alignment

Mohammed Alser\textsuperscript{1,2,3†}, Jeremy Rotman\textsuperscript{4†}, Dhrithi Deshpande\textsuperscript{5}, Kodi Taraszka\textsuperscript{4}, Huwenbo Shi\textsuperscript{6,7}, Pelin Icer Baykal\textsuperscript{8}, Harry Taegyun Yang\textsuperscript{4,9}, Victor Xue\textsuperscript{4}, Sergey Knyazev\textsuperscript{8}, Benjamin D. Singer\textsuperscript{10,11,12}, Brunilda Balliu\textsuperscript{13}, David Koslicki\textsuperscript{14,15,16}, Pavel Skums\textsuperscript{8}, Alex Zelikovsky\textsuperscript{8,17}, Can Alkan\textsuperscript{2,18}, Onur Mutlu\textsuperscript{1,2,3†} and Serghei Mangul\textsuperscript{5++}
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 **

2 Million bases/minute
Read Mapping *

150x slower

* 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

Problem with (Genome) Analysis Today

Special-Purpose Machine for Data Generation

General-Purpose Machine for Data Analysis

FAST

SLOW

Slow and inefficient processing capability

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)
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
Computational Cost is Mathematically Proven

arXiv.org > cs > arXiv:1412.0348

Computer Science > Computational Complexity

[Submitted on 1 Dec 2014 (v1), last revised 15 Aug 2017 (this version, v4)]

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
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

1. **Indexing**
   - Reference Genome
   - k-mers
   - Index
   - k-mer content
   - k-mer locations
   - 1, 4, 8, 3, 5, 12, 50, 52, 2, 100, 17, 19, 23, 90, ...

2. **Pre-Alignment Filtering**
   - Read
   - k-mers
   - Locating common k-mers
   - Reference subsequences extracted at each common k-mer location

3. **Sequence Alignment**
   - Read
   - Dynamic Programming (DP) Matrix
   - Output
   - SAM file (alignment score, edit distance, type and location of each edit)

**Accelerating Indexing**
- Reducing the number of seeds
- Reducing data movement during indexing

**Accelerating Pre-Alignment Filtering**
- q-gram filtering
- Pigeonhole principle
- Base counting
- Sparse DP

**Accelerating Alignment**
- Accurate alignment accelerators
- Heuristic-based alignment accelerators

Detailed Analysis of Tackling the Bottleneck

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

“Accelerating Genome Analysis: A Primer on an Ongoing Journey”
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, Oguz 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** = ~13.3 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
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, 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
(Source Code)
[Online link at Bioinformatics Journal]
Hardware Implementation

Counting is performed concurrently for all bit-vectors and all sliding windows in a single clock cycle using multiple 4-input LUTs.
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,*

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

**Key idea:**
- Reduce the approximate string matching problem to the Single Net Routing problem in VLSI chip layout
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" 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 Alser\textsuperscript{1,2,*}, Taha Shahroodi\textsuperscript{1}, Juan Gómez-Luna\textsuperscript{1,2}, Can Alkan\textsuperscript{4,*}, and Onur Mutlu\textsuperscript{1,2,3,4,*}

\textsuperscript{1}Department of Computer Science, ETH Zurich, Zurich 8006, Switzerland
\textsuperscript{2}Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland
\textsuperscript{3}Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh 15213, PA, USA
\textsuperscript{4}Department 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

SAFARI
GenASM: ASM Framework for GSA

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
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**

**Key Results – Area and Power**

- **Total (1 vault):** 0.334 mm$^2$ 0.101 W
- **Total (32 vaults):** 10.69 mm$^2$ 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

GenASM has low area and power overheads
Use Cases of GenASM

1. **Indexing**
   - Reference genome
   - Hash table based index

2. **Seeding**
   - Reads from sequenced genome
   - Candidate mapping locations

3. **Pre-Alignment Filtering**
   - Remaining candidate mapping locations

4. **Read Alignment**
   - Optimal alignment
# 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**)

