Accelerating Genome Analysis

A Primer on an Ongoing Journey

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
6 September 2022
Barcelona Supercomputing Center

SAFARI ETH zürich Carnegie Mellon

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

Our Dream (circa 2007)

- 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?
  - . . .

A Bright Future for Intelligent Genome Analysis

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

MinION from ONT

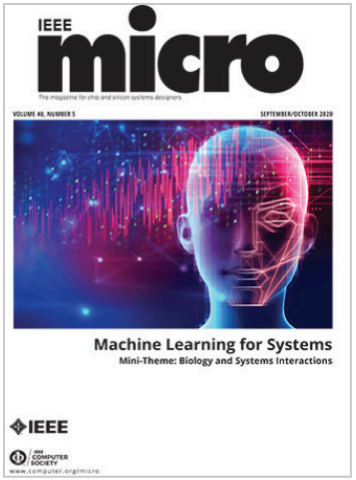
Accelerating Genome Analysis: A Primer on an Ongoing Journey
Sept.-Oct. 2020, pp. 65-75, vol. 40
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
July-Aug. 2021, pp. 39-48, vol. 41
DOI Bookmark: 10.1109/MM.2021.3088396

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, August 2020.



Previous | Next

Table of Contents

Past Issues

Home / Magazines / IEEE Micro / 2020.05

IEEE Micro

Accelerating Genome Analysis: A Primer on an Ongoing Journey

Sept.-Oct. 2020, pp. 65-75, vol. 40
DOI Bookmark: 10.1109/MM.2020.3013728

Authors

Mohammed Alser, ETH Zürich
Zulal Bingol, 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

https://arxiv.org/pdf/2008.00961.pdf

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"
IEEE Micro, 2021.
[Source Code]

Home / Magazines / IEEE Micro / 2021.04

IEEE Micro

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

July-Aug. 2021, pp. 39-48, vol. 41
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

https://arxiv.org/pdf/2106.06433.pdf

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]

ELSEVIER

COMPUTATIONAL ANDSTRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

Review

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

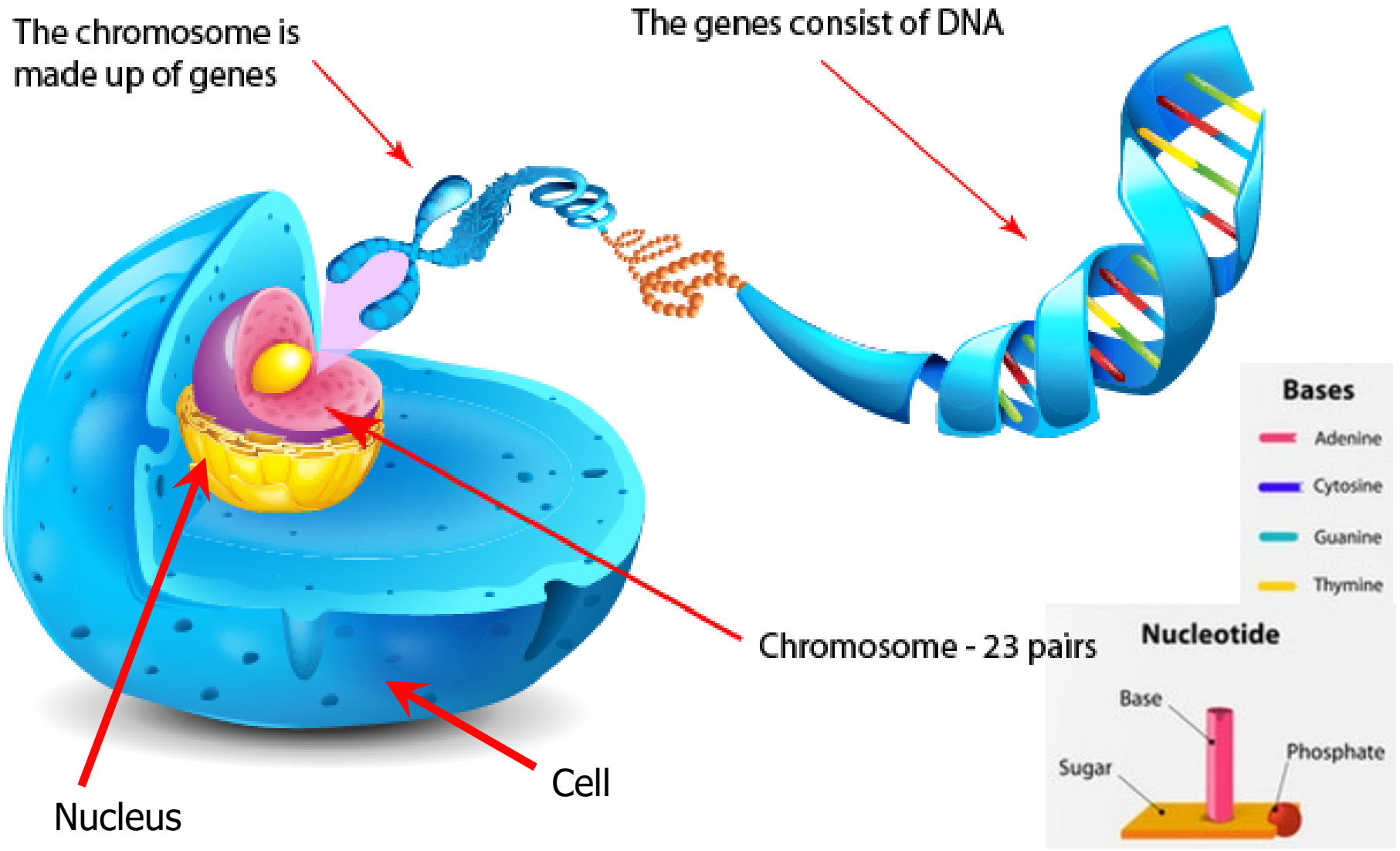
Mohammed Alser *, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu *

ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland

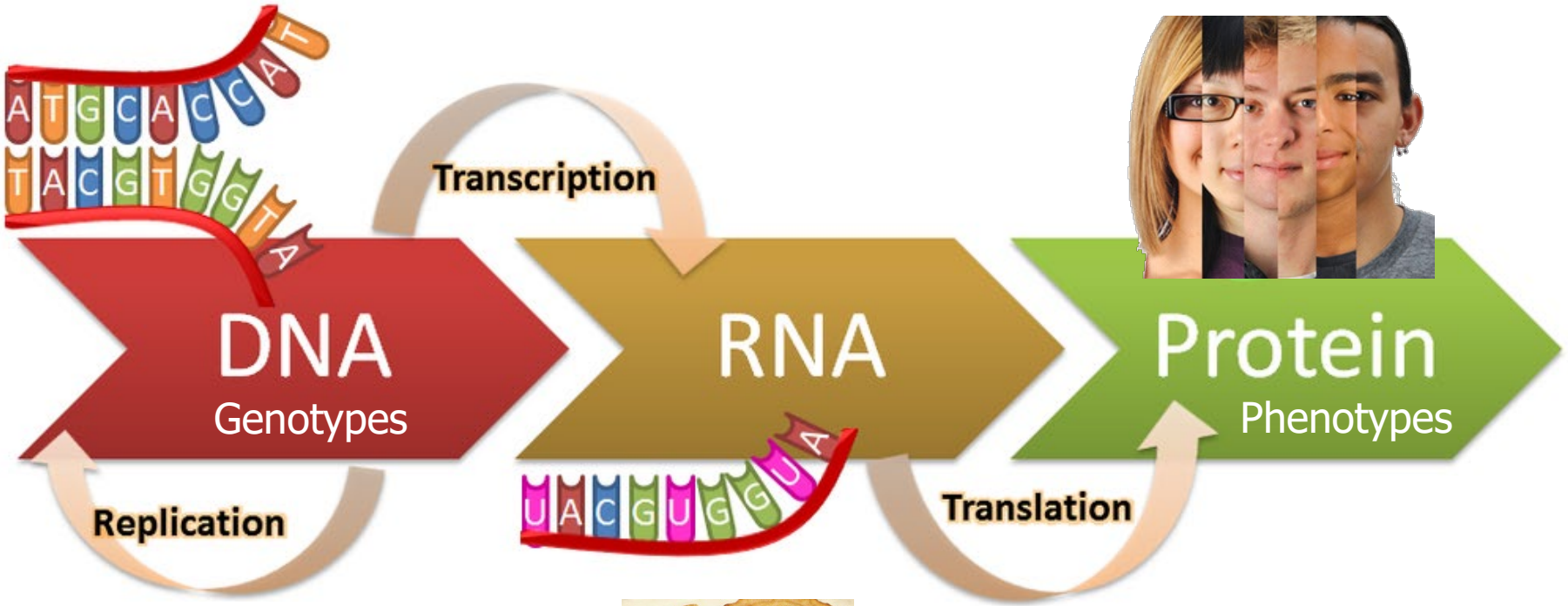
Agenda

- 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

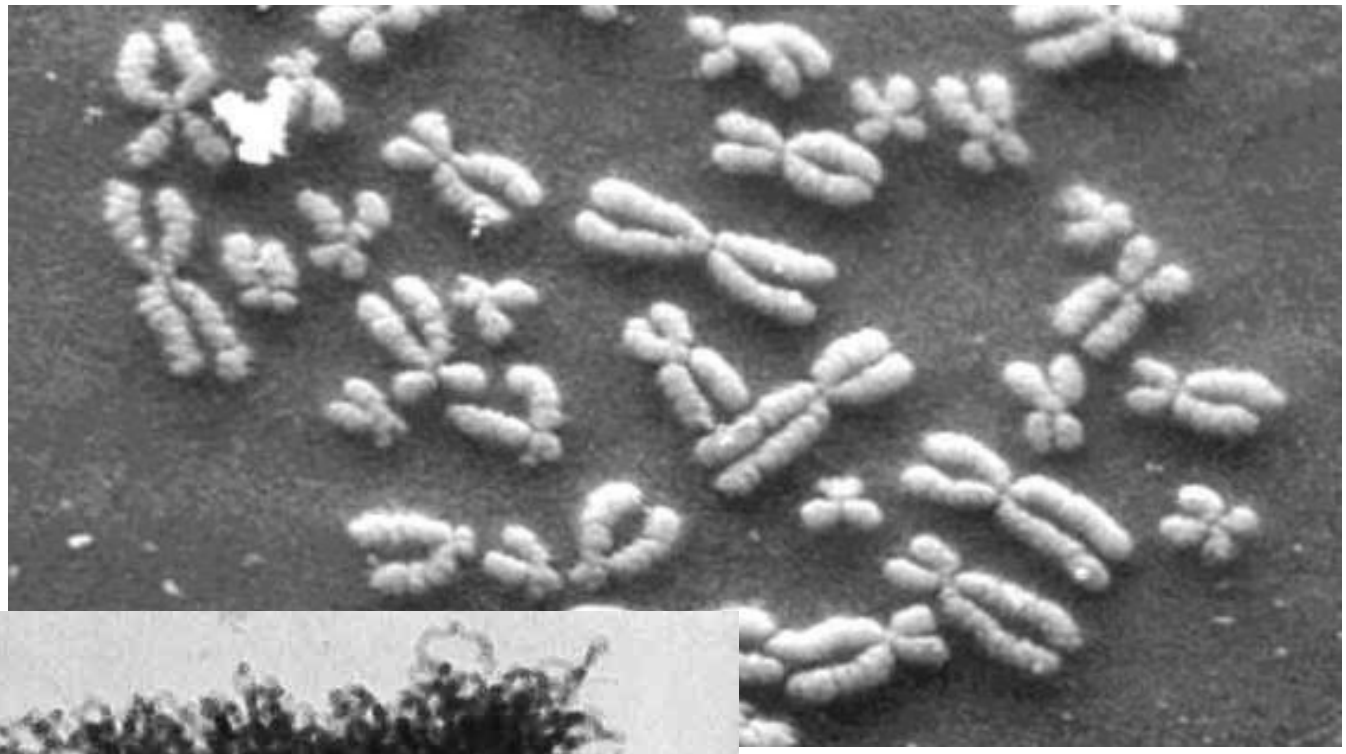
What Is a Genome Made Of?

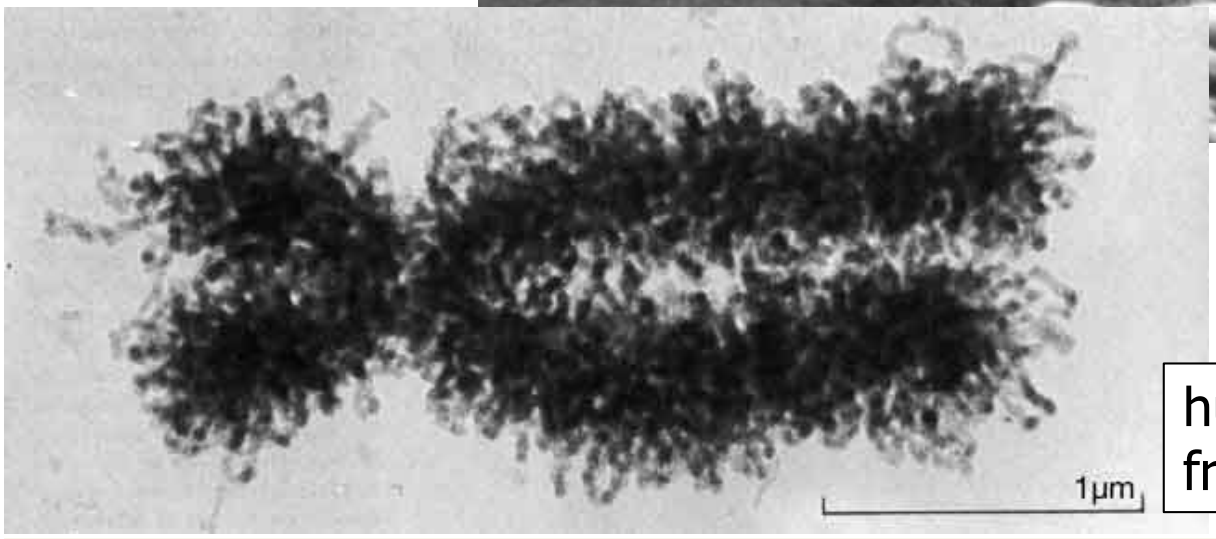


The Central Dogma of Molecular Biology



DNA Under Electron Microscope

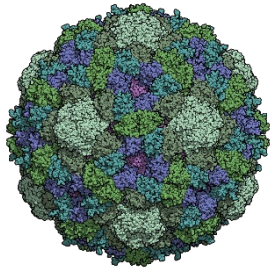


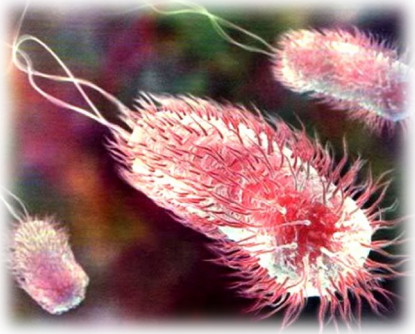


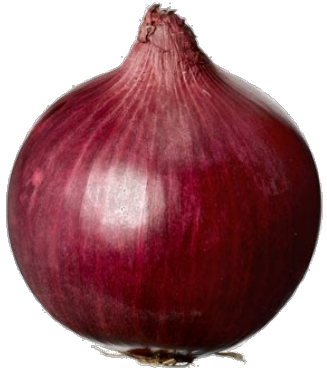
human chromosome #12
from HeLa's cell

CCTCCTCAGTGCCACCCAGCCCACTGGCAGCTCCCAAACA
GGCTCTTATTAAAACACCCTGTTCCCTGCCCCTTGGAGTG
AGGTGTCAAGGACCTAAACTAAAAAAAAAAAAAGAAAA
AGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAA
AAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATG
TGCTAAACAGCACTTTT**TTGACCATTAT**TTTGGATCTGAAA
GAAATCAAGAATAAATGAAGGACTTGATACATTGGAAGA
GGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAGAAA
AAGAAAAGAAAAAGA**A**TTTAAAATTTAAGTAATTCTTTGA
AAAAAACTAATTTCTAAGCTTCTT**C**ATGTCAAGGACCTAAT
GTCTGTGTTGCAGGTCTTCTTGCATTTCCCTGTCAAAAGA
AAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTA
ATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCAGGCC
GGCTCTTATTAAAACACCCTGTTCCCTGCCCCTTGGAGTG

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

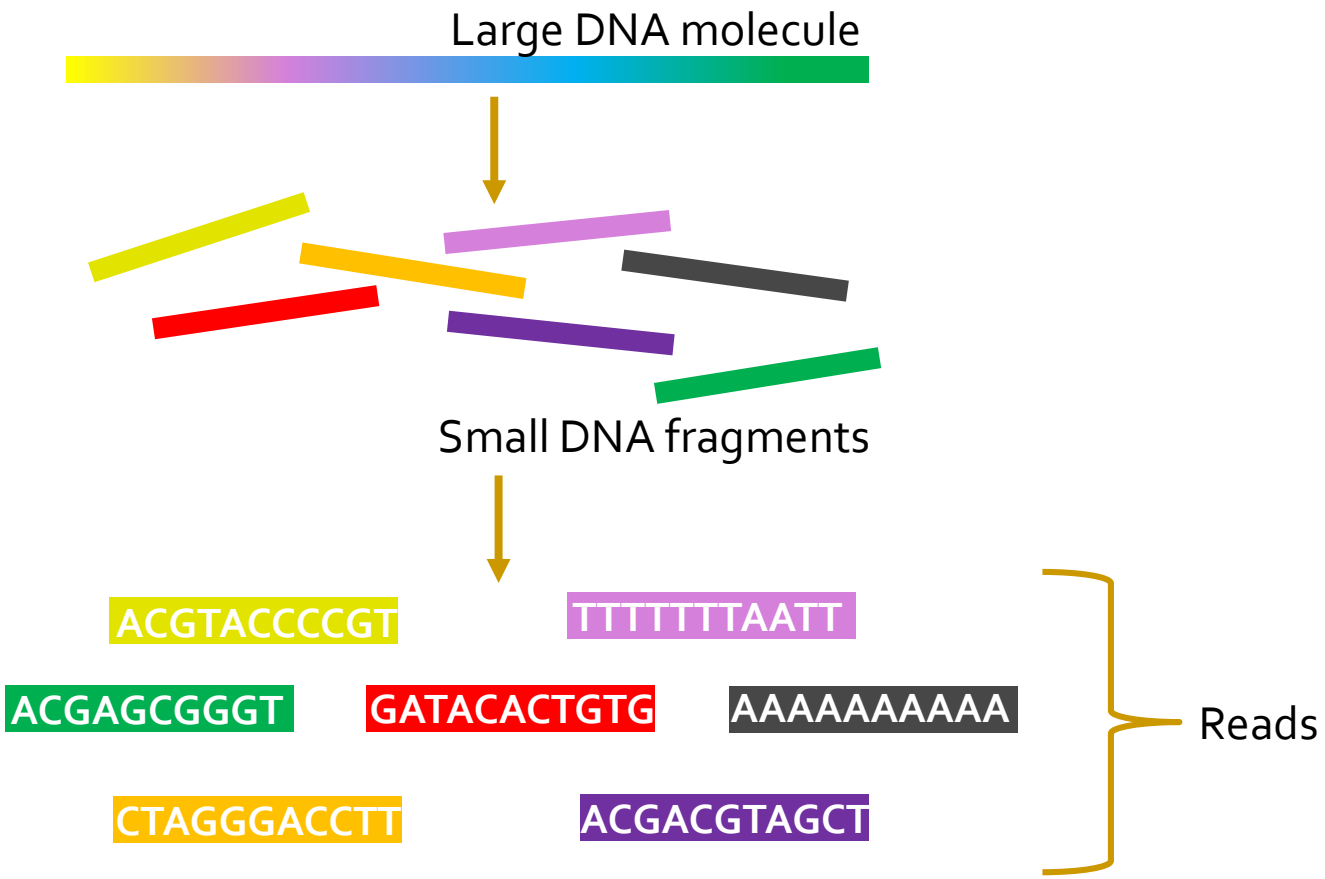
Onion, Allium Cepa
16 Billion bp

Paris Japonica
149 Billion bp

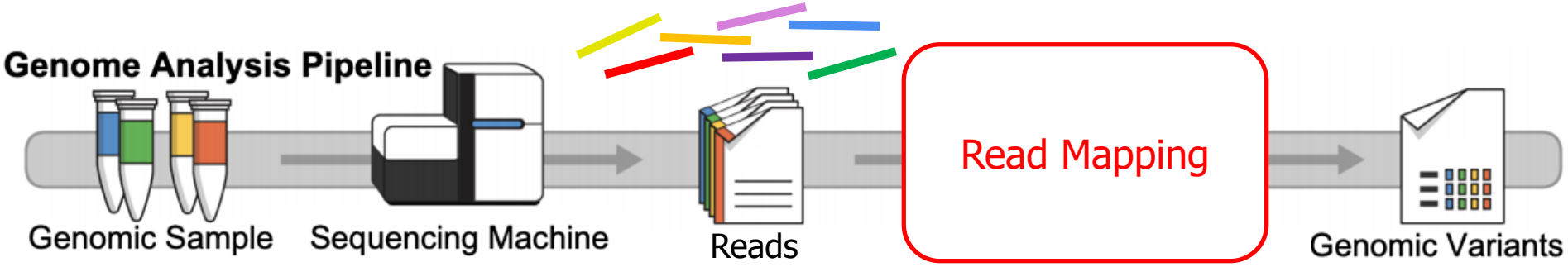
DNA Sequencing

- 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



Genome Sequencing and Analysis



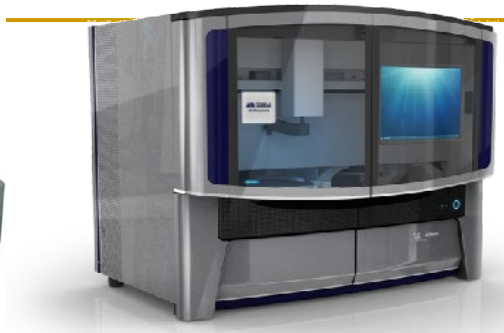
Current sequencing machines provide
small randomized fragments
of the original DNA sequence

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

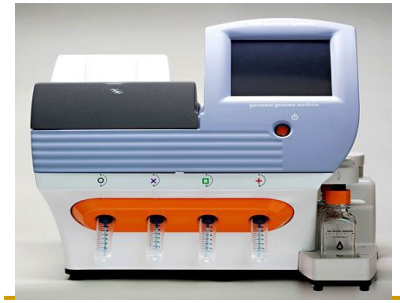
Oxford Nanopore MinION

Illumina HiSeq2000

Pacific Biosciences RS

Illumina NovaSeq 6000

Oxford Nanopore GridION



Ion Torrent PGM

Ion Torrent Proton

… and more! All produce data with different properties.

High-Throughput Sequencers

Illumina MiSeq

Pacific Biosciences Sequel II

Oxford Nanopore PromethION

Oxford Nanopore MinION

Illumina NovaSeq 6000

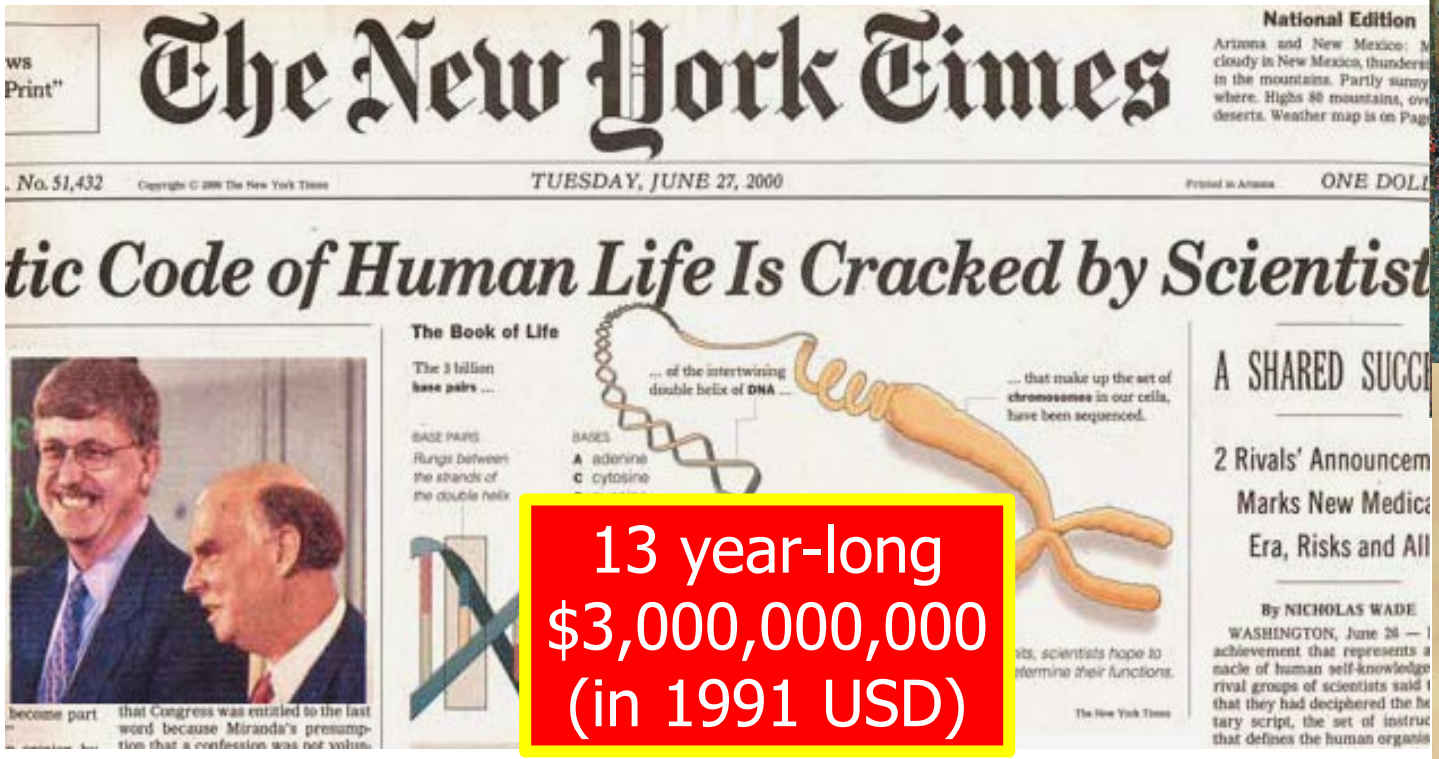
Pacific Biosciences RS II

Oxford Nanopore SmidgION

… and more! All produce data with different properties.

The Genomic Era

- 1990-2003: The Human Genome Project (HGP) provides a complete and accurate sequence of all **DNA base pairs** that make up the human genome and finds 20,000 to 25,000 human genes.


The New York Times

National Edition

No. 51,432 TUESDAY, JUNE 27, 2000

tic Code of Human Life Is Cracked by Scientist

The Book of Life

A SHARED SUCC

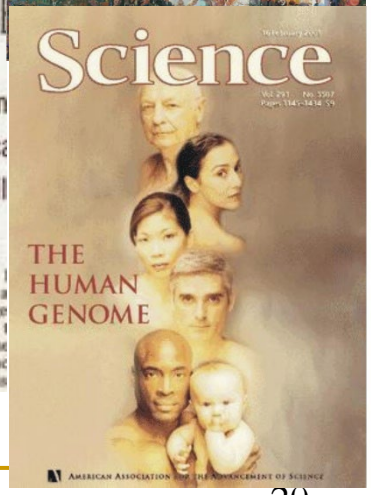
2 Rivals' Announcem

Marks New Medica

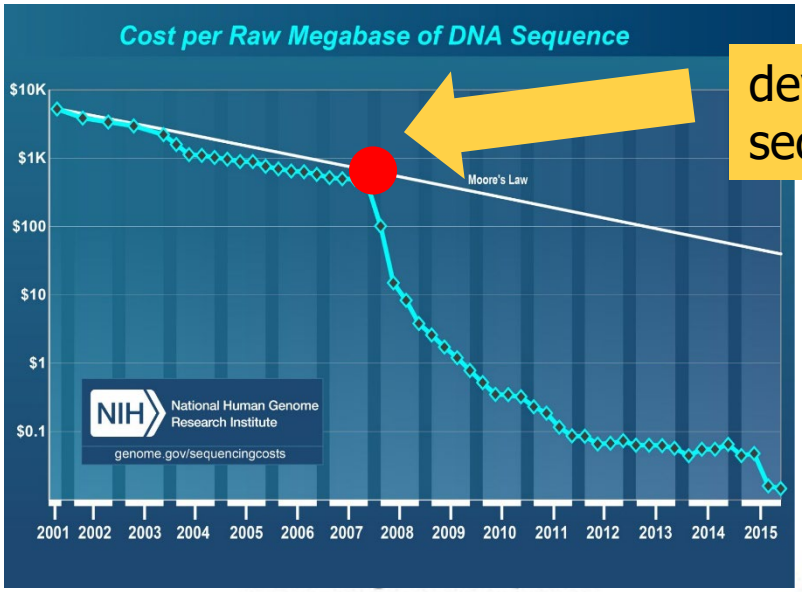
Era, Risks and All

By NICHOLAS WADE

13 year-long
$3,000,000,000
(in 1991 USD)

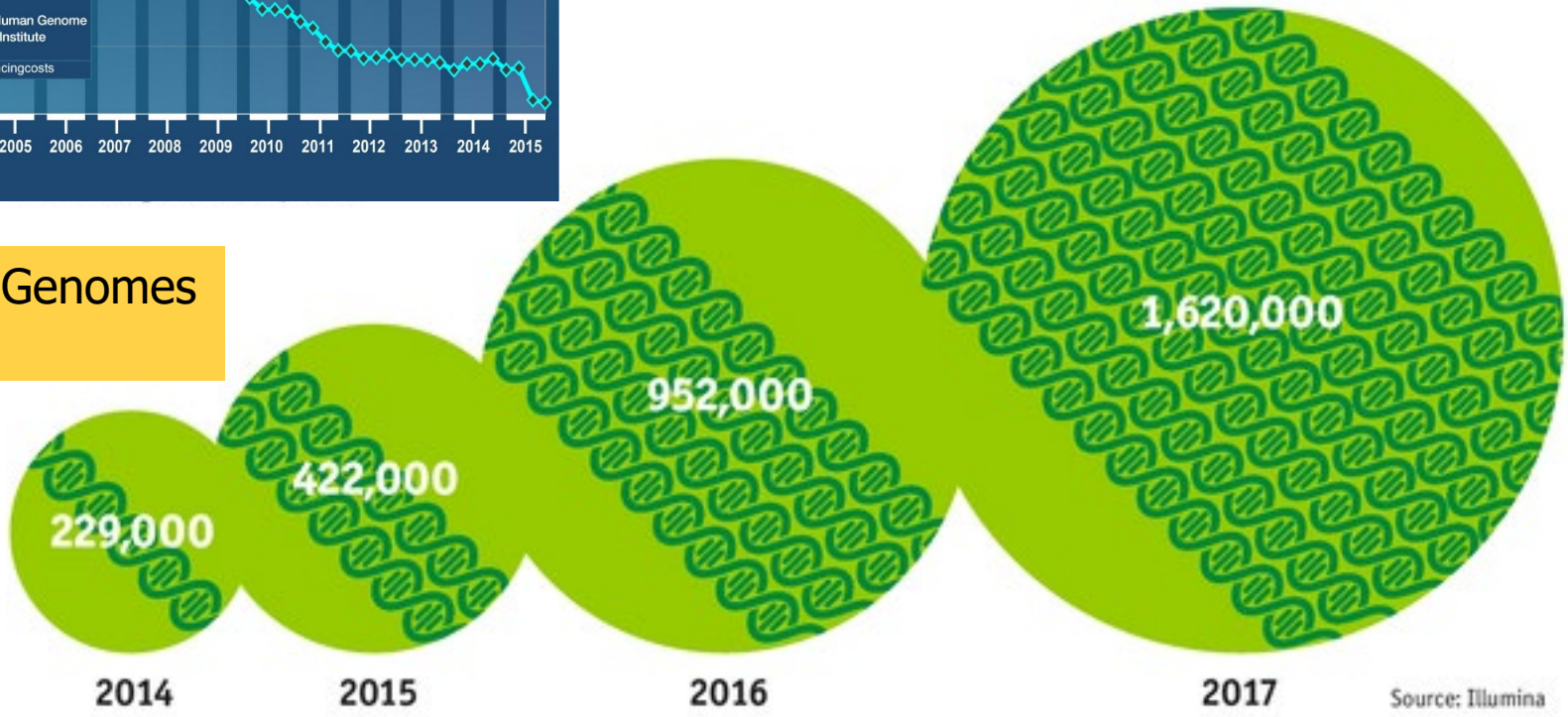


The Genomic Era (continued)



development of high-throughput sequencing (HTS) technologies

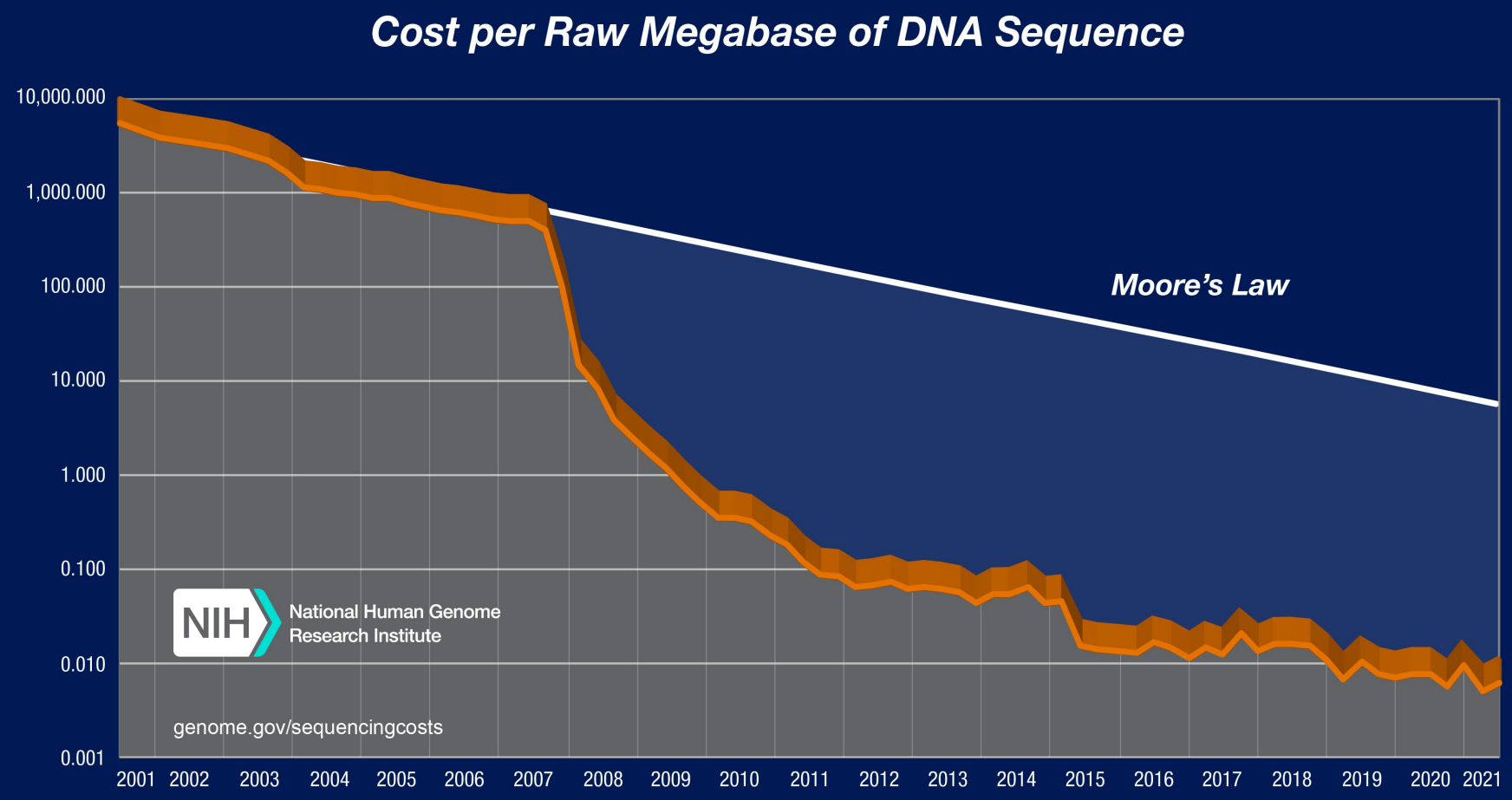
Number of Genomes Sequenced



The Economist

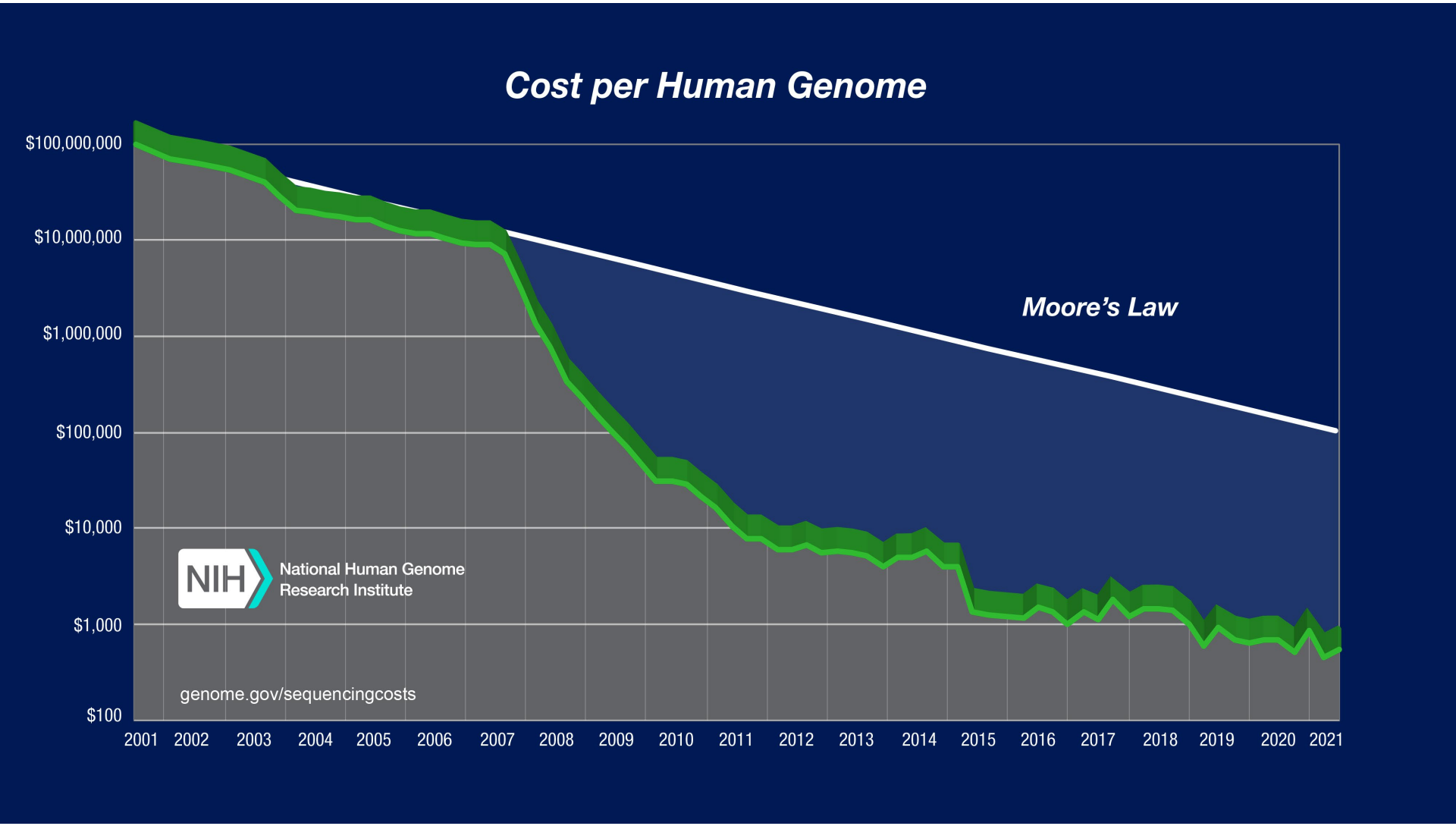
http://www.economist.com/news/21631808-so-much-genetic-data-so-many-uses-genes-unzipped

Genome Sequencing Cost Is Reducing



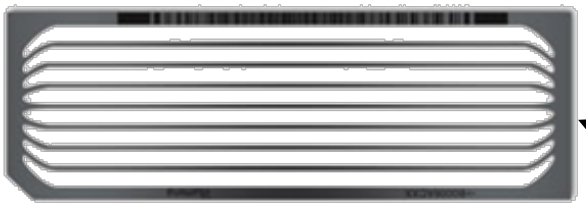
*From NIH (https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data)

Genome Sequencing Cost Is Reducing

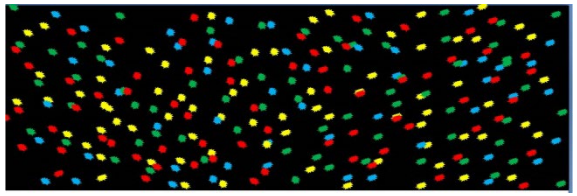


*From NIH (https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data)

High-Throughput Sequencing (HTS)



flow cell



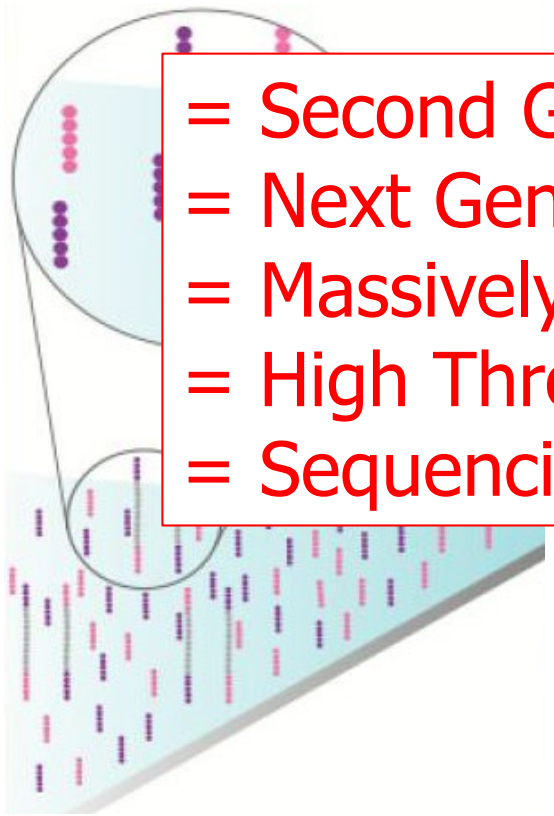
computer

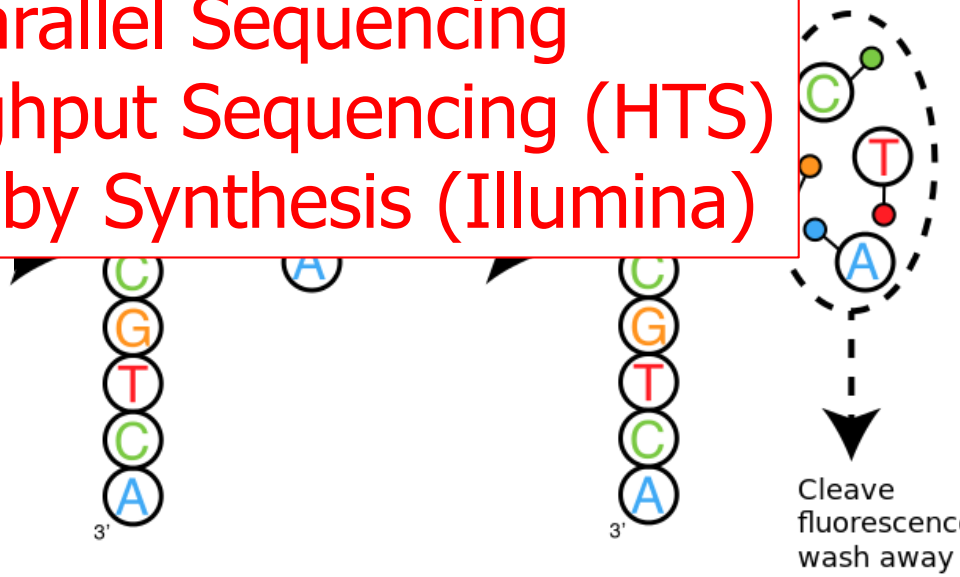
orange = G

readout

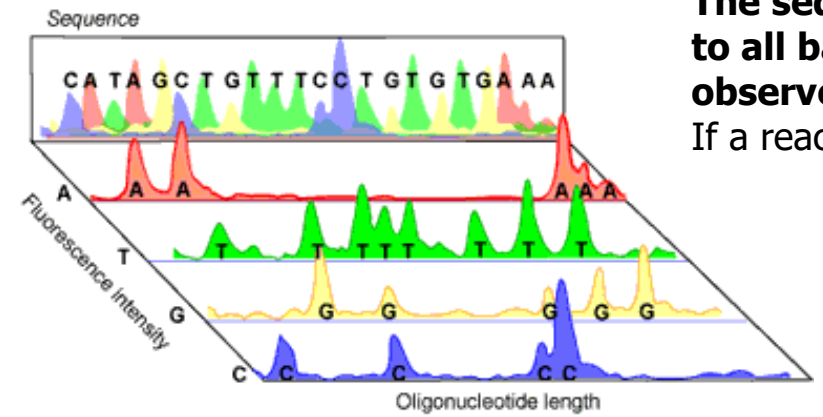
AGTG

= Second Generation
= Next Generation
= Massively Parallel Sequencing
= High Throughput Sequencing (HTS)
= Sequencing by Synthesis (Illumina)

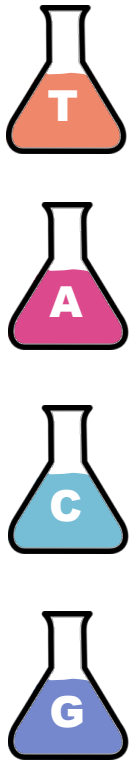


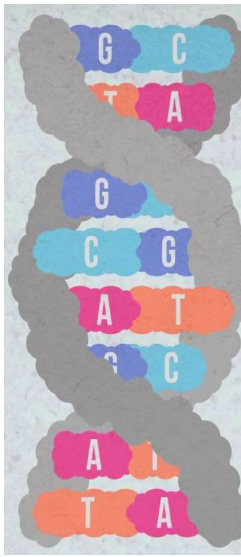


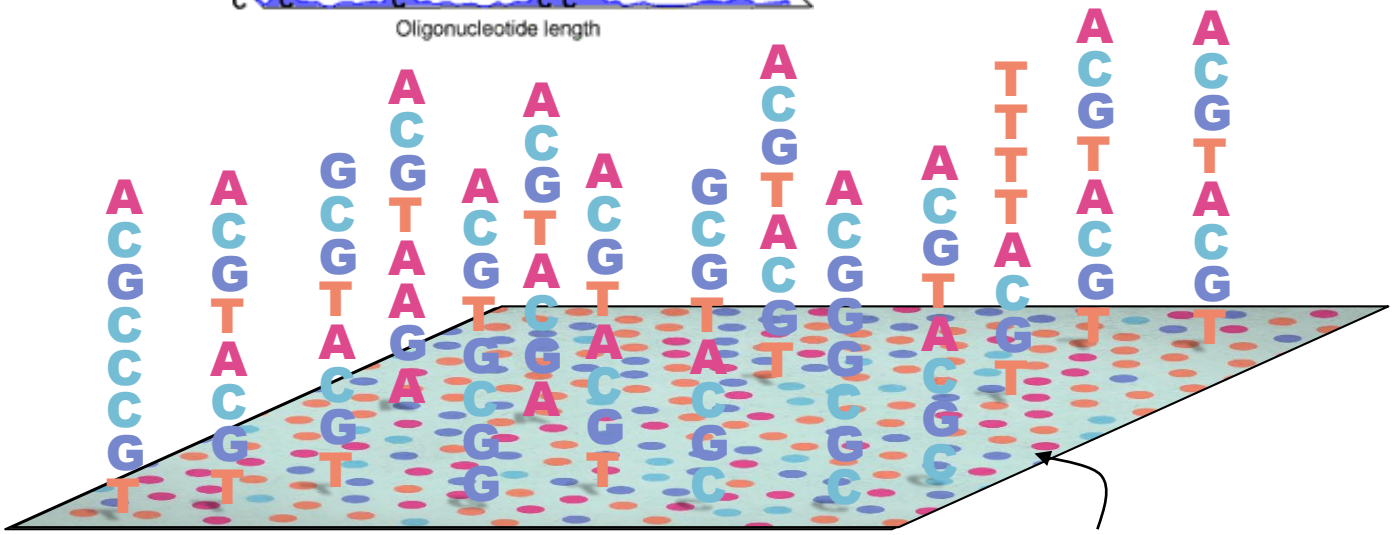
High-Throughput Sequencing (HTS)



The sequencer adds the molecule "T" to all bases near the flow cell surface and observes the chemical reaction via a CMOS sensor.
If a reaction happens then the base is "A"







As a workaround, HTS technologies sequence random short DNA fragments (75-300 basepairs long) of copies of the original molecule.