**SAFARI**
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\textsuperscript{†\textcopyright} Gurpreet S. Kalsi\textsuperscript{\textcopyright} Zülal Bingöl\textsuperscript{\textcopyright} Can Firtina\textsuperscript{\textcopyright} Lavanya Subramanian\textsuperscript{‡} Jeremie S. Kim\textsuperscript{‡}\textsuperscript{†}
Rachata Ausavarungnirun\textsuperscript{©} Mohammed Alser\textsuperscript{\textcopyright} Juan Gomez-Luna\textsuperscript{\textcopyright} Amirali Boroumand\textsuperscript{‡} Anant Nori\textsuperscript{\textcopyright}
Allison Scibisz\textsuperscript{‡} Sreenivas Subramoney\textsuperscript{\textsuperscript{\textcopyright}} Can Alkan\textsuperscript{\textsuperscript{\textcopyright}} Saugata Ghose\textsuperscript{\textsuperscript{\textsuperscript{\textcopyright}†} Onur Mutlu\textsuperscript{\textsuperscript{\textsuperscript{\textcopyright}†\textsuperscript{†}}}

\textsuperscript{†}Carnegie Mellon University \textsuperscript{\textcopyright} Processor Architecture Research Lab, Intel Labs \textsuperscript{©} Bilkent University \textsuperscript{§}ETH Zürich
\textsuperscript{‡}Facebook \textsuperscript{©} King Mongkut’s University of Technology North Bangkok \textsuperscript{\textcopyright}University of Illinois at Urbana–Champaign
SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping

Damla Senol Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Züal 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
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)**

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

0.2 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)

**Linear reference genome**

**Known genetic variations**

**Reads from sequenced genome**

**Genome graph**

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

**Candidate mapping locations (subgraphs)**

**Remaining candidate mapping locations (subgraphs)**

**Optimal alignment between read & subgraph**
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 Talk Video

SeGraM: A Universal HW Accelerator for Genomic Sequence-to-Graph Mapping - Damla Senol Cali (ISCA)

136 views • Premiered 21 hours ago

https://www.youtube.com/watch?v=qyjqYoyDP9s
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 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 Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Zülal Bingöl³
- Gurpreet S. Kalsi⁴ Ziyi Zuo⁵ Can Firtina² Meryem Banu Cavlak² Jeremie Kim² Nika Mansouri Ghiasi² Gagandeep Singh² Juan Gómez-Luna² Nour Almadhoun Alserr² Mohammed Alser² Sreenivas Subramoney⁴ Can Alkan³ Saugata Ghose⁶ Onur Mutlu²

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

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
Read Mapping & Filtering in Memory

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 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]

IEEE Micro

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

Authors
Gagandeep Singh, ETH Zürich, Zürich, Switzerland
Mohammed Alser, ETH Zürich, Zürich, Switzerland
Damla Senol Cali, Carnegie Mellon University, Pittsburgh, PA, USA
Dionysios Diamantopoulos, Zürich Lab, IBM Research Europe, Rüschlikon, Switzerland
Juan Gomez-Luna, ETH Zürich, Zürich, Switzerland
Henk Corporaal, Eindhoven University of Technology, Eindhoven, The Netherlands
Onur Mutlu, ETH Zürich, Zürich, Switzerland
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\(^1\) Jisung Park\(^1\) Harun Mustafa\(^1\) Jeremie Kim\(^1\) Ataberk Olgun\(^1\) Arvid Gollwitzer\(^1\) Damla Senol Cali\(^2\) Can Firtina\(^1\) Haiyu Mao\(^1\) Nour Almadhoun Alserr\(^1\) Rachata Ausavarungnirun\(^3\) Nandita Vijaykumar\(^4\) Mohammed Alser\(^1\) Onur Mutlu\(^1\)

\(^1\)ETH Zürich \(^2\)Bionano Genomics \(^3\)KMUTNB \(^4\)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

SAFARI
**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

- 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
A Modern Primer on Processing in Memory

Onur Mutlu\textsuperscript{a,b}, Saugata Ghose\textsuperscript{b,c}, Juan Gómez-Luna\textsuperscript{a}, Rachata Ausavarungnirun\textsuperscript{d}

SAFARI Research Group

\textsuperscript{a}ETH Zürich
\textsuperscript{b}Carnegie Mellon University
\textsuperscript{c}University of Illinois at Urbana-Champaign
\textsuperscript{d}King Mongkut’s University of Technology North Bangkok

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun, 
"A Modern Primer on Processing in Memory"

Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gómez-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
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 HAJJ, American University of Beirut, Lebanon
IVAN FERNANDEZ, ETH Zürich and University of Malaga, Spain
CHRISTINA GIANNOLA, 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, reducing 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 simulation or, at best, with simplified hardware prototypes. In contrast, the UPMEM company has designed and manufactured the first publicly available real-world PIM architecture. The UPMEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called DRAM Processing Units (DPU), integrated in 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 evaluation of the UPMEM-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) benchmarks, 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 energy consumption of the UPMEM-based PIM system with 140 and 2,560 DPU processors. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 140 and 2,560 DPU processors 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)
  - Control/Status Interface
  - DDR4 Interface
  - PIM Chip
  - 24-KB Instruction Memory
  - 64-KB WRAM
  - 64-MB DRAM Bank
  - DMA Engine

ETH ZURICH HAUPTGEBÄUDE
Computer Architecture - Lecture 12d: Real Processing-in-DRAM with UPMEM (ETH Zürich, Fall 2020)

https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xdylQxBu7xRPS-wisBN&index=26
A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems

Safaa Diab¹, Amir Nassereldine¹, Mohammed Alser², Juan Gómez Luna², Onur Mutlu², Izzat El Hajj¹

¹American University of Beirut, Lebanon ²ETH Zürich, Switzerland
Accelerating Genome Analysis w/ Processing using NVM

Appears at MICRO 2022

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

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

¹ETH Zürich  ²Bionano 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


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

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

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

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

Jeremie S. Kim¹, Can Firtina¹, Meryem Banu Cavlak², Damla Senol Cali³, Nastaran Hajinazar¹,⁴, Mohammed Alser¹, Can Alkan² and Onur Mutlu¹,²,³*

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"


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

**METHOD**

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

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

SAFARI
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
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

FPGA board(s)

[Links]
[emea.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html](emea.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html)
A University of Michigan startup in 2018 joined NVIDIA in 2020
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.
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
A Longer Version of This Talk

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
Accelerating Genome Analysis

A Primer on an Ongoing Journey

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
3 November 2022
Montenegro Academy of Sciences Conference
Backup Slides for Further Info
Resources & Acknowledgments
Accelerating Genome Analysis: Overview

- Mohammed Alser, Zulal Bingol, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,

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


[Slides (pptx)(pdf)]
[Talk Video (1 hour 2 minutes)]

---

Accelerating Genome Analysis: A Primer on an Ongoing Journey

**Mohammed Alser**
ETH Zürich

**Zülal Bingöl**
Bilkent University

**Damla Senol Cali**
Carnegie Mellon University

**Jeremie Kim**
ETH Zurich and Carnegie Mellon University

**Saugata Ghose**
University of Illinois at Urbana–Champaign and Carnegie Mellon University

**Can Alkan**
Bilkent University

**Onur Mutlu**
ETH Zurich, Carnegie Mellon University, and Bilkent University
A Modern Primer on Processing in Memory

Onur Mutlu\textsuperscript{a,b}, Saugata Ghose\textsuperscript{b,c}, Juan Gómez-Luna\textsuperscript{a}, Rachata Ausavarungnirun\textsuperscript{d}

\textit{SAFARI Research Group}

\textsuperscript{a}ETH Zürich  
\textsuperscript{b}Carnegie Mellon University  
\textsuperscript{c}University of Illinois at Urbana-Champaign  
\textsuperscript{d}King Mongkut’s University of Technology North Bangkok

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun, "A Modern Primer on Processing in Memory"  

PIM Review and Open Problems (II)

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 Memory-Centric System Design