High-Throughput Sequencing

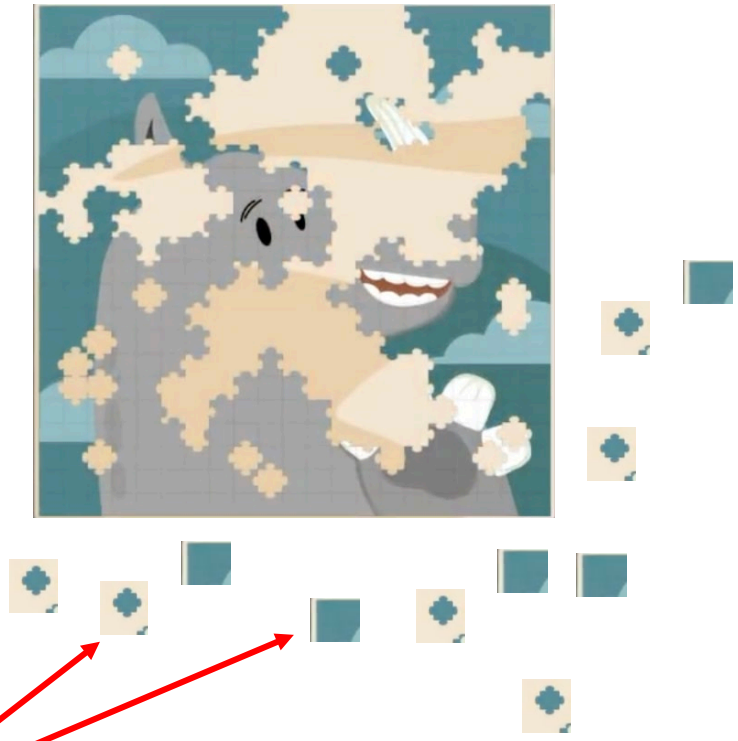
- Massively parallel sequencing technology
  - Illumina, Roche 454, Ion Torrent, SOLID…
- Small DNA fragments are first amplified and then sequenced in parallel, leading to
  - High throughput
  - High speed
  - Low cost
  - Short reads
- Sequencing is done by either reading optical signals as each base is added, or by detecting hydrogen ions instead of light, leading to:
  - Low error rates (relatively)
  - Reads lack information about their order and which part of genome they are originated from

Solving the Puzzle

.FASTA file

Reference genome

.FASTQ file



Sequenced Reads

https://www.pacb.com/smrt-science/smrt-sequencing/hifi-reads-for-highly-accurate-long-read-sequencing/

Newer Genome Sequencing Technologies

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

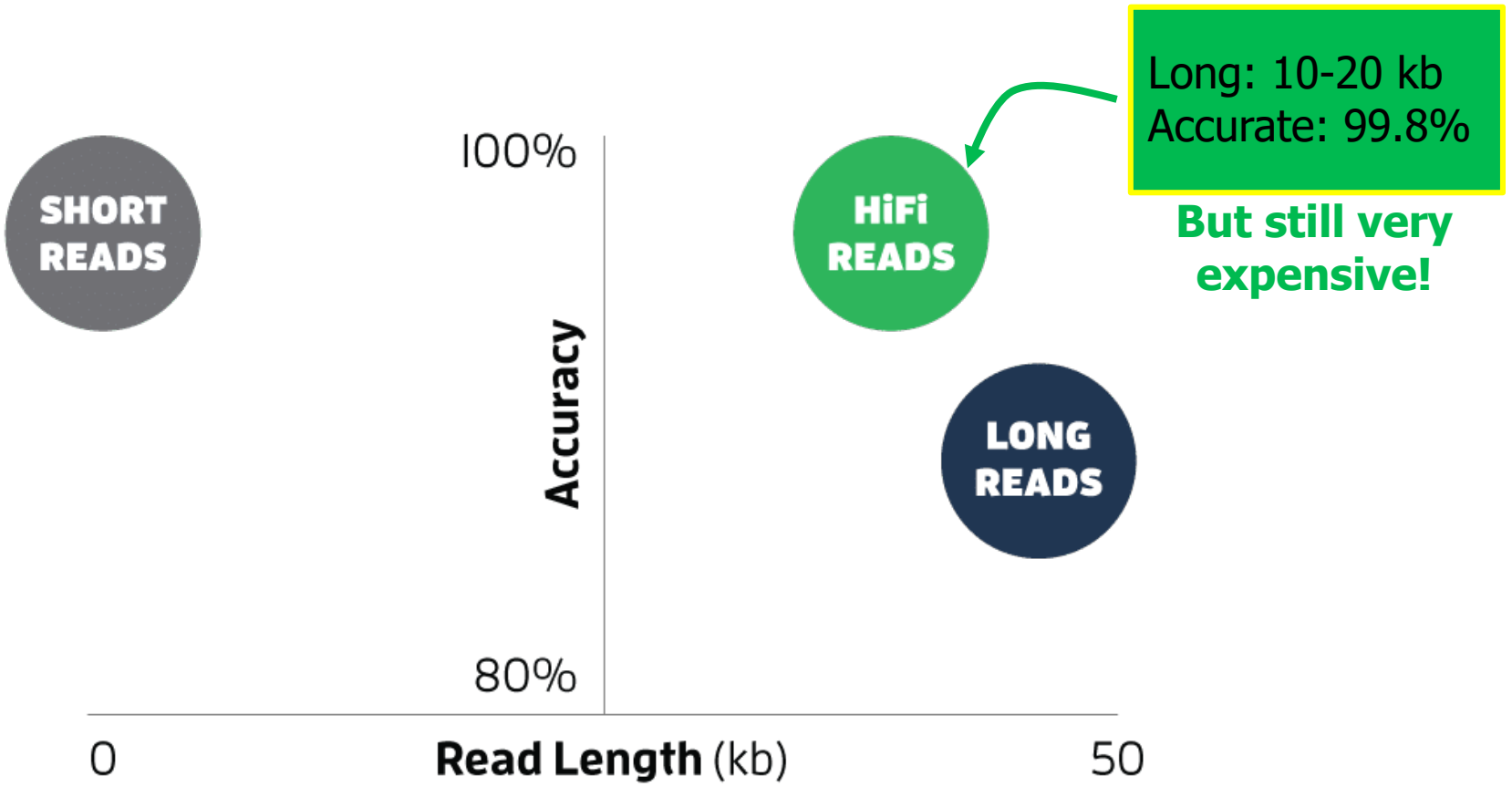
Damla Senol Cali ✉, Jeremie S Kim, Saugata Ghose, Can Alkan, Onur Mutlu

Briefings in Bioinformatics, bby017, https://doi.org/10.1093/bib/bby017
Published: 02 April 2018 **Article history** ▾

Oxford Nanopore MinION

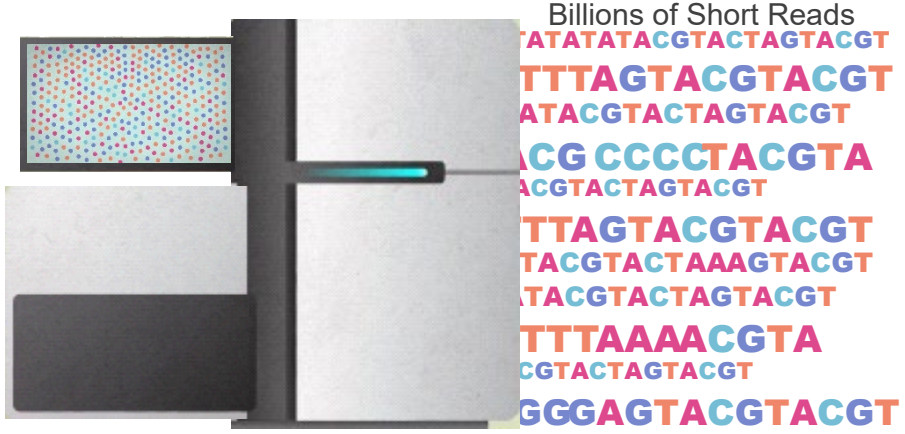
Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions**," Briefings in Bioinformatics, 2018.
[Open arxiv.org version] [Slides (pptx) (pdf)] [Talk Video at AACBB 2019]

Types of Genomic Reads

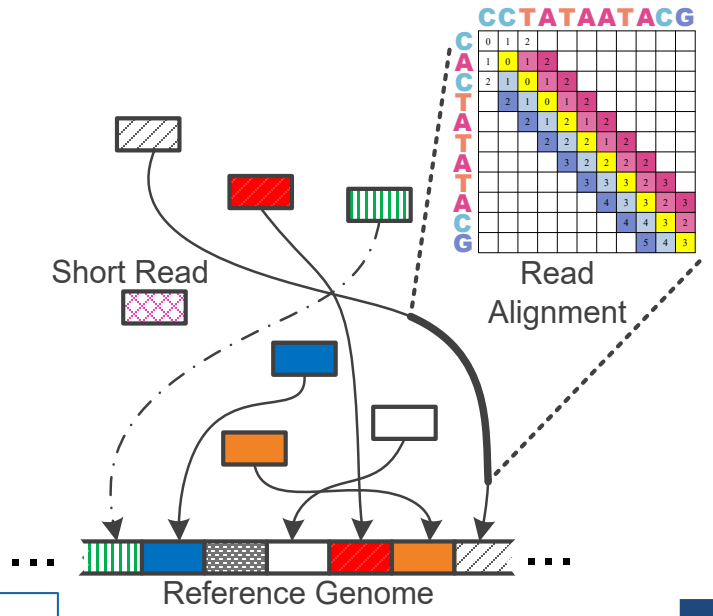


Wenger+, "Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome", *Nature Biotechnology*, 2019

Genome Analysis



1 Sequencing



Read Mapping 2

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

3 Variant Calling

Scientific Discovery 4

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]

Alser *et al. Genome Biology* (2021) 22:249
https://doi.org/10.1186/s13059-021-02443-7

Genome Biology

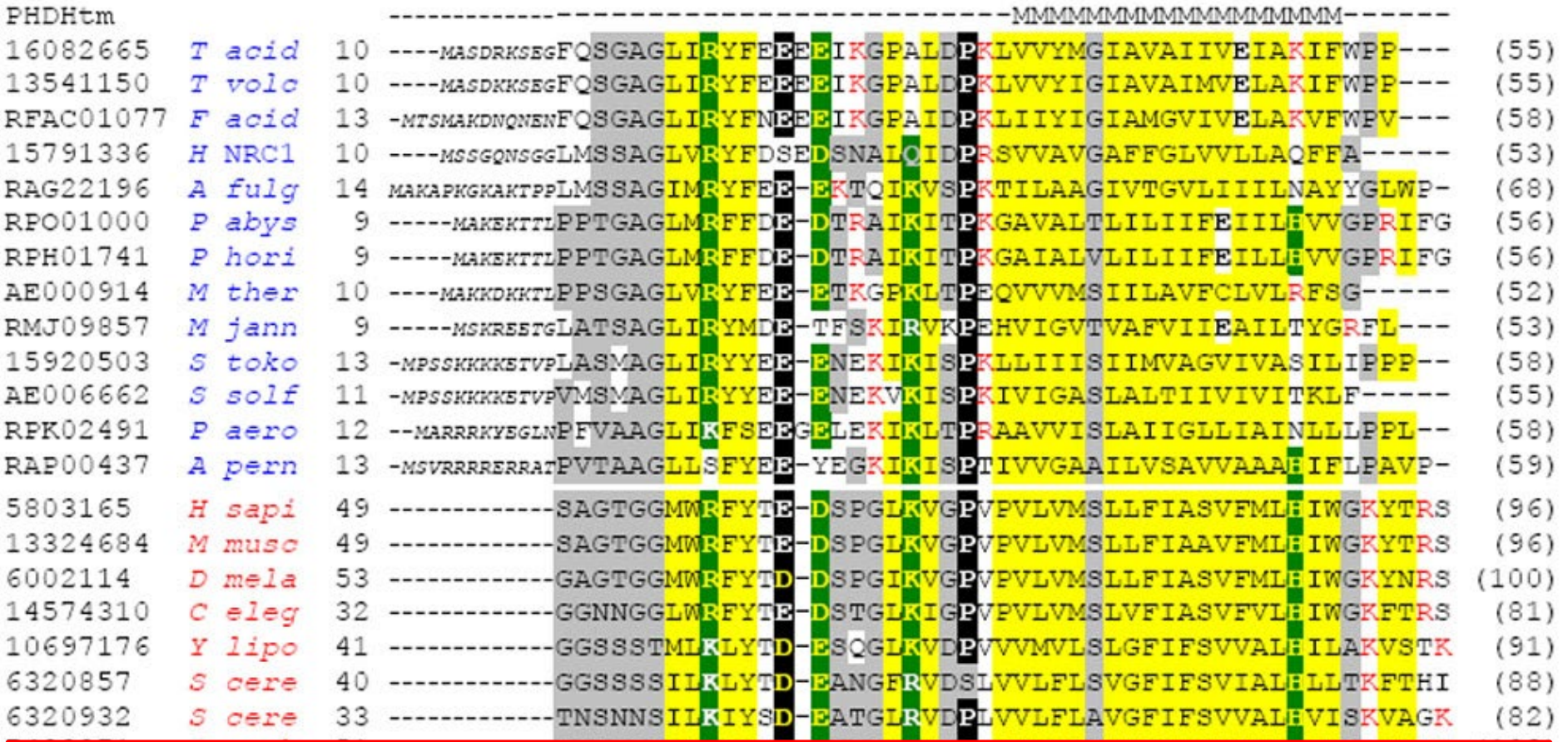
REVIEW Open Access

Technology dictates algorithms: recent developments in read alignment

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

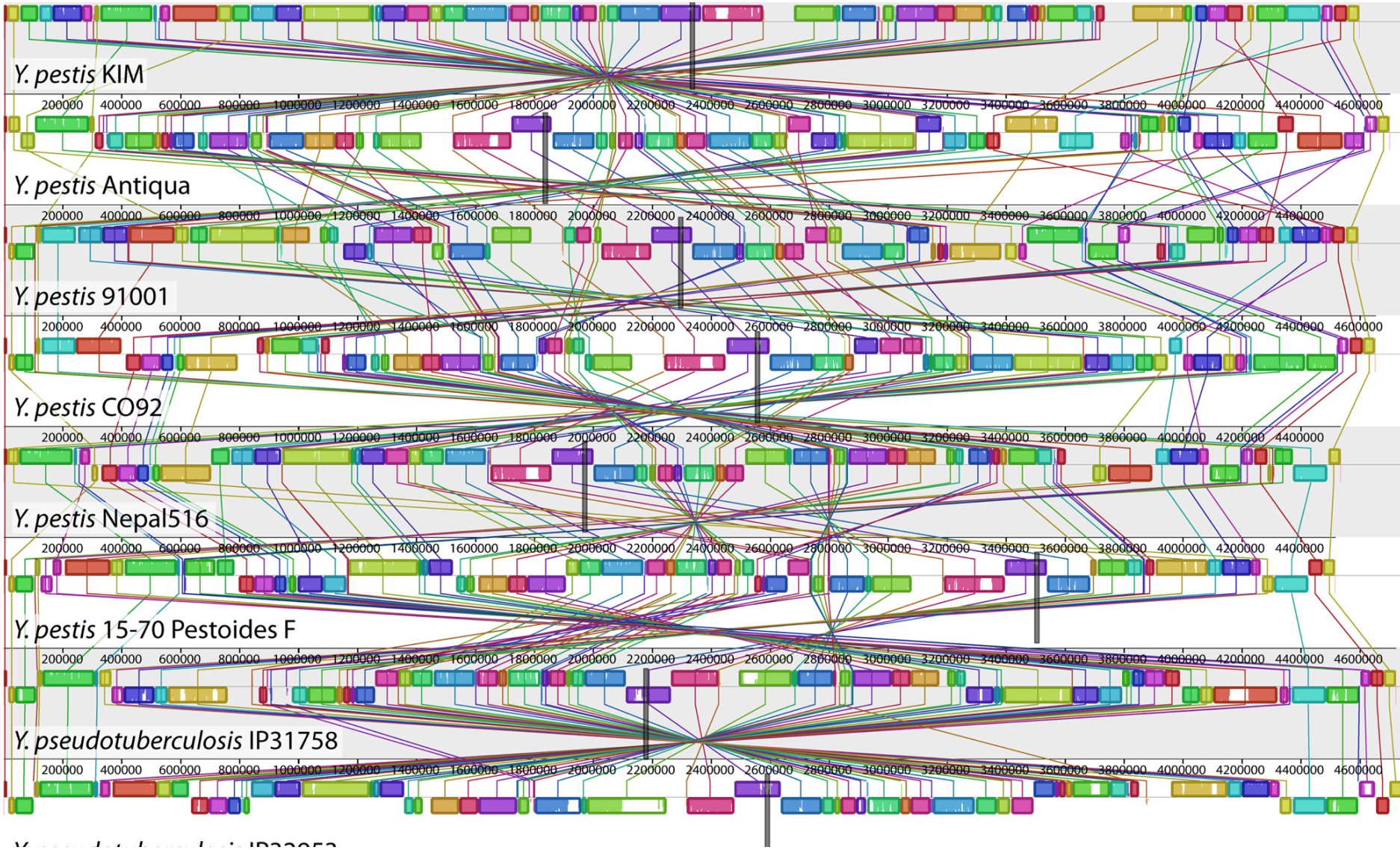
Why Do We Care?

Multiple sequence alignment



Example Question: If I give you a bunch of sequences, tell me where they are the same and where they are different.

Genome Sequence Alignment: Example



Source: By Aaron E. Darling, István Miklós, Mark A. Ragan - Figure 1 from Darling AE, Miklós I, Ragan MA (2008). "Dynamics of Genome Rearrangement in Bacterial Populations". PLOS Genetics. DOI:10.1371/journal.pgen.1000128., CC BY 2.5, https://commons.wikimedia.org/w/index.php?curid=30550950

The Genetic Similarity Between Species

Human ~ Human

99.9%

Human ~ Chimpanzee

96%

Human ~ Cat

90%

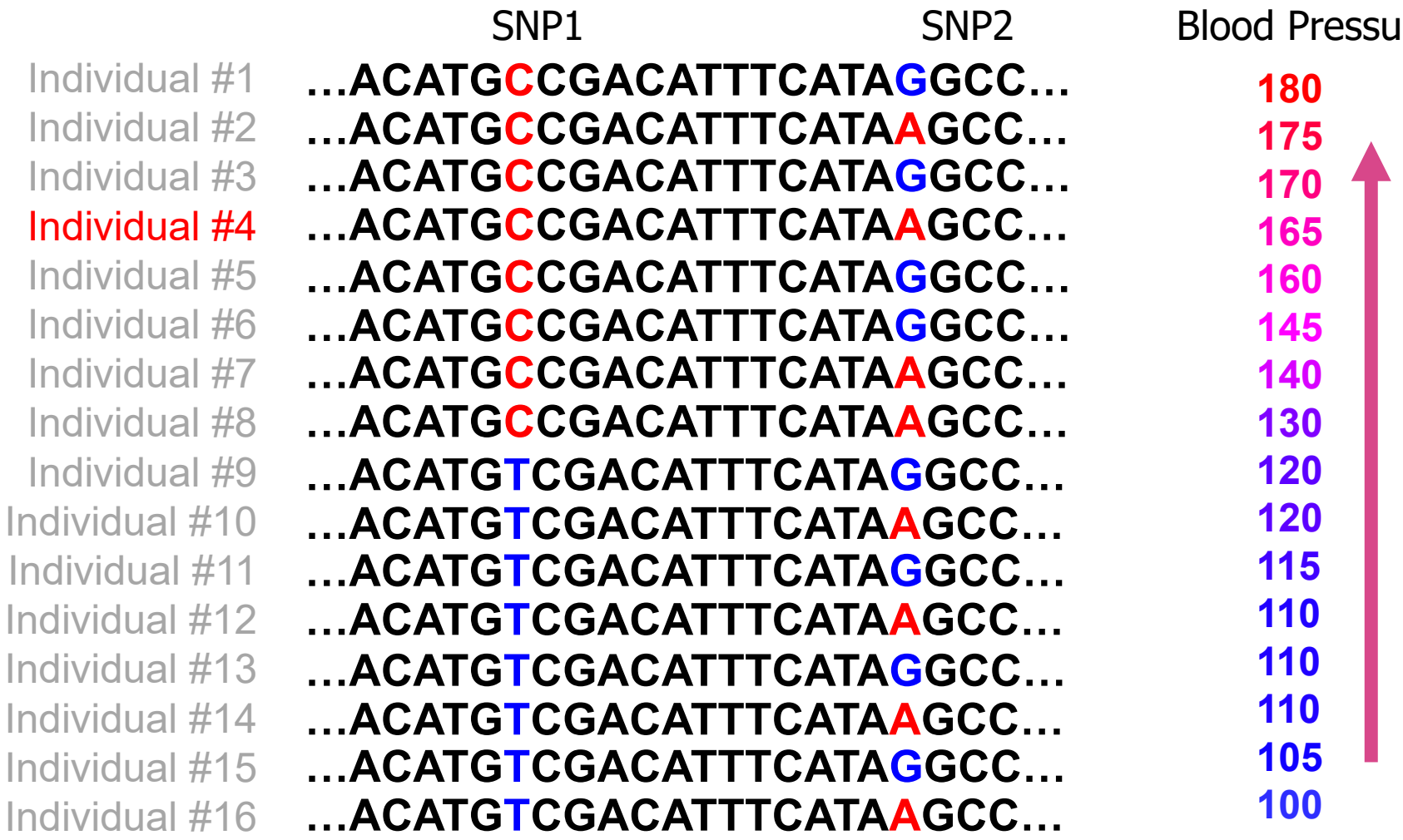
Human ~ Cow

80%

Human ~ Banana

50-60%

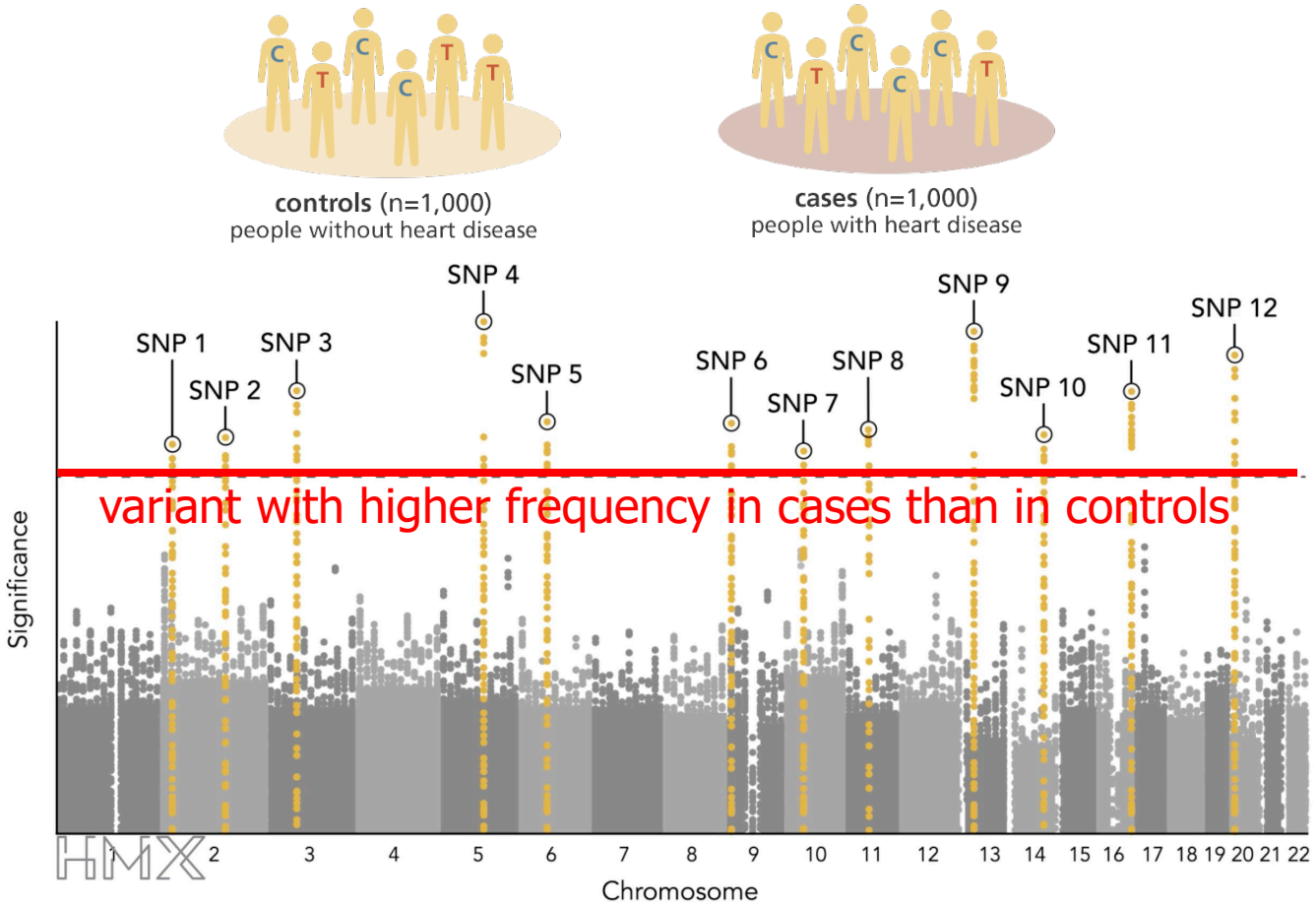
Finding Variations Associated with Traits



SNP: single nucleotide polymorphism

Genome-Wide Association Studies (GWAS)

- Enables detection of genetic variants associated with phenotypes using two groups of people.



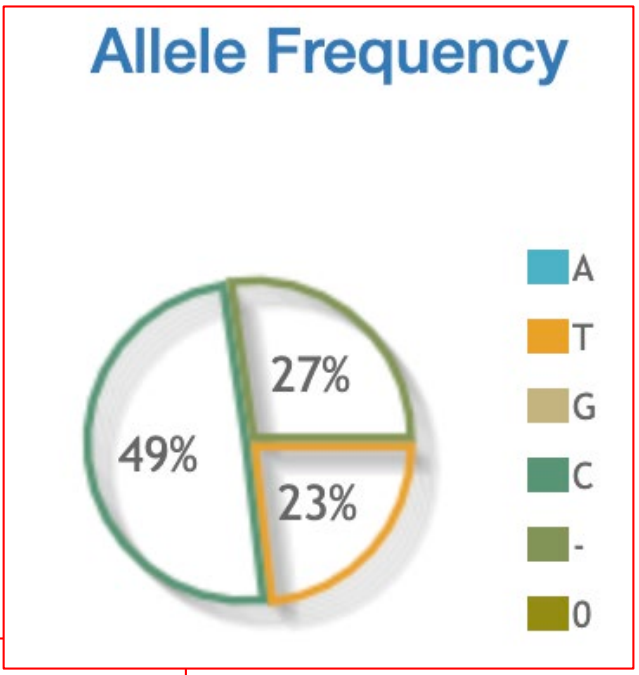
SNPs and Personalized Medicine

openSNP Search

SNP rs12979860

Basic Information

Name	rs12979860
Chromosome	19
Position	39248147
Weight of evidence	926



Links to SNPedia

Title	Summary
rs12979860 T/T	~20-25% of such hepatitis c patients respond to treatment
rs12979860 C/C	~80% of such hepatitis c patients respond to treatment
rs12979860 C/T	~20-40% of such hepatitis c patients respond to treatment

https://opensnp.org/snps/rs12979860

Much Larger Structural Variations

AUTISM
Weiss, *N Eng J Med* 2008
Deletion of 593 kb

SCHIZOPHRENIA
McCarthy, *Nat Genet* 2009
Duplication of 593 kb

OBESITY
Walters, *Nature* 2010
Deletion of 593 kb

UNDERWEIGHT
Jacquemont, *Nature* 2011
Duplication of 593 kb

Deletion in the short arm
of chromosome 16 (16p11.2)

Duplication in the short arm
of chromosome 16 (16p11.2)

Personalized Medicine for Critically Ill Infants

- rWGS can be performed in 2-day (costly) or 5-day time to interpretation.
- Diagnostic rWGS for infants
  - Avoids morbidity
  - Reduces hospital stay length by 6%-69%
  - Reduces inpatient cost by $800,000-$2,000,000.

Article | Open Access | Published: 04 April 2018

Rapid whole-genome sequencing decreases infant morbidity and cost of hospitalization

Lauge Farnaes, Amber Hildreth, Nathaly M. Sweeney, Michelle M. Clark, S
Chowdhury, Shareef Nahas, Julie A. Cakici, Wendy Benson, Robert H. Kap
Richard Kronick, Matthew N. Bainbridge, Jennifer Friedman, Jeffrey J. Go
Ding, Narayanan Veeraraghavan, David Dimmock & Stephen F. Kingsmore

npj Genomic Medicine **3**, Article number: 10 (2018) | Cite this article

Article | Open Access | Published: 05 May 2020

Clinical utility of 24-h rapid trio-exome sequencing for critically ill infants

Huijun Wang, Yanyan Qian, Yulan Lu, Qian Qin, Guoping Lu, Guoqiang Cheng, Ping Zhang, Lin Yang, Bingbing Wu ✉ & Wenhao Zhou ✉

npj Genomic Medicine **5**, Article number: 20 (2020) | Cite this article

Farnaes+, "Rapid whole-genome sequencing decreases infant morbidity and cost of hospitalization", NPJ Genomic Medicine, 2018

Recommended Reading

nature reviews genetics

Explore our content ⌄ Journal information ⌄

nature › nature reviews genetics › review articles › article

Review Article | Published: 15 November 2019

Structural variation in the sequencing era

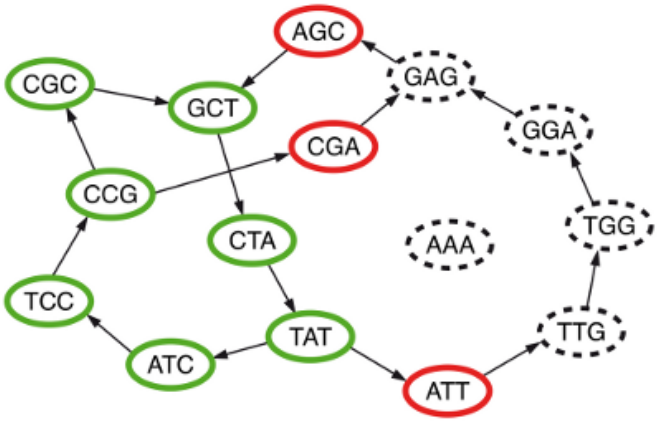
Steve S. Ho, Alexander E. Urban & Ryan E. Mills ✉

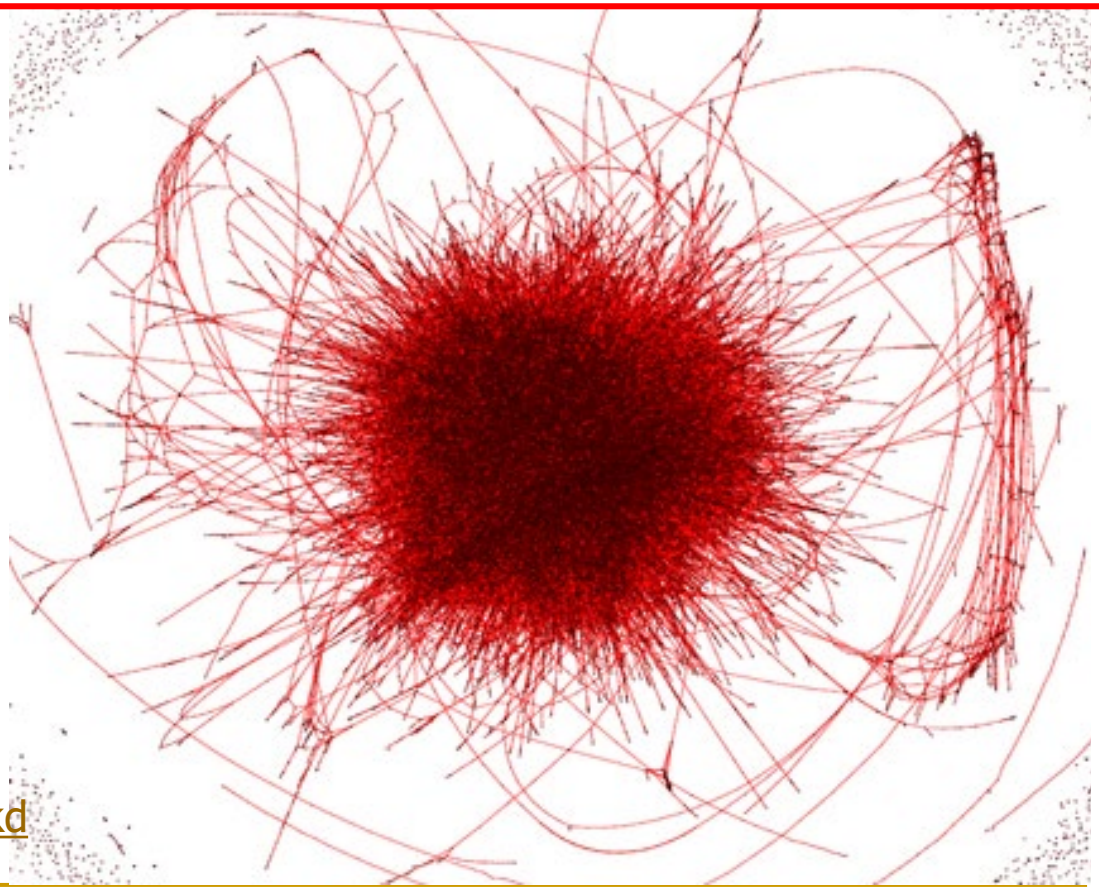
Nature Reviews Genetics **21**, 171–189(2020) | Cite this article

15k Accesses | **16** Citations | **309** Altmetric | Metrics

Ho+, "Structural variation in the sequencing era", Nature Reviews Genetics, 2020

Question 2: Given a bunch of short sequences, Can you identify the approximate species cluster for genomically unknown organisms (bacteria)?



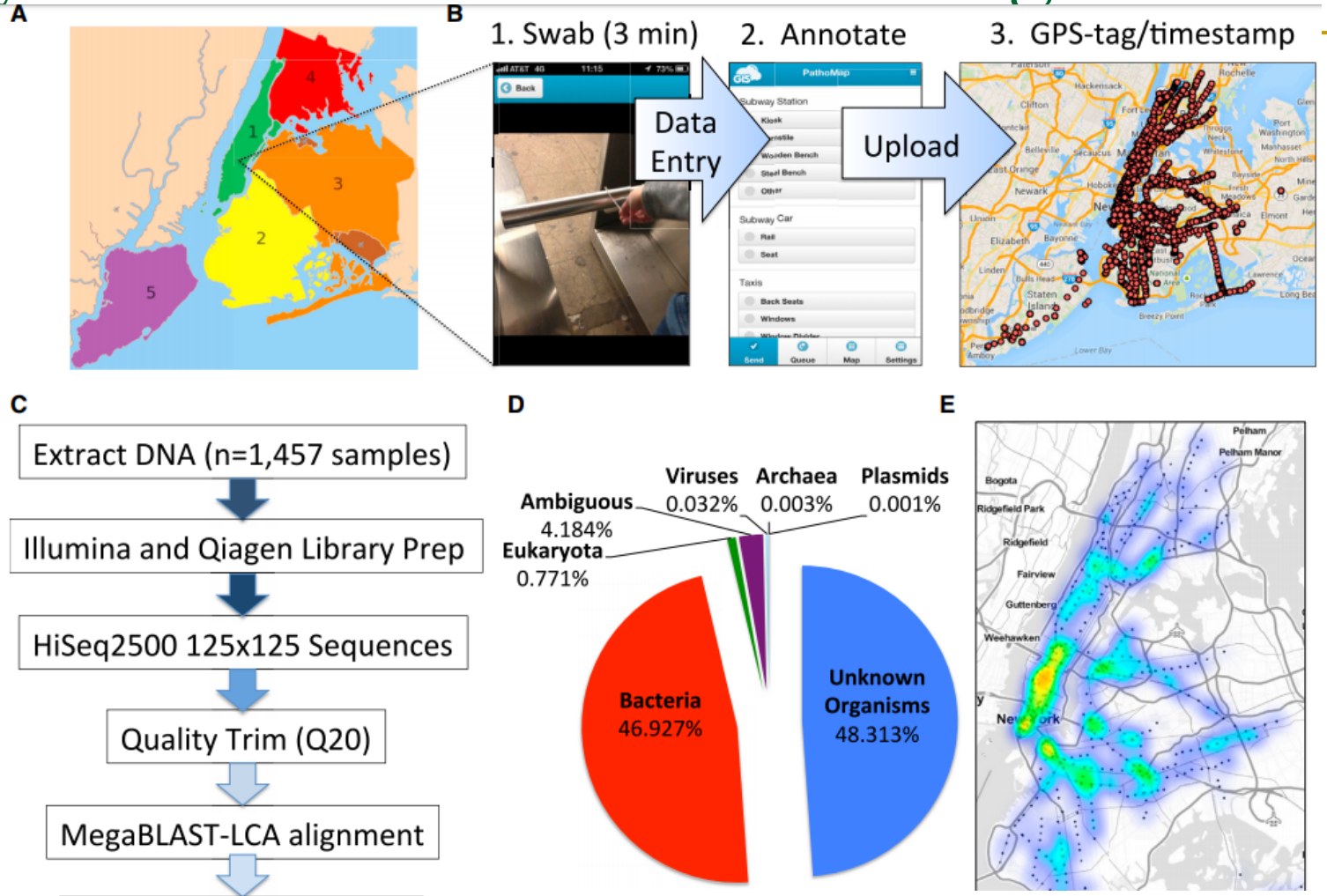


uncleaned de Bruijn graph

http://math.oregonstate.edu/~koslickd

Population-Scale Microbiome Profiling

City-Scale Microbiome Profiling



Afshinnekoo+, "Geospatial Resolution of Human and Bacterial Diversity with City-Scale Metagenomics", Cell Systems, 2015

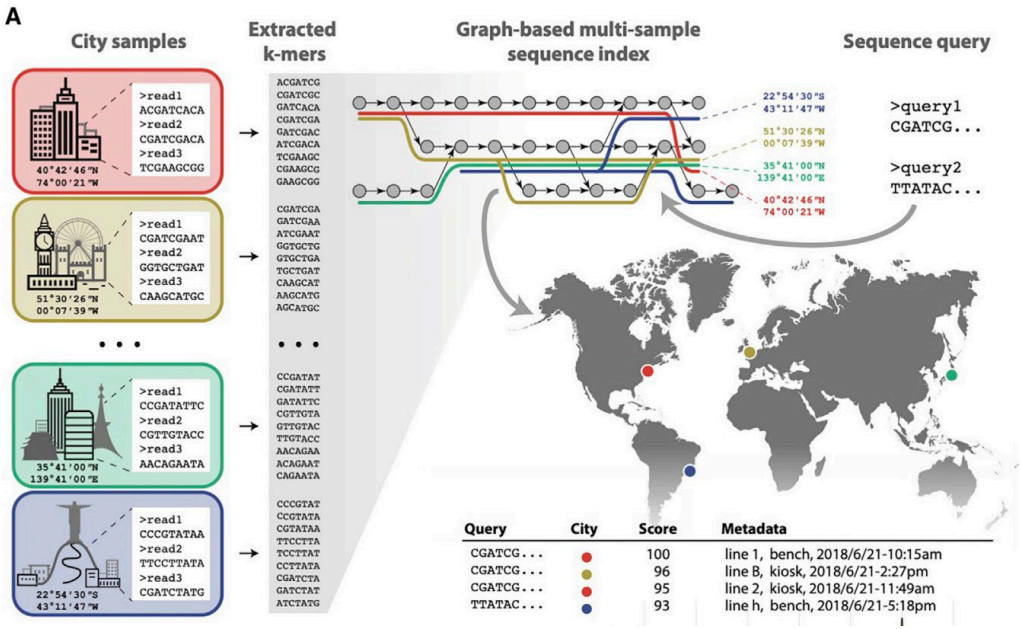
Figure 1. The Metagenome of New York City

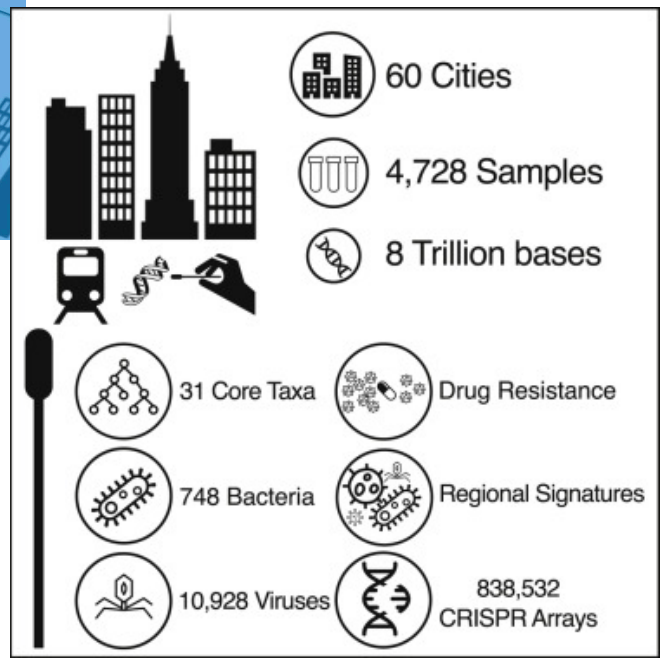
(A) The five boroughs of NYC include (1) Manhattan (green)

(B) The collection from the 466 subway stations of NYC across the 24 subway lines involved three main steps: (1) collection with Copan Elution swabs, (2) data entry into the database, and (3) uploading of the data. An image is shown of the current collection database, taken from http://pathomap.giscloud.com.

(C) Workflow for sample DNA extraction, library preparation, sequencing, quality trimming of the FASTQ files, and alignment with MegaBLAST and MetaPhlAn to discern taxa present

Global-Scale Microbiome Profiling





Danko+, "A global metagenomic map of urban microbiomes and antimicrobial resistance", Cell, 2021

A Tsunami of Sequencing Data

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

Source: @kyrpides

Another Question: Example from 2020-…

200 Oxford Nanopore sequencers have left UK for China, to support rapid, near-sample coronavirus sequencing for outbreak surveillance

Fri 31st January 2020

Following extensive support of, and collaboration with, public health professionals in China, Oxford Nanopore has shipped an additional 200 MinION sequencers and related consumables to China. These will be used to support the ongoing surveillance of the current coronavirus outbreak, adding to a large number of the devices already installed in the country.

Each MinION sequencer is approximately the size of a stapler, and can provide rapid sequence information about the coronavirus.

700Kg of Oxford Nanopore sequencers and consumables are on their way for use by Chinese scientists in understanding the current coronavirus outbreak.

Source: https://nanoporetech.com/about-us/news/200-oxford-nanopore-sequencers-have-left-uk-china-support-rapid-near-sample

Example: Scalable SARS-CoV-2 Testing

BMJ Yale

HOME | ABOU

Search

Comments (1)

Swab-Seq: A high-throughput platform for massively scaled up SARS-CoV-2 testing

Joshua S. Bloom, Eric M. Jones, Molly Gasperini, Nathan B. Lubock, Laila Sathe, Chetan Munugala, A. Sina Booeshaghi, Oliver F. Brandenberg, Longhua Guo, James Boocock, Scott W. Simpkins, Isabella Lin, Nathan LaPierre, Duke Hong, Yi Zhang, Gabriel Oland, Bianca Judy Choe, Sukantha Chandrasekaran, Evann E. Hilt, Manish J. Butte, Robert Damoiseaux, Aaron R. Cooper, Yi Yin, Lior Pachter, Omai B. Garner, Jonathan Flint, Eleazar Eskin, Chongyuan Luo, Sriram Kosuri, Leonid Kruglyak, Valerie A. Arboleda

doi: https://doi.org/10.1101/2020.08.04.20167874

Bloom+, "Swab-Seq: A high-throughput platform for massively scaled up SARS-CoV-2 testing", *medRxiv*, 2020

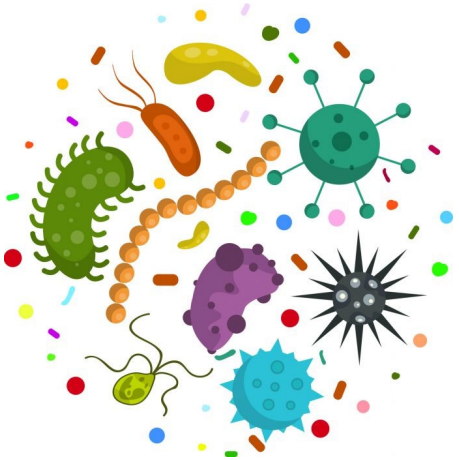
Example: Rapid Surveillance of Ebola Outbreak

Figure 1: Deployment of the portable genome surveillance system in Guinea.