- 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
Memory-Centric Computing Systems

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
12 December 2020
IEDM Tutorial
Special Research Sessions & Courses

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

[Video thumbnail showing a presentation slide titled "In-Memory Processing: ISVLSI 2022 Special Session" and a link to the YouTube video "ISVLSI 2022 Special Session on Processing-in-Memory"]

https://www.youtube.com/watch?v=qeukNs5XI3g
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]
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

Detailed Lectures on Genome Analysis

- **Computer Architecture, Fall 2020, Lecture 3a**
  - *Introduction to Genome Sequence Analysis* (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5](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)
  - [https://www.youtube.com/watch?v=ygmQpdDL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14](https://www.youtube.com/watch?v=ygmQpdDL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14)

- **Computer Architecture, Fall 2020, Lecture 9a**
  - *GenASM: Approx. String Matching Accelerator* (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15](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)
  - [https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqlgwiDRQDTyId](https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqlgwiDRQDTyId)

---

*SAFARI*  
[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Genomics (Spring 2022)

- **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:**
  - [https://www.youtube.com/watch?v=DEL5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18](https://www.youtube.com/watch?v=DEL5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18)

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings
Genomics (Fall 2021)

- **Fall 2021 Edition:**

- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=MnogTeMjY8k&list=PL5Q2soXY2Zi8sngHTrNZnDhDkPq55J9J

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

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Livestream</th>
<th>Meeting</th>
<th>Learning Materials</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>5.10 Tue.</td>
<td>YouTube Live</td>
<td>M1: P&amp;S Accelerating Genomics Course Introduction &amp; Project Proposals (PDF) (PPT)</td>
<td>Required Materials</td>
<td>Recommended Materials</td>
</tr>
<tr>
<td>W2</td>
<td>20.10 Wed.</td>
<td>YouTube Live</td>
<td>M2: Introduction to Sequencing (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W3</td>
<td>27.10 Wed.</td>
<td>YouTube Live</td>
<td>M3: Read Mapping (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W4</td>
<td>3.11 Wed.</td>
<td>YouTube Live</td>
<td>M4: GateKeeper (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W5</td>
<td>10.11 Wed.</td>
<td>YouTube Live</td>
<td>M5: MAGNET &amp; Shouji (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W6</td>
<td>17.11 Wed.</td>
<td>YouTube Live</td>
<td>M6:1: SneakySnake (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>YouTube Live</td>
<td>M6:2: GRIM-Filter (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>YouTube Live</td>
<td>M7: GenASM (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W8</td>
<td>01.12 Wed.</td>
<td>YouTube Live</td>
<td>M8: Genome Assembly (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W9</td>
<td>13.12 Mon.</td>
<td>YouTube Live</td>
<td>M9: GRIM-Filter (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W10</td>
<td>15.12 Wed.</td>
<td>YouTube Live</td>
<td>M10: Genomic Data Sharing Under Differential Privacy (PDF) (PPT)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fall 2021 Edition:
- 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

Youtube Livestream (2020):
- https://www.youtube.com/watch?v=c3mPdZAFmc&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
DDCA (Spring 2022)

- **Spring 2022 Edition:**

- **Spring 2021 Edition:**

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=cpXdE3Hwyk0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6](https://www.youtube.com/watch?v=cpXdE3Hwyk0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6)

- **Youtube Livestream (Spring 2021):**
  - [https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN](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](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_hxizriwKmFHgcoe2Q8-m0](https://www.youtube.com/watch?v=rS9UPk509AQ&list=PL5Q2soXY2Zi_hxizriwKmFHgcoe2Q8-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
PIM Course (Spring 2022)

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

- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=9e4Chndovo&list=PL5Q2soXY2Zi841fUYYUK9EsXKhQKRPyX

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Processing-in-Memory lectures
  - Hands-on research exploration
  - Many research readings
Hetero. Systems (Spring’22)

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

- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=oFO5fTrqFIY&list=PL5Q2soXY2Zi9XrgXR38IM_FTjmY6h7Gzm