Quick+, "Real-time, portable genome sequencing for Ebola surveillance", *Nature*, 2016

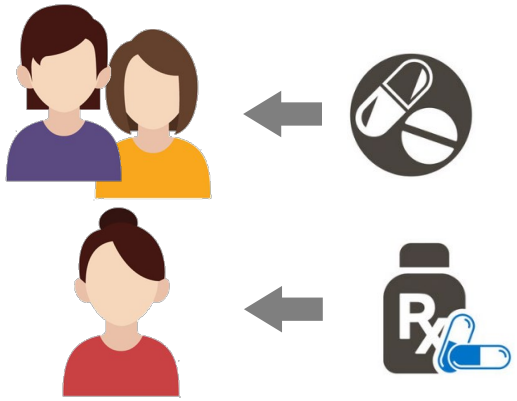
We Need Faster & Scalable Genome Analysis

Understanding **genetic variations, species, evolution, …**



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

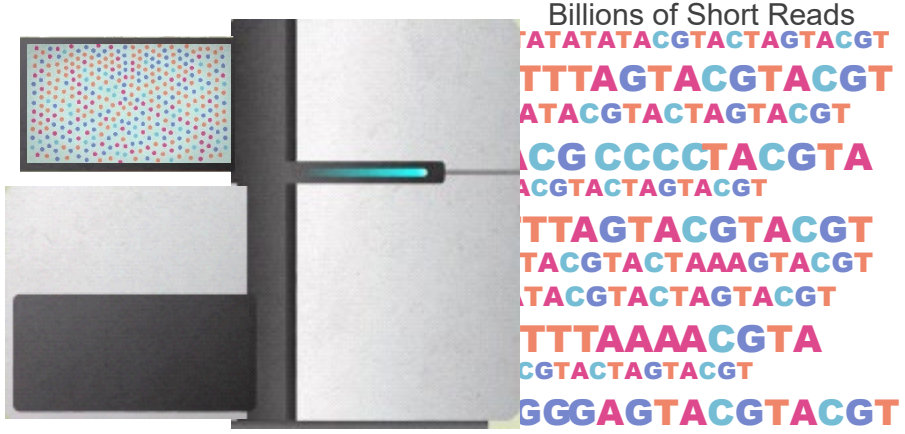
Rapid surveillance of **disease outbreaks**



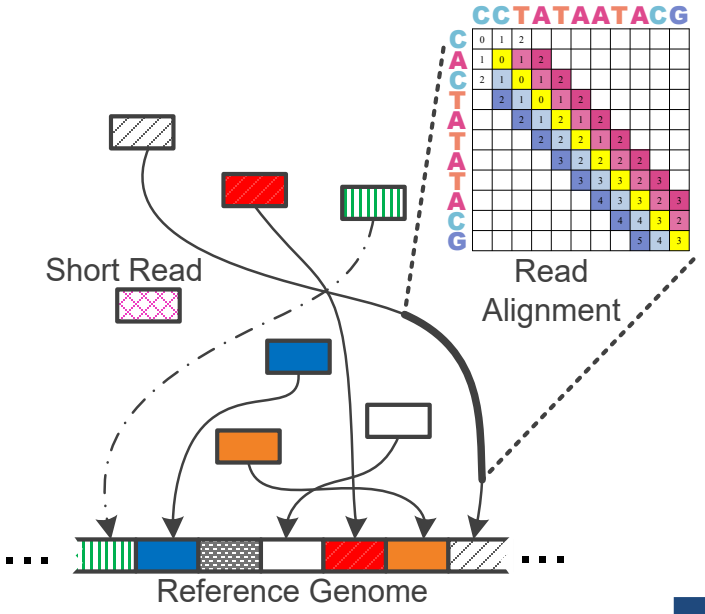
Developing **personalized medicine**

And, many, many other applications …

One Problem

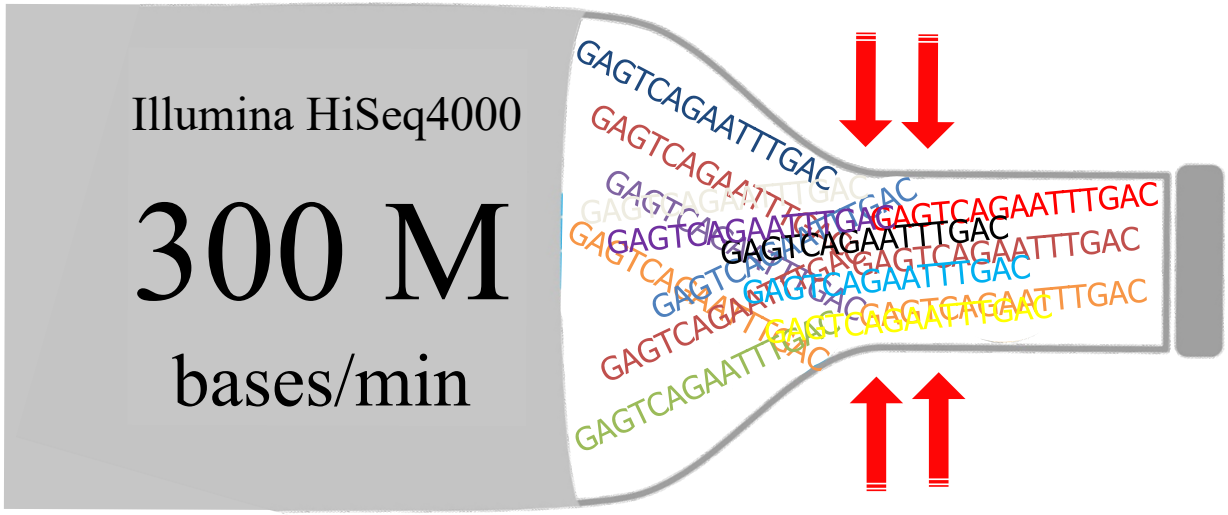


1 Sequencing



Read Mapping 2

We Are Bottlenecked in Read Mapping



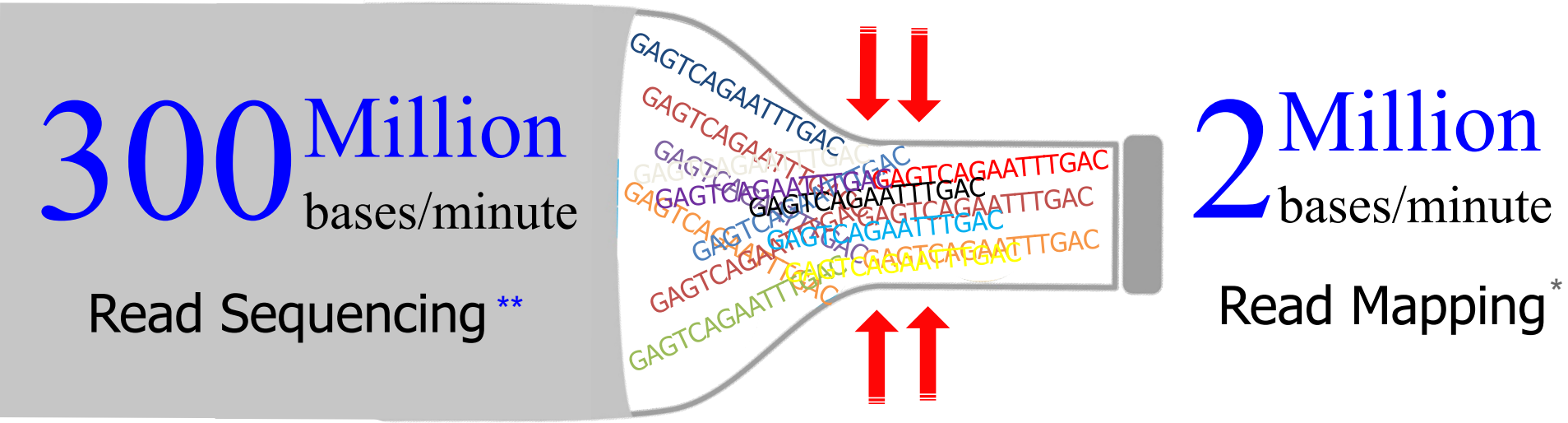
on average

2 M

bases/min

(0.6%)

The Read Mapping Bottleneck



150x slower

* BWA-MEM
** HiSeqX10, MinION

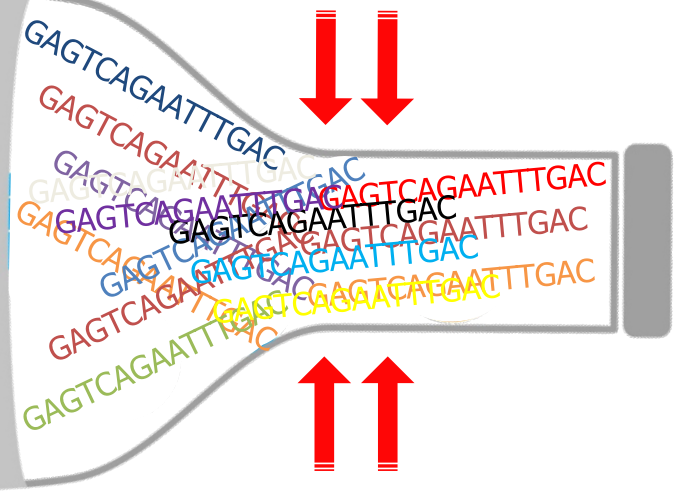
The Read Mapping Bottleneck

48 Human whole genomes

at 30× coverage

in about 2 days

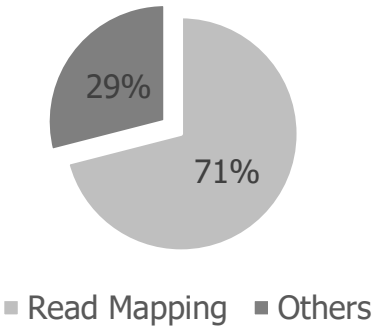
Illumina NovaSeq 6000



1 Human genome

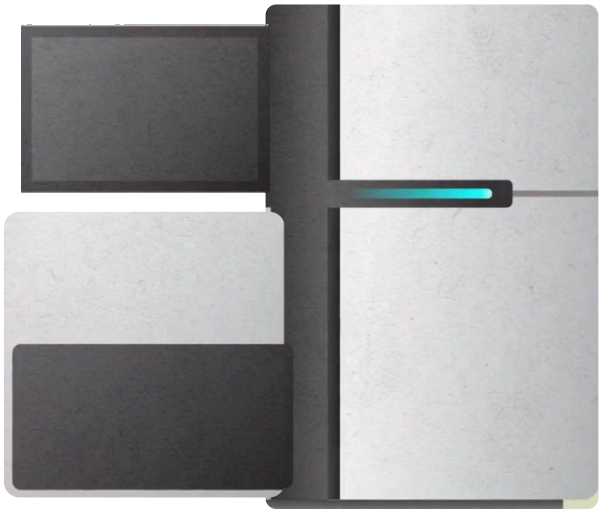
32 CPU hours

on a 48-core processor



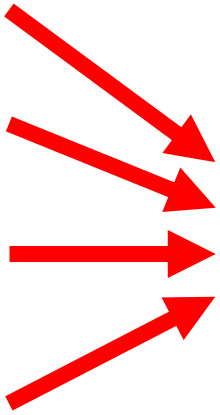
Goyal+, "Ultra-fast next generation human genome sequencing data processing using DRAGENTM bio-IT processor for precision medicine", *Open Journal of Genetics*, 2017.

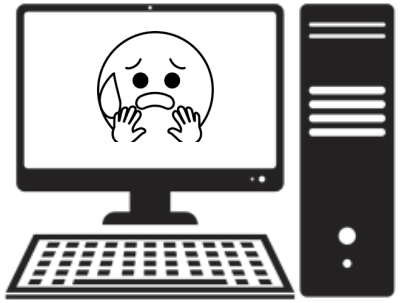
Problem with (Genome) Analysis Today



Special-Purpose Machine
for **Data Generation**

FAST





General-Purpose Machine
for **Data Analysis**

SLOW

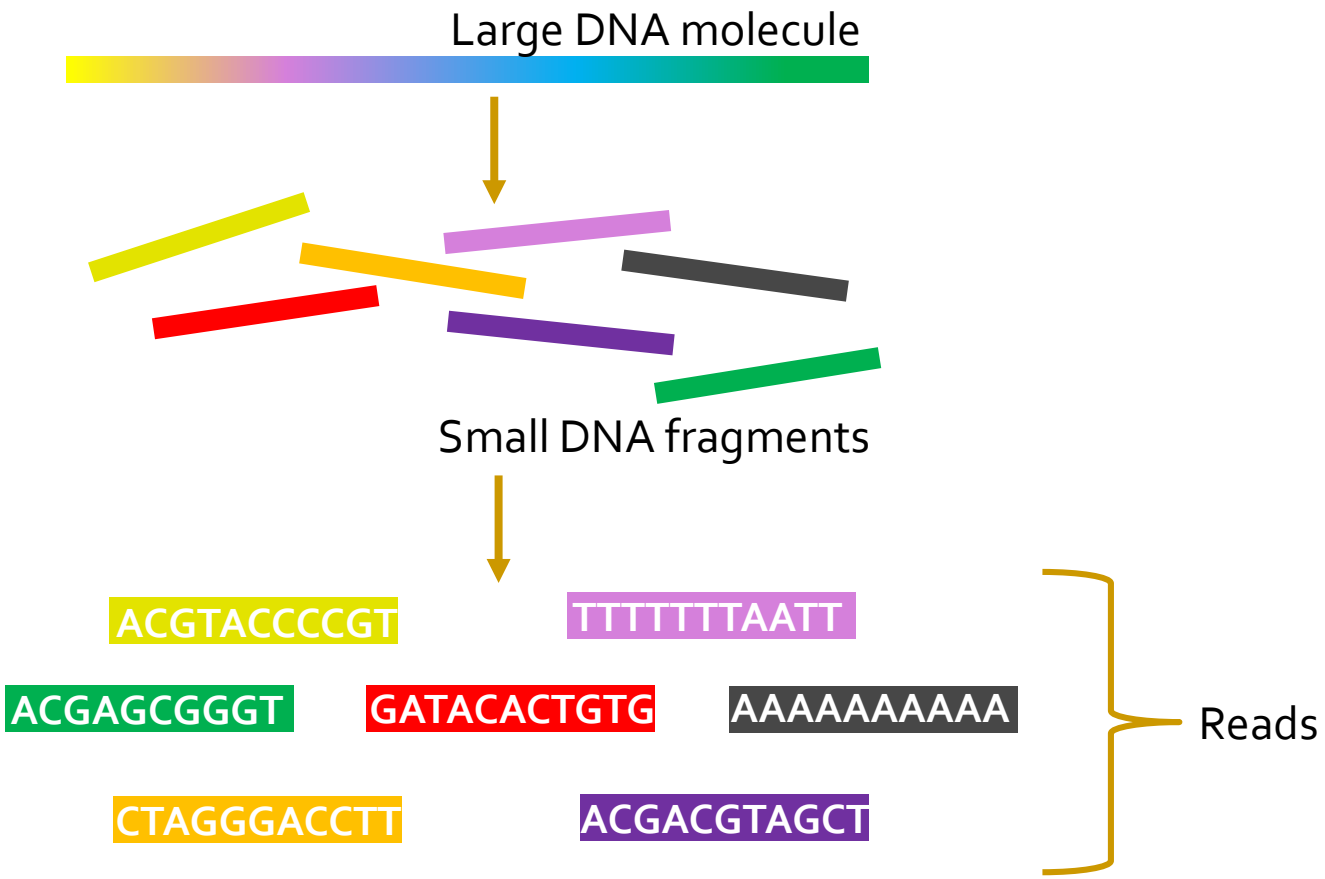
Slow and inefficient processing capability

This picture is similar for many "data generators & analyzers" today

One Problem

Need to construct
the entire genome
from many sequenced reads

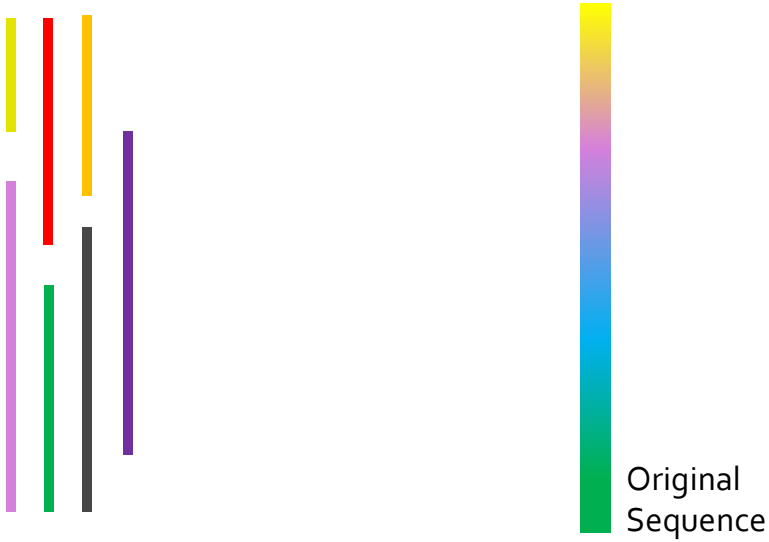
Genome Sequencing



Genome Sequence Analysis

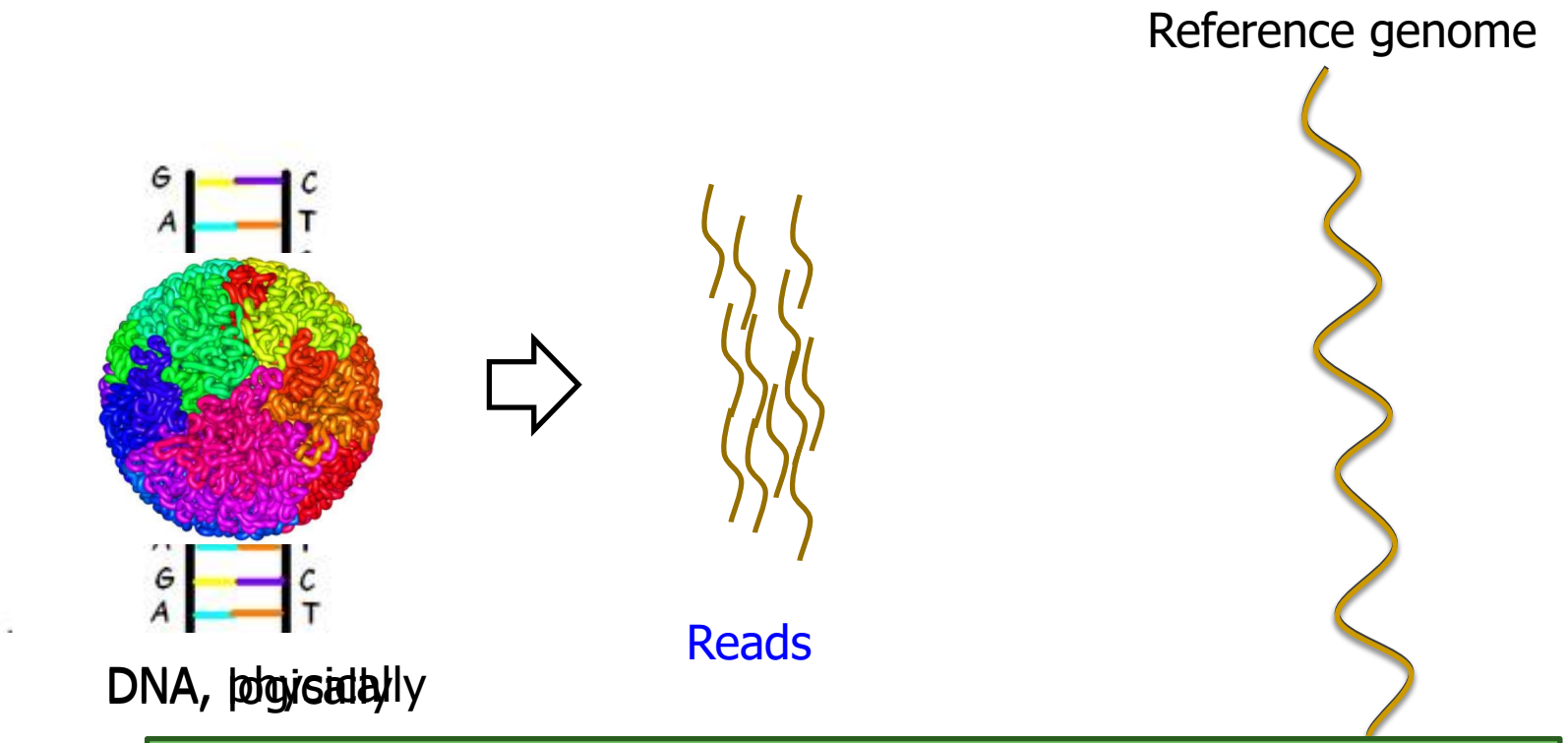
Read Mapping, method of aligning the reads against a **known reference genome** to **detect matches and variations**

***De novo* Assembly,** method of merging the reads in order to **construct** the original sequence (reference genome)



Read Mapping

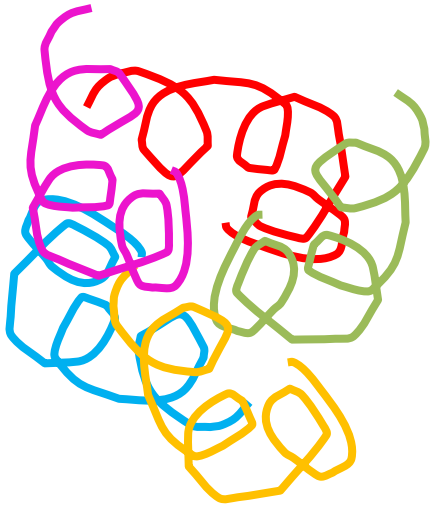
- Map many short DNA fragments (reads) to a known reference genome with some differences allowed



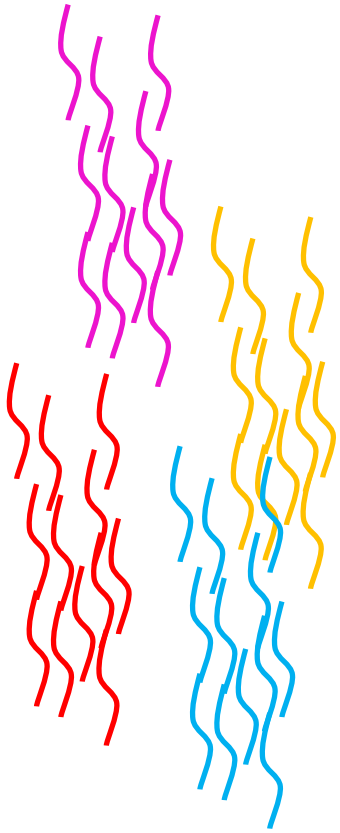
Mapping short reads to reference genome is challenging (billions of 50-300 base pair reads)

Read Mapping for Metagenomic Analysis

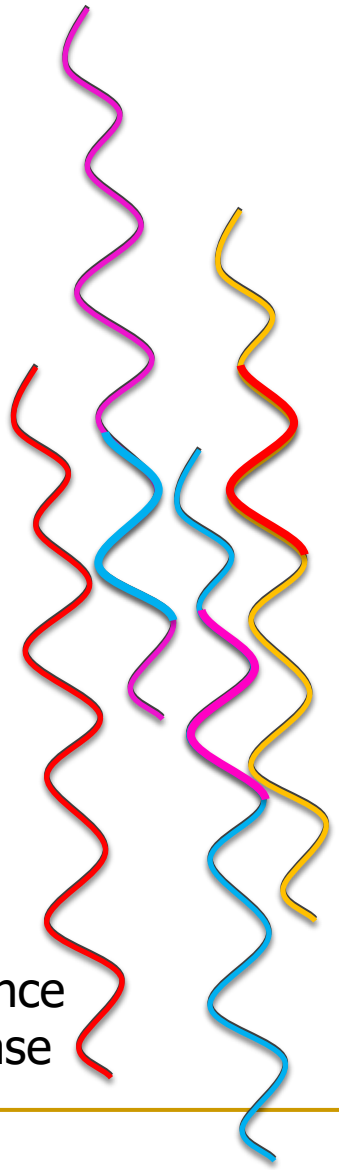
Reads from different unknown donors at sequencing time are mapped to many known reference genomes



Genetic material recovered directly from environmental samples

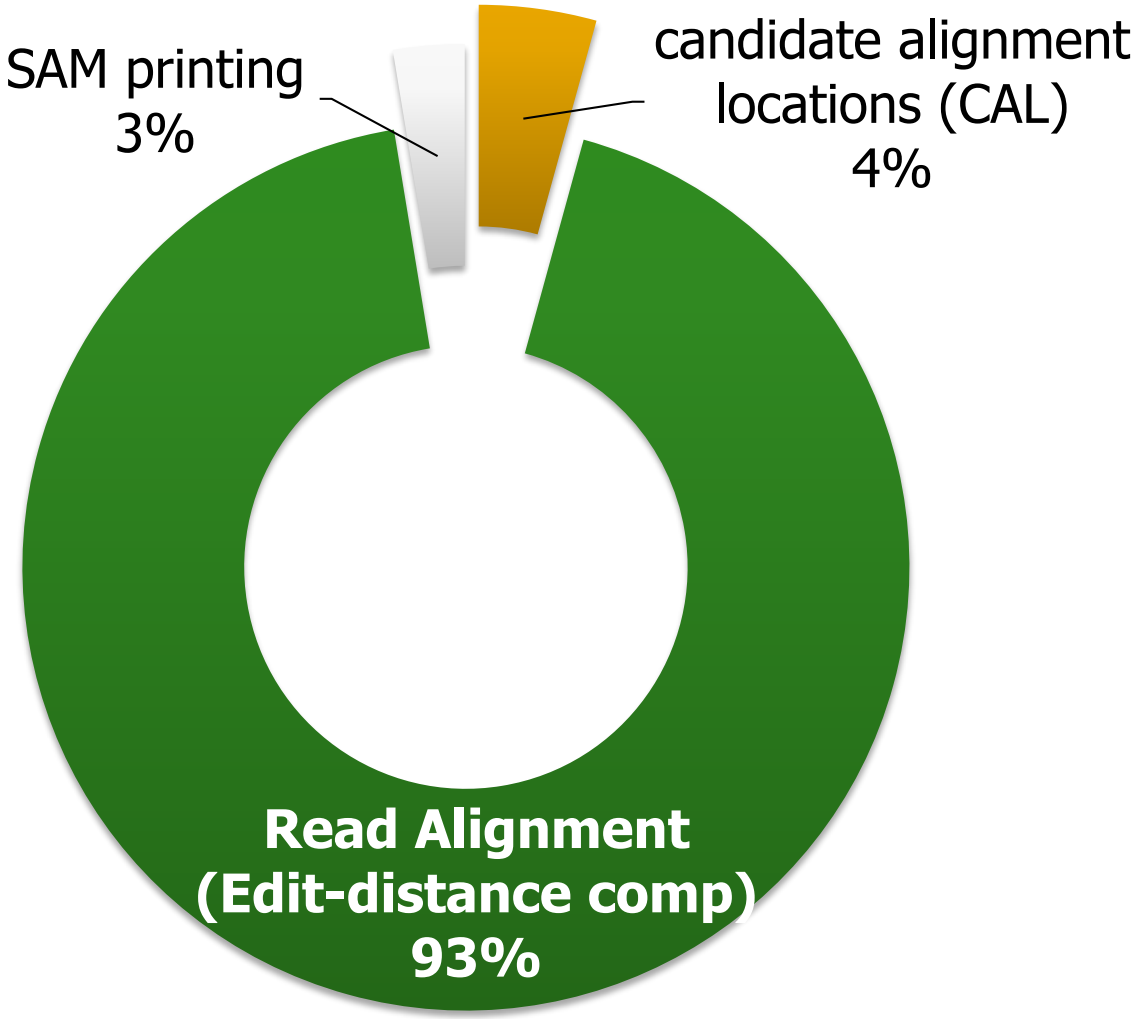


Reads in "text format"



Reference Database

Read Mapping Execution Time (Old Times)



Matching Each Read to Reference Genome

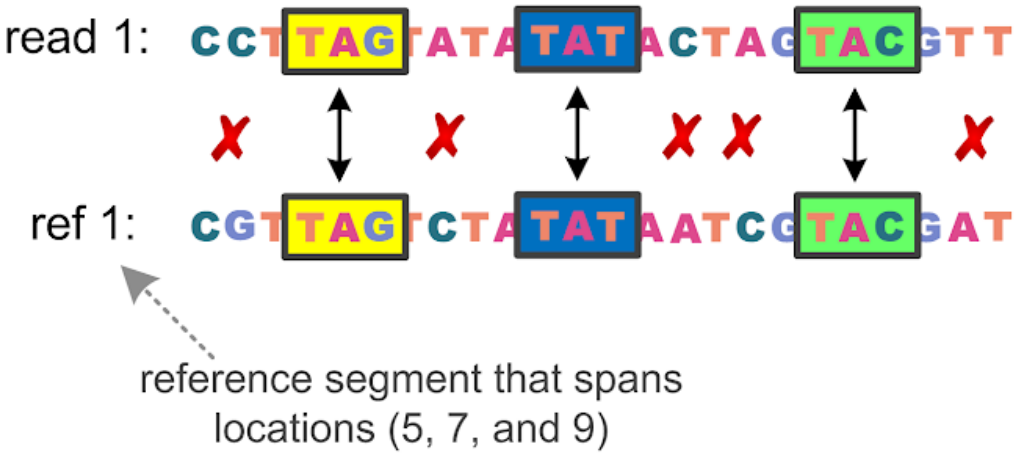
Reference Genome .FASTA file:

```
>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCTTCATTGACATTTAAACTCTGGGGCAGGTGAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGGAACCGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACCGCCCCCCCGGCCCGGCTCGGGGCCCGCGGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCGCCCCAAGTGGCCCCGGGGCTTGATTTTTGCTTTTAAAAG
GAGGCATACAAAGATGGAAGCGAGTTACTGAGGGAGGGATAGGAAGGGGGGTGGAGGAGGGACTTGTCTT
TGCCGAGTGTCAAAAGTAGCACTCCTAATCCAGTCCGGCCCT
GAGCTGGGAGTAGGGGGCGGGAGTCTGCTGCTGCTGTCTGCTAAAGCCACTCGCGACCGCGAAAAATGCA
GGAGGTGGGGACGCACTTTGCATCCAGACCTCCTCTGCATCGCAGTTCCGCTTGGGAAAG
TCCGTACCCGCGCCTAAAGACACCCTGCCGCGGGTCGGGCGAGGTGCAGCAGAAGTTTCCC
GCGGTTGCAAAGTGCAGATGGCTGGACCGCAACAAAGTCTAGAGATGGGGTTCGTTTCTCAGAAAGACGC
```

Sequenced Reads .FASTQ file:

```
@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
TAATAAATCTTTAGATNNNNNNNNNTAG
+HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
efcfffffcfeefffcffffffddf`feed]`]_Ba_^__[YBBBBBBBBBRTT
```

Base-by-Base Comparison



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.

NETHERLANDS x SWITZERLAND

N	E	-	T	H	E	R	L	A	N	D	S
S	W	I	T	Z	E	R	L	A	N	D	-

match
deletion
insertion
mismatch

Challenges in Read Mapping

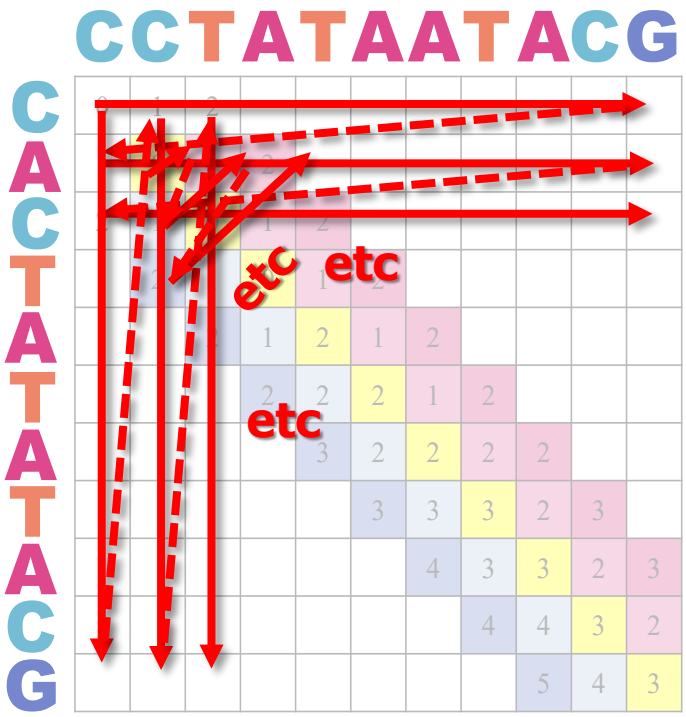
- Need to find many mappings of each read
  - A short read may map to many locations, especially with High-Throughput DNA Sequencing technologies
  - How can we find all mappings efficiently?

- Need to tolerate small variances/errors in each read
  - Each individual is different: Subject's DNA may slightly differ from the reference (Mismatches, insertions, deletions)
  - How can we efficiently map each read with up to *e* errors present?

- Need to map each read very fast (i.e., performance is important)
  - Human DNA is 3.2 billion base pairs long → Millions to billions of reads (State-of-the-art mappers take weeks to map a human's DNA)
  - How can we design a much higher performance read mapper?

Why Is Read Alignment Slow?

- Quadratic-time dynamic-programming algorithm(s)
- Data dependencies limit the computation parallelism
- Entire matrix computed even though strings may be dissimilar



Read Alignment

Example: Dynamic Programming Table

immediate left,
upper left,
upper entries of its own

NETHERLANDS x SWITZERLAND

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

Example: Dynamic Programming Table

NETHERLANDS x SWITZERLAND

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

- Matrix-filling is O(mn) time and space.
- Backtrace is O(m + n) time.

Example: Dynamic Programming

- Quadratic-time dynamic-programming algorithm

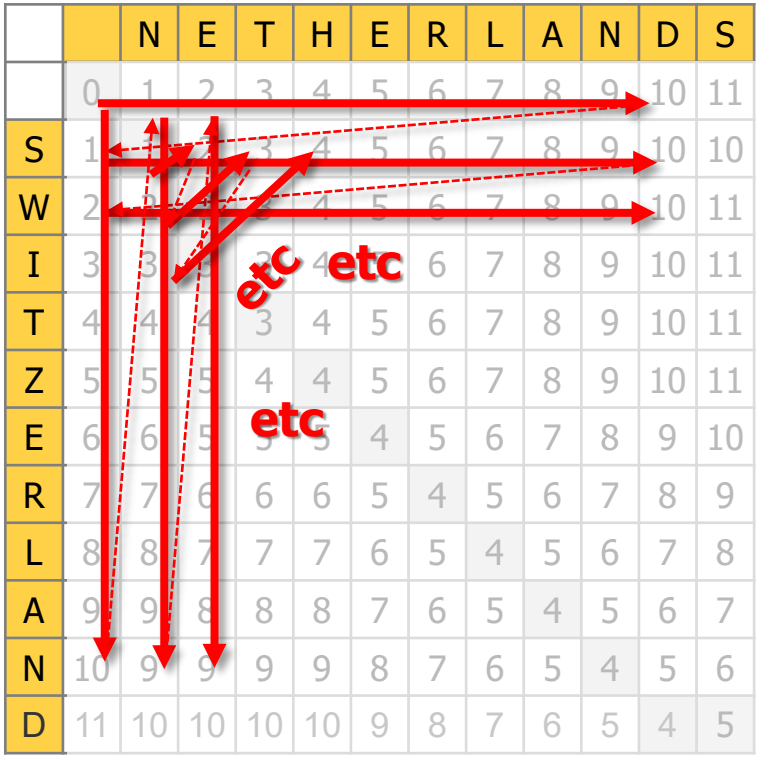
WHY?!

Enumerate all possible prefixes

NETHERLANDS x SWITZERLAND

- D
C
- E
C

NETHERLANDS x S
NETHERLANDS x SW
NETHERLANDS x SWI
NETERLANDS x SWIT
NETHERLANDS x SWITZ
NETHERLANDS x SWITZE
NETHERLANDS x SWITZER
NETHERLANDS x SWITZERL
NETHERLANDS x SWITZERLA
NETHERLANDS x SWITZERLAN
NETHERLANDS x SWITZERLAND



Computational Cost is Mathematically Proven

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

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]

Alser *et al. Genome Biology* (2021) 22:249
https://doi.org/10.1186/s13059-021-02443-7

Genome Biology

REVIEW Open Access

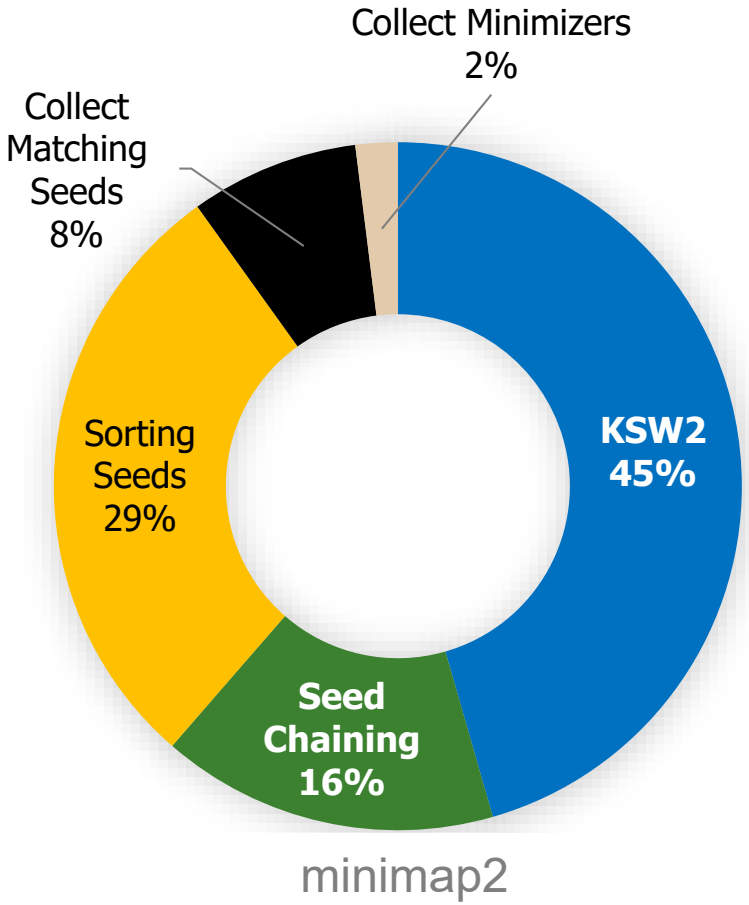
Technology dictates algorithms: recent developments in read alignment

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

Read Mapping Execution Time (Modern)

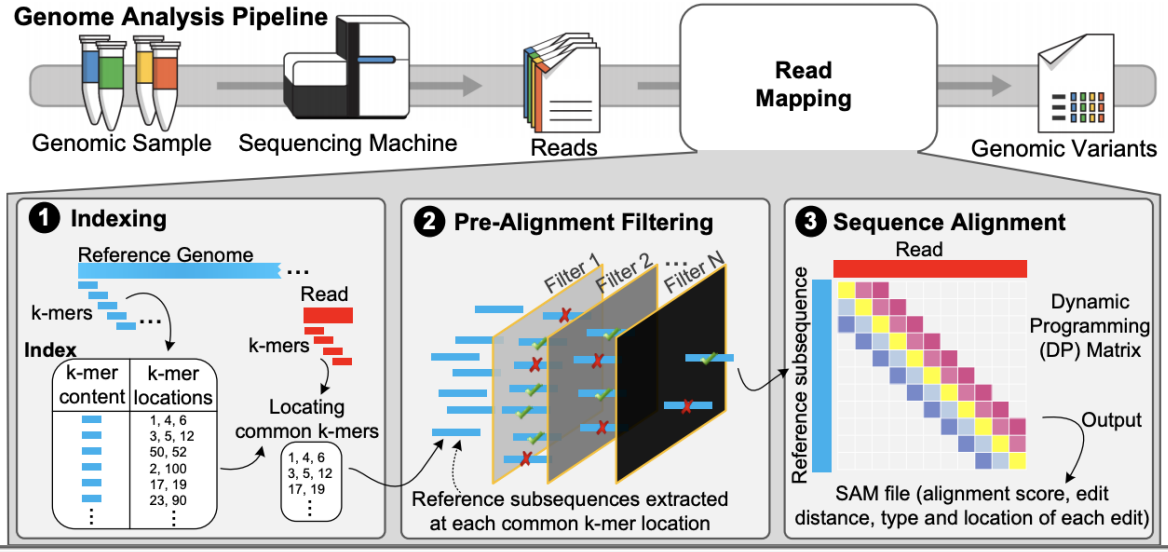
>60%

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



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

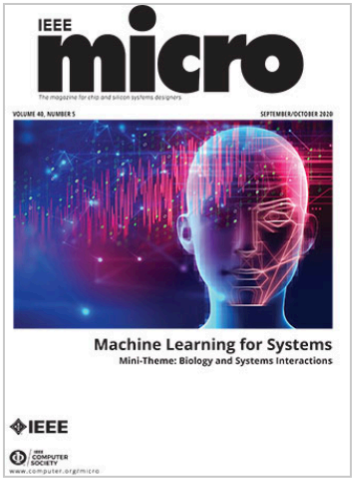
Alser+, "Accelerating Genome Analysis: A Primer on an Ongoing Journey", IEEE Micro, 2020.

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"

IEEE Micro, August 2020.



Previous | Next
Table of Contents
Past Issues

Home / Magazines / IEEE Micro / 2020.05

IEEE Micro

Accelerating Genome Analysis: A Primer on an Ongoing Journey

Sept.-Oct. 2020, pp. 65-75, vol. 40
DOI Bookmark: 10.1109/MM.2020.3013728

Authors

Mohammed Alser, ETH Zürich
Zulal Bingol, 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

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 Algorithms: Two Styles

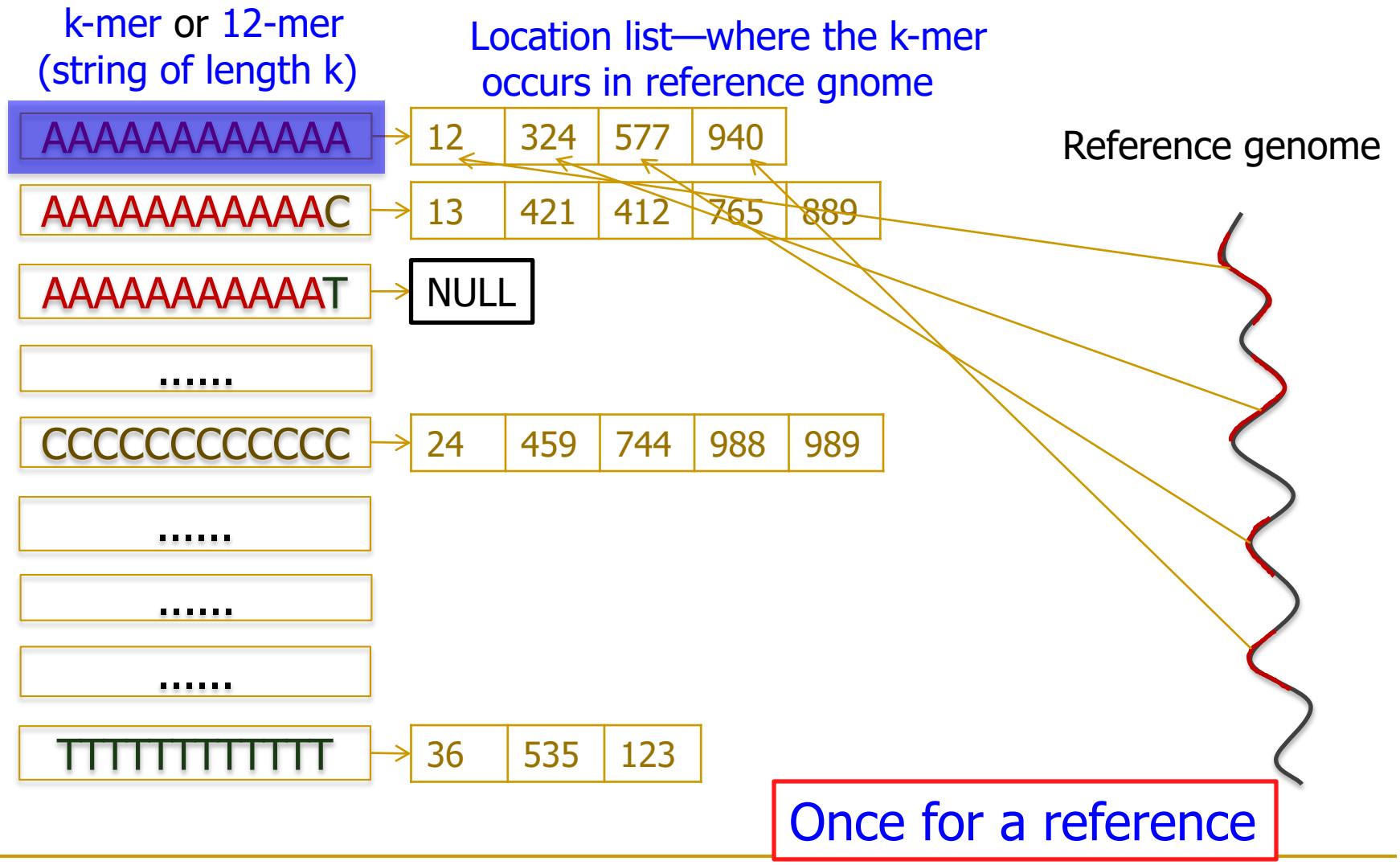
- Hash based seed-and-extend (hash table, suffix array, suffix tree)
  - Index the "k-mers" in the genome into a hash table (pre-processing)
  - When searching a read, find the location of a k-mer in the read; then extend through alignment
  - More sensitive (can find all mapping locations), but slow
  - Requires large memory; this can be reduced with cost to run time

- Burrows-Wheeler Transform & Ferragina-Manzini Index based aligners
  - BWT is a compression method used to compress the genome index
  - Perfect matches can be found very quickly, memory lookup costs increase for imperfect matches
  - Reduced sensitivity

Hash Table Based Read Mappers

- Key Idea
  - Preprocess the reference into a *Hash Table*
  - Use *Hash Table* to map reads

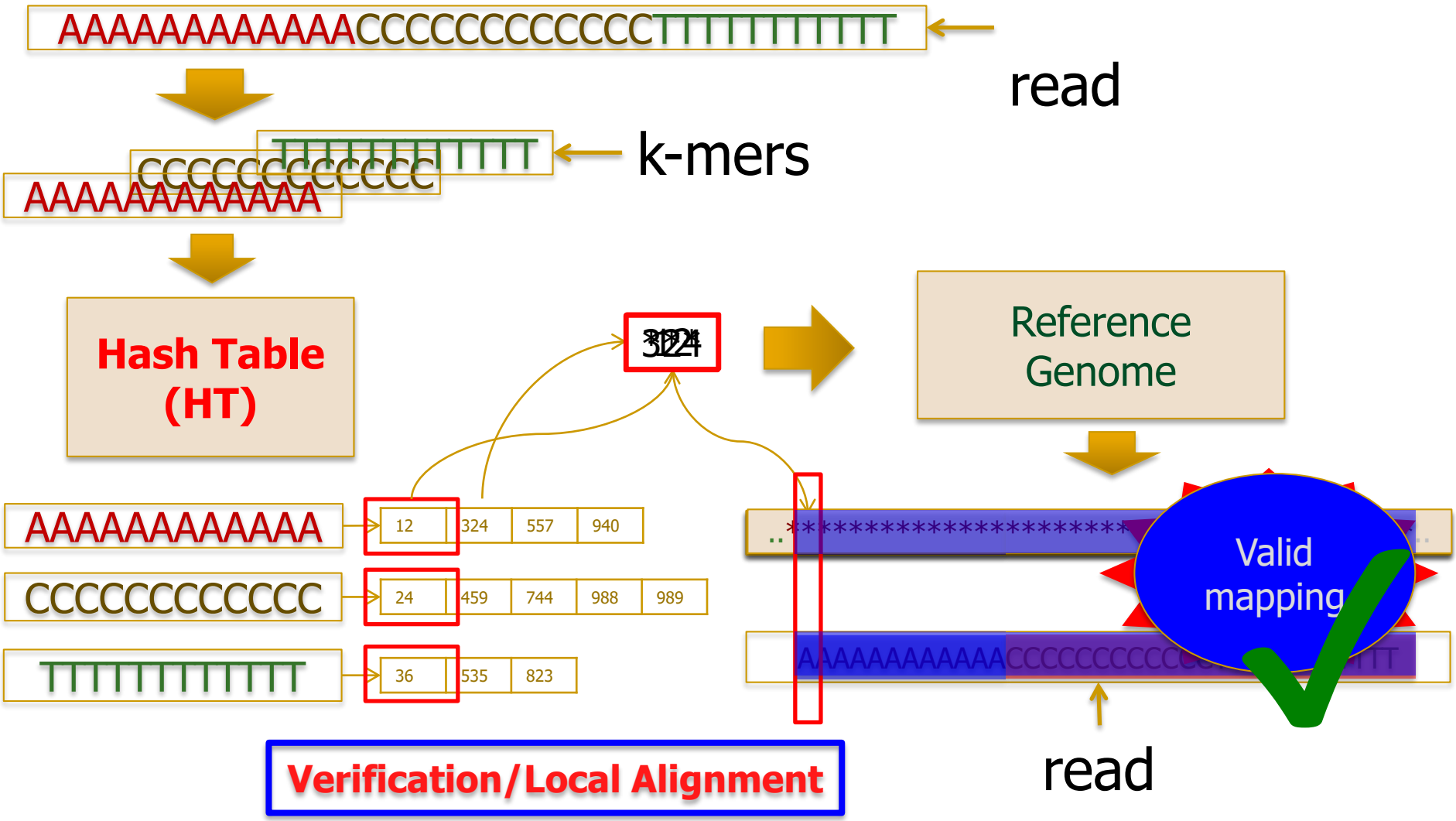
Hash Table-Based Mappers [Alkan+ Nature Gen'09]



Hash Table Based Read Mappers

- Key Idea
  - Preprocess the reference into a *Hash Table*
  - Use *Hash Table* to map reads

Hash Table-Based Mappers [Alkan+ Nature Gen'09]



Our First Step: Comprehensive Mapping

- + Guaranteed to find *all* mappings → sensitive
- + Can tolerate up to *e* errors

nature genetics

http://mrfast.sourceforge.net/

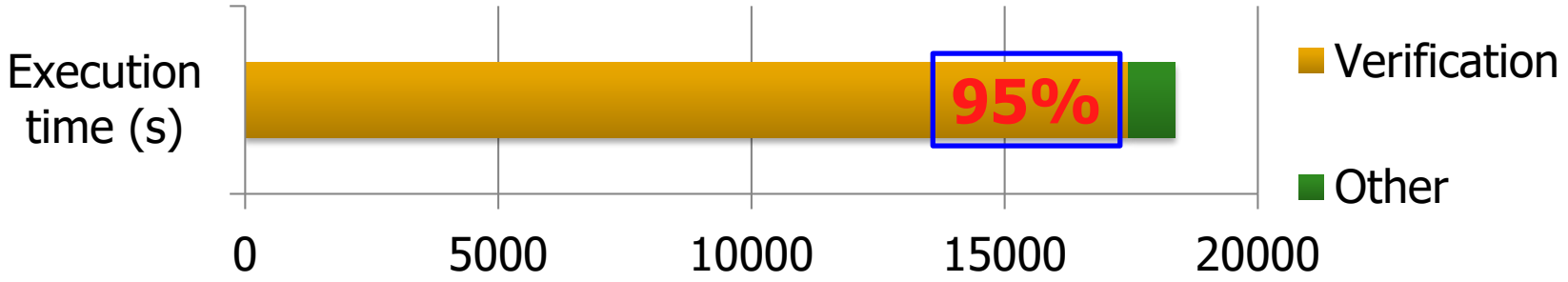
Personalized copy number and segmental duplication maps using next-generation sequencing

Can Alkan[1,2], Jeffrey M Kidd[1], Tomas Marques-Bonet[1,3], Gozde Aksay[1], Francesca Antonacci[1], Fereydoun Hormozdiari[4], Jacob O Kitzman[1], Carl Baker[1], Maika Malig[1], Onur Mutlu[5], S Cenk Sahinalp[4], Richard A Gibbs[6] & Evan E Eichler[1,2]

Alkan+, **"Personalized copy number and segmental duplication maps using next-generation sequencing"**, Nature Genetics 2009.

Problem and Goal

- Poor performance of existing read mappers: Very slow
  - Verification/alignment takes too long to execute
  - Verification requires a memory access for reference genome + many base-pair-wise comparisons between the reference and the read (edit distance computation)



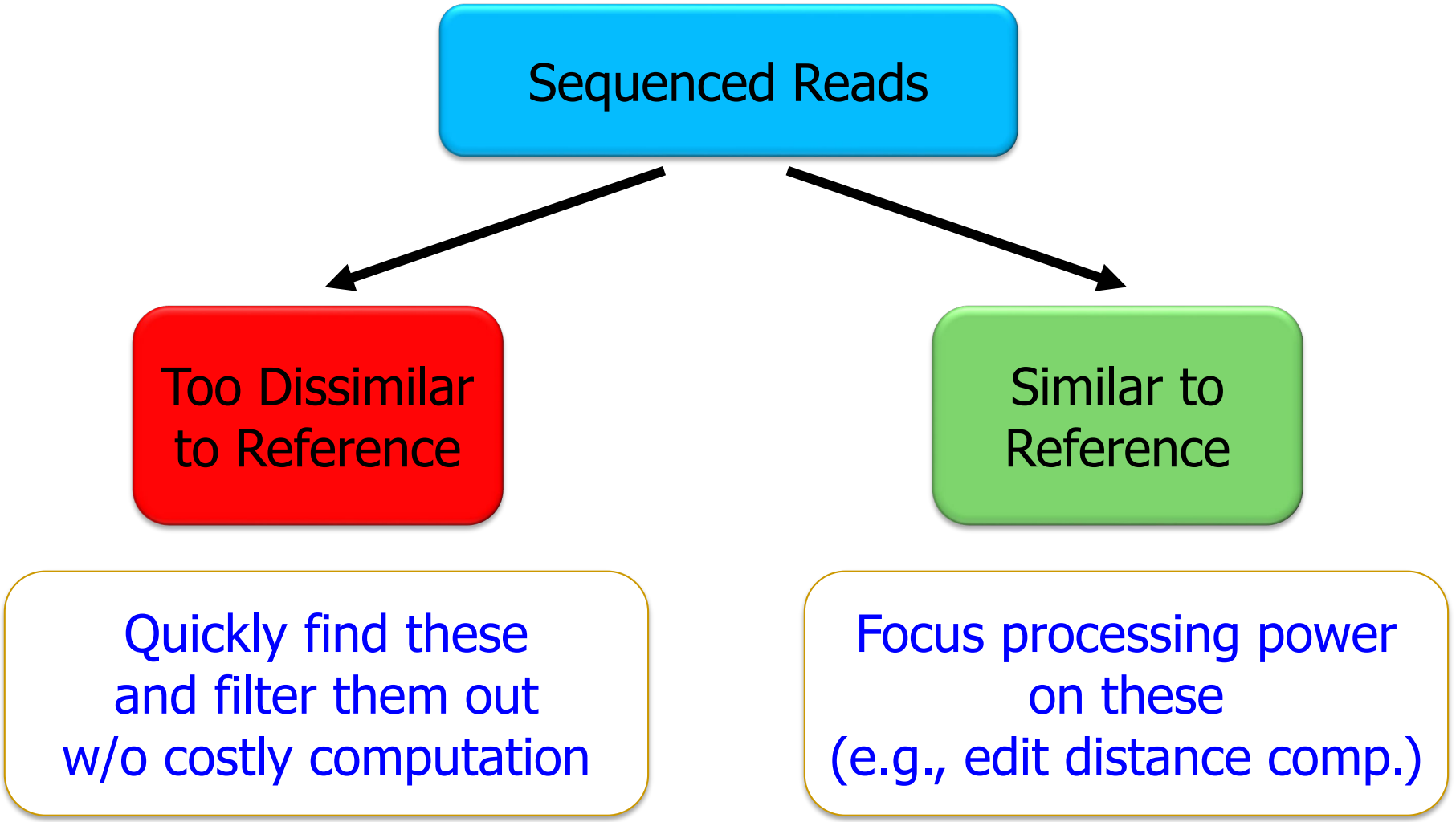
- Goal: Speed up the mapper by reducing the cost of verification

Overarching Key Idea

Filter fast before you align

Minimize costly
edit distance computations
("approximate string comparisons")

Overarching Key Idea



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"
  IEEE Micro (*IEEE MICRO*), Vol. 40, No. 5, pages 65-75, September/October 2020.
  [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

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

Our First Filter: Pure Software Approach

- Download the source code and try for yourself
  - Download link to FastHASH

Xin *et al. BMC Genomics* 2013, **14**(Suppl 1):S13
http://www.biomedcentral.com/1471-2164/14/S1/S13

BMC Genomics

PROCEEDINGS Open Access

Accelerating read mapping with FastHASH

Hongyi Xin[1], Donghyuk Lee[1], Farhad Hormozdiari[2], Samihan Yedkar[1], Onur Mutlu[1*], Can Alkan[3*]

From The Eleventh Asia Pacific Bioinformatics Conference (APBC 2013)
Vancouver, Canada. 21-24 January 2013

Xin+, **"Accelerating Read Mapping with FastHASH"**, BMC Genomics 2013.

Reducing the Cost of Verification

- Most verification (edit distance computation) calculations are unnecessary
  - 1 out of 1000 potential locations passes the verification process

- We can get rid of unnecessary verification calculations by
  - *Detecting and rejecting* *early* invalid mappings (filtering)
  - *Reducing* the *number* of potential mappings to examine

Key Observations [Xin+, BMC Genomics 2013]

- Observation 1
  - Adjacent k-mers in the read should also be adjacent in the reference genome
  - Read mapper can quickly reject mappings that do **not** satisfy this property

- Observation 2
  - Some k-mers are cheaper to verify than others because they have shorter location lists (they occur less frequently in the reference genome)
    - Mapper needs to examine only *e+1* k-mers' locations to tolerate *e* errors
  - Read mapper can choose the cheapest *e+1* k-mers and verify their locations

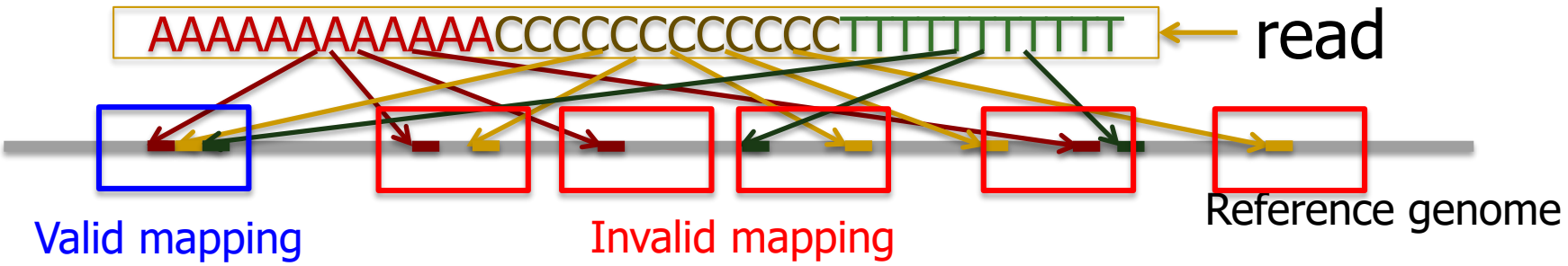
FastHASH Mechanisms [Xin+, BMC Genomics 2013]

- **Adjacency Filtering (AF)**: Rejects obviously invalid mapping locations at early stage to avoid unnecessary verifications

- **Cheap K-mer Selection (CKS):** Reduces the absolute number of potential mapping locations to verify

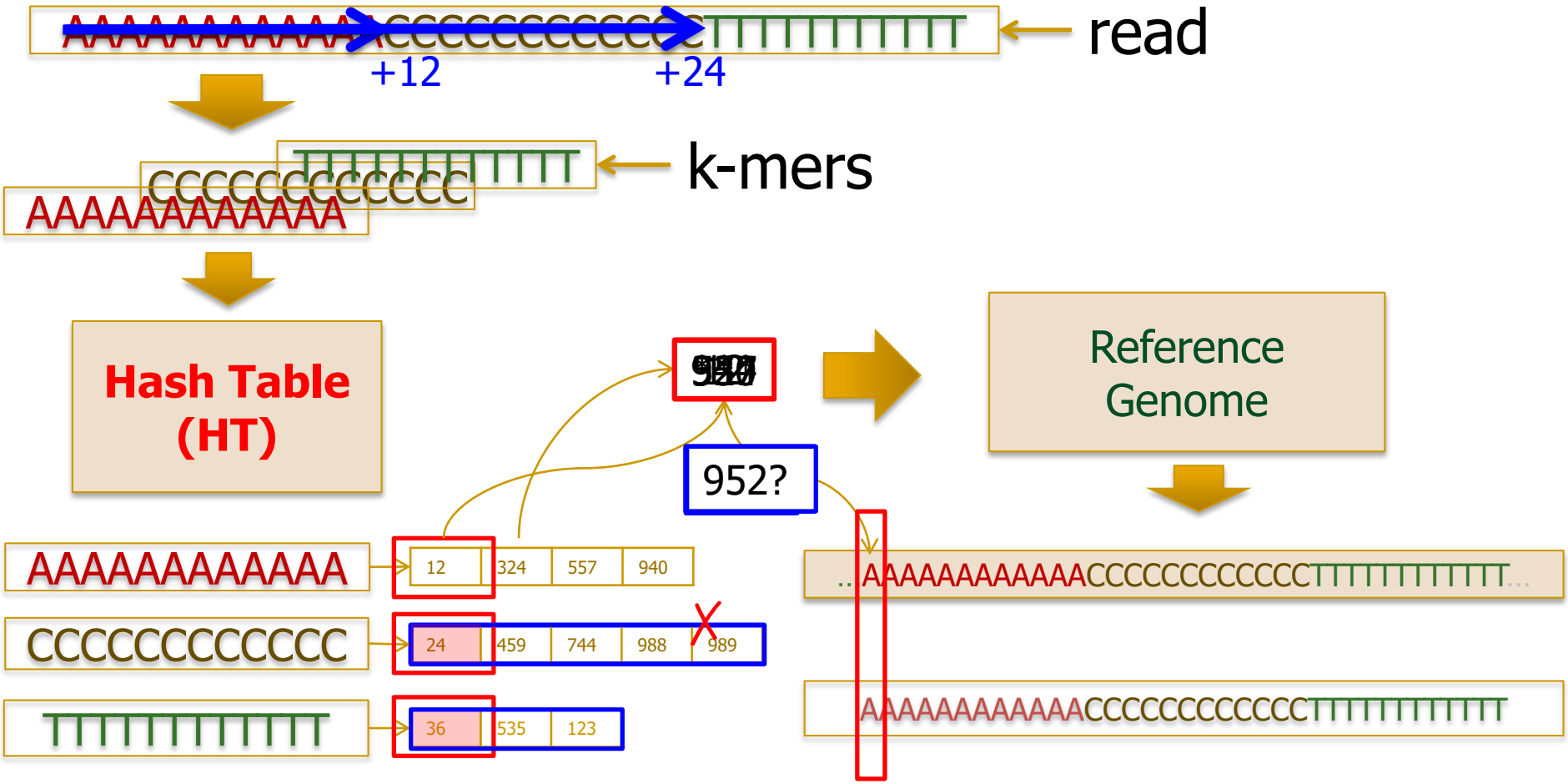
Adjacency Filtering (AF)

- **Goal:** detect and filter out invalid mappings at early stage
- **Key Insight:** For a valid mapping, adjacent k-mers in the read are also adjacent in the reference genome



- **Key Idea:** search for adjacent locations in the k-mers' location lists (in the index)
  - If more than *e* k-mers fail → there must be more than e errors → invalid mapping

Adjacency Filtering (AF)



FastHASH Mechanisms [Xin+, BMC Genomics 2013]

- **Adjacency Filtering (AF)**: Rejects obviously invalid mapping locations at early stage to avoid unnecessary verifications

- **Cheap K-mer Selection (CKS):** Reduces the absolute number of potential mapping locations to verify

Cheap K-mer Selection (CKS)

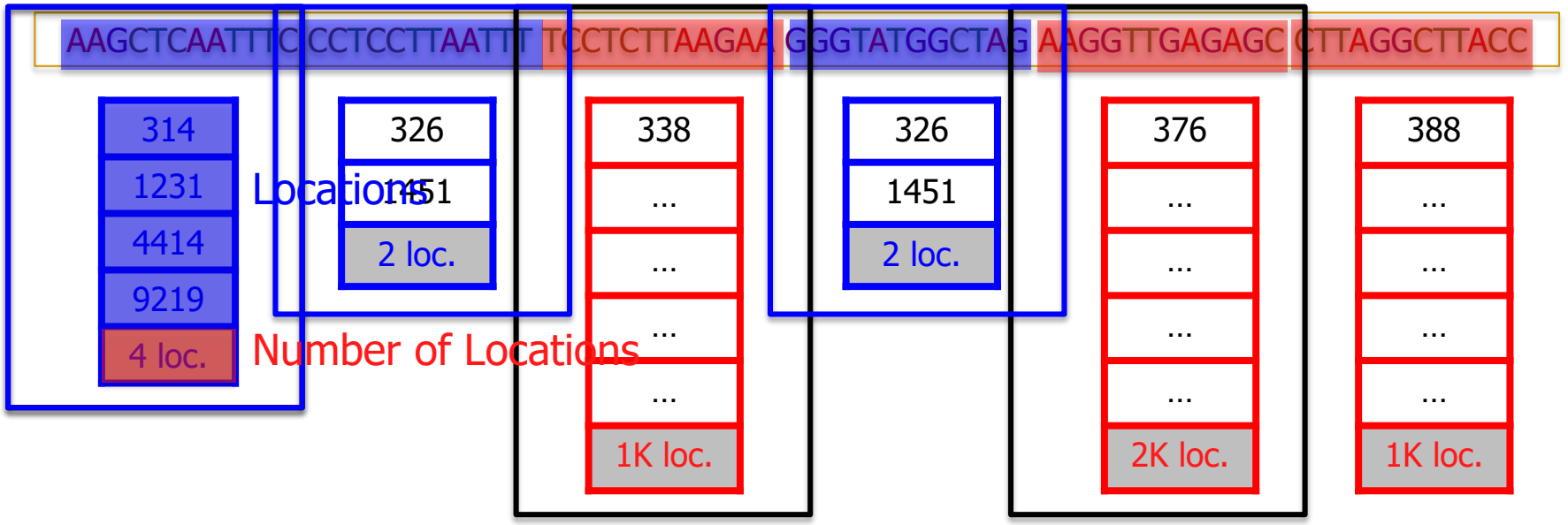
- **Goal:** Reduce the number of potential mappings to examine

- **Key insight:**
  - K-mers have different cost to examine: Some k-mers are *cheaper* as they have fewer locations than others (occur less frequently in reference genome)

- **Key idea:**
  - Sort the k-mers based on their number of locations
  - Select the k-mers with the fewest number locations to verify

Cheap K-mer Selection

- *e*=2 (examine 3 k-mers)



Expensive 3 k-mers

Previous work needs to verify:

3004 locations

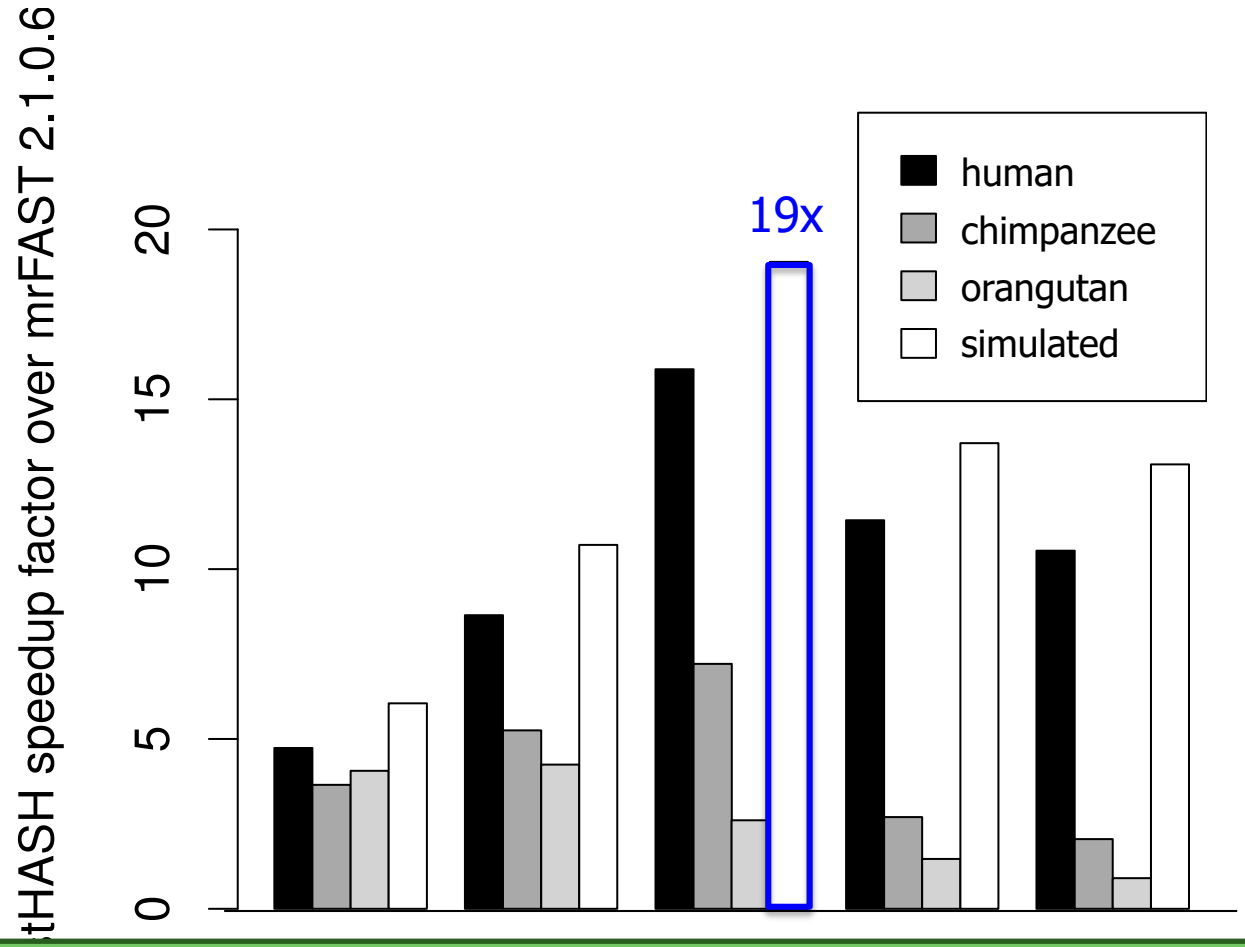
FastHASH verifies only:

8 locations

Methodology

- Implemented FastHASH on top of state-of-the-art mapper: mrFAST
  - New version mrFAST-2.5.0.0 over mrFAST-2.1.0.6
- Tested with real read sets generated from Illumina platform
  - 1M reads of a human (160 base pairs)
  - 500K reads of a chimpanzee (101 base pairs)
  - 500K reads of a orangutan (70 base pairs)
- Tested with simulated reads generated from reference genome
  - 1M simulated reads of human (180 base pairs)
- Evaluation system
  - Intel Core i7 Sandy Bridge machine
  - 16 GB of main memory

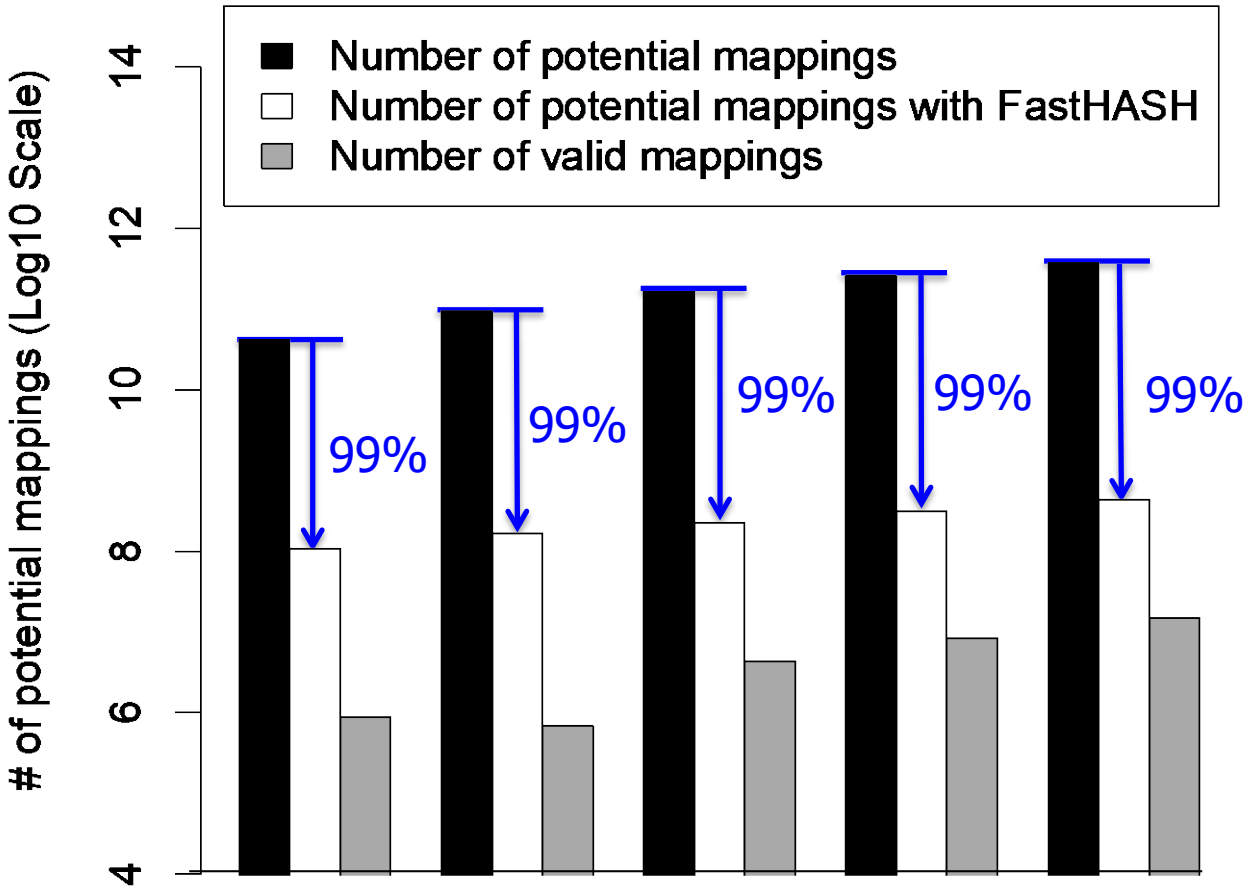
FastHASH Speedup: Entire Read Mapper



With FastHASH, new mrFAST obtains up to 19x speedup over previous version, without losing valid mappings

Analysis

- Reduction of potential mappings with FastHASH



FastHASH filters out over 99% of the potential mappings without sacrificing any valid mappings

FastHASH Summary & Conclusion

- Problem: Existing read mappers perform poorly, especially in the presence of errors
- Observation: Most of the verification (edit distance) calculations are unnecessary → filter them out
- Key Idea: Exploit the structure of the genome to
  - Reject invalid mappings early (Adjacency Filtering)
  - Reduce the number of possible mappings to examine (Cheap K-mer Selection)
- Key Result: FastHASH obtains up to 19x speedup over the state-of-the-art mapper without losing valid mappings

More on FastHASH

- Download source code and try for yourself
  - Download link to FastHASH

Xin *et al. BMC Genomics* 2013, **14**(Suppl 1):S13
http://www.biomedcentral.com/1471-2164/14/S1/S13

BMC Genomics

PROCEEDINGS Open Access

Accelerating read mapping with FastHASH

Hongyi Xin[1], Donghyuk Lee[1], Farhad Hormozdiari[2], Samihan Yedkar[1], Onur Mutlu[1*], Can Alkan[3*]

From The Eleventh Asia Pacific Bioinformatics Conference (APBC 2013)
Vancouver, Canada. 21-24 January 2013

Xin+, **"Accelerating Read Mapping with FastHASH"**, BMC Genomics 2013.

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

Shifted Hamming Distance: SIMD Acceleration

https://github.com/CMU-SAFARI/Shifted-Hamming-Distance

Bioinformatics, 31(10), 2015, 1553–1560
doi: 10.1093/bioinformatics/btu856
Advance Access Publication Date: 10 January 2015
Original Paper

Sequence analysis

Shifted Hamming distance: a fast and accurate SIMD-friendly filter to accelerate alignment verification in read mapping

Hongyi Xin[1,*], John Greth[2], John Emmons[2], Gennady Pekhimenko[1], Carl Kingsford[3], Can Alkan[4,*] and Onur Mutlu[2,*]

Xin+, **"Shifted Hamming Distance: A Fast and Accurate SIMD-friendly Filter to Accelerate Alignment Verification in Read Mapping", Bioinformatics 2015.**

Shifted Hamming Distance

- **Key observation:**
  - If two strings differ by *E* edits, then every bp match can be aligned in at most *2E* shifts (of one of the strings).
    - Insight: Shifting a string by one "corrects" for one "error"

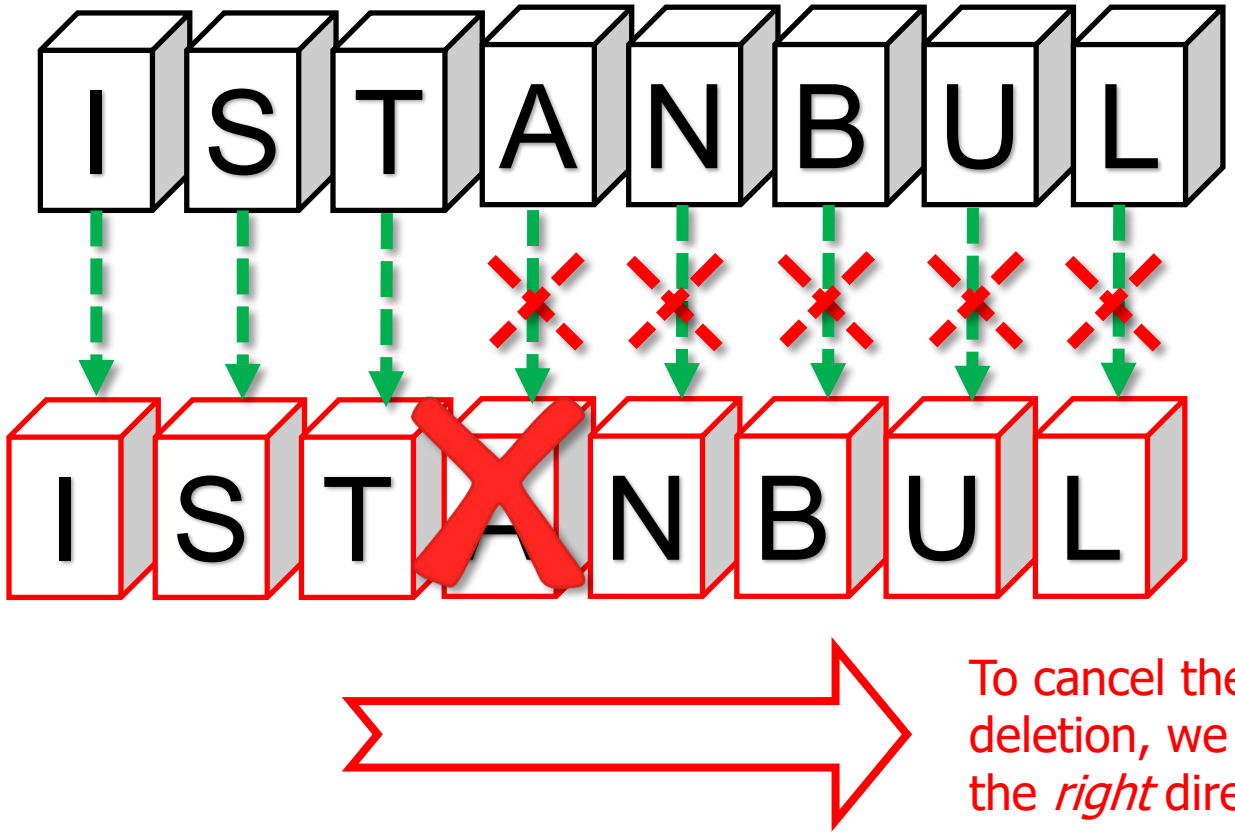
- **Key idea:**
  - Compute "Shifted Hamming Distance": AND of 2E Hamming Distances of two strings, to filter out invalid mappings
    - Uses bit-parallel operations that nicely map to SIMD instructions

- **Key result:**
  - SHD is 3x faster than SeqAn (the best implementation of Gene Myers' bit-vector algorithm), with only a 7% false positive rate
  - The fastest CPU-based filtering (pre-alignment) mechanism

Hamming Distance (∑⊕)

3 matches 5 mismatches

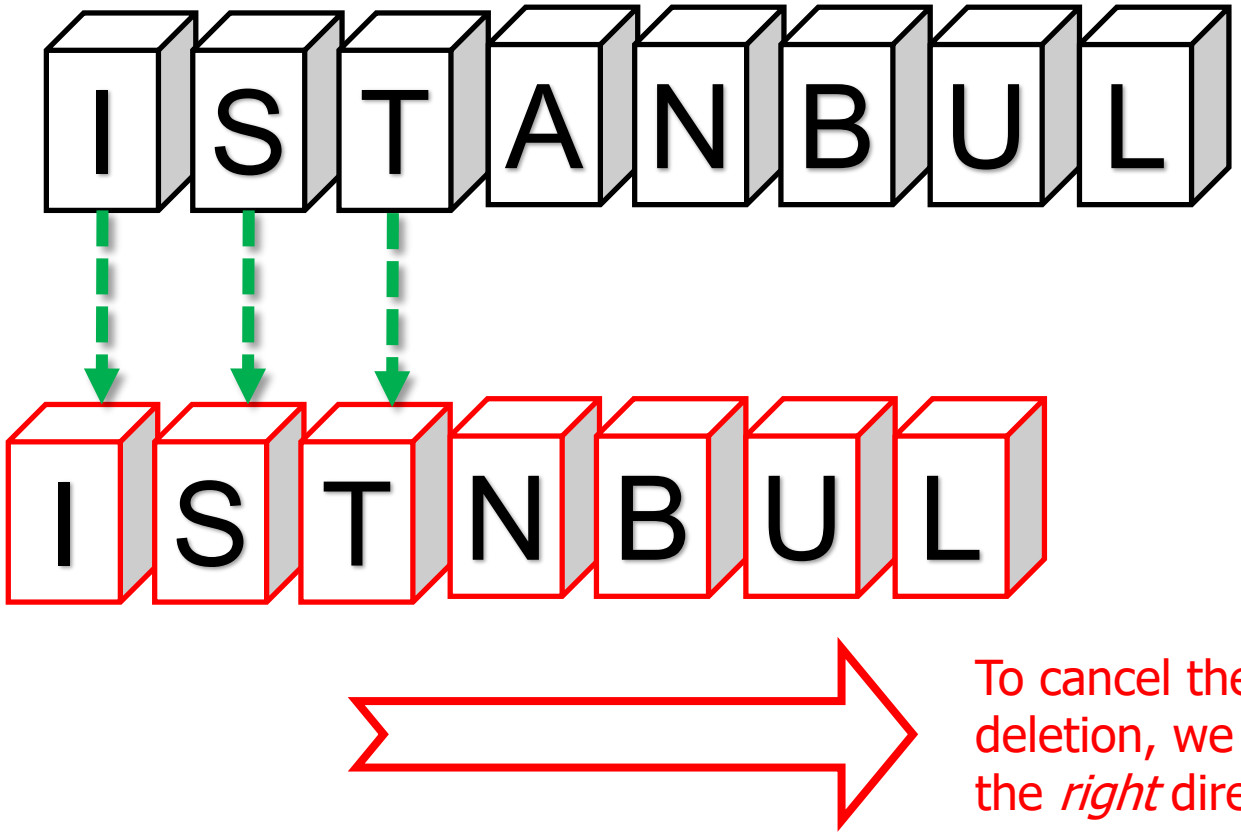
Edit = 1 Deletion



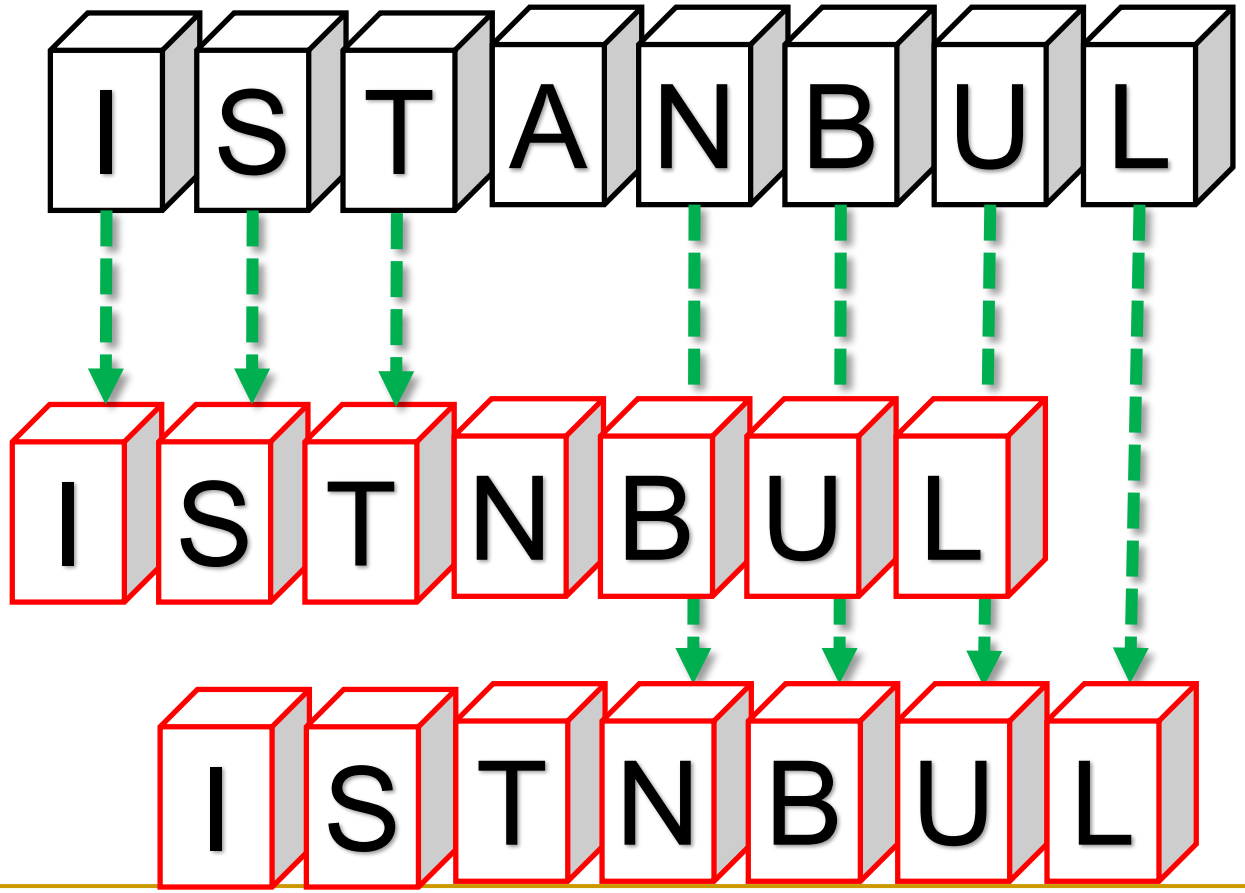
To cancel the effect of a deletion, we need to shift in the *right* direction

Insight: Shifting a String Helps Similarity Search

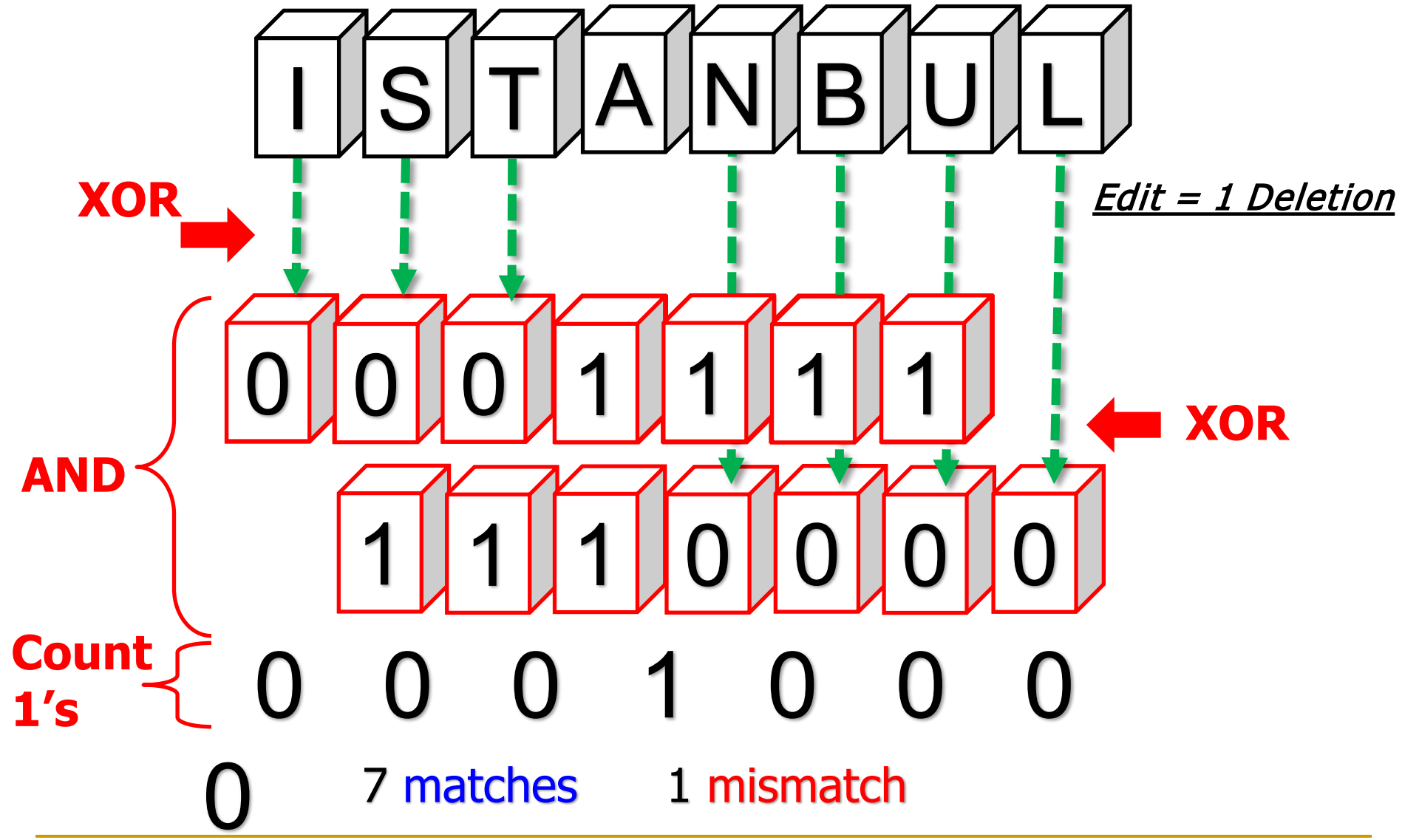
3 matches 5 mismatches



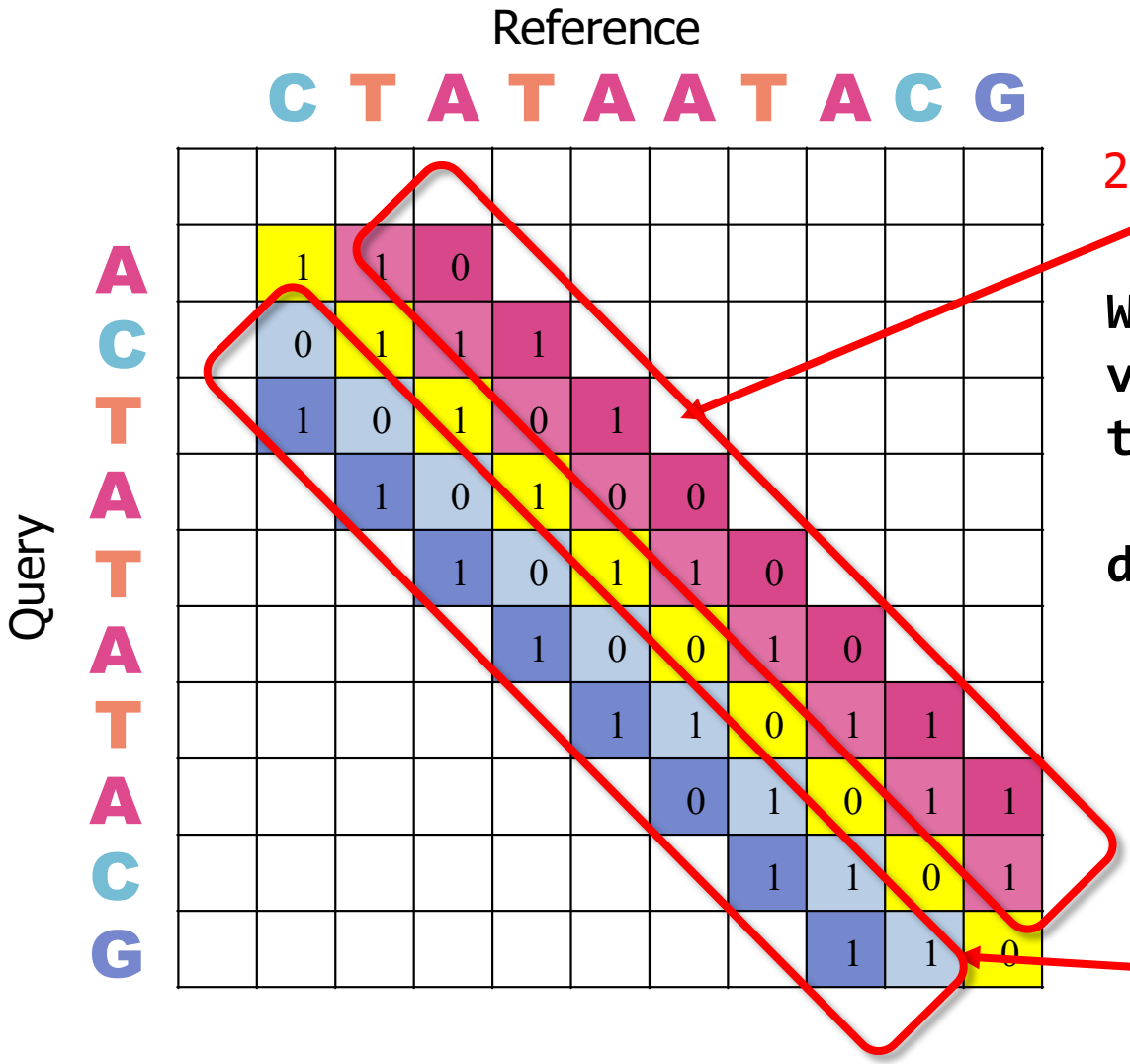
Insight: Shifting a String Helps Similarity Search



Shifted Hamming Distance



Highly Parallel Matrix Computation



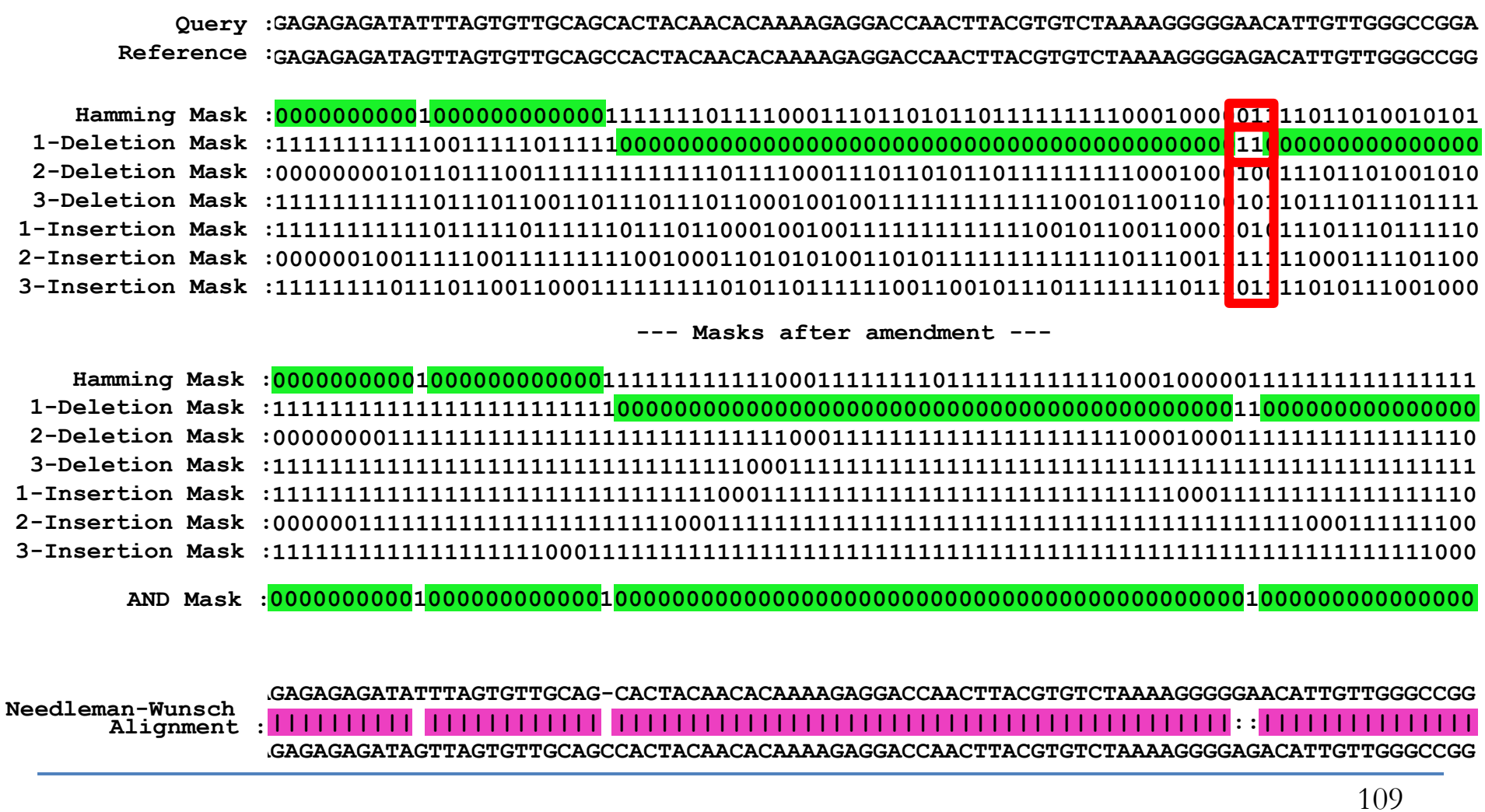
We need to compute 2E+1 vectors, E=edit distance threshold

dp[i][j]= 0 if X[i]=Y[j]
1 if X[i]≠Y[j]

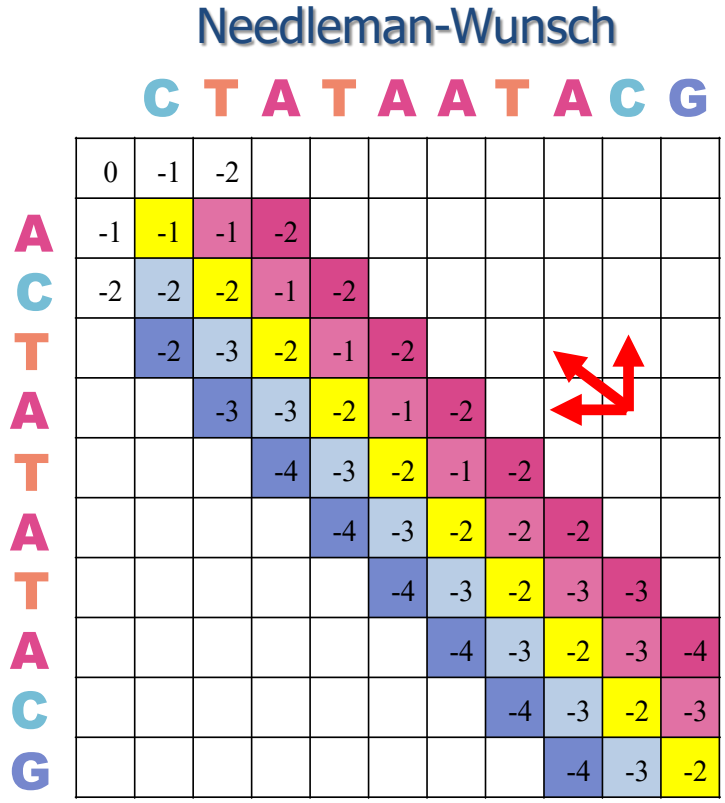
No data dependencies!