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - GPU and Parallelism lectures
  - Hands-on research exploration
  - Many research readings
**Spring 2022 Edition:**
- [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=hw_sw_co_design](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=hw_sw_co_design)

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

```
<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Livestream</th>
<th>Meeting</th>
<th>Materials</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>W0</td>
<td>16.03</td>
<td><a href="#">YouTube Live</a></td>
<td>Intro to HW/SW Co-Design (PPTX)</td>
<td>Required</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W1</td>
<td>23.03</td>
<td></td>
<td>Project selection</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>W2</td>
<td>30.03</td>
<td><a href="#">YouTube Live</a></td>
<td>Virtual Memory (I)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W3</td>
<td>13.04</td>
<td><a href="#">YouTube Live</a></td>
<td>Virtual Memory (II)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
SSD Course (Spring 2022)

- **Spring 2022 Edition:**

- **Youtube Livestream:**
  - [https://www.youtube.com/watch?v=_q4rm71DsY4&list=PL5Q2soXY2zi8vabcse1kL22DEcgMI2RAq](https://www.youtube.com/watch?v=_q4rm71DsY4&list=PL5Q2soXY2zi8vabcse1kL22DEcgMI2RAq)

- Project course
  - Taken by Bachelor’s/Master’s students
  - SSD Basics and Advanced Topics
  - 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

SAFARI
SAFARI Research Group
safari.ethz.ch

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

Pinned

- **ramulator** Public
  A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...
  - C++ 311 161

- **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 53 21

- **DAMOV** Public
  DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processin...
  - C++ 26 4

- **SneakySnake** Public
  SneakySnake is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude...
  - VHDL 41 8

- **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++ 146 93

- **rowhammer** Public
  - C 189 41

[GitHub Link](https://github.com/CMU-SAFARI/)
Some Recent Papers
Connecting Basecalling and Read Mapping in 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"


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

Haiyu Mao\textsuperscript{1} Mohammed Alser\textsuperscript{1} Mohammad Sadrosadati\textsuperscript{1} Can Firtina\textsuperscript{1} Akanksha Baranwal\textsuperscript{1}
Damla Senol Cali\textsuperscript{2} Aditya Manglik\textsuperscript{1} Nour Almadhoun Alserr\textsuperscript{1} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \hspace{3cm} \textsuperscript{2}Bionano Genomics
Finding Approximate Seed Matches

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

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


[arXiv preprint]
[BLEND Source Code and Data]

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

Can Firtina\textsuperscript{1} Jisung Park\textsuperscript{1} Mohammed Alser\textsuperscript{1} Jeremie S. Kim\textsuperscript{1} Damla Senol Cali\textsuperscript{2}
Taha Shahroodi\textsuperscript{3} Nika Mansouri-Ghiasi\textsuperscript{1} Gagandeep Singh\textsuperscript{1} Konstantinos Kanellopoulos\textsuperscript{1}
Can Alkan\textsuperscript{4} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zurich \hspace{2cm} \textsuperscript{2}Bionano Genomics \hspace{2cm} \textsuperscript{3}TU Delft \hspace{2cm} \textsuperscript{4}Bilkent University
Hardware Acceleration for pHMMs

  [Source Code]

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

Can Firtina\textsuperscript{1} \quad Kamlesh Pillai\textsuperscript{2} \quad Gurpreet S. Kalsi\textsuperscript{2} \quad Bharathwaj Suresh\textsuperscript{2} \quad Damla Senol Cali\textsuperscript{3}