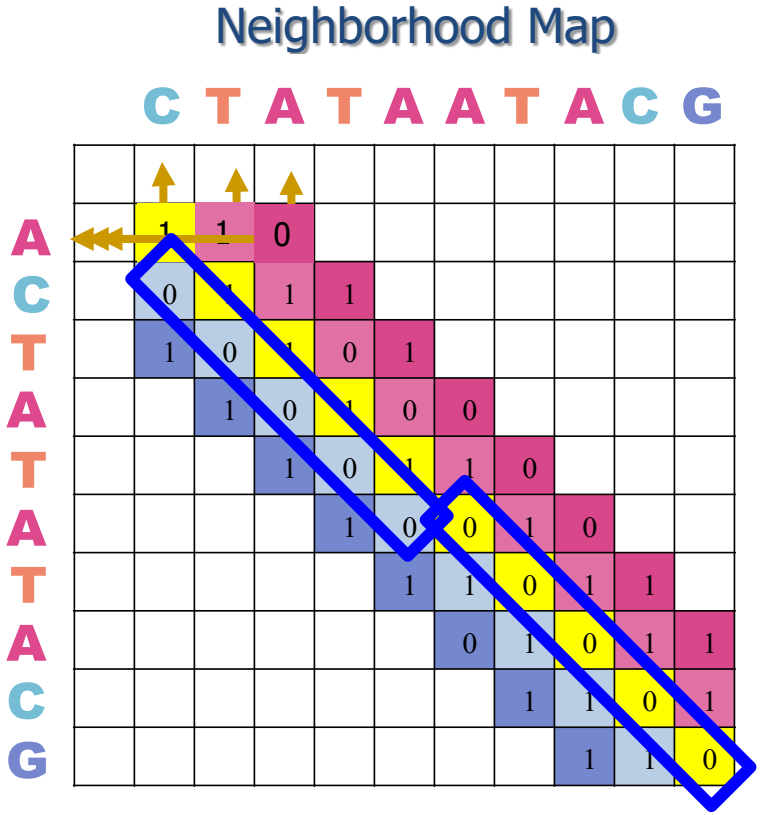
Key Idea of SHD Filtering

Generate 2E+1 masks → Amend random zeros: 101 → 111 & 1001 → 1111 → AND all masks, ACCEPT iff number of '1' ≤ Threshold



Alignment vs. Pre-alignment (Filtering)






```
|dp[i][j-1]   -1 // Inser.
```

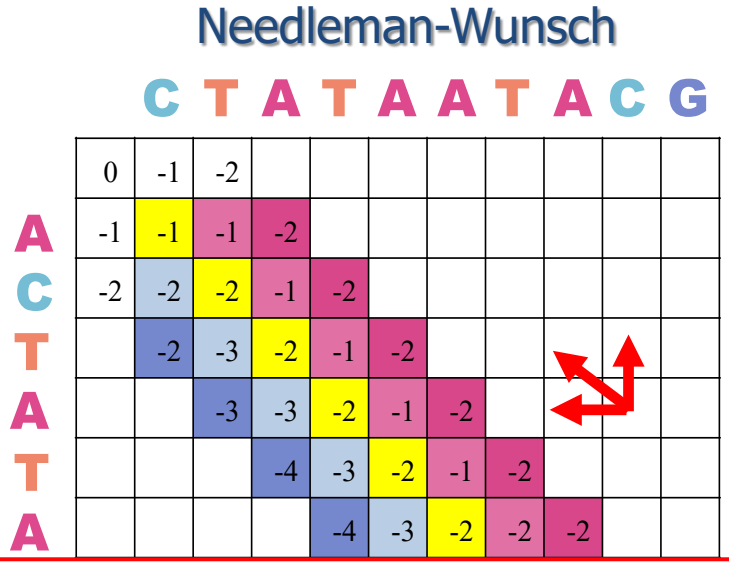
```
dp[i][j]=|0 if X[i]=Y[j]
```

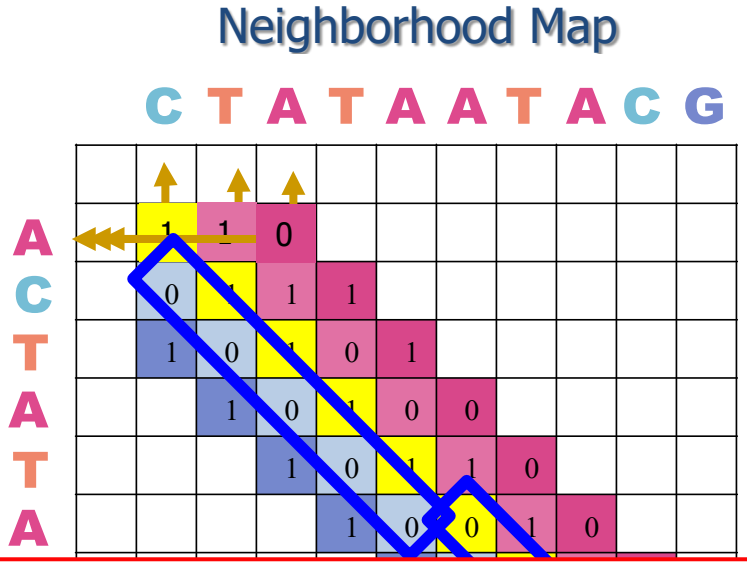
Our goal is to track the diagonally consecutive matches in the neighborhood map

pre-computed cells!

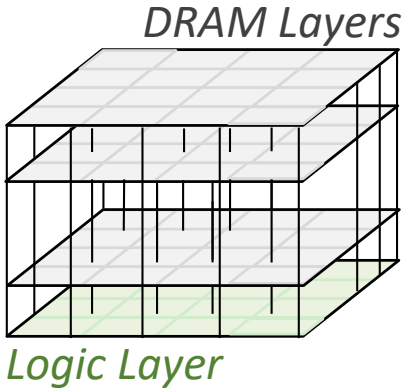
No data dependencies!

Alignment Matrix vs. Neighborhood Map





Independent vectors can be processed in parallel using hardware technologies



New Bottleneck: Filtering (Pre-Alignment)

Sequencing generates many reads, each of which potentially mapping to many locations

→

Filtering (Pre-alignment) eliminates the need to verify/align read to invalid mapping locations

→

Alignment/verification (costly edit distance computation) is performed **only** on reads that pass the filter

- New bottleneck in read mapping becomes the "filtering (pre-alignment)" step

More on Shifted Hamming Distance

https://github.com/CMU-SAFARI/Shifted-Hamming-Distance

Bioinformatics, 31(10), 2015, 1553–1560
doi: 10.1093/bioinformatics/btu856
Advance Access Publication Date: 10 January 2015
Original Paper

Sequence analysis

Shifted Hamming distance: a fast and accurate SIMD-friendly filter to accelerate alignment verification in read mapping

Hongyi Xin[1,*], John Greth[2], John Emmons[2], Gennady Pekhimenko[1], Carl Kingsford[3], Can Alkan[4,*] and Onur Mutlu[2,*]

Xin+, **"Shifted Hamming Distance: A Fast and Accurate SIMD-friendly Filter to Accelerate Alignment Verification in Read Mapping", Bioinformatics 2015.**

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

Location Filtering (Pre-alignment)

- **Alignment** is expensive
  - We need to align millions to billions of reads

- Modern read mappers reduce the time spent on alignment for increased performance. Can be done in two ways:
  1. Optimize the algorithm for alignment
  2. Reduce the number of alignments necessary by **filtering** out mismatches quickly

- Both methods are used by mappers today, but filtering has replaced alignment as the bottleneck **[Xin+, BMC Genomics 2013]**

Location Filtering (Pre-alignment)

- **Alignment** is expensive
  - We need to align millions to billions of reads

- M… t
  f…
  1…
  2…
  out mismatches quickly

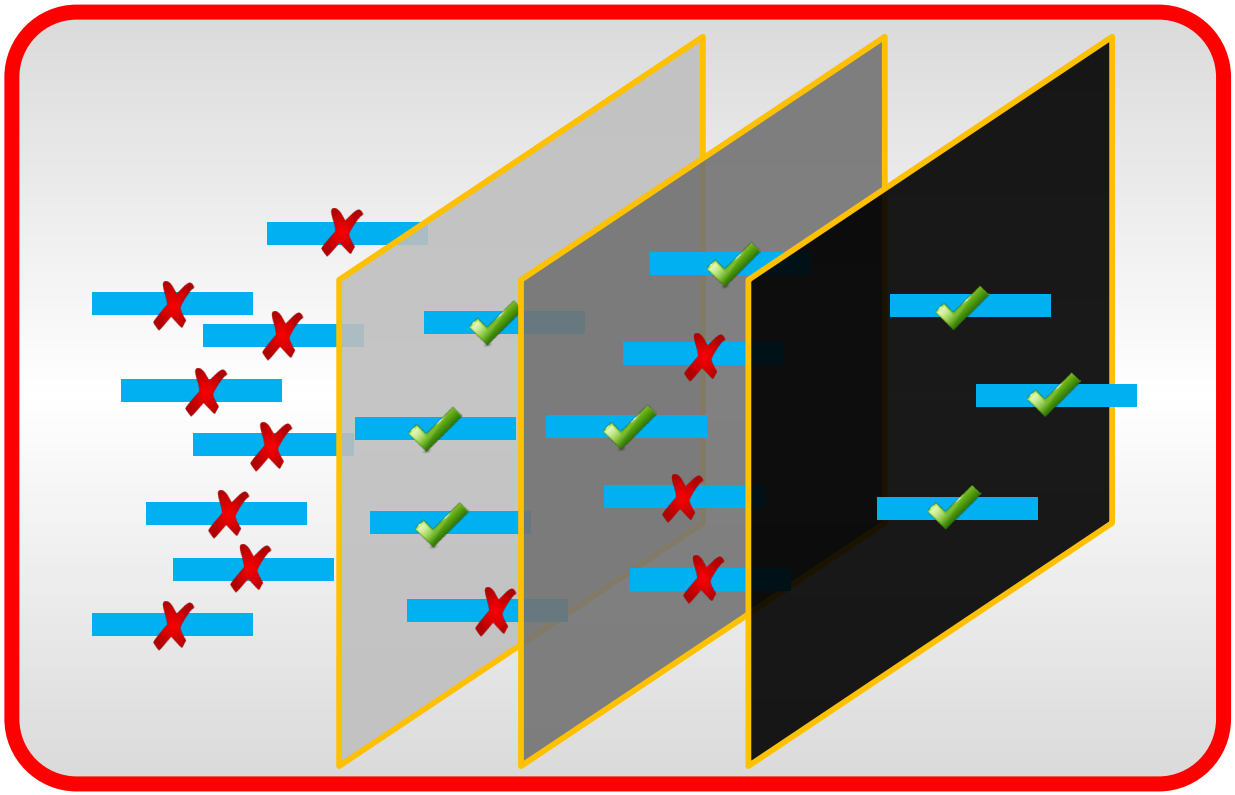
Our goal is to accelerate read mapping by improving the filtering step

- Both methods are used by mappers today, but filtering has replaced alignment as the bottleneck **[Xin+, BMC Genomics 2013]**

Ideal Location Filtering Algorithm

Step 2

Query the Index

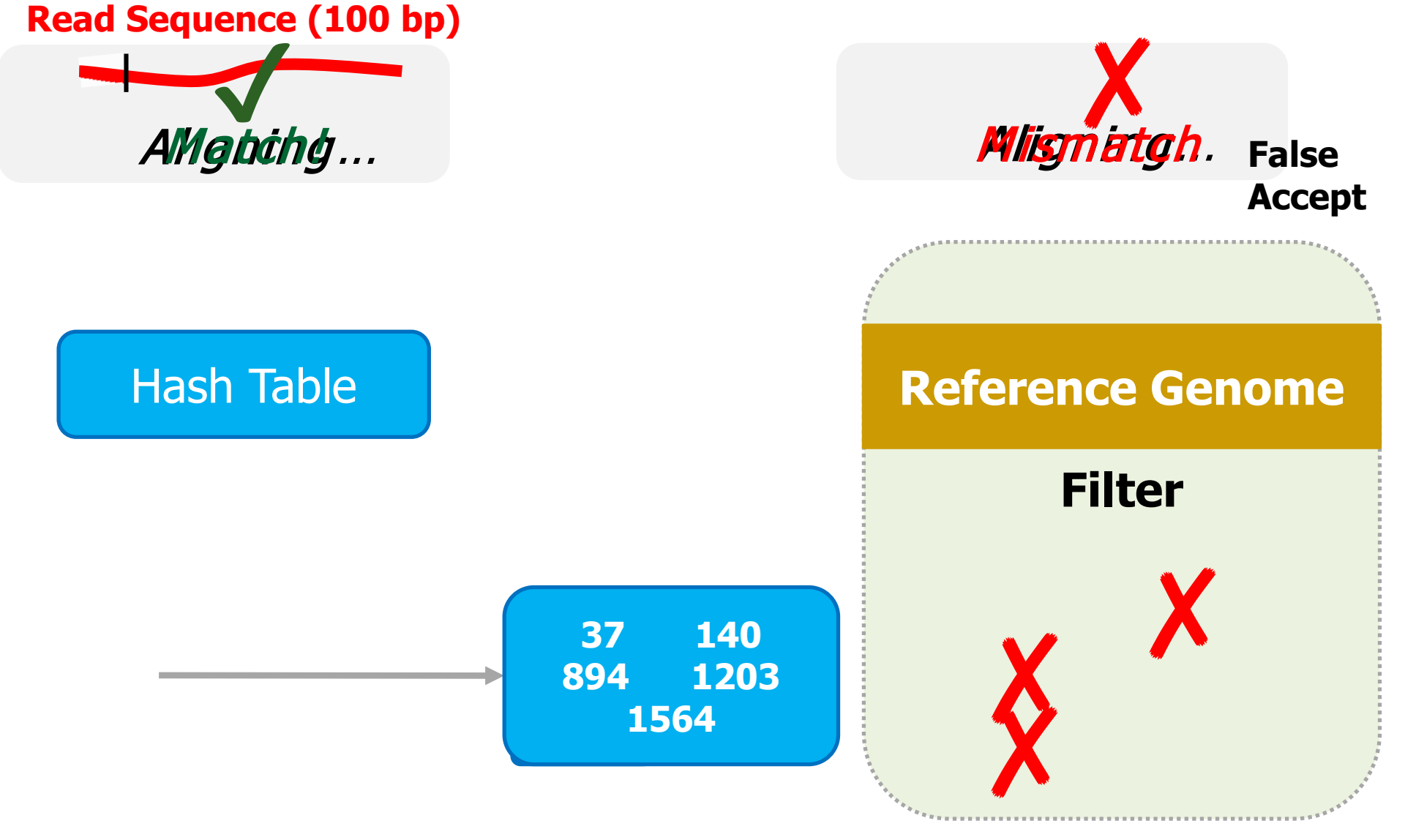


Step 3

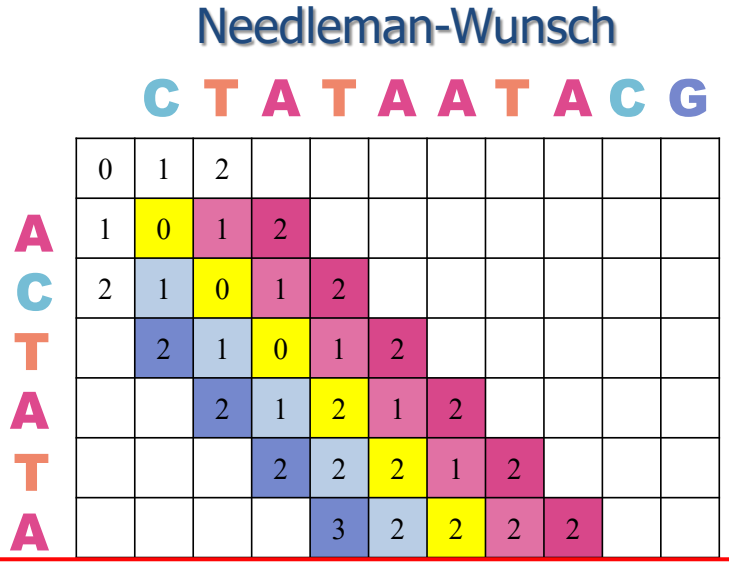
Read Alignment

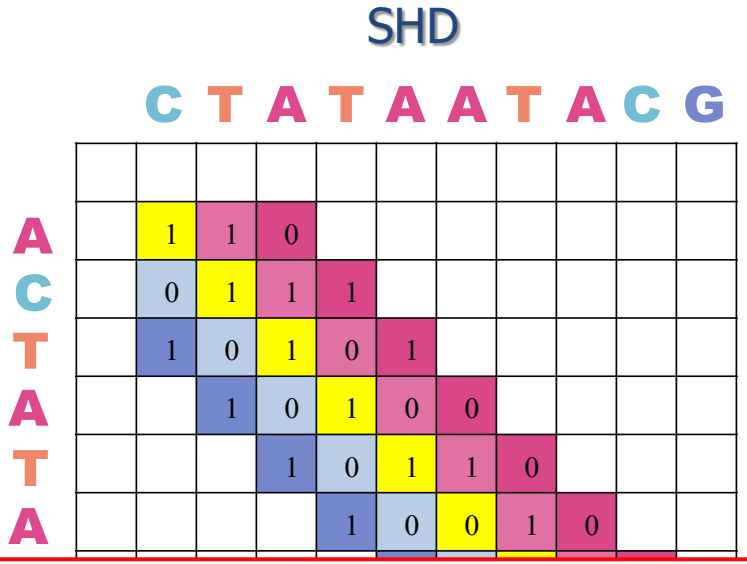
1. Filters out most of the incorrect mappings
2. Preserves all correct mappings
3. Does this quickly

Location Filtering Example

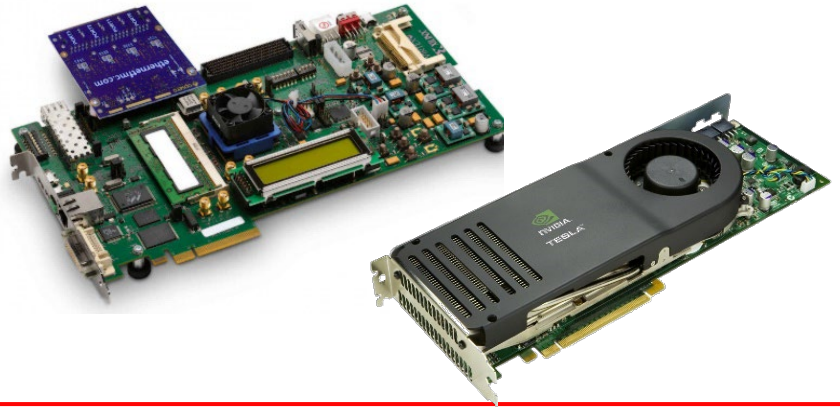


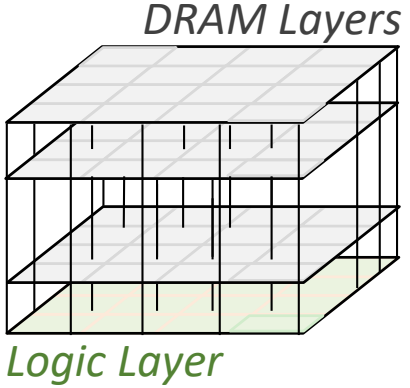
Alignment vs. Pre-alignment (Filtering)



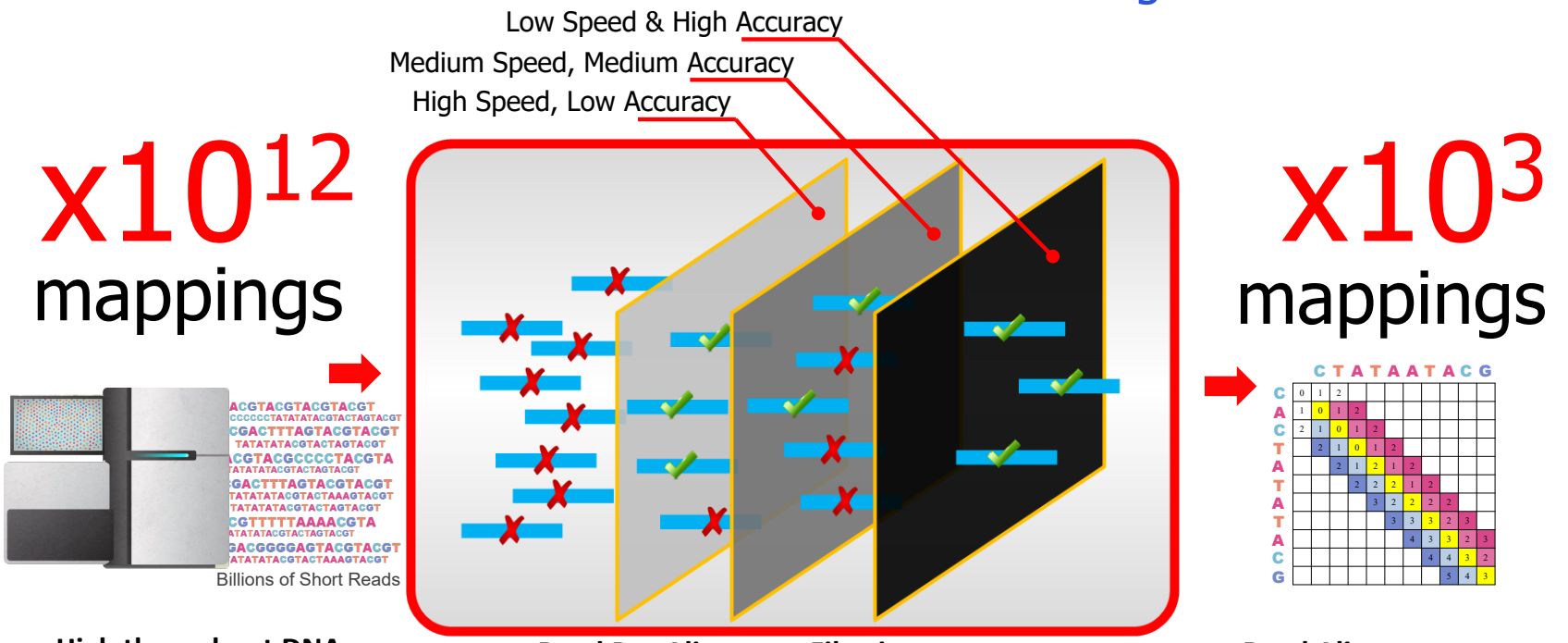


Independent vectors can be processed in parallel using hardware technologies





GateKeeper: FPGA-Based Alignment Filtering



1 **High throughput DNA sequencing (HTS) technologies**

2 **Read Pre-Alignment Filtering**
Fast & Low False Positive Rate

3 **Read Alignment**
Slow & Zero False Positives

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 Walkthrough

Generate 2E+1 masks → Amend random zeros: 101 → 111 & 1001 → 1111 → AND all masks, ACCEPT iff number of '1' ≤ Threshold

```
                Query :GAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGGAACATTGTTGGGCCGGA
            Reference :GAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGGCCGG

          Hamming Mask :00000000001000000000000111111101111000111011010110111111111000100000111101101001010
    1-Deletion Mask :11111111111001111101111100000000000000000000000000000000000000000011000000000000000
    2-Deletion Mask :00000000101101110011111111111111011110001110110101101111111100010001001110110100101
    3-Deletion Mask :11111111111011101100110111011101100010010011111111111111100101100110010110111011101111
  1-Insertion Mask :11111111111011111011111101110110001001001111111111111100101100110001010111011101111110
  2-Insertion Mask :00000010011111001111111110010001101010100110101111111111111011100111111100011110110 0
  3-Insertion Mask :11111111011101100110001111111110101101111110011001011101111111101110111101011100100 0

                                                --- Masks after amendment ---

          Hamming Mask :00000000001000000000000111111111111000111111110111111111111000100000111111111111111
    1-Deletion Mask :11111111111111111111111100000000000000000000000000000000000000000011000000000000000
    2-Deletion Mask :00000000111111111111111111111111111111000111111111111111111111000100011111111111111110
    3-Deletion Mask :11111111111111111111111111111111111000111111111111111111111111111111111111111111111111
  1-Insertion Mask :11111111111111111111111111111111000111111111111111111111111111000111111111111111110
  2-Insertion Mask :00000011111111111111111111111000111111111111111111111111111111111111111000111111100
  3-Insertion Mask :11111111111111111111000111111111111111111111111111111111111111111111111111111111000

           AND Mask :00000000001000000000000100000000000000000000000000000000000000000001000000000000000


                     GAGAGAGATATTTAGTGTTGCAG-CACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGGAACATTGTTGGGCCGG
Needleman-Wunsch
       Alignment :||||||||||| |||||||||||| ||||||||||||||||||||||||||||||||||||||||||::||||||||||||||||
                     GAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGGCCGG
```

GateKeeper Walkthrough (cont'd)

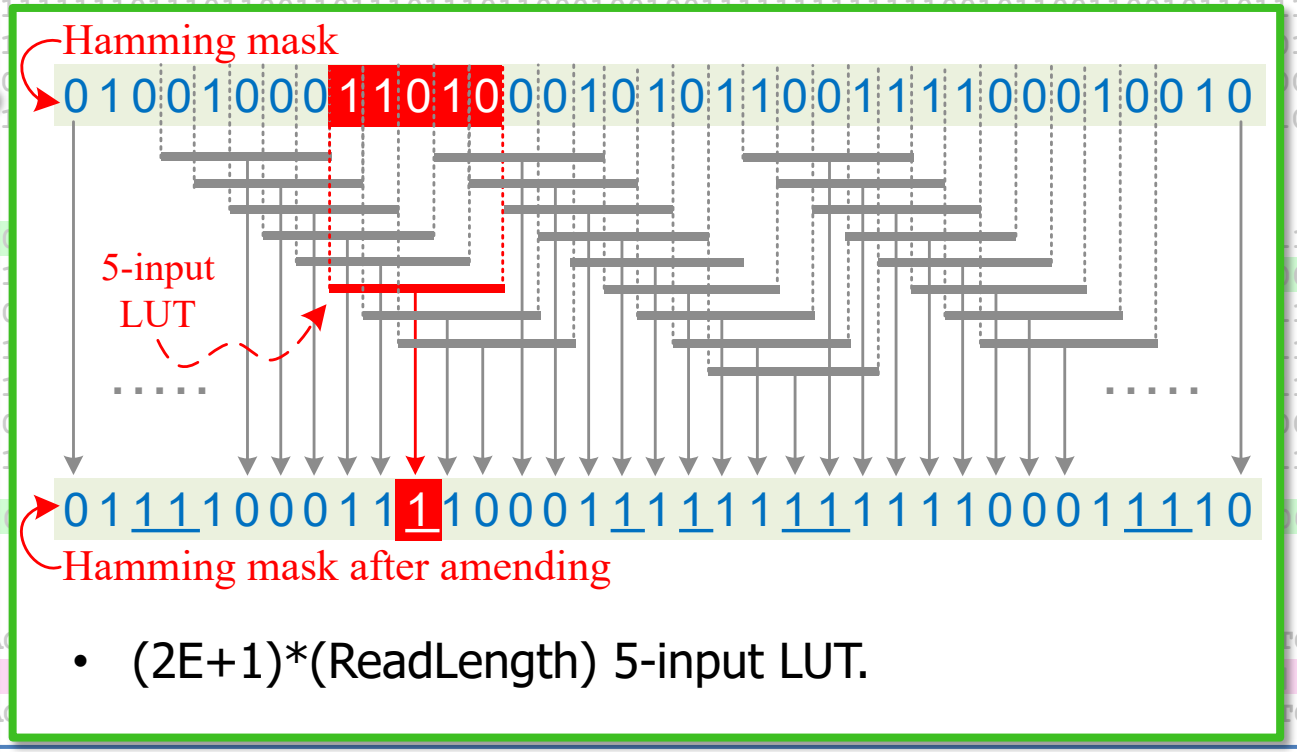
Generate 2E+1 masks

Amend random zeros: 101 → 111 & 1001 → 1111

AND all masks, ACCEPT iff number of '1' ≤ Threshold

- E right-shift registers (length=ReadLength)
- E left-shift registers (length=ReadLength)
- (2E+1) * (ReadLength) 2-XOR operations.

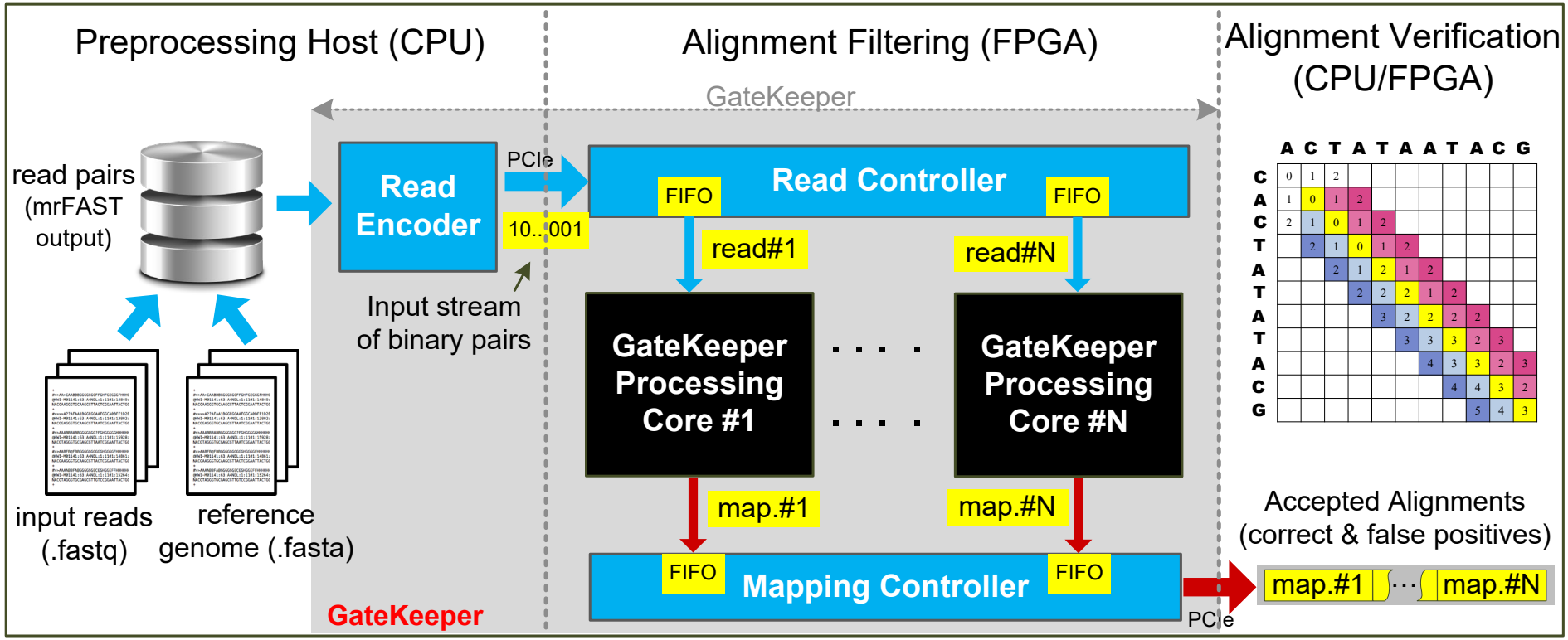
- (2E)*(ReadLength) 2-AND operations.
- (ReadLength/4) 5-input LUT.
- log_2ReadLength-bit counter.



- (2E+1)*(ReadLength) 5-input LUT.

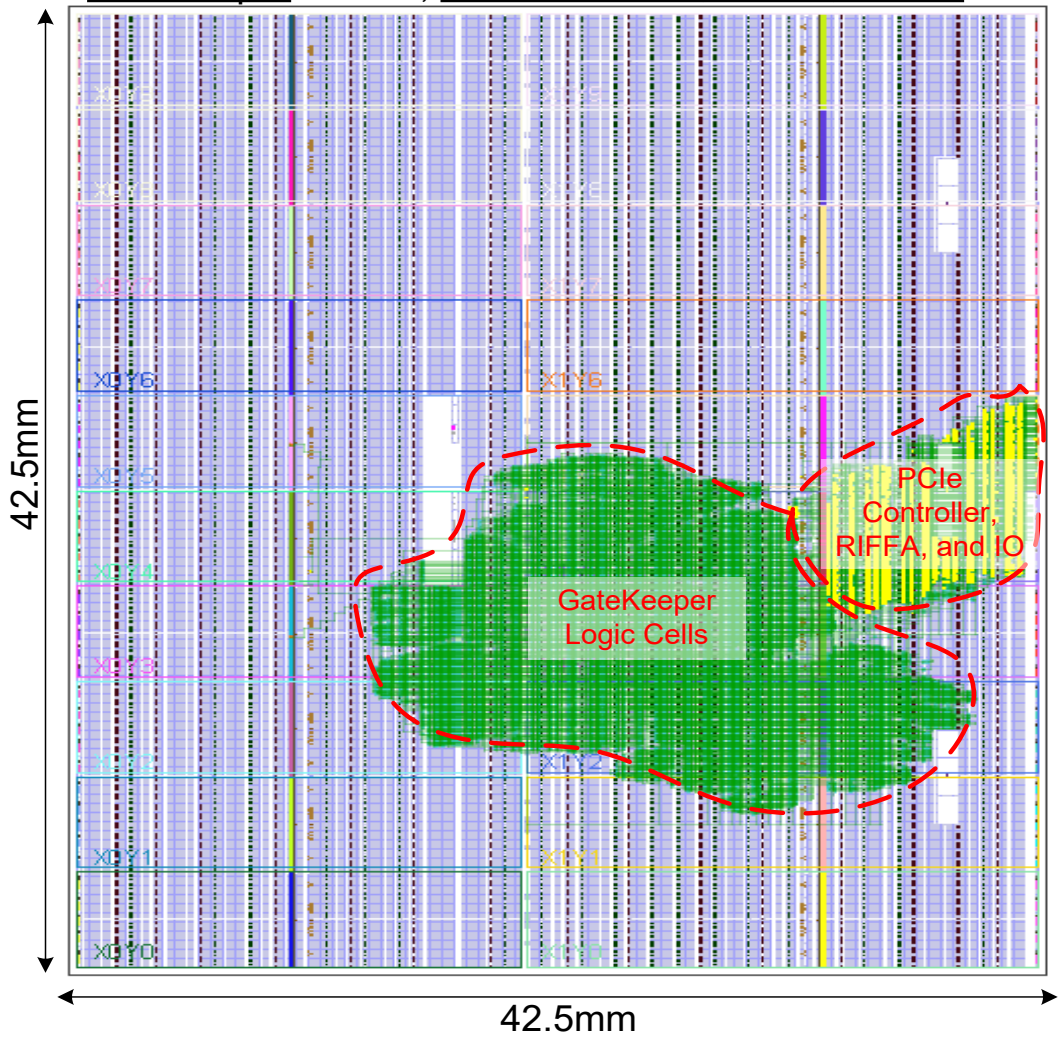
GateKeeper Accelerator Architecture

- **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 vs. SHD

GateKeeper	SHD

▪ FPGA (Xilinx VC709)
▪ Multi-core (parallel)
▪ Examines a single mapping @ 125 MHz
▪ Limited to PCIe Gen3(4x) transfer rate (128 bits @ 250MHz)
▪ Amending requires:
  ❑ (2E+1) 5-input LUT.
	▪ Intel SIMD
	▪ Single-core (sequential)
	▪ Examines a single mapping @ ~2MHz
	▪ Limited to a read length of 128 bp (SSE register size)
	▪ Amending requires:
	  ❑ 4(2E+1) bitwise OR.
	  ❑ 4(2E+1) packed shuffle.
	  ❑ 3(2E+1) shift.

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

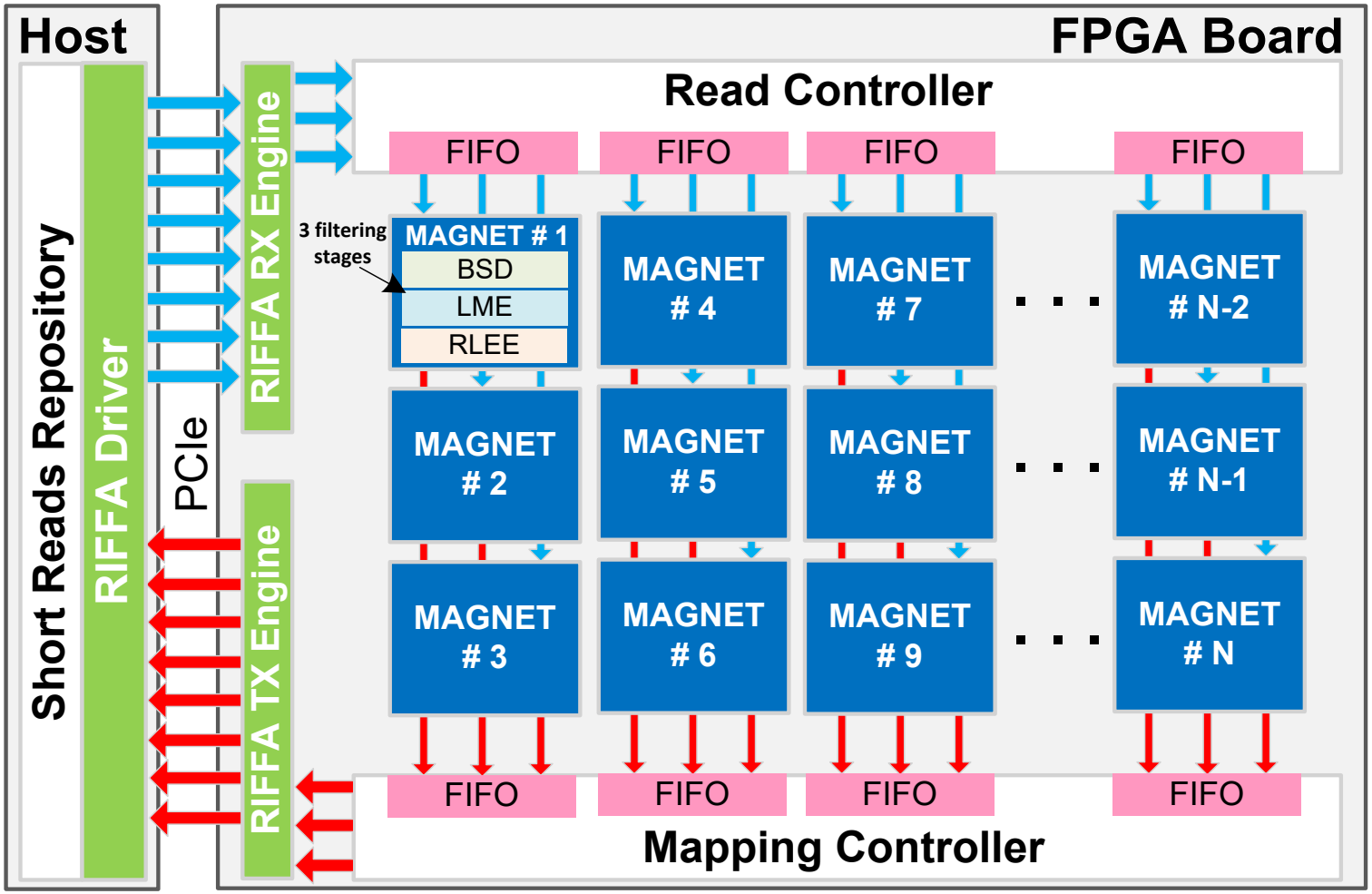
- Mohammed Alser, Hasan Hassan, Hongyi Xin, Oguz Ergin, Onur Mutlu, and Can Alkan
  "GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping"
  Bioinformatics, [published online, May 31], 2017.
  [Source Code]
  [Online link at Bioinformatics Journal]

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** ▾

MAGNET Accelerator [Alser+, TIR 2017]

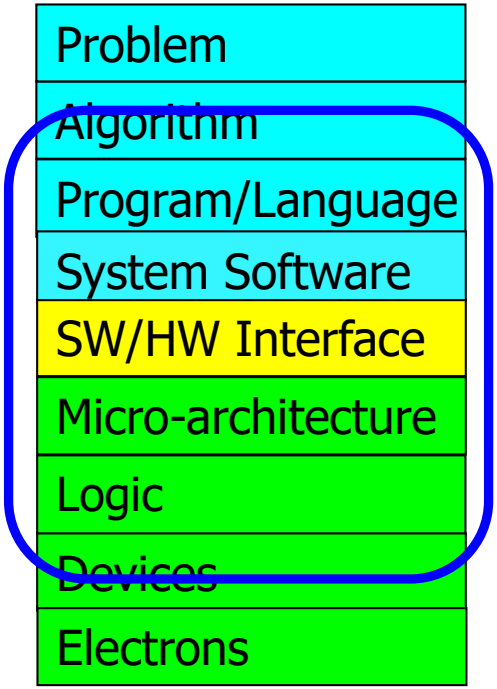


Can We Do Better?

Faster, More Accurate,
More Scalable
Pre-Alignment Filtering

Algorithm-Arch-Device Co-Design is Critical

Computer Architecture (expanded view)



Shouji (障子) [Alser+, Bioinformatics 2019]

Mohammed Alser, Hasan Hassan, Akash Kumar, Onur Mutlu, and Can Alkan,
"Shouji: A Fast and Efficient Pre-Alignment Filter for Sequence Alignment"
Bioinformatics, [published online, March 28], 2019.
[Source Code]
[Online link at Bioinformatics Journal]

Bioinformatics, 2019, 1–9
doi: 10.1093/bioinformatics/btz234
Advance Access Publication Date: 28 March 2019
Original Paper

OXFORD

Sequence alignment

Shouji: a fast and efficient pre-alignment filter for sequence alignment

Mohammed Alser[1,2,3,*], Hasan Hassan[1], Akash Kumar[2], Onur Mutlu[1,3,*] and Can Alkan[3,*]

[1]Computer Science Department, ETH Zürich, Zürich 8092, Switzerland, [2]Chair for Processor Design, Center For Advancing Electronics Dresden, Institute of Computer Engineering, Technische Universität Dresden, 01062 Dresden, Germany and [3]Computer Engineering Department, Bilkent University, 06800 Ankara, Turkey

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

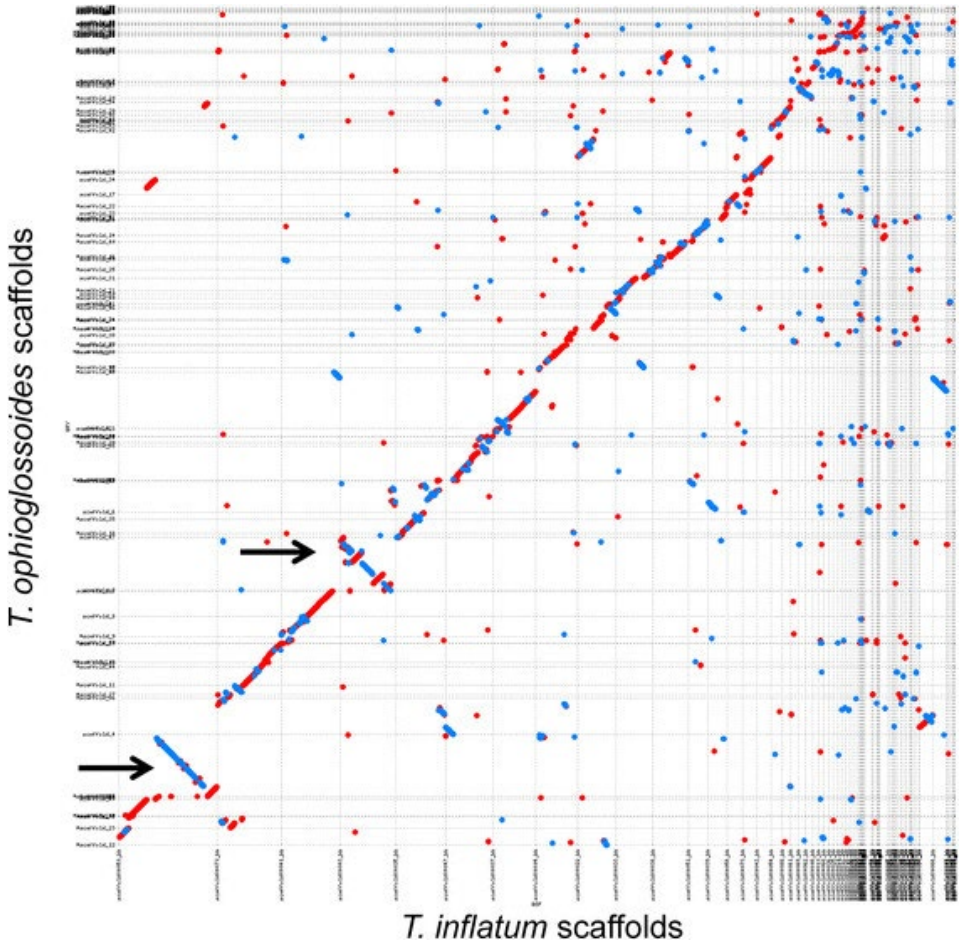
Received on September 13, 2018; revised on February 27, 2019; editorial decision on March 7, 2019; accepted on March 27, 2019

Shouji

- **Key observation:**
  - Correct alignment always includes long identical subsequences
  - Processing the entire sequence at once is ineffective for hardware design
- **Key idea:**
  - Use an **overlapping** sliding window approach to quickly and accurately find all long identical subsequences (consecutive zeros)
- **Key result:**
  - Shouji accelerates the **best-performing CPU read aligner** Edlib (Bioinformatics 2017) by up to 18.8x using 16 filtering units that work in parallel
  - Shouji on FPGA is up to 10,000x faster than on CPU
  - Shouji is 2.4x to 467x more accurate than GateKeeper (Bioinformatics 2017) and SHD (Bioinformatics 2015)

Shouji

- **Key observation:**
  - Correct alignment always includes long identical subsequences

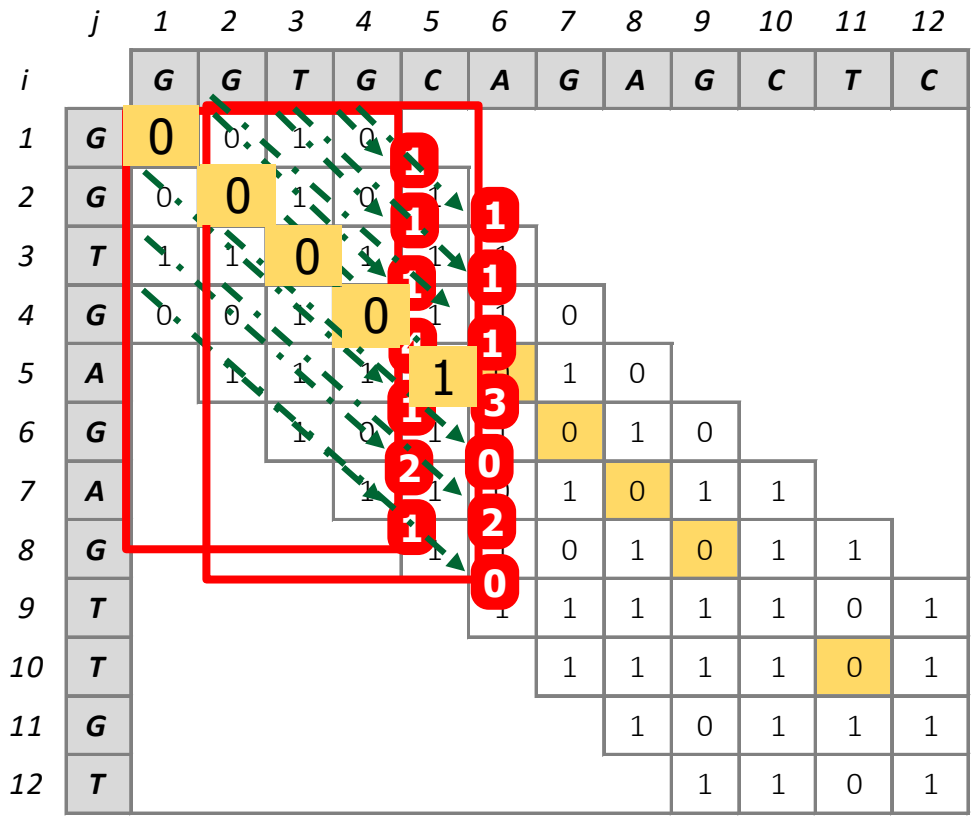


Dot plot, dot matrix
(Lipman and Pearson, 1985)

Shouji Walkthrough

Build the Neighborhood Map

Find all common subsequences (diagonal segments of consecutive zeros) shared between two given sequences



Store longest subsequence in Shouji Bit-vector

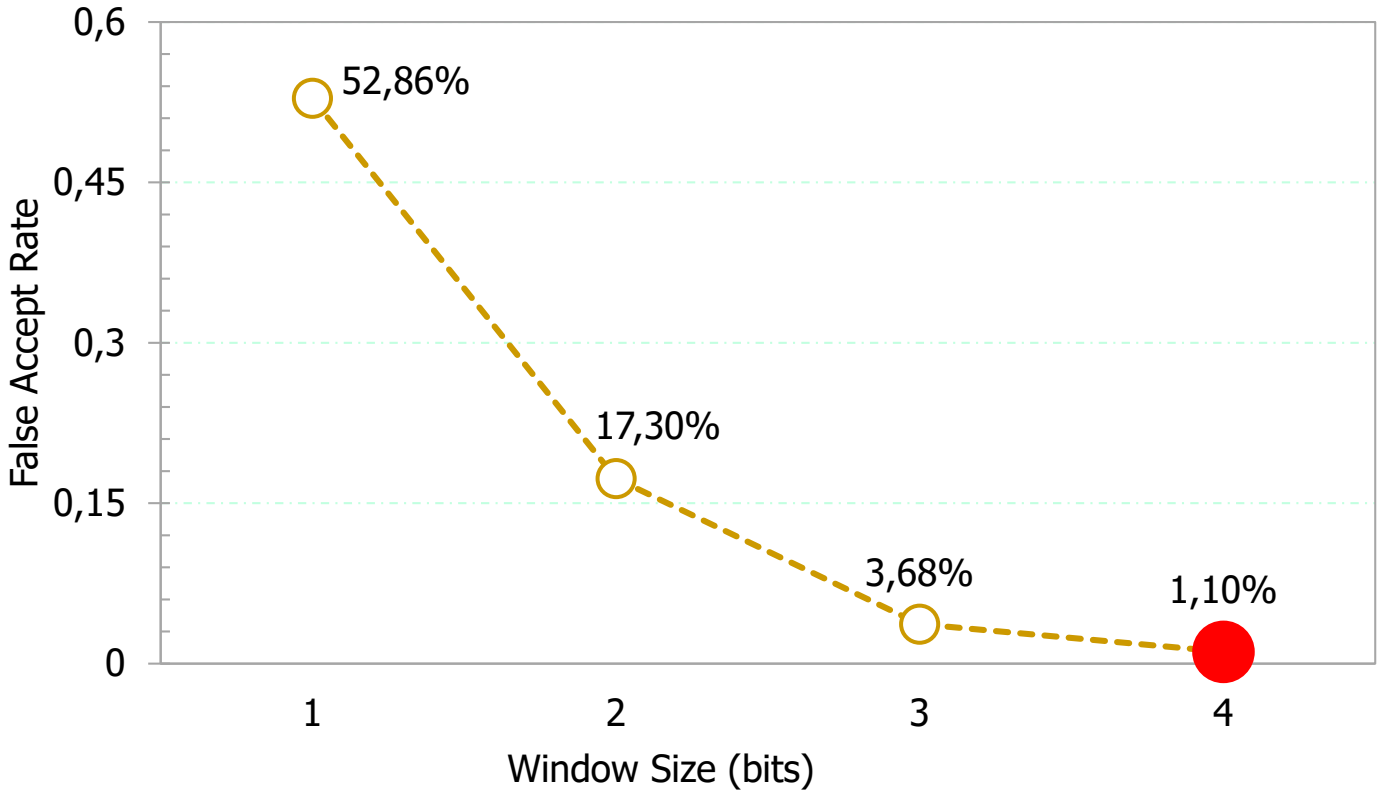
0	0	0	0	1	0	0	0	0	1	0	1

ACCEPT iff number of '1's ≤ Threshold

Shouji: a fast and efficient pre-alignment filter for sequence alignment, *Bioinformatics* 2019, https://doi.org/10.1093/bioinformatics/btz234

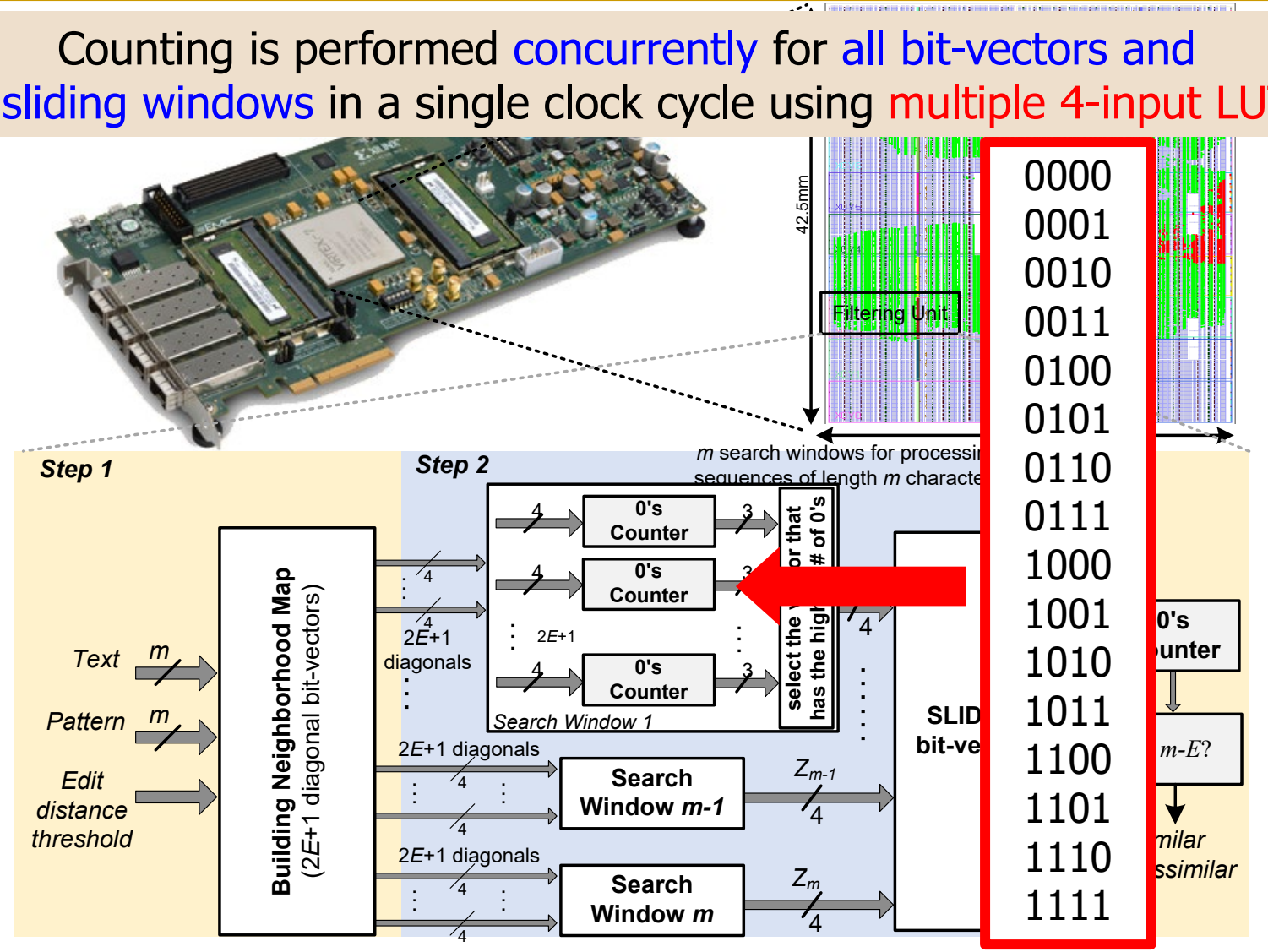
Effect of Sliding Window Size

- Large enough window to accurately capture longer streaks of matches → lower false positives
- Small enough window to perform fast computation



Hardware Implementation

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



More on Shouji (障子) [Alser+, Bioinformatics 2019]

Mohammed Alser, Hasan Hassan, Akash Kumar, Onur Mutlu, and Can Alkan,
"Shouji: A Fast and Efficient Pre-Alignment Filter for Sequence Alignment"
Bioinformatics, [published online, March 28], 2019.
[Source Code]
[Online link at Bioinformatics Journal]

Bioinformatics, 2019, 1–9
doi: 10.1093/bioinformatics/btz234
Advance Access Publication Date: 28 March 2019
Original Paper

OXFORD

Sequence alignment

Shouji: a fast and efficient pre-alignment filter for sequence alignment

Mohammed Alser[1,2,3,*], Hasan Hassan[1], Akash Kumar[2], Onur Mutlu[1,3,*] and Can Alkan[3,*]

[1]Computer Science Department, ETH Zürich, Zürich 8092, Switzerland, [2]Chair for Processor Design, Center For Advancing Electronics Dresden, Institute of Computer Engineering, Technische Universität Dresden, 01062 Dresden, Germany and [3]Computer Engineering Department, Bilkent University, 06800 Ankara, Turkey

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on September 13, 2018; revised on February 27, 2019; editorial decision on March 7, 2019; accepted on March 27, 2019

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]

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

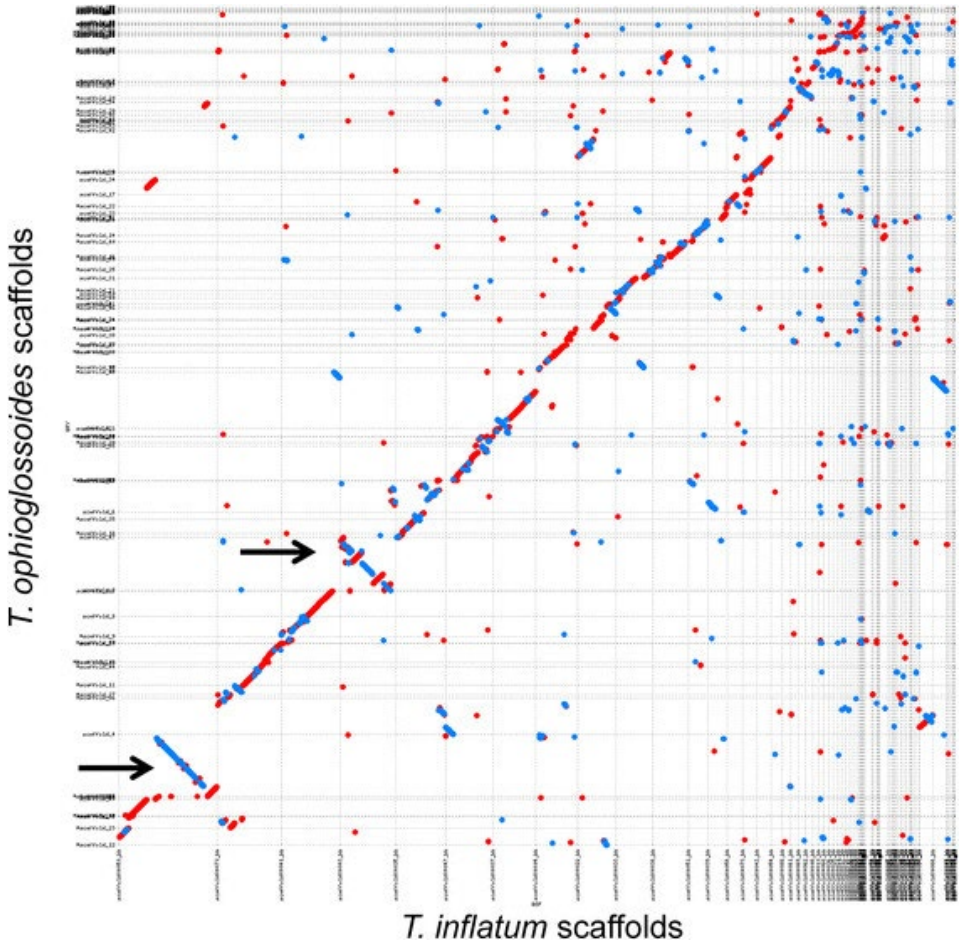
SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs

Mohammed Alser [1,2,*], Taha Shahroodi [1], Juan Gómez-Luna [1,2], Can Alkan [4,*], and Onur Mutlu [1,2,3,4,*]

[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

SneakySnake

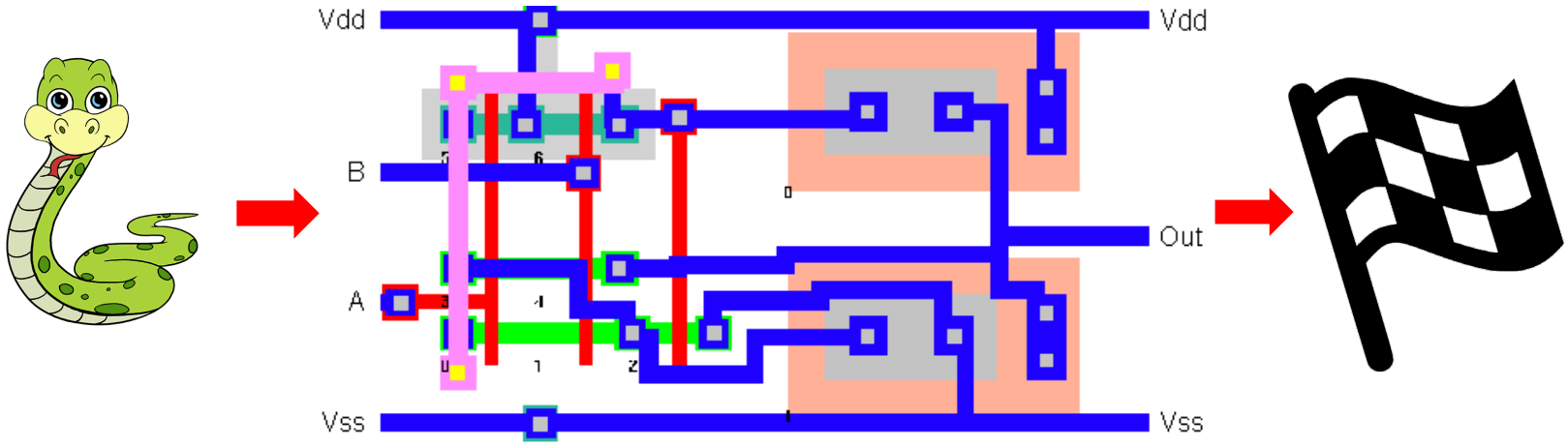
- **Key observation:**
  - Correct alignment is a sequence of non-overlapping long matches



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



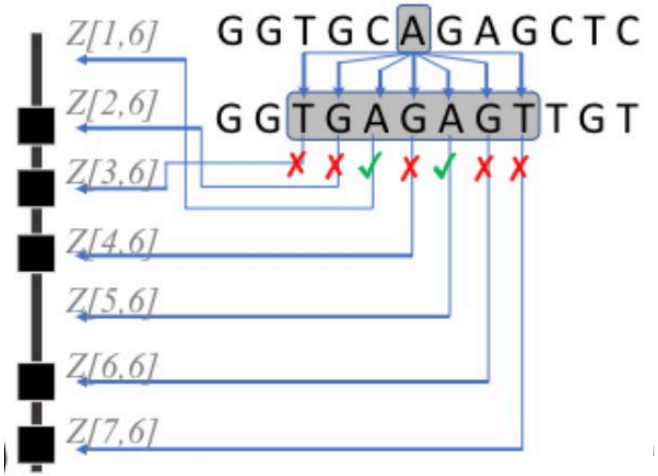
VLSI chip layout

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*

SneakySnake Walkthrough

Building Neighborhood Map | Finding the Optimal Routing Path | Examining the Snake Survival



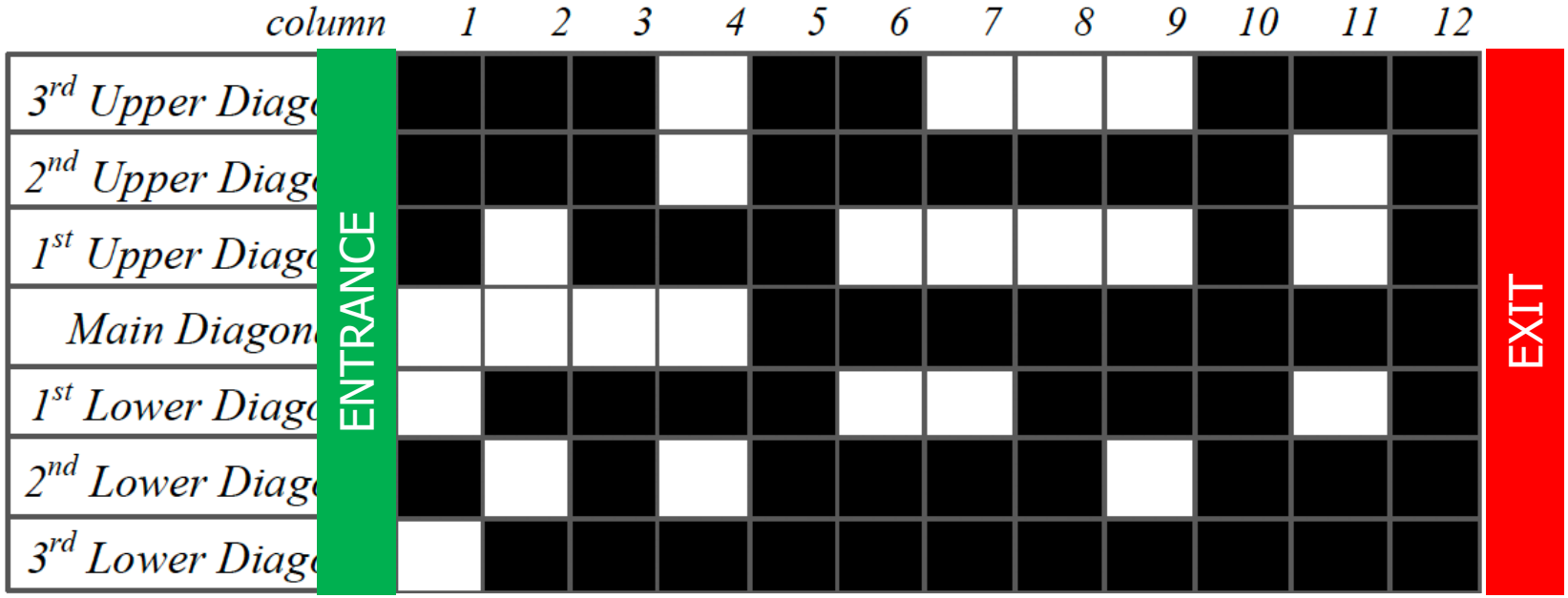
$E = 3$

column	1	2	3	4	5	6	7	8	9	10	11	12
3^{rd} Upper Diagonal	1	1	1	0	1	1	0	0	0	1	1	1
2^{nd} Upper Diagonal	1	1	1	0	1	1	1	1	1	1	0	1
1^{st} Upper Diagonal	1	0	1	1	1	0	0	0	0	1	0	1
Main Diagonal	0	0	0	0	1	1	1	1	1	1	1	1
1^{st} Lower Diagonal	0	1	1	1	1	0	0	1	1	1	0	1
2^{nd} Lower Diagonal	1	0	1	0	1	1	1	1	0	1	1	1
3^{rd} Lower Diagonal	0	1	1	1	1	1	1	1	1	1	1	1

SneakySnake Walkthrough

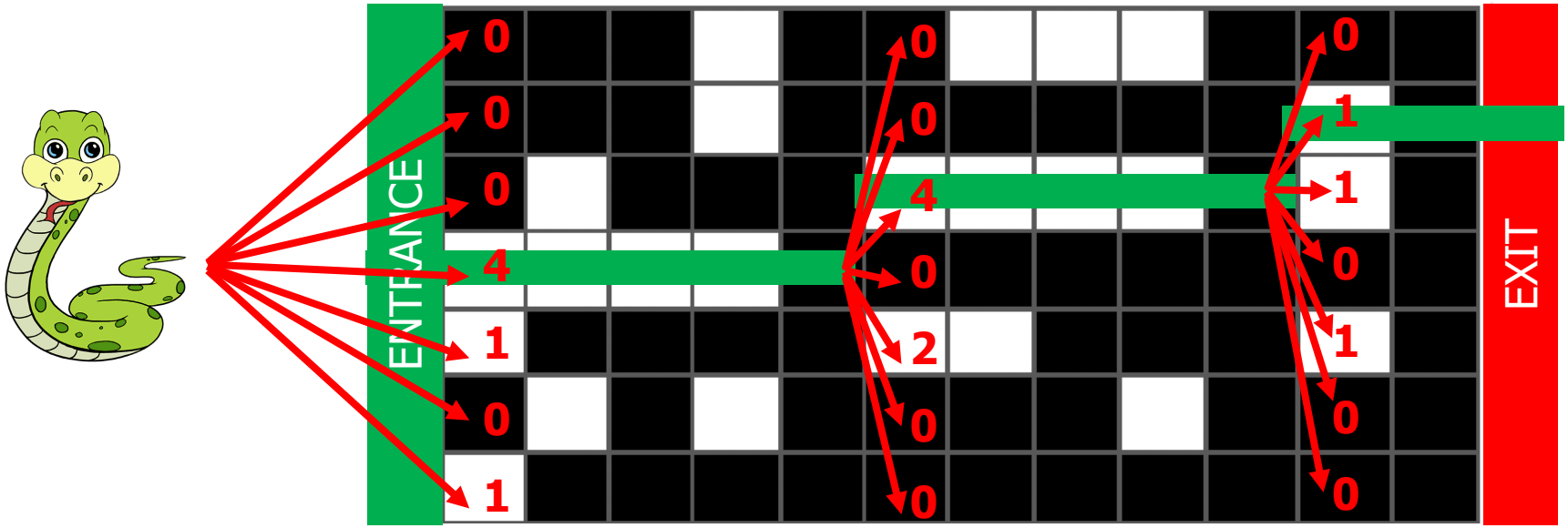
Building Neighborhood Map | Finding the Optimal Routing Path | Examining the Snake Survival

$E = 3$



SneakySnake Walkthrough

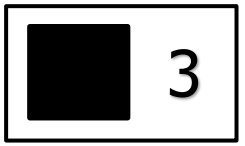
Building Neighborhood Map | Finding the Optimal Routing Path | Examining the Snake Survival

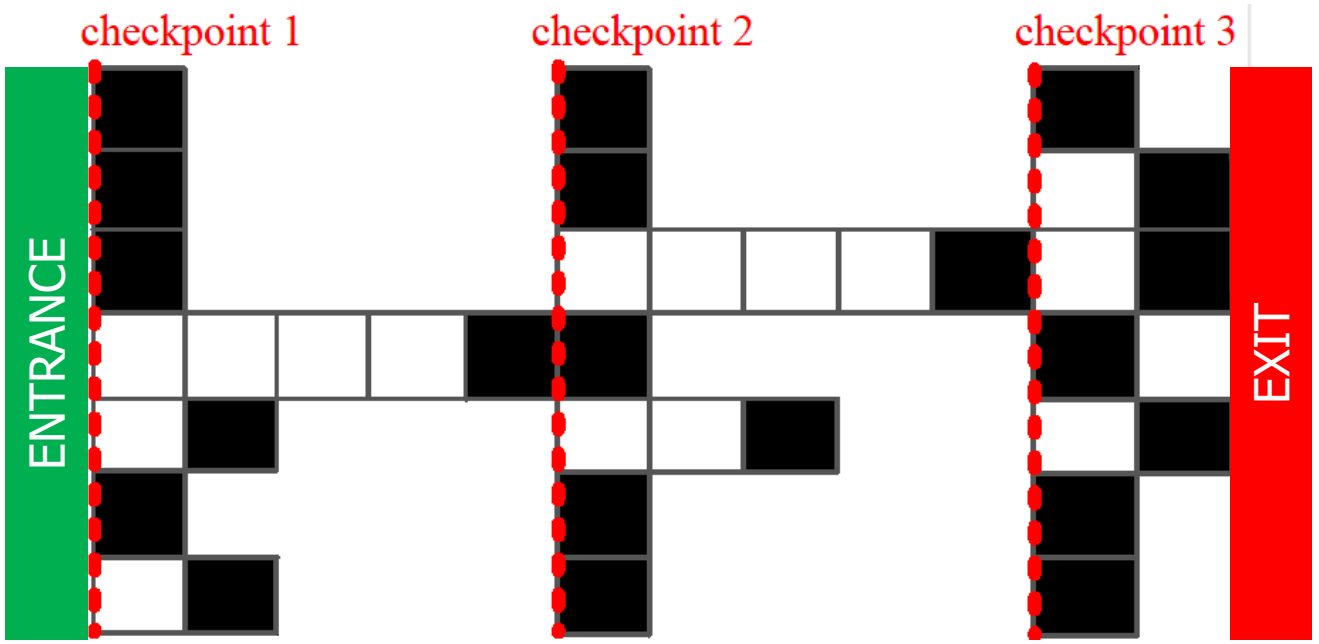


SneakySnake Walkthrough

Building Neighborhood Map | Finding the Routing Travel Path | Examining the Snake Survival

This is what you actually need to build and it can be done on-the-fly!





FPGA Resource Analysis

- FPGA resource usage for a single filtering unit of GateKeeper, Shouji, and Snake-on-Chip for a sequence length of 100 and under different edit distance thresholds (E).

	E (bp)	Slice LUT	Slice Register	No. of Filtering Units
GateKeeper	2	0.39%	0.01%	16
	5	0.71%	0.01%	16
Shouji	2	0.69%	0.08%	16
	5	1.72%	0.16%	16
Snake-on-Chip	2	0.68%	0.16%	16
	5	1.42%	0.34%	16

Key Results of SneakySnake

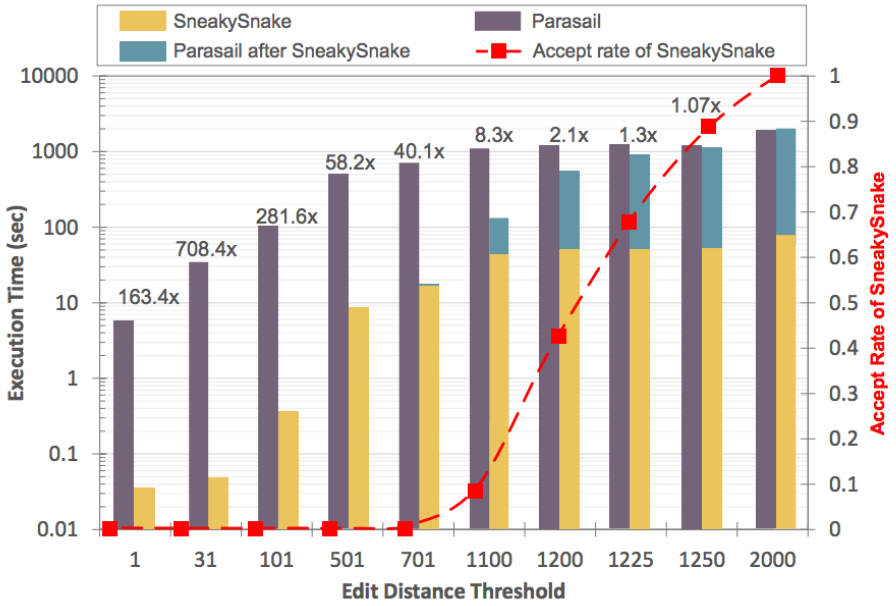
- SneakySnake is up to four orders of magnitude more accurate than Shouji (Bioinformatics'19) and GateKeeper (Bioinformatics'17)

- Short reads:
  - SneakySnake accelerates Edlib (Bioinformatics'17) and Parasail (BMC Bioinformatics'16) by
  - up to 37.7× and 43.9× (>12× on average), on CPUs
  - up to 413× and 689× (>400× on average) using *FPGAs/GPUs*

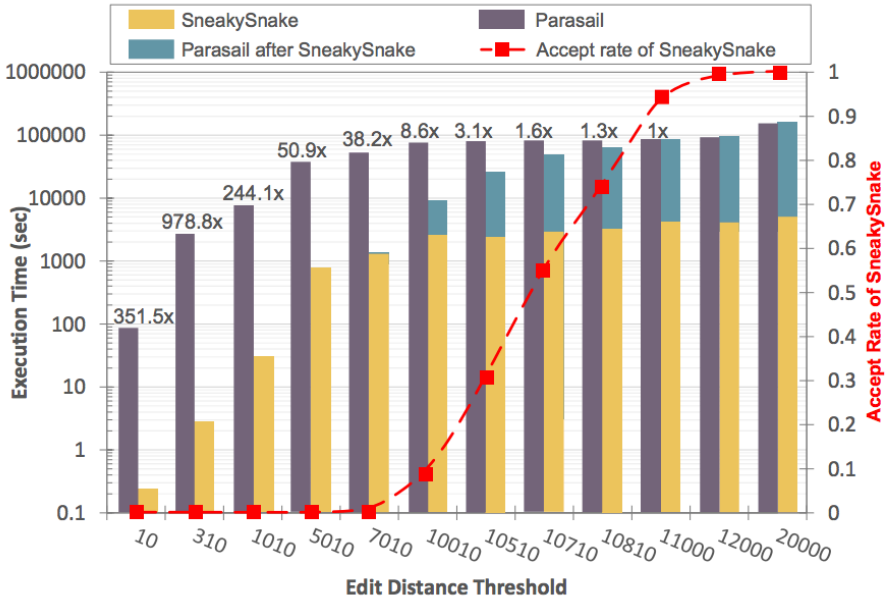
- Long reads:
  - SneakySnake accelerates Parasail and KSW2 by 140.1× and 17.1× on average, respectively, on CPUs

Long Read Mapping (SneakySnake vs Parasail)

10K bp reads **100K bp reads**



(a)



(b)

Fig. 10: The execution time of SneakySnake, Parasail, and SneakySnake integrated with Parasail using long sequences, (a) 10Kbp and (b) 100Kbp, and 40 CPU threads. The left y-axes of (a) and (b) are on a logarithmic scale. For each edit distance threshold value, we provide in the right y-axes of (a) and (b) the rate of accepted pairs (out of 100,000 pairs for 10Kbp and out of 74,687 pairs for 100Kbp) by SneakySnake that are passed to Parasail. We present the end-to-end speedup values obtained by integrating SneakySnake with Parasail.

Long Read Mapping (SneakySnake vs KSW2)

10K bp reads **100K bp reads**

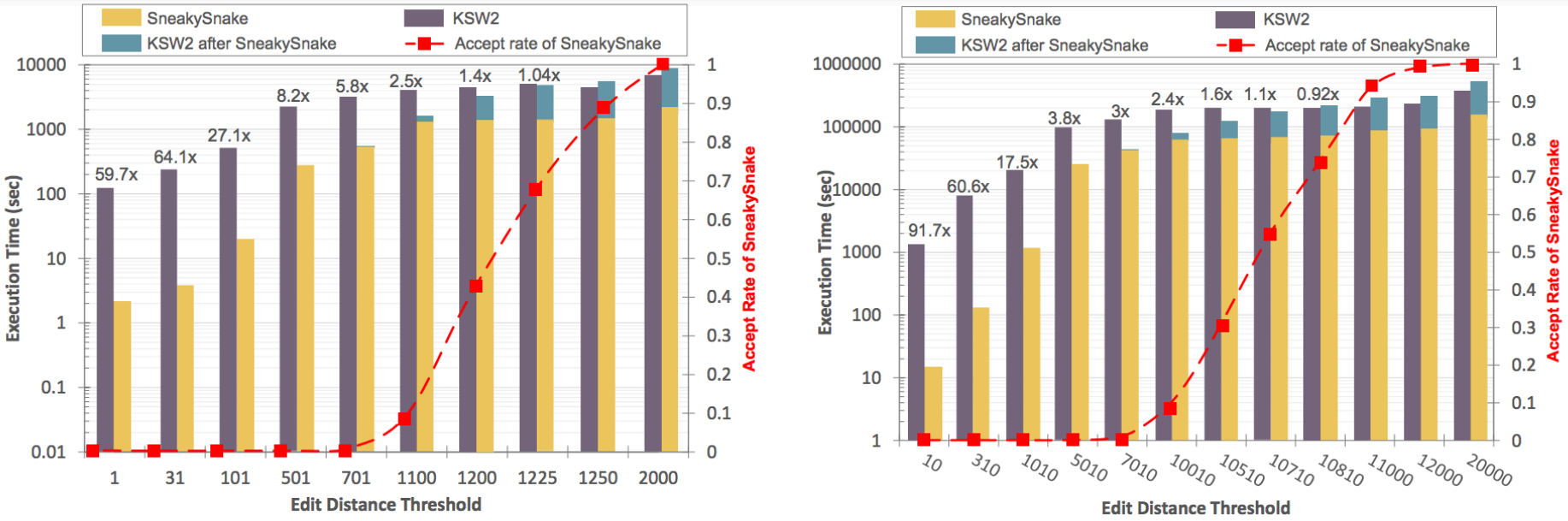


Fig. 11: The execution time of SneakySnake, KSW2, and SneakySnake integrated with KSW2 using long sequences, (a) 10Kbp and (b) 100Kbp, and a single CPU thread. The left y-axes of (a) and (b) are on a logarithmic scale. For each edit distance threshold value, we provide in the right y-axes of (a) and (b) the rate of accepted pairs (out of 100,000 pairs for 10Kbp and out of 74,687 pairs for 100Kbp) by SneakySnake that are passed to KSW2. We present the end-to-end speedup values obtained by integrating SneakySnake with KSW2.

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]

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

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

Mohammed Alser [1,2,*], Taha Shahroodi [1], Juan Gómez-Luna [1,2], Can Alkan [4,*], and Onur Mutlu [1,2,3,4,*]

[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

GenASM Framework [MICRO 2020]

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu, **"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"**
  Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.
  [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*

Problem & Our Goal

- Multiple steps of read mapping require *approximate string matching*
  - ASM enables read mapping to account for sequencing errors and genetic variations in the reads
- ASM makes up a significant portion of read mapping (more than 70%)
- One of the major bottlenecks of genome sequence analysis

Our Goal:

Accelerate approximate string matching by designing a fast and flexible framework, which can be used to accelerate *multiple steps* of the genome sequence analysis pipeline

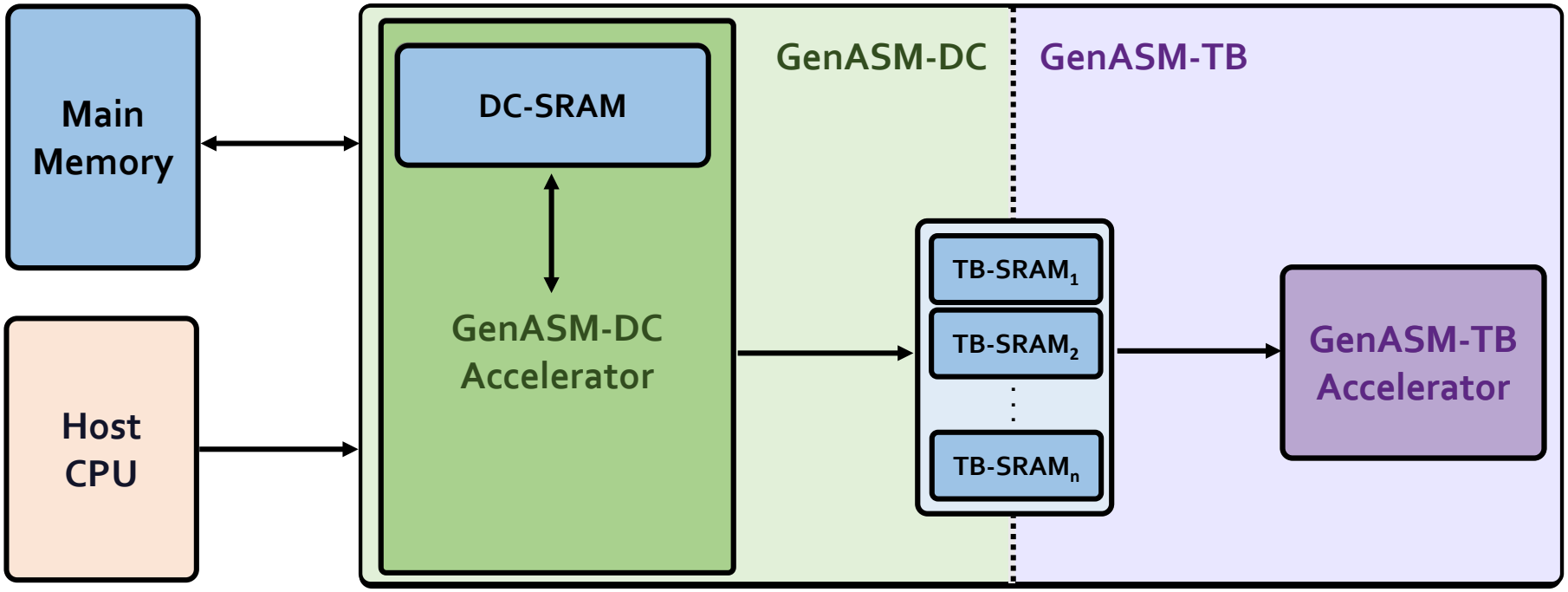
GenASM: ASM Framework for GSA

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

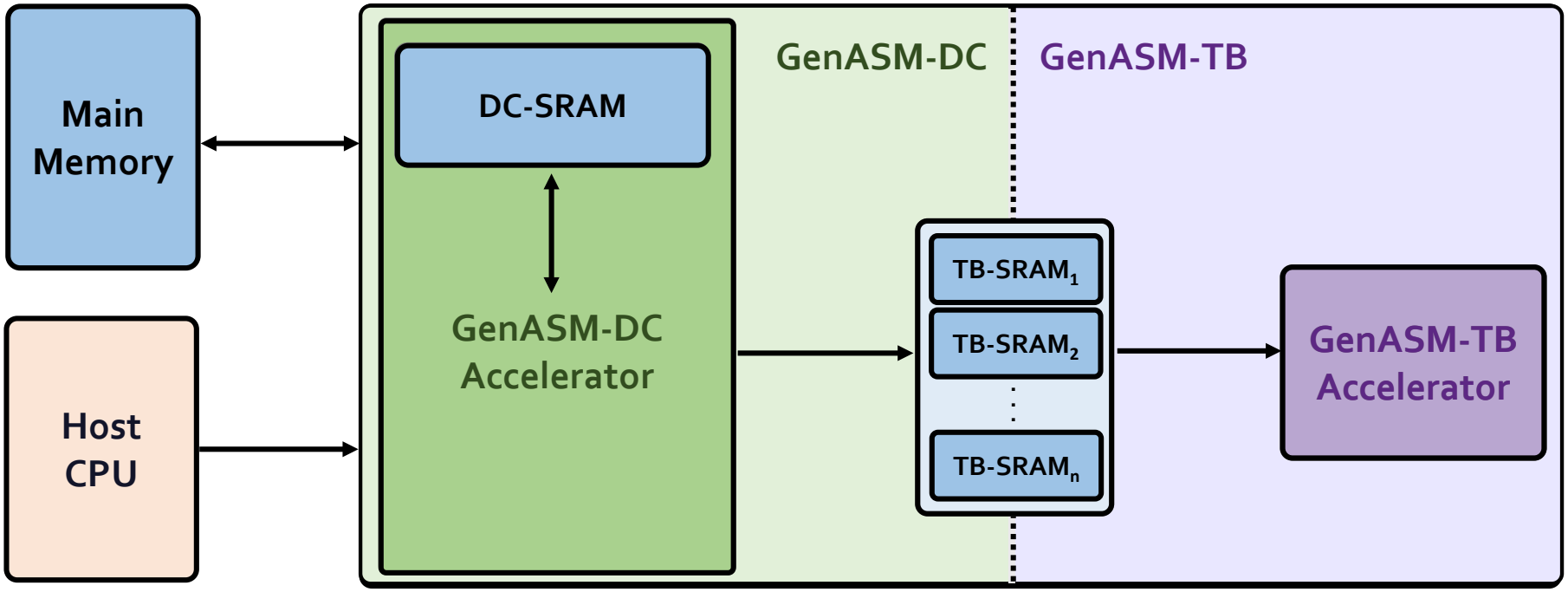
GenASM: Hardware Design



GenASM-DC:
generates bitvectors and performs edit **D**istance **C**alculation

GenASM-TB:
performs **T**race**B**ack and assembles the optimal alignment

GenASM: Hardware Design



Our *specialized compute units* and *on-chip SRAMs* help us to:

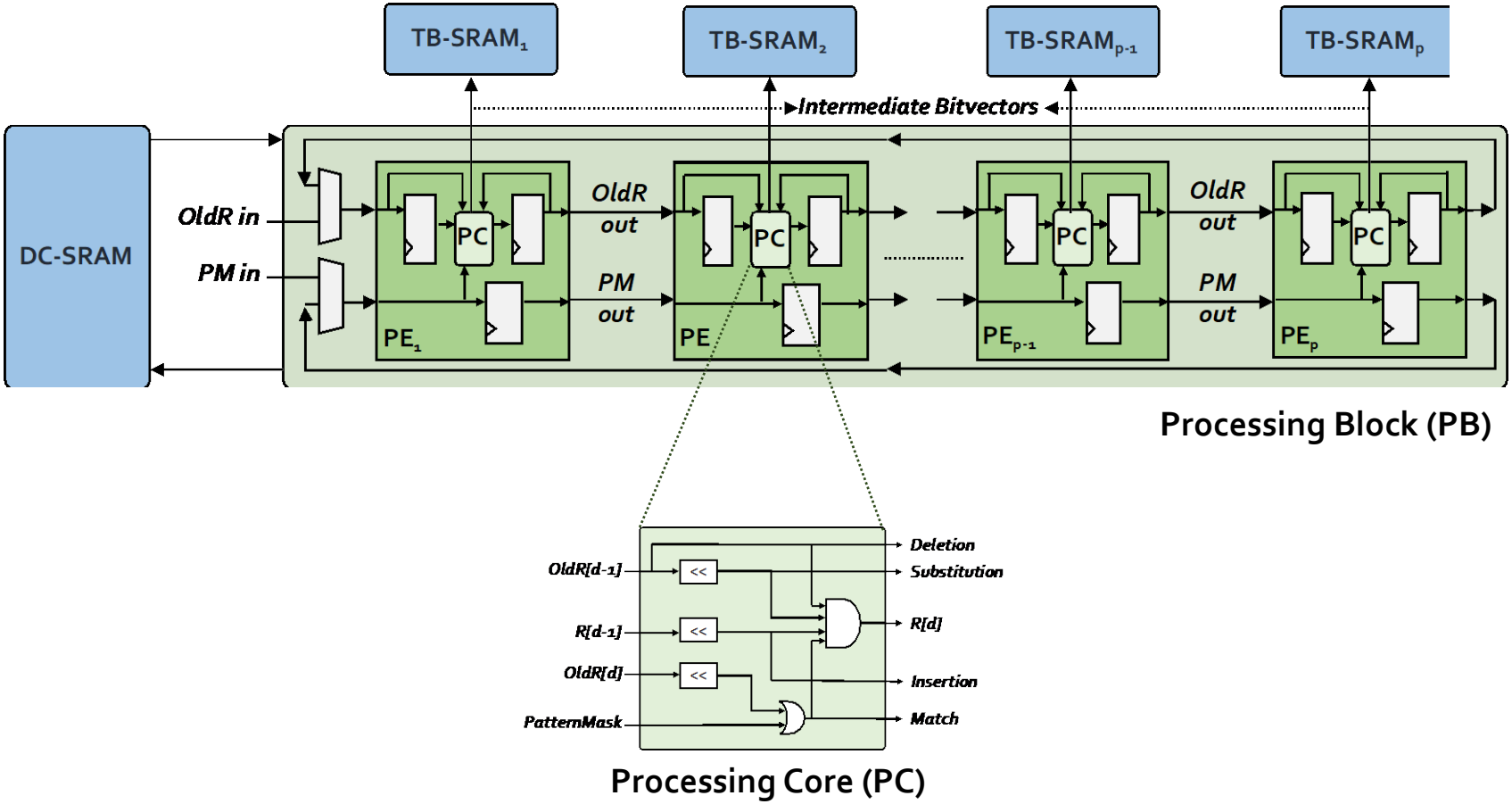
→ Match the rate of computation with memory capacity and bandwidth

→ Achieve high performance and power efficiency

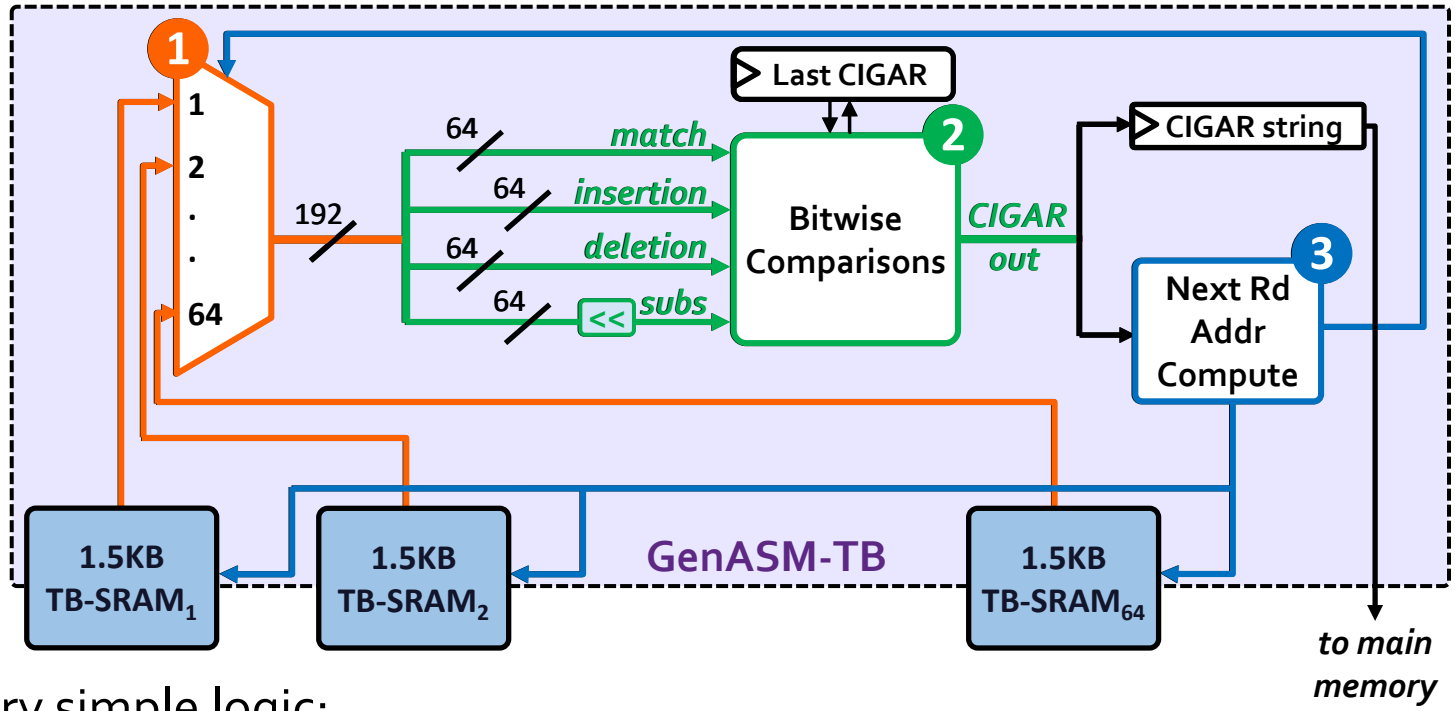
→ Scale linearly in performance with the number of parallel compute units that we add to the system

GenASM-DC: Hardware Design

- **Linear cyclic systolic array** based accelerator
  - Designed to maximize parallelism and minimize memory bandwidth and memory footprint