Jeremie S. Kim\textsuperscript{1} \quad Taha Shahroodi\textsuperscript{4} \quad Meryem Banu Cavlak\textsuperscript{1} \quad Joel Lindegger\textsuperscript{1} \quad Mohammed Alser\textsuperscript{1}
Juan Gómez Luna\textsuperscript{1} \quad Sreenivas Subramoney\textsuperscript{2} \quad Onur Mutlu\textsuperscript{1}
\textsuperscript{1}ETH Zurich \quad \textsuperscript{2}Intel Labs \quad \textsuperscript{3}Bionano Genomics \quad \textsuperscript{4}TU Delft
Remapping Reads Between References

  [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,\textasteriskcentered4}, Mohammed Alser\textsuperscript{1}, Can Alkan\textsuperscript{2} and Onur Mutlu\textsuperscript{1,2,3\textasteriskcentered}
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
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 ALSER2, STEPHAN WONG1, (Senior Member, IEEE), ONUR MUTLU2, (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
A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems

Safaa Diab¹, Amir Nassereldine¹, Mohammed Alser², Juan Gómez Luna², Onur Mutlu², Izzat El Hajj¹

¹American University of Beirut, Lebanon   ²ETH Zürich, Switzerland
Scrooge

Joël Lindegger, Damla Senol Cali, Mohammed Alser, Juan Gómez-Luna, Nika Mansouri Ghiasi, Onur Mutlu

“Scrooge: A Fast and Memory-Frugal Genomic Sequence Aligner for CPUs, GPUs, and ASICs”
arXiv, 2022
[Source code]

Genome analysis

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

Joël Lindegger¹,*, Damla Senol Cali², Mohammed Alser¹, Juan Gómez-Luna¹, Nika Mansouri Ghiasi¹ and Onur Mutlu¹,*

¹Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland and
²Bionano Genomics, San Diego, CA 92121, USA.
Intelligent Genome Analysis

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
Pairwise Sequence Alignment using PIM

- Safaa Diab, Amir Nassereldine, Mohammed Alser, Juan Gómez Luna, Onur Mutlu, and Izzat El Hajj,
  "High-throughput Pairwise Alignment with the Wavefront Algorithm using Processing-in-Memory"

High-throughput Pairwise Alignment with the Wavefront Algorithm using Processing-in-Memory

Safaa Diab¹, Amir Nassereldine¹, Mohammed Alser², Juan Gómez Luna², Onur Mutlu², Izzat El Hajj¹
  ¹American University of Beirut, Lebanon ²ETH Zürich, Switzerland
Detailed Lectures on PIM (I)

- Computer Architecture, Fall 2020, Lecture 6
  - Computation in Memory (ETH Zürich, Fall 2020)
    - https://www.youtube.com/watch?v=oGcZAGwfEUE&list=PL5Q2soXY2Zi9xyIgBxUz7xRPS-wisBN&index=12

- Computer Architecture, Fall 2020, Lecture 7
  - Near-Data Processing (ETH Zürich, Fall 2020)
    - https://www.youtube.com/watch?v=j2GIigqn1Qw&list=PL5Q2soXY2Zi9xyIgBxUz7xRPS-wisBN&index=13

- Computer Architecture, Fall 2020, Lecture 11a
  - Memory Controllers (ETH Zürich, Fall 2020)
    - https://www.youtube.com/watch?v=TeG773OgiMQ&list=PL5Q2soXY2Zi9xyIgBxUz7xRPS-wisBN&index=20

- Computer Architecture, Fall 2020, Lecture 12d
  - Real Processing-in-DRAM with UPMEM (ETH Zürich, Fall 2020)
    - https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xyIgBxUz7xRPS-wisBN&index=25

SAFARI

https://www.youtube.com/onurmutlulectures
Detailed Lectures on PIM (II)

- Computer Architecture, Fall 2020, Lecture 15
  - Emerging Memory Technologies (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=AlE1rD9G_YU&list=PL5Q2soXY2Zi9idyIgBxUz7xRPS-wisBN&index=28

- Computer Architecture, Fall 2020, Lecture 16a
  - Opportunities & Challenges of Emerging Memory Technologies (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9idyIgBxUz7xRPS-wisBN&index=29

- Computer Architecture, Fall 2020, Guest Lecture
  - In-Memory Computing: Memory Devices & Applications (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=wNmqQHiEZNk&list=PL5Q2soXY2Zi9idyIgBxUz7xRPS-wisBN&index=41

SAFARI

https://www.youtube.com/onurmutlulectures
End of Backup Slides