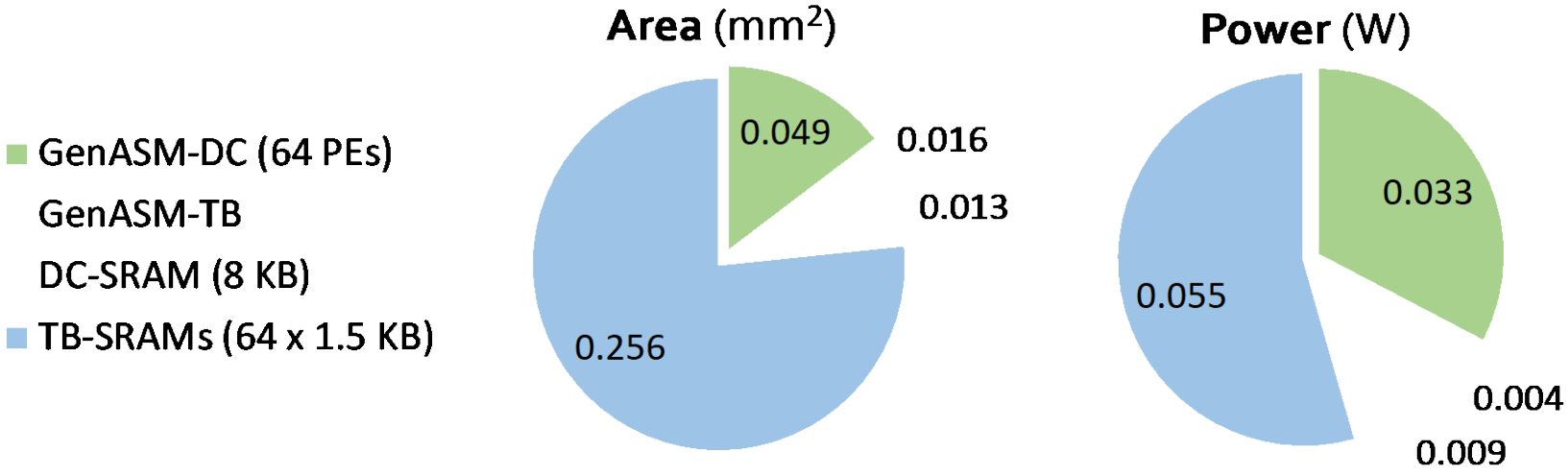
GenASM-TB: Hardware Design



- 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

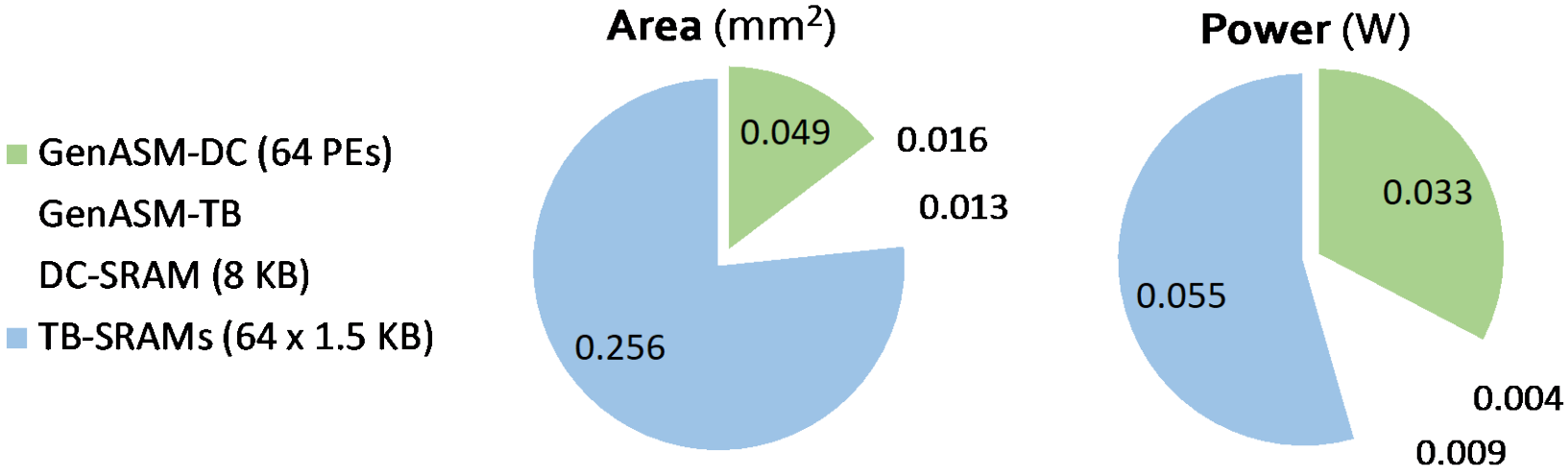
- Based on our **synthesis** of **GenASM-DC** and **GenASM-TB** accelerator datapaths using the Synopsys Design Compiler with a **28nm** LP process:
  - Both GenASM-DC and GenASM-TB operate **@ 1GHz**



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%**

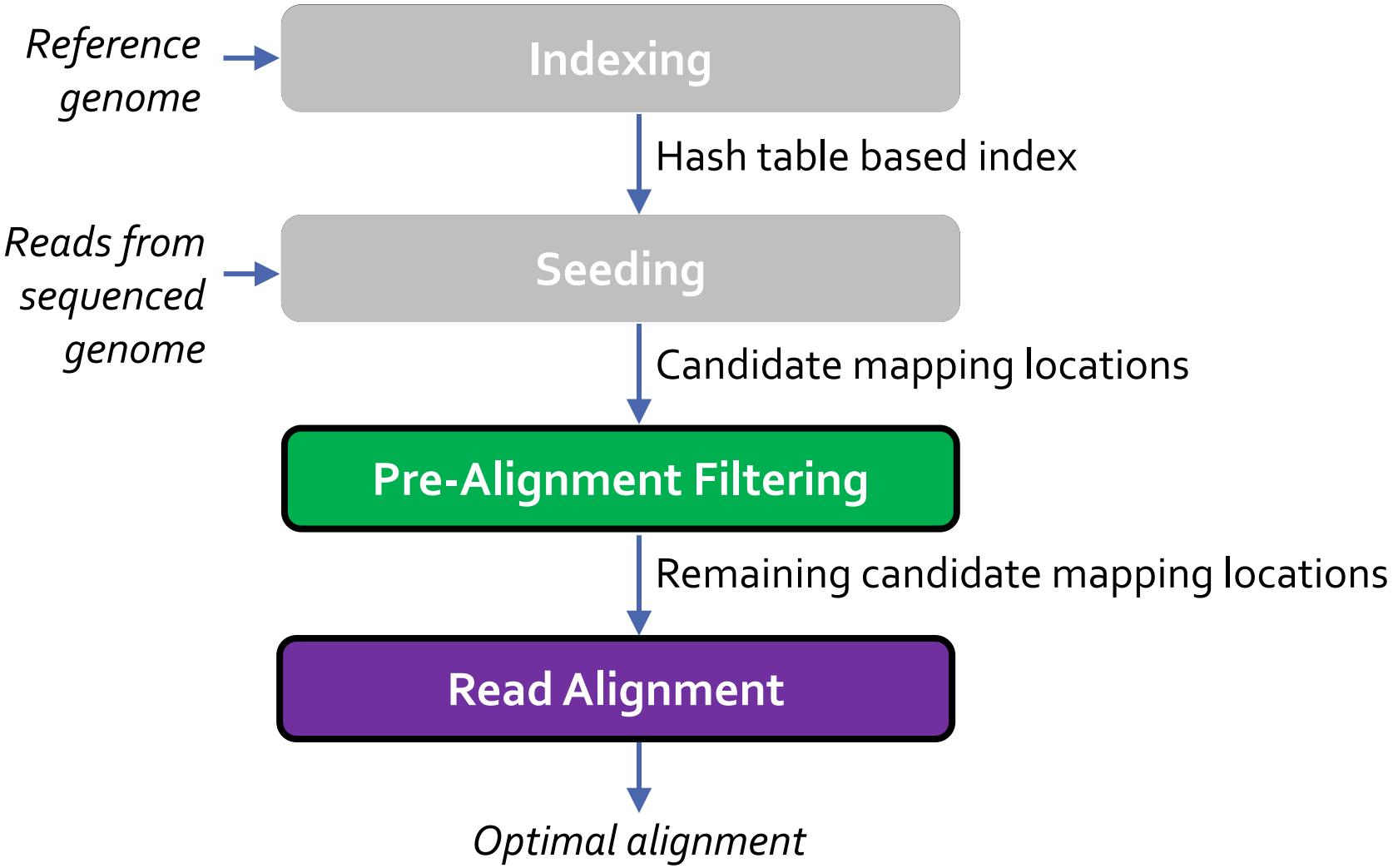
Key Results – Area and Power

- 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



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]

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu, **"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"**
  Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.
  [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*

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"
  *Proceedings of the 49th International Symposium on Computer Architecture (**ISCA**)*, New York, June 2022.
  [arXiv version]

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

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

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

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

Damla Senol Cali, Ph.D.

damlasenolcali@gmail.com
https://damlasenolcali.github.io/

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

Carnegie Mellon
ETH zürich
Bilkent University
intel
University of Illinois Urbana-Champaign
SAFARI

Genome Sequence Analysis

- Mapping the reads to a reference genome (i.e., ***read mapping***) is a *critical step* in genome sequence analysis

Linear Reference: ACGTACGT

Read: ACGG

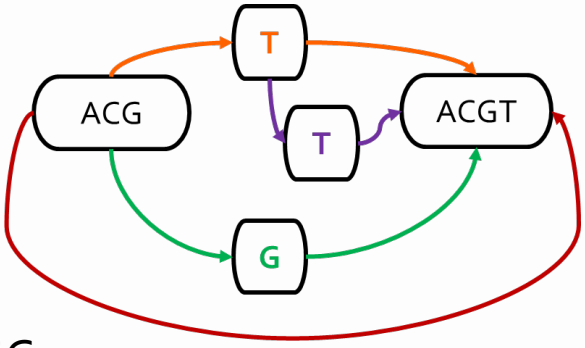
Alternative Sequence: ACGGACGT

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

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**)

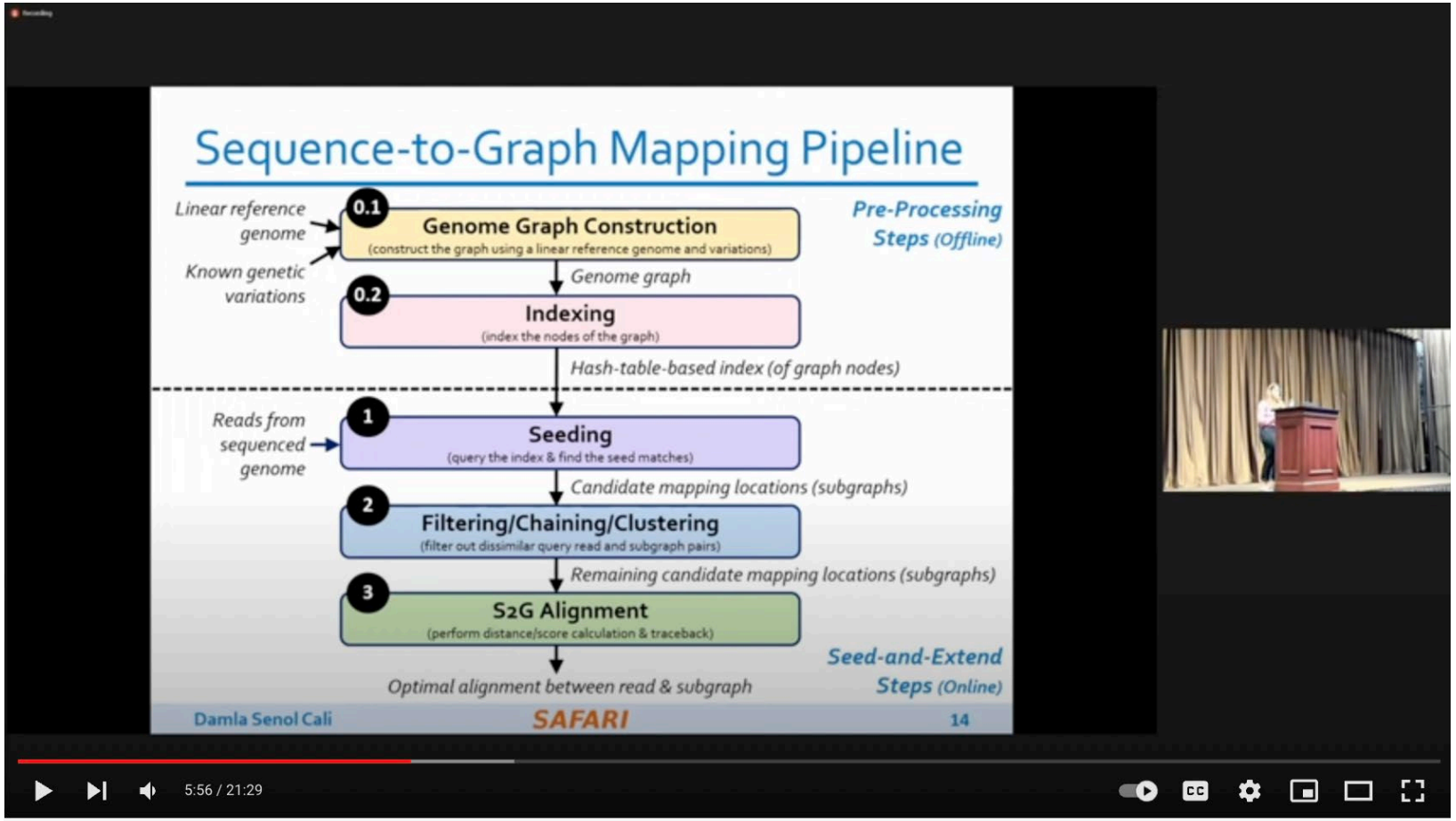
(2) Sequence-to-Graph (S2G) Alignment

- **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



https://www.youtube.com/watch?v=gyjqYoyDP9s

Accelerating Sequence-to-Graph Mapping

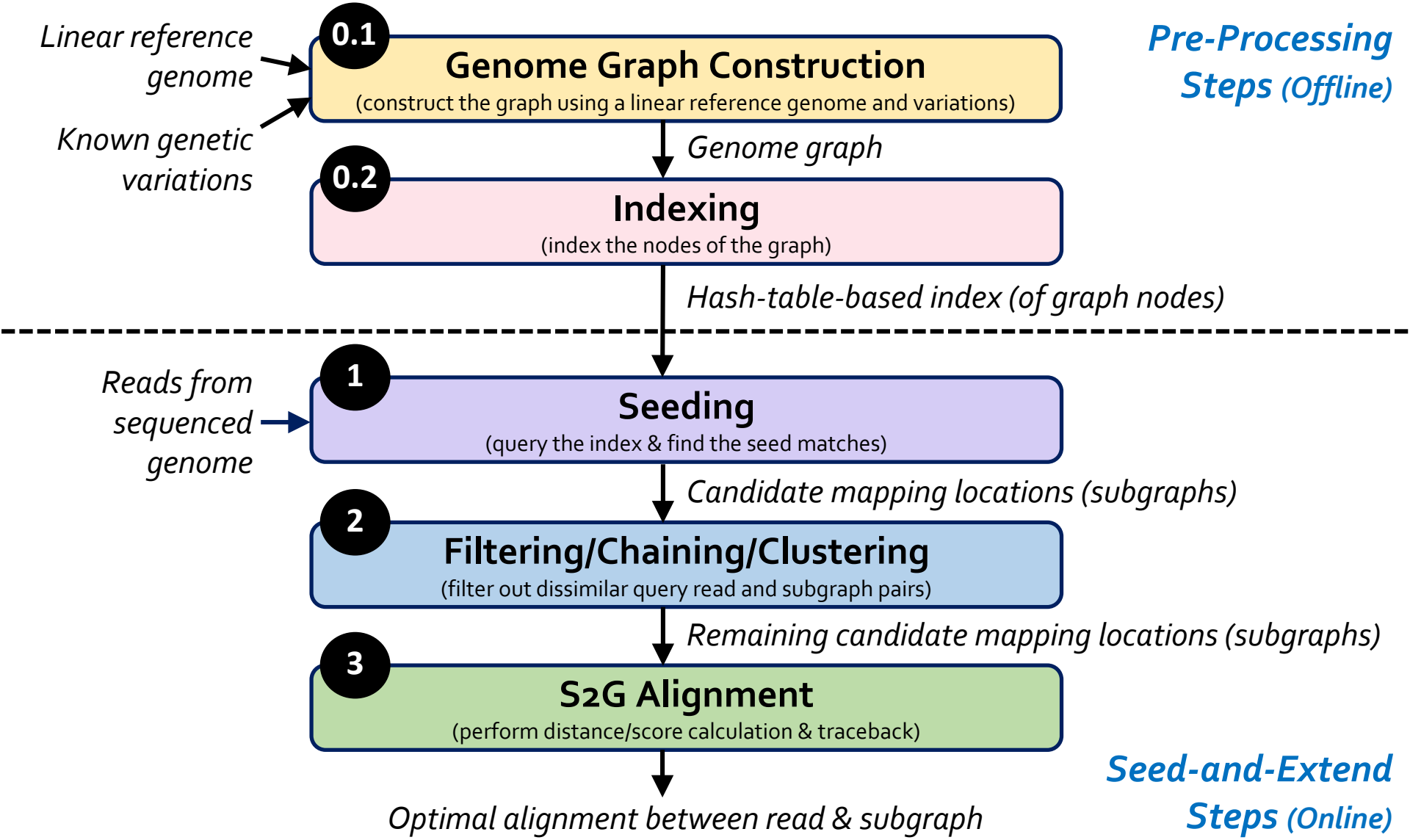
- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika MansouriGhiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur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

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

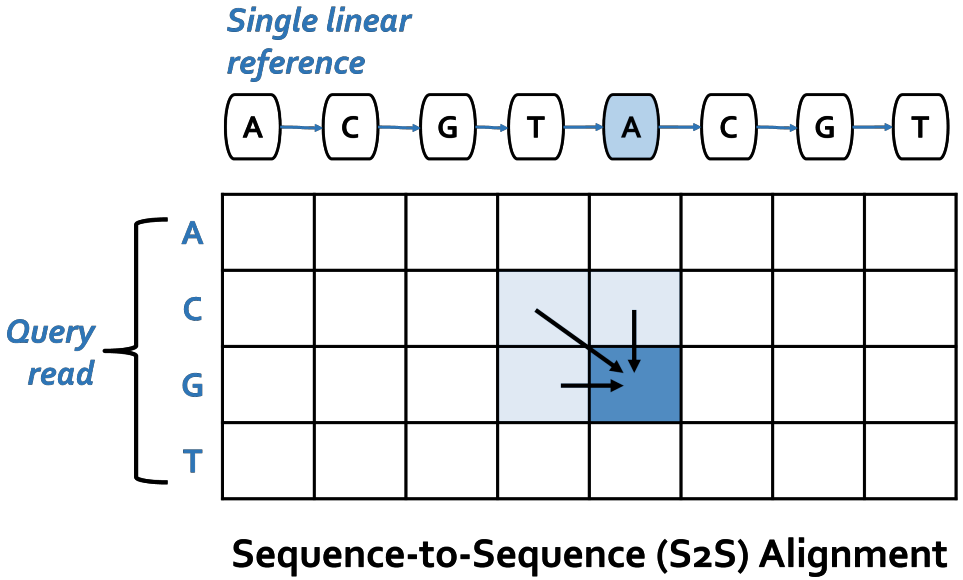
Damla Senol Cali[1] Konstantinos Kanellopoulos[2] Joël Lindegger[2] Zülal Bingöl[3]
Gurpreet S. Kalsi[4] Ziyi Zuo[5] Can Firtina[2] Meryem Banu Cavlak[2] Jeremie Kim[2]
Nika Mansouri Ghiasi[2] Gagandeep Singh[2] Juan Gómez-Luna[2] Nour Almadhoun Alserr[2]
Mohammed Alser[2] Sreenivas Subramoney[4] Can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

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

Sequence-to-Graph Mapping Pipeline

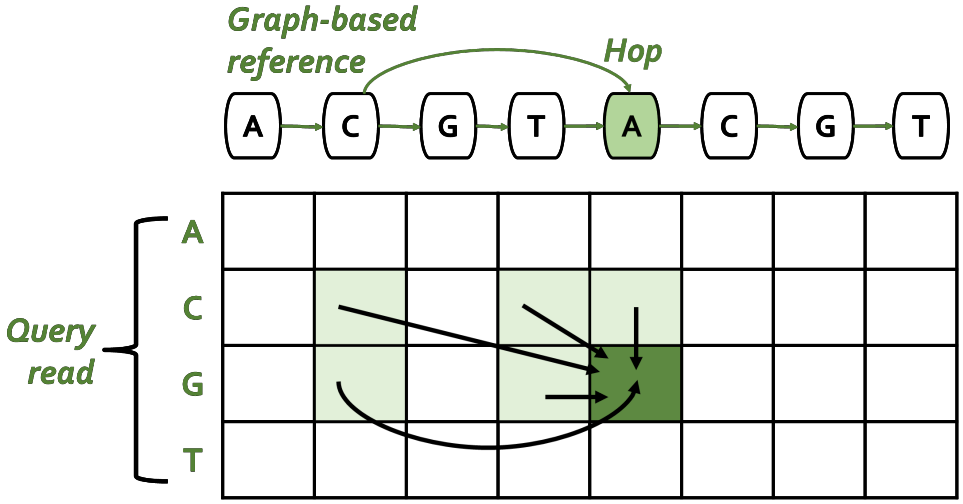


S2S vs. S2G Alignment



Sequence-to-Sequence (S2S) Alignment

S2S vs. S2G Alignment



Sequence-to-Graph (S2G) Alignment

In contrast to S2S alignment,

S2G alignment must incorporate **non-neighboring characters**

as well whenever there is an edge (i.e., ***hop***)

from the non-neighboring character to the current character

Analysis of State-of-the-Art Tools

SW

Based on our analysis with **GraphAligner** and **vg:**

Observation 1: Alignment step is the bottleneck

Observation 2: Alignment suffers from high cache miss rates

Observation 3: Seeding suffers from the DRAM latency bottleneck

Observation 4: Baseline tools scale sublinearly

HW

Observation 5: Existing S2S mapping accelerators are unsuitable for the S2G mapping problem

Observation 6: Existing graph accelerators are unable to handle S2G alignment

SeGraM: Universal Genomic Mapping Accelerator

- ***First universal genomic mapping accelerator*** that can support *both* sequence-to-graph mapping and sequence-to-sequence mapping, for *both* short and long reads

- ***First algorithm/hardware co-design*** for accelerating sequence-to-graph mapping

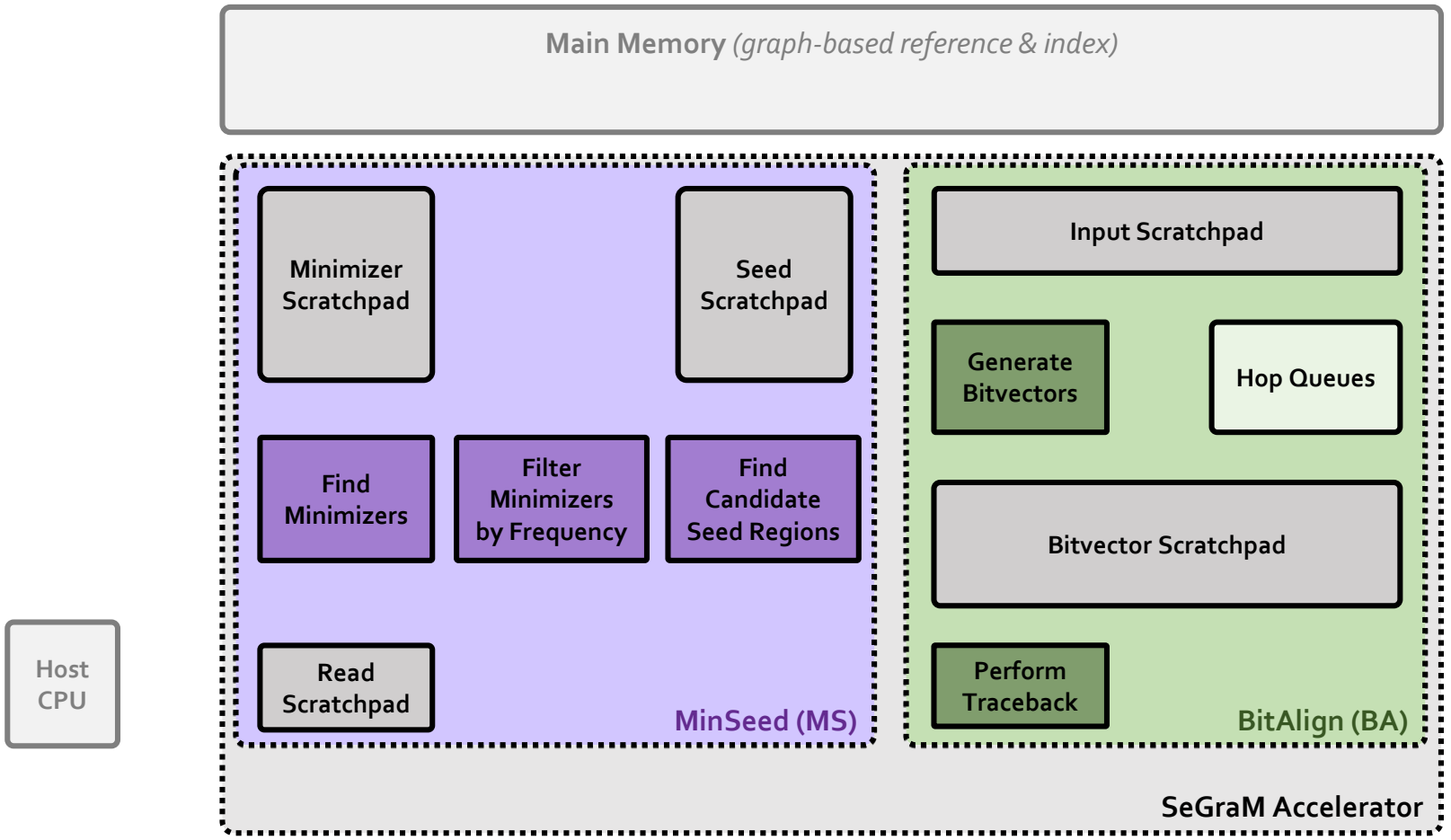
- We base SeGraM upon a minimizer-based seeding algorithm
- We propose a novel bitvector-based alignment algorithm to perform approximate string matching between a read and a graph-based reference genome

SW

- We co-design both algorithms with high-performance, scalable, and efficient hardware accelerators

HW

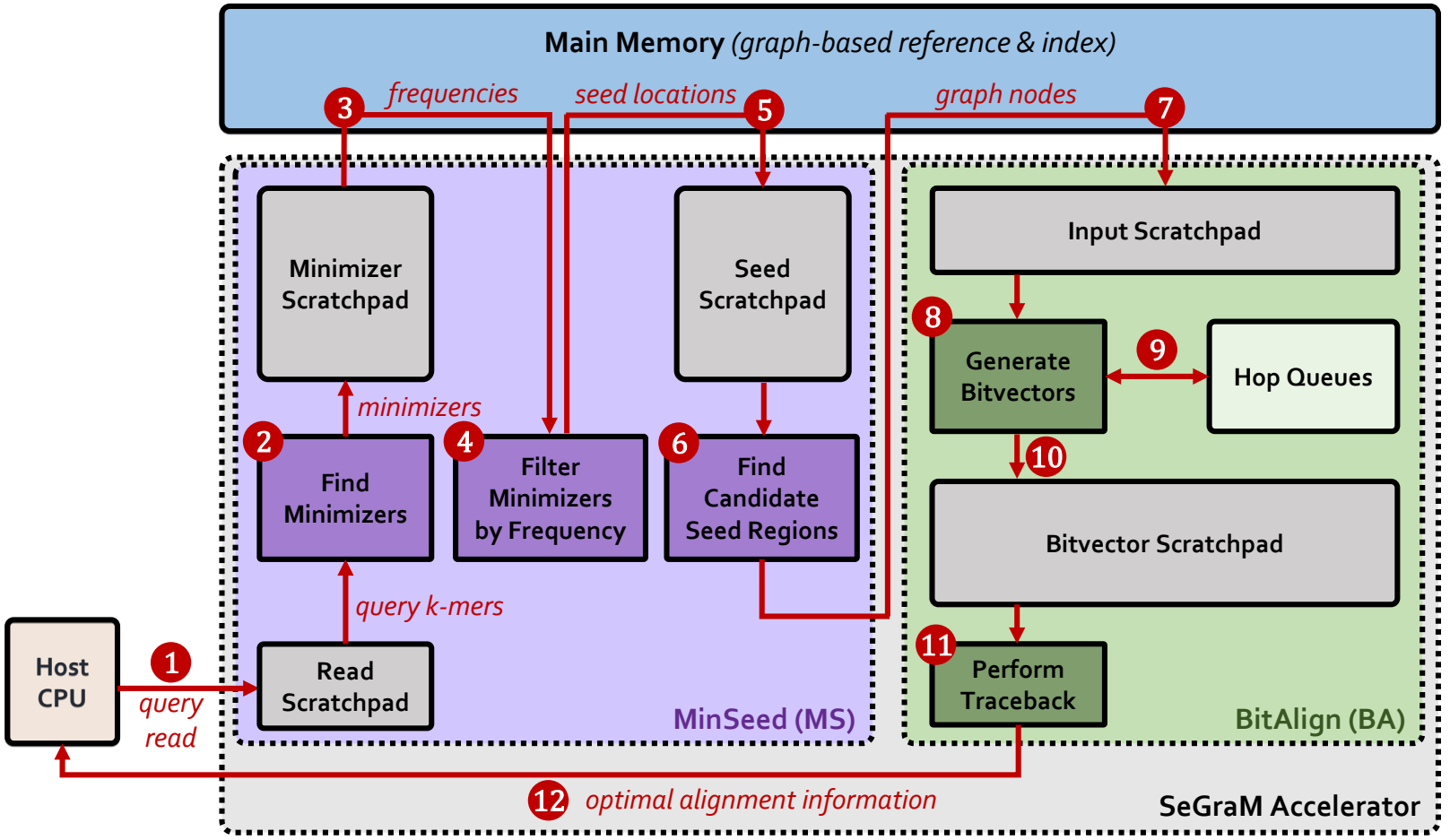
SeGraM Hardware Design



MinSeed: *first* hardware accelerator for **Min**imizer-based **Seed**ing

BitAlign: *first* hardware accelerator for (**Bit**vector-based) sequence-to-graph **Align**ment

SeGraM Hardware Design

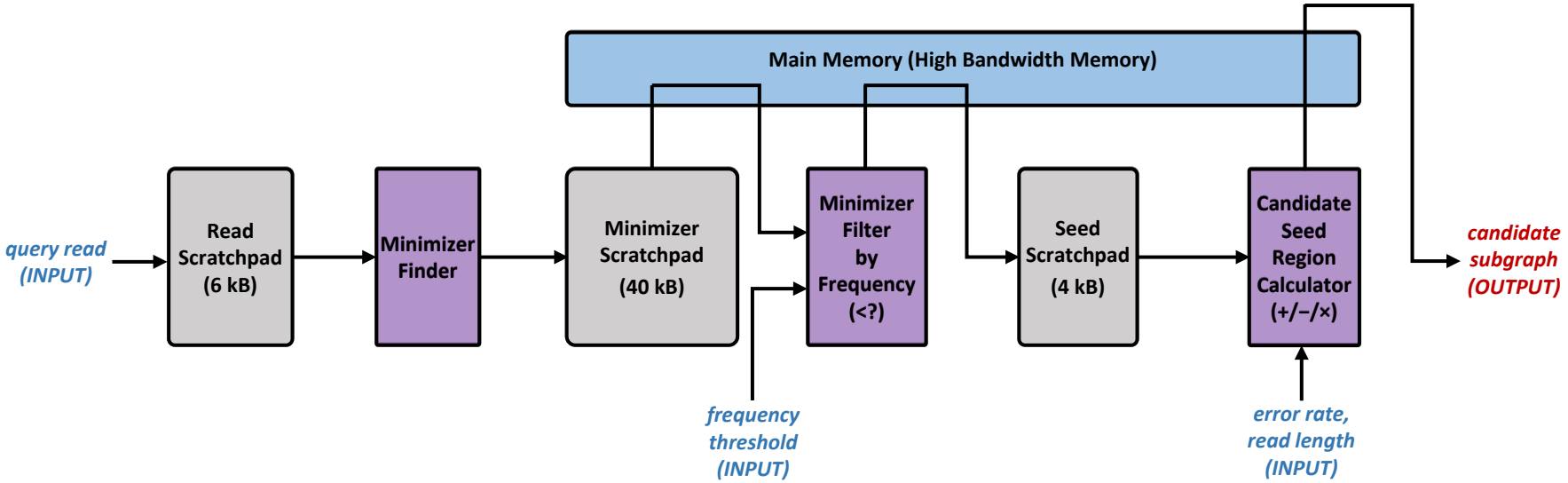


MinSeed: *first* hardware accelerator for **Min**imizer-based **Seed**ing

BitAlign: *first* hardware accelerator for (**Bit**vector-based) sequence-to-graph **Align**ment

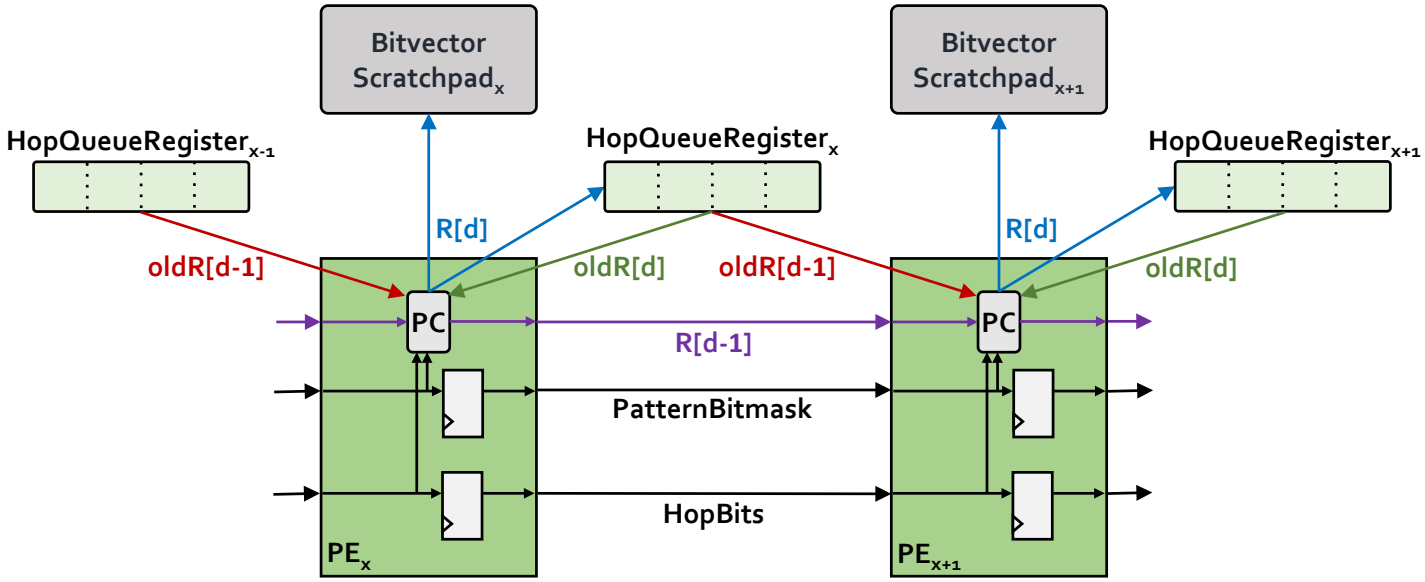
MinSeed HW

- MinSeed = 3 computation modules + 3 scratchpads + memory interface
  - Computation modules: Implemented with simple logic
  - Scratchpads: 50kB in total; employ double buffering technique to hide the latency of MinSeed
  - High-Bandwidth Memory (HBM): Enables low-latency and highly-parallel memory access



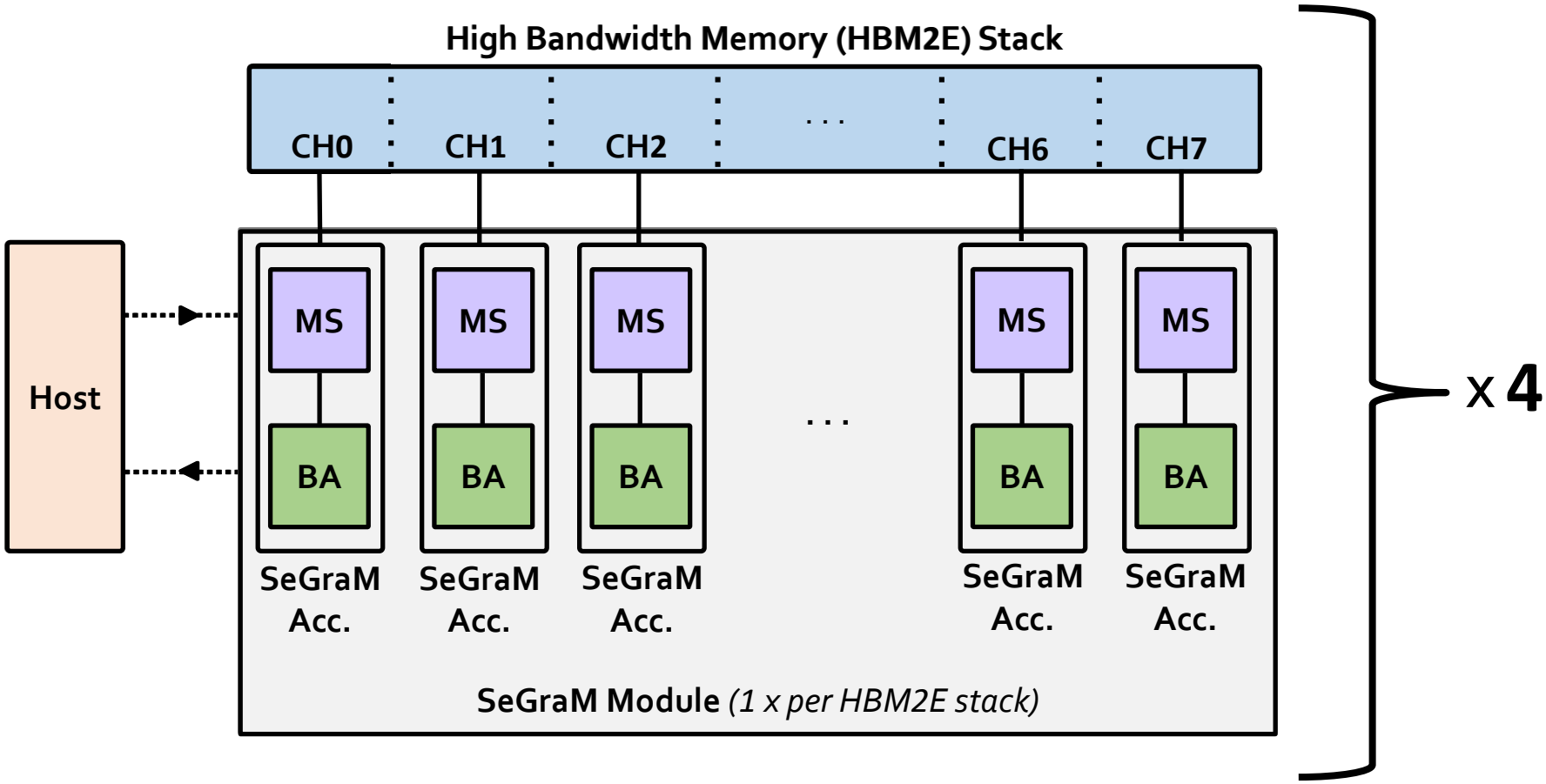
BitAlign HW

- **Linear cyclic systolic array-based accelerator**
- Based on the GenASM hardware design*
- Incorporates ***hop queue registers*** to feed the bitvectors of non-neighboring characters/nodes (i.e., *hops*)



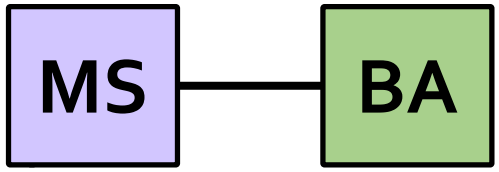
[*] D. Senol Cali *et al.* "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis" (MICRO'20)

Overall System Design of SeGraM

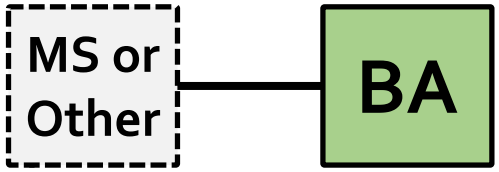


Use Cases of SeGraM

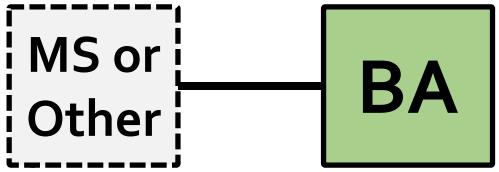
(1) Sequence-to-Graph Mapping



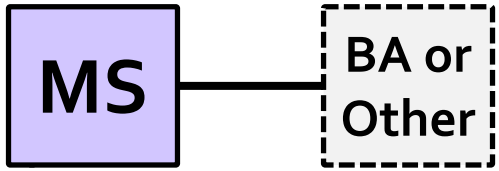
(2) Sequence-to-Graph Alignment



(3) Sequence-to-Sequence Alignment



(4) Seeding

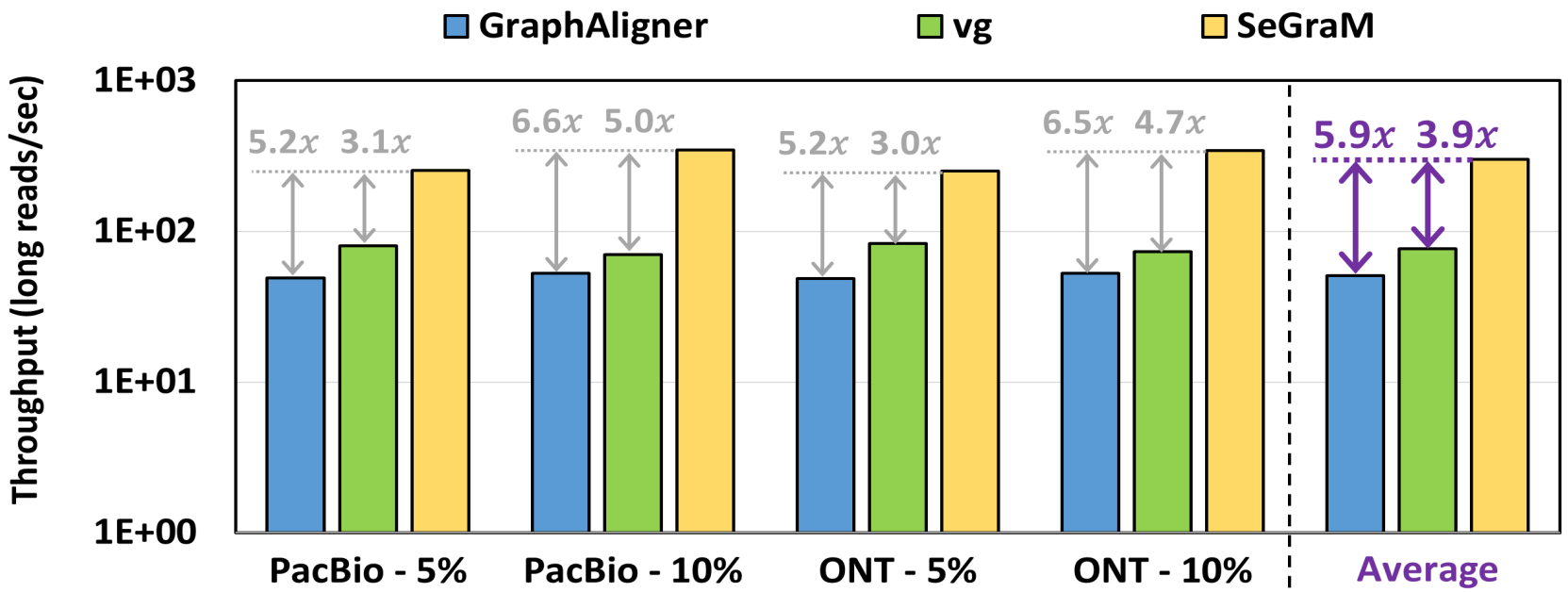


Key Results – Area & Power

- Based on our **synthesis** of **MinSeed** and **BitAlign** accelerator datapaths using the Synopsys Design Compiler with a **28nm** process (**@ 1GHz):**

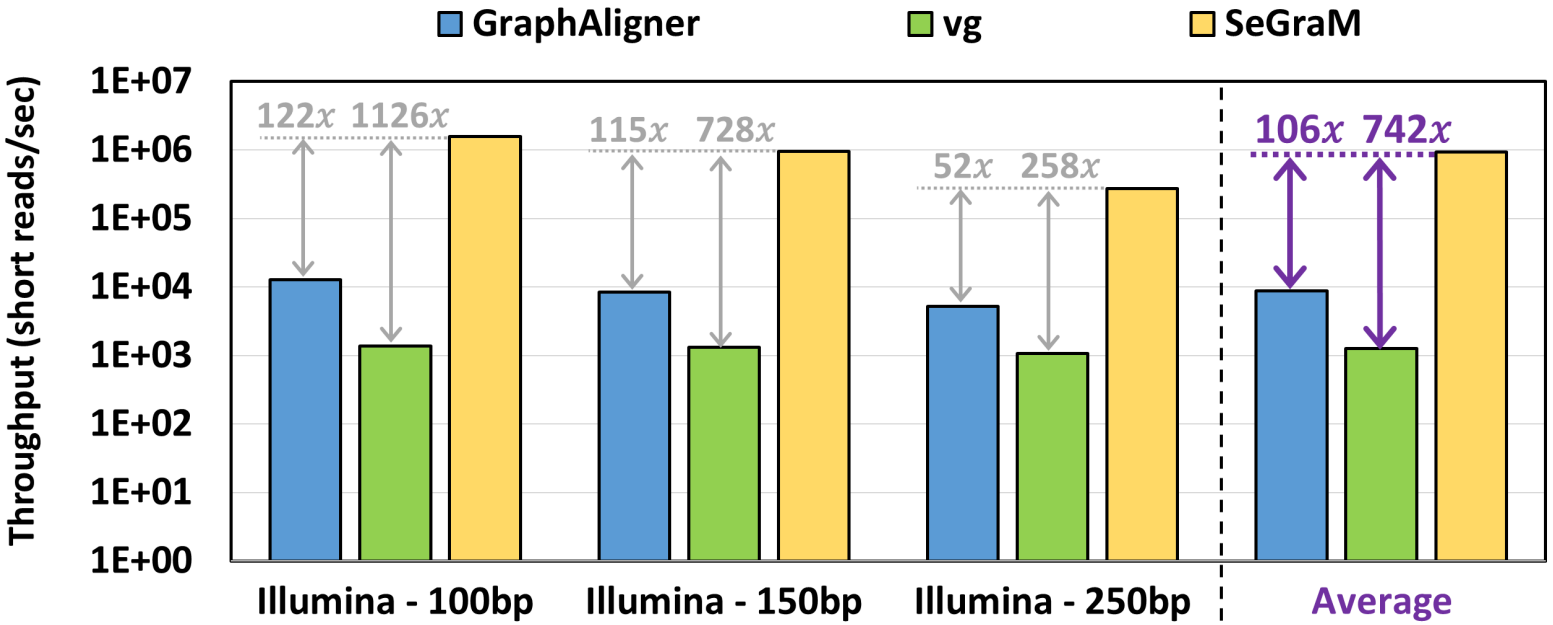
Component	Area (mm^2)	Power (mW)
MinSeed – Logic	0.017	10.8
Read Scratchpad (6 kB)	0.012	7.9
Minimizer Scratchpad (40 kB)	0.055	22.7
Seed Scratchpad (4 kB)	0.008	6.4
BitAlign – Edit Distance Calculation Logic with Hop Queue Registers (64 PEs)	0.393	378.0
BitAlign – Traceback Logic	0.020	2.7
Input Scratchpad (24 kB)	0.033	13.3
Bitvector Scratchpads (128 kB)	0.329	316.2
Total – 1 SeGraM Accelerator	**0.867**	**758.0 (0.8 W)**
Total – 4 SeGraM Modules (32 SeGraM Accelerators)	**27.744**	**24.3 W**
HBM2E (4 stacks)	**--**	**3.8 W**

Key Results – SeGraM with Long Reads



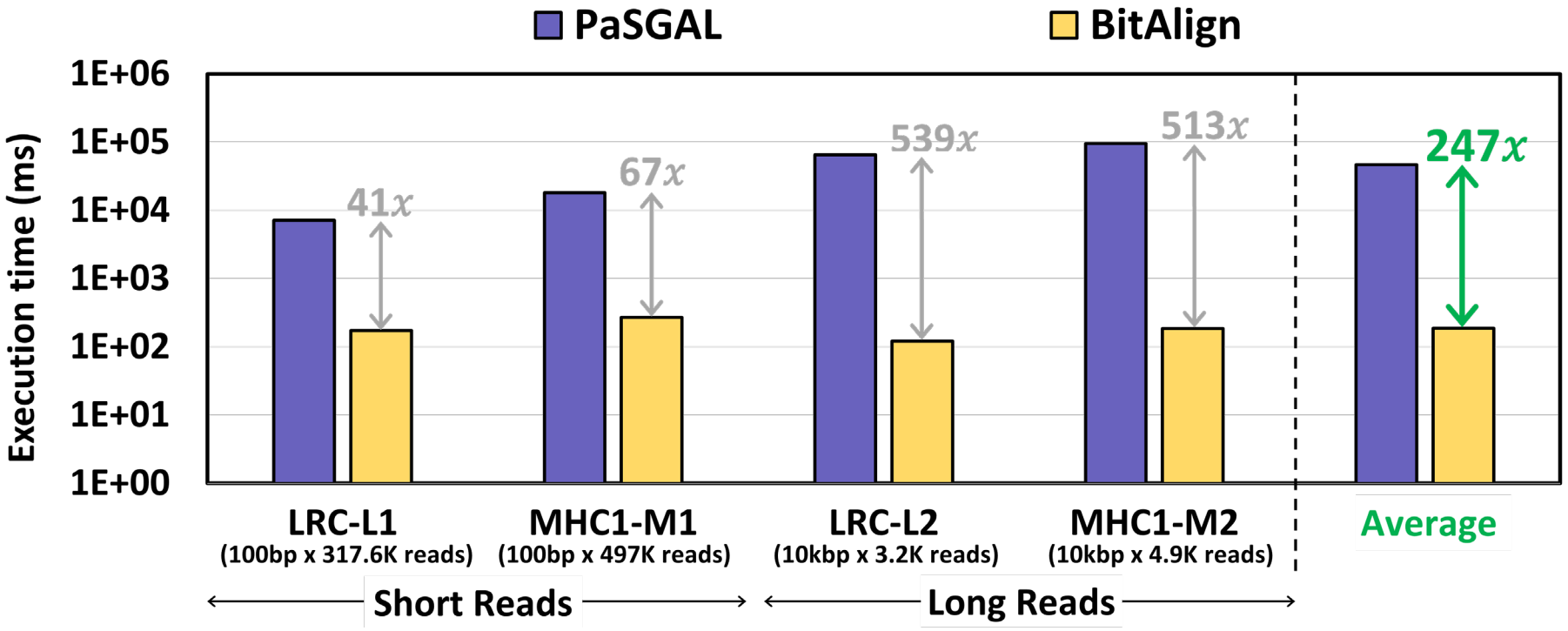
SeGraM provides **5.9× and 3.9× throughput improvement** over GraphAligner and vg, while **reducing the power consumption by 4.1× and 4.4×**

Key Results – SeGraM with Short Reads



SeGraM provides **106× and 742× throughput improvement** over GraphAligner and vg, while **reducing the power consumption by 3.0× and 3.2×**

Key Results – BitAlign (S2G Alignment)



BitAlign provides **41×-539× speedup** over PaSGAL

Conclusion

- **SeGraM:** *First universal algorithm/hardware co-designed genomic mapping accelerator that supports:*
  - Sequence-to-graph (S2G) & sequence-to-sequence (S2S) mapping
  - Short & long reads
  - **MinSeed:** *First minimizer-based seeding accelerator*
  - **BitAlign:** *First (bitvector-based) S2G alignment accelerator*
- SeGraM **supports multiple use cases:**
  - End-to-end S2G mapping
  - S2G alignment
  - S2S alignment
  - Seeding
- SeGraM outperforms state-of-the-art software & hardware solutions

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"
  *Proceedings of the 49th International Symposium on Computer Architecture (**ISCA**)*, New York, June 2022.
  [arXiv version]

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

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

[1]Bionano Genomics [2]ETH Zürich [3]Bilkent University [4]Intel Labs
[5]Carnegie Mellon University [6]University 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

- Problem: Heavily bottlenecked by Data Movement
- GateKeeper, Shouji, SneakySnake performance limited by DRAM bandwidth [Alser+, Bioinformatics 2017,2019,2020]
- Ditto for SHD [Xin+, Bioinformatics 2015]
- Solution: Processing-in-memory can alleviate the bottleneck
- We need to design mapping & filtering algorithms to fit processing-in-memory

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"
IEEE Micro, 2021.
[Source Code]

Previous | Next

Table of Contents

Past Issues

Home / Magazines / IEEE Micro / 2021.04

IEEE Micro

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

July-Aug. 2021, pp. 39-48, vol. 41
DOI Bookmark: 10.1109/MM.2021.3088396

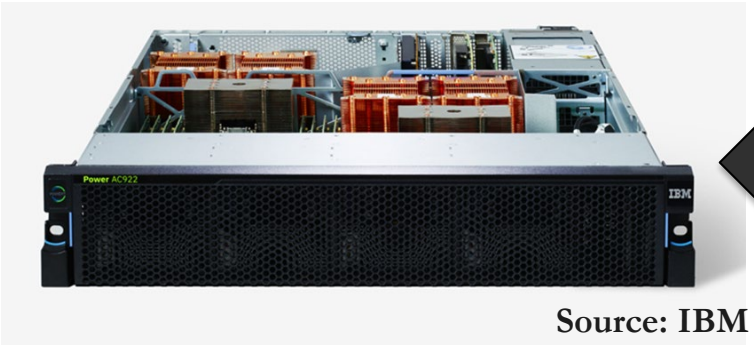
Authors

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

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

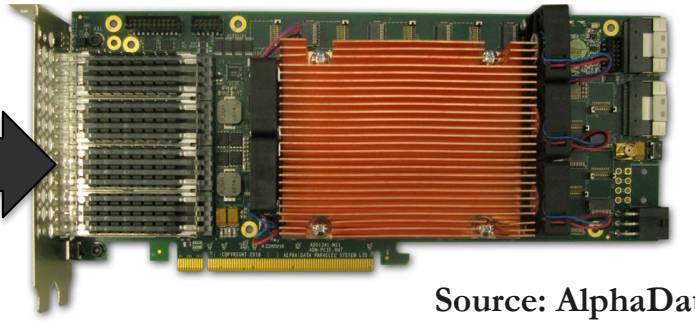
Near-Memory Acceleration using FPGAs



Source: IBM

IBM POWER9 CPU

OCAPI



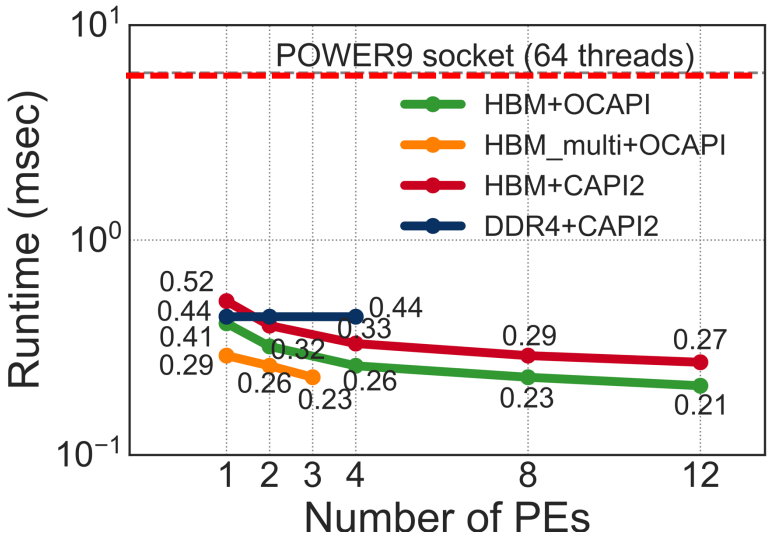
Source: AlphaData

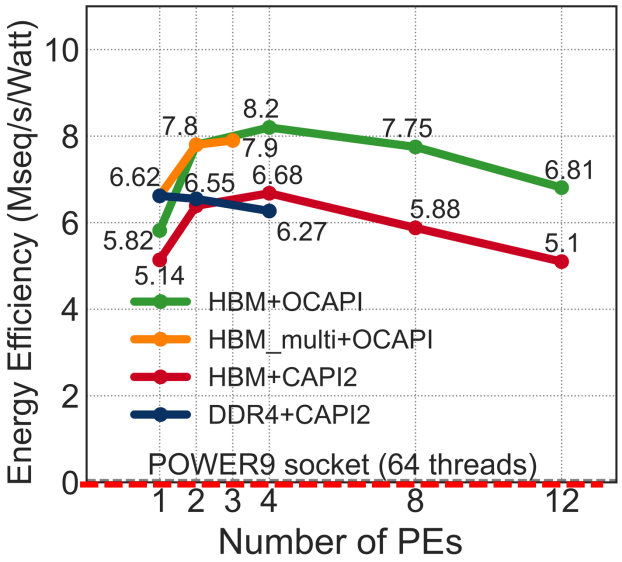
HBM-based FPGA board

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"
IEEE Micro, 2021.
[Source Code]

Previous | Next

Table of Contents

Past Issues

Home / Magazines / IEEE Micro / 2021.04

IEEE Micro

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

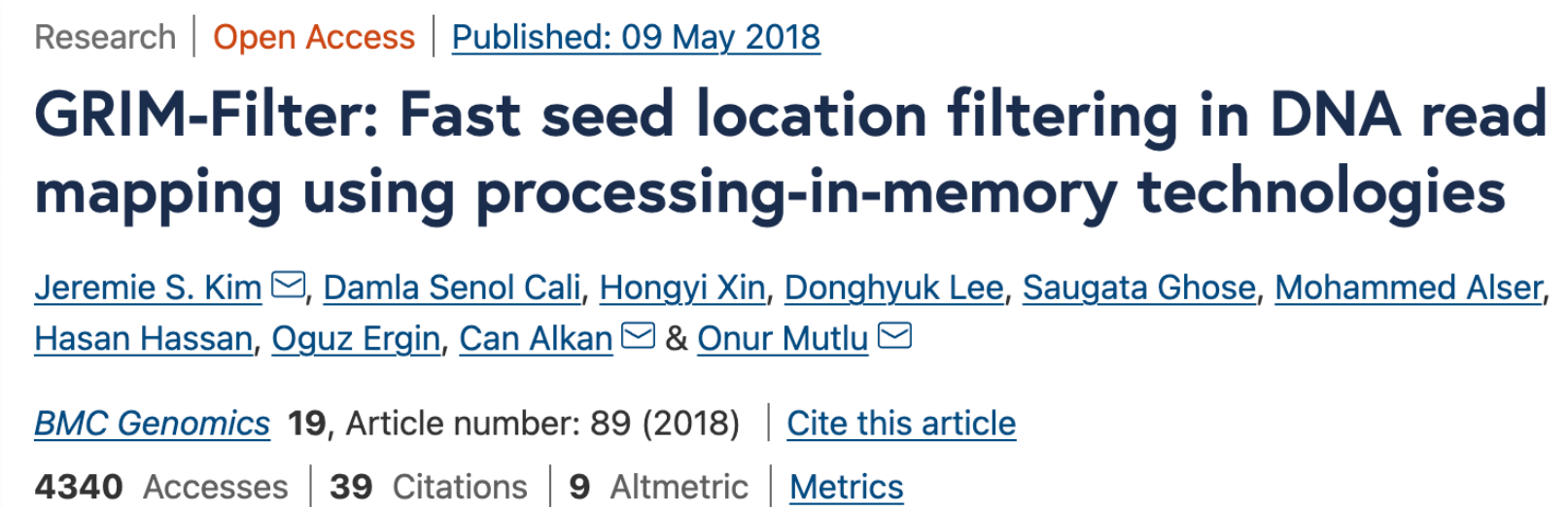
July-Aug. 2021, pp. 39-48, vol. 41
DOI Bookmark: 10.1109/MM.2021.3088396

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

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.
  Proceedings of the 16th Asia Pacific Bioinformatics Conference (APBC), Yokohama, Japan, January 2018.
  [Slides (pptx) (pdf)]
  [Source Code]
  [arxiv.org Version (pdf)]
  [Talk Video at AACBB 2019]



Research | Open Access | Published: 09 May 2018

GRIM-Filter: Fast seed location filtering in DNA read mapping using processing-in-memory technologies

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

BMC Genomics **19**, Article number: 89 (2018) | Cite this article

4340 Accesses | **39** Citations | **9** Altmetric | Metrics

GRIM-Filter

- **Key observation:** FPGA and GPU accelerators are heavily bottlenecked by data movement
- **Key idea:** exploit the high memory bandwidth and the logic layer of 3D-stacked memory to perform highly-parallel filtering in the DRAM chip itself
- **GRIM-Filter, an algorithm-hardware co-designed PIM system for pre-alignment filtering**
- **Key results:**
  - GRIM-Filter is 1.8x-3.7x (2.1x on average) faster than the FastHASH filter (BMC Genomics'13) across real data sets
  - GRIM-Filter has 5.6x-6.4x (6.0x on average) lower false accept rate than the FastHASH filter (BMC Genomics'13) across real data sets

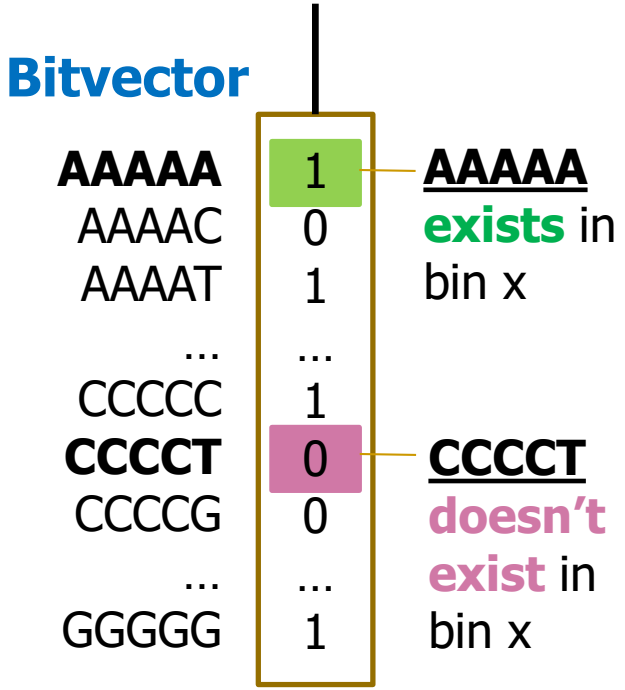
Our Proposal: GRIM-Filter

1. **Data Structures: Bins & Bitvectors**
2. Checking a Bin
3. Integrating GRIM-Filter into a Mapper

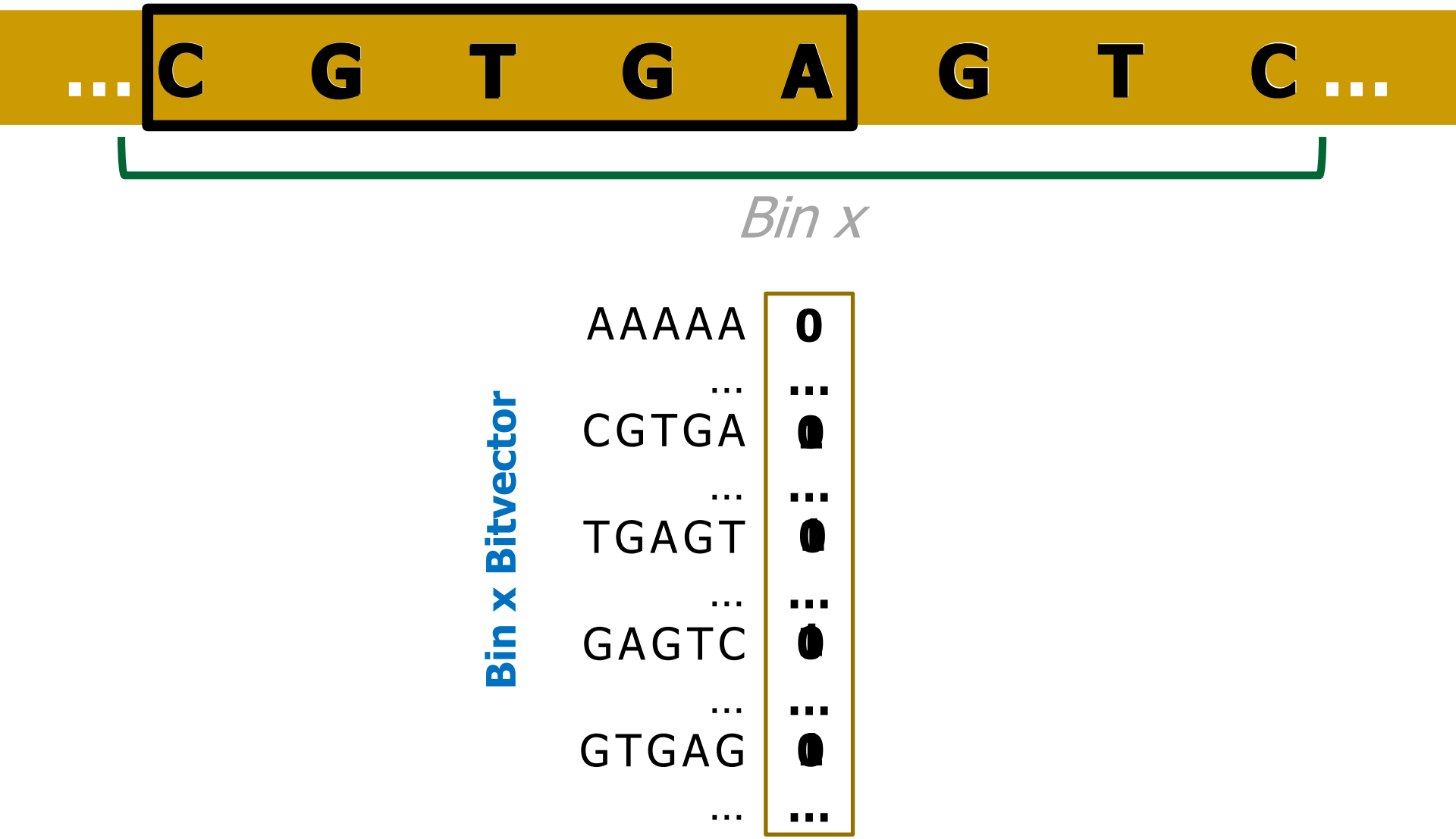
GRIM-Filter: Bins

- We partition the genome into large sequences (**bins**).

- Represent each bin with a **bitvector** that holds the occurrence of all permutations of a small string (**token**) in the bin
- To account for matches that straddle bins, we employ overlapping bins
  - A read will now always completely fall within a single bin



GRIM-Filter: Bitvectors



GRIM-Filter: Bitvectors

Storing all bitvectors requires $4^n * t$ bits in memory, where t = number of bins.

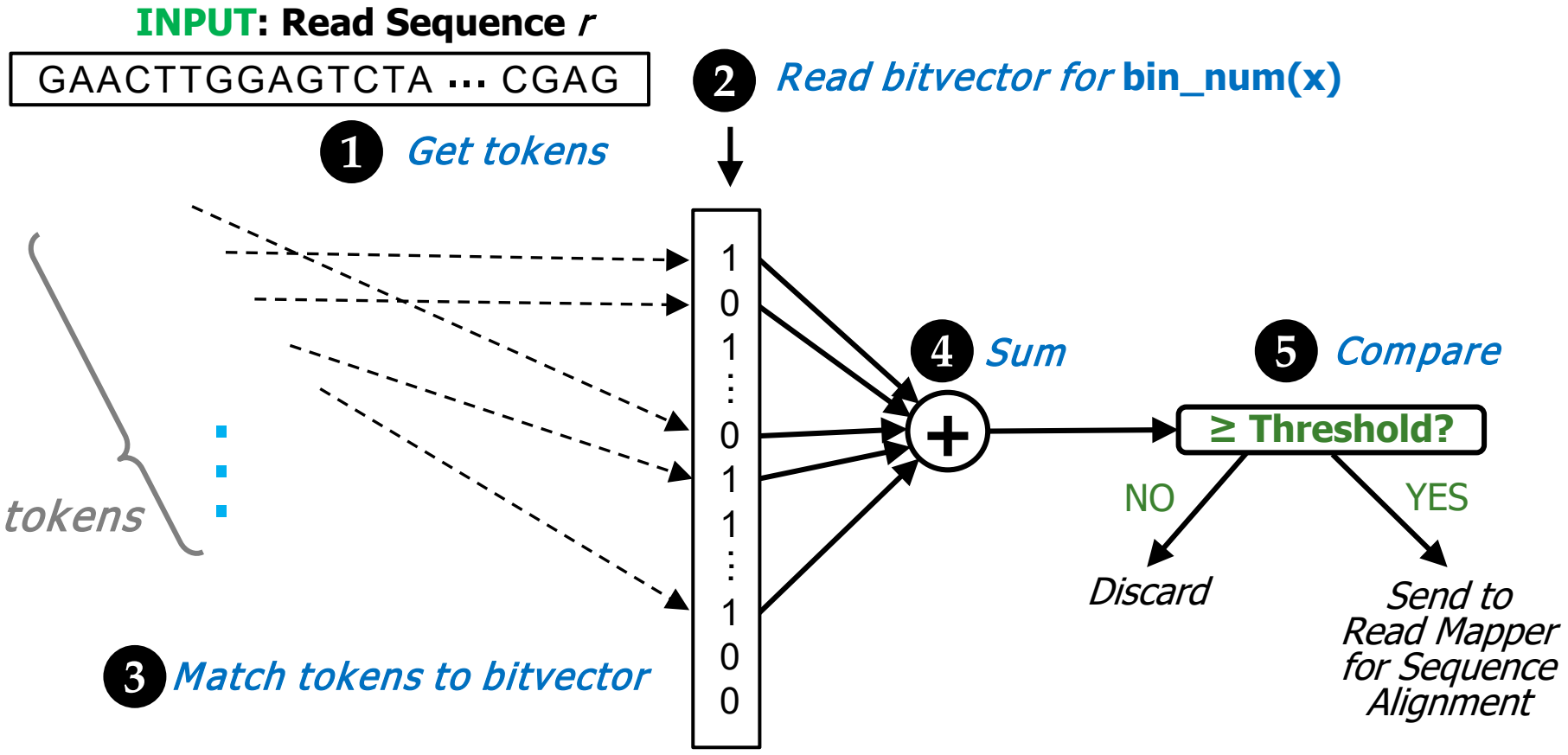
For **bin size** ~200, and **n** = 5, **memory footprint** ~3.8 GB

Our Proposal: GRIM-Filter

1. Data Structures: Bins & Bitvectors
2. **Checking a Bin**
3. Integrating GRIM-Filter into a Mapper

GRIM-Filter: Checking a Bin

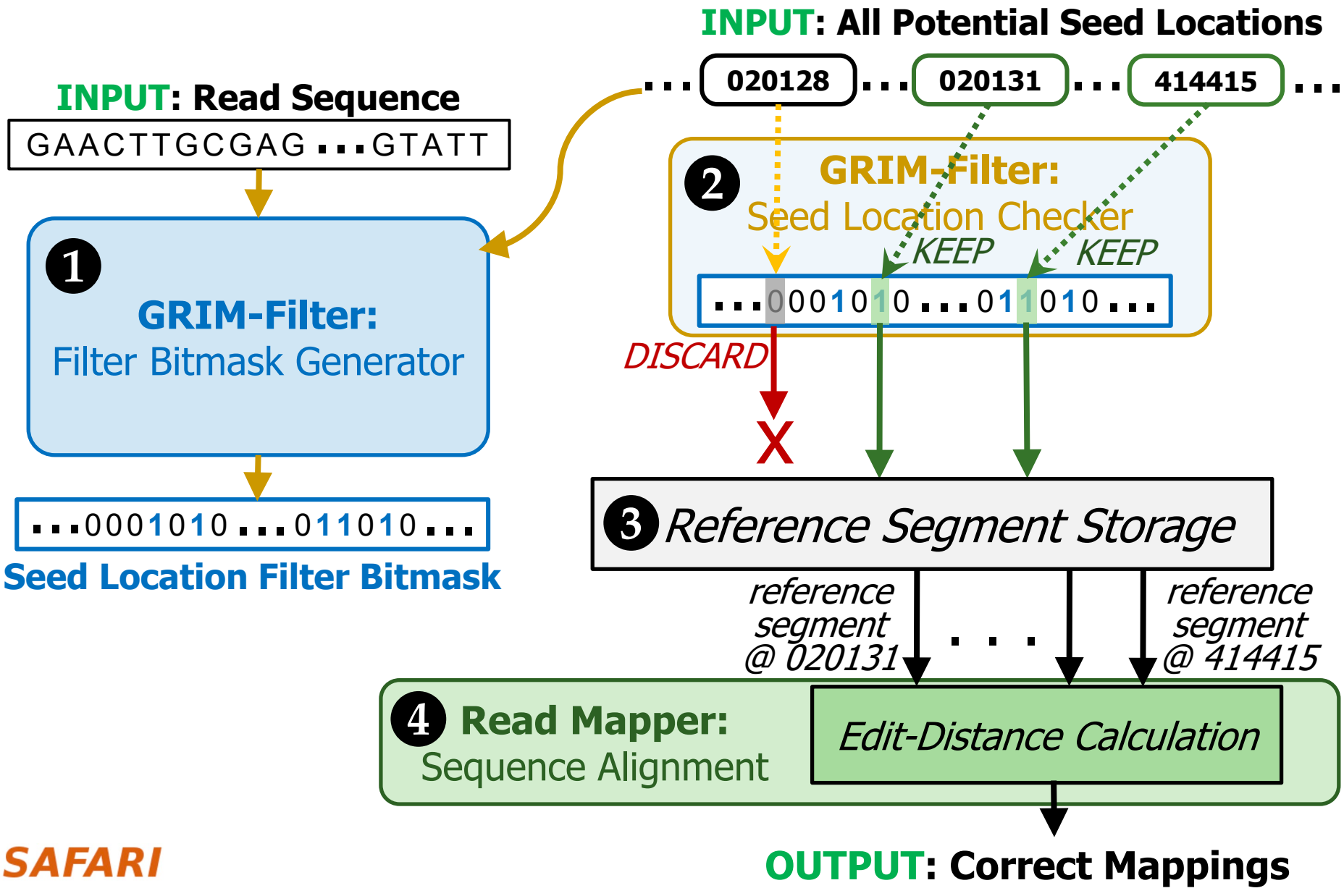
How GRIM-Filter determines whether to **discard** potential match locations in a given bin **prior** to alignment



Our Proposal: GRIM-Filter

1. Data Structures: Bins & Bitvectors
2. Checking a Bin
3. **Integrating GRIM-Filter into a Mapper**

Integrating GRIM-Filter into a Read Mapper



Key Properties of GRIM-Filter

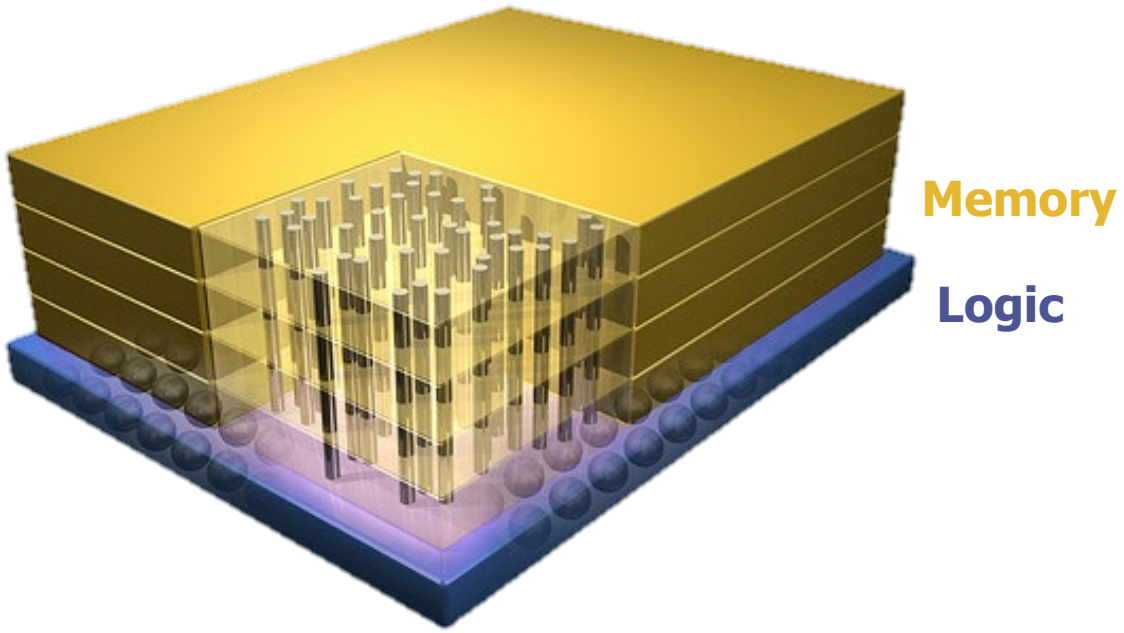
1. **Simple Operations**:
   - To check a given bin, find the **sum** of all bits corresponding to each token in the read
   - **Compare** against threshold to determine whether to align

2. **Highly Parallel**: Each bin is operated on independently and there are many many bins

3. **Memory Bound**: Given the frequent accesses to the large bitvectors, we find that GRIM-Filter is memory bound

These properties together make GRIM-Filter a good algorithm to be run in 3D-Stacked DRAM

Opportunity: 3D-Stacked Logic+Memory



Other "True 3D" technologies under development

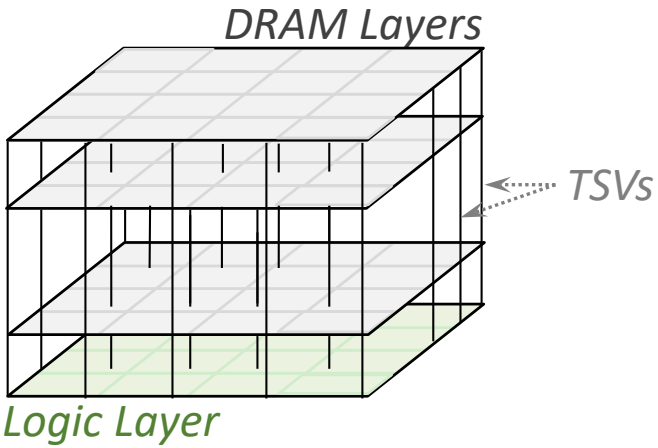
DRAM Landscape (circa 2015)

Segment	*DRAM Standards & Architectures*
Commodity	DDR3 (2007) [14]; DDR4 (2012) [18]
Low-Power	LPDDR3 (2012) [17]; LPDDR4 (2014) [20]
Graphics	GDDR5 (2009) [15]
Performance	eDRAM [28], [32]; RLDRAM3 (2011) [29]
3D-Stacked	WIO (2011) [16]; WIO2 (2014) [21]; MCDRAM (2015) [13]; HBM (2013) [19]; HMC1.0 (2013) [10]; HMC1.1 (2014) [11]
Academic	SBA/SSA (2010) [38]; Staged Reads (2012) [8]; RAIDR (2012) [27]; SALP (2012) [24]; TL-DRAM (2013) [26]; RowClone (2013) [37]; Half-DRAM (2014) [39]; Row-Buffer Decoupling (2014) [33]; SARP (2014) [6]; AL-DRAM (2015) [25]

Table 1. Landscape of DRAM-based memory

Kim+, "Ramulator: A Flexible and Extensible DRAM Simulator", IEEE CAL 2015.

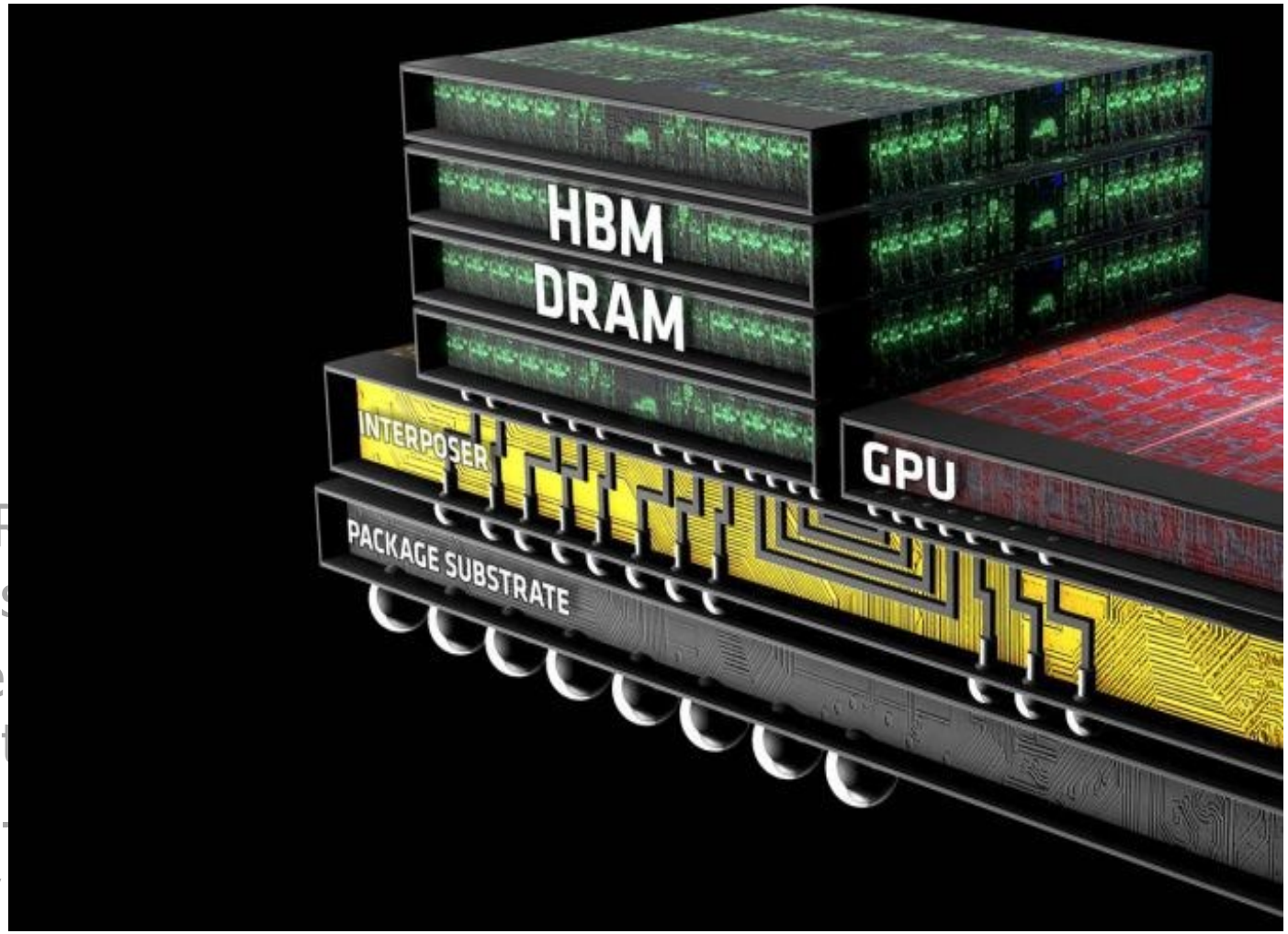
3D-Stacked Memory



- 3D-Stacked DRAM architecture has **extremely high bandwidth** as well as a stacked customizable logic layer
  - Logic Layer enables **Processing-in-Memory**, via high-bandwidth low-latency access to DRAM layers
  - Embed GRIM-Filter operations into **DRAM logic layer** and appropriately distribute bitvectors throughout memory

3D-Stacked Memory

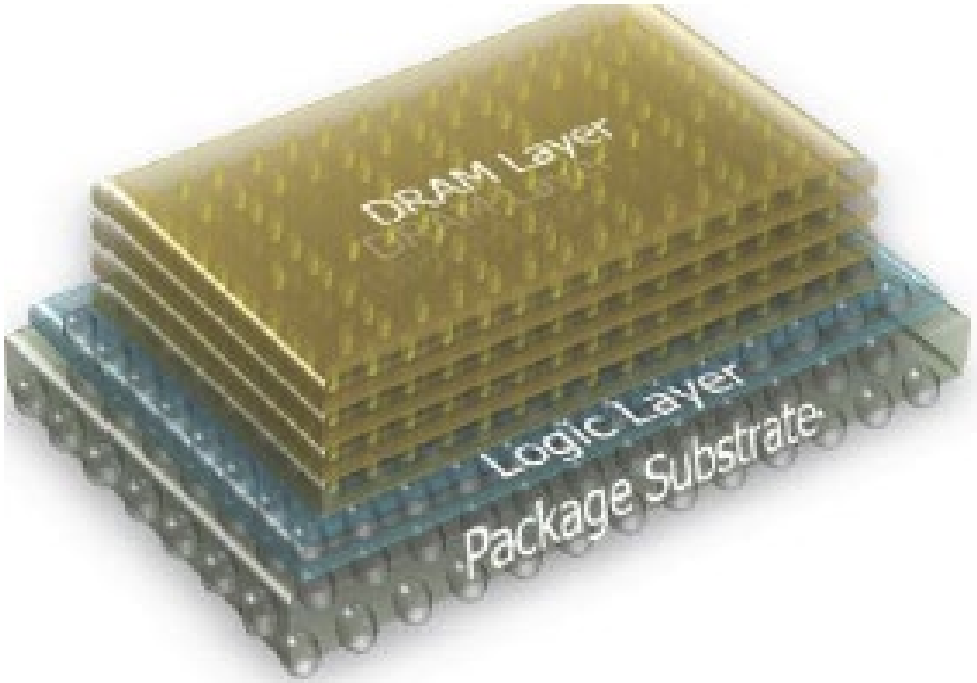
- 3D-Stacked DR
  bandwidth as
  - Logic Layer e
    computation t
  - Embed GRIM-
    appropriately



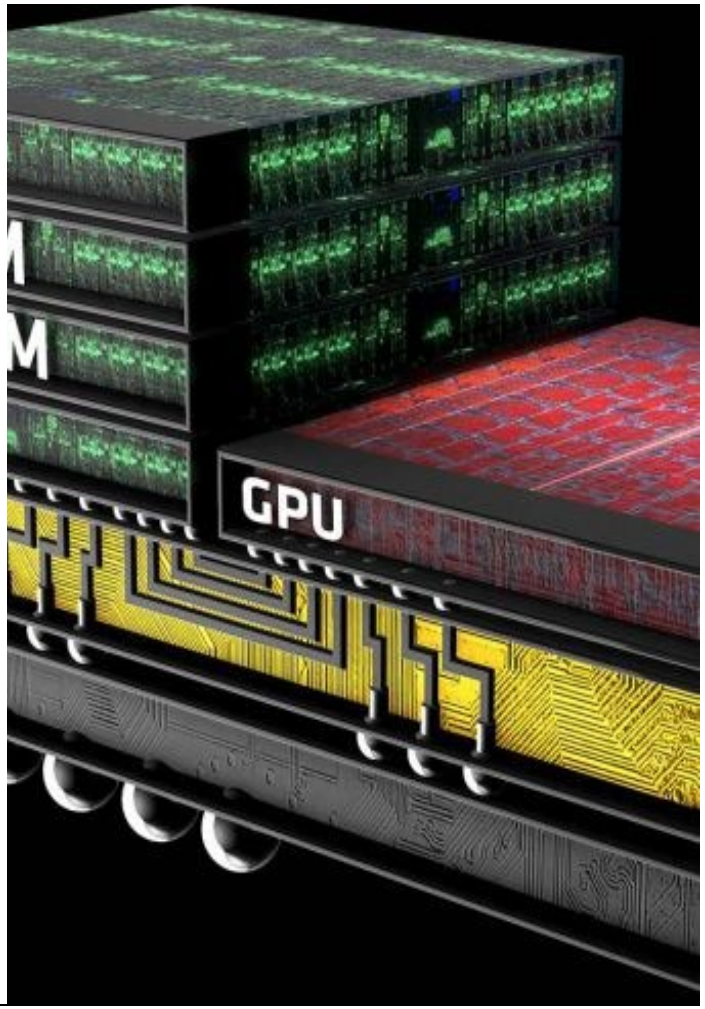
http://i1-news.softpedia-static.com/images/news2/Micron-and-Samsung-Join-Force-to-Create-Next-Gen-Hybrid-Memory-2.png

3D-Stacked Memory

Micron's HMC



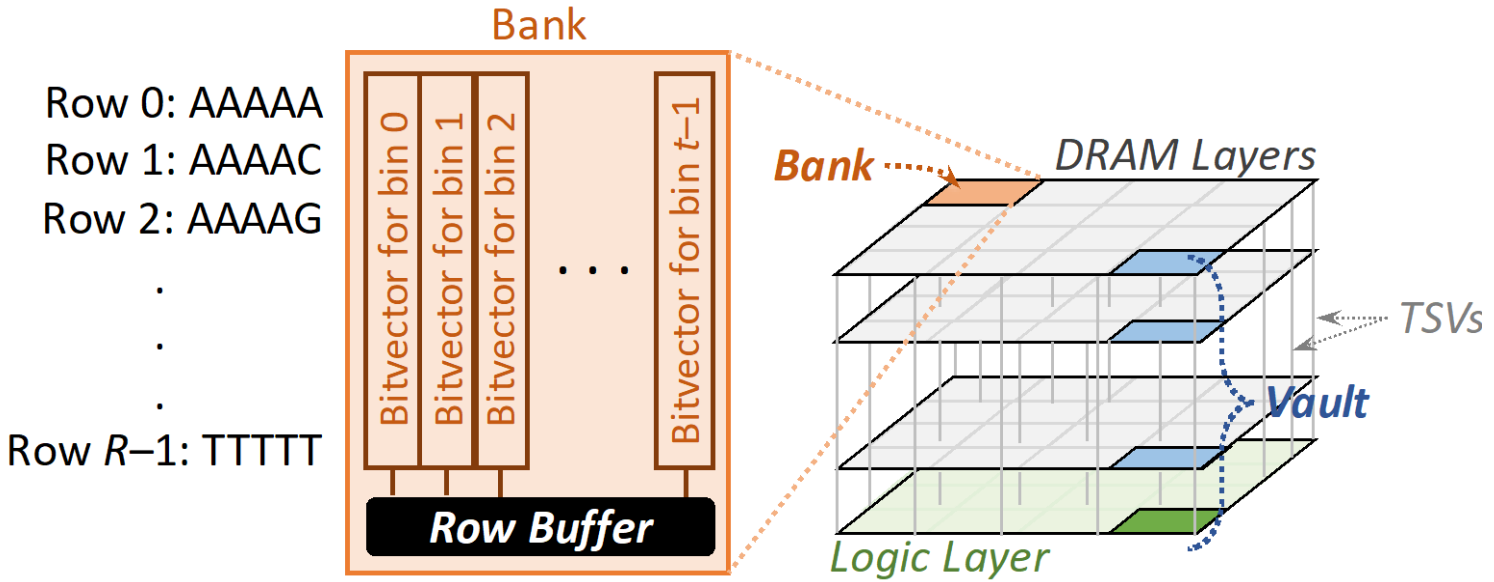
Micron has working demonstration components



http://images.anandtech.com/doci/9266/HBMCar_678x452.jpg

http://i1-news.softpedia-static.com/images/news2/Micron-and-Samsung-Join-Force-to-Create-Next-Gen-Hybrid-Memory-2.png

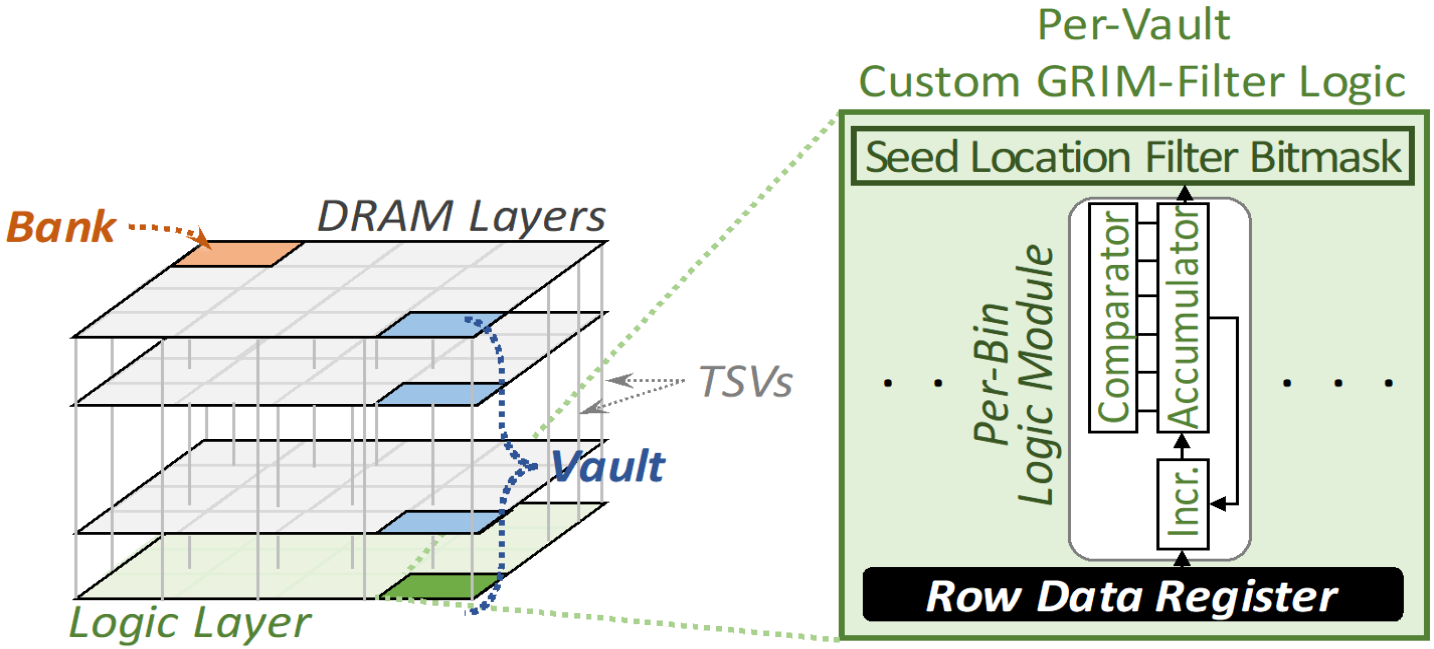
GRIM-Filter in 3D-Stacked DRAM



- Each DRAM layer is organized as an array of **banks**
  - A **bank** is an array of cells with a row buffer to transfer data

- The layout of bitvectors in a bank enables filtering many bins in parallel

GRIM-Filter in 3D-Stacked DRAM



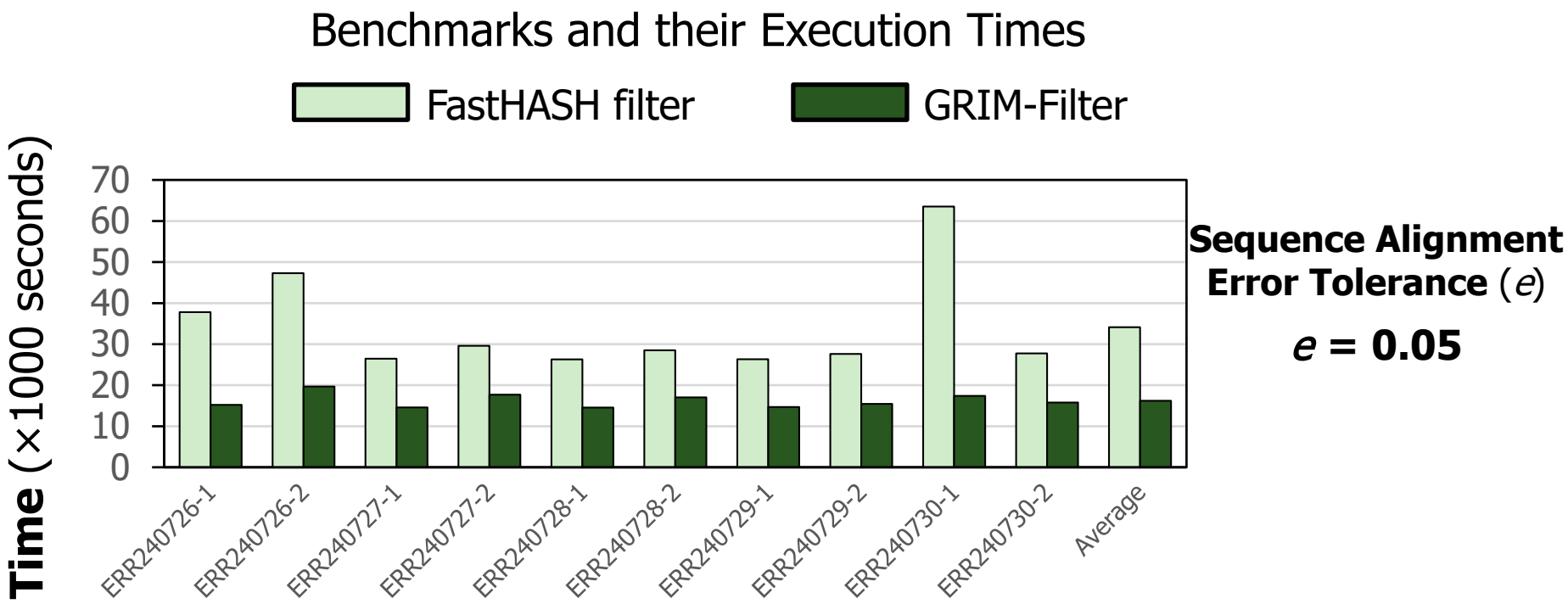
- Customized logic for accumulation and comparison per genome segment
  - Low area overhead, simple implementation
  - For HBM2, we use 4096 incrementer LUTs, 7-bit counters, and comparators in logic layer

Details are in [Kim+, BMC Genomics 2018]

Methodology

- Performance simulated using an in-house 3D-Stacked DRAM simulator
- Evaluate 10 real read data sets (From the 1000 Genomes Project)
  - Each data set consists of 4 million reads of length 100
- Evaluate two key metrics
  - Performance
  - False negative rate
    - The fraction of locations that pass the filter but result in a mismatch
- Compare against a state-of-the-art filter, FastHASH **[Xin+, BMC Genomics 2013]** when using mrFAST, but **GRIM-Filter can be used with ANY read mapper**

GRIM-Filter Performance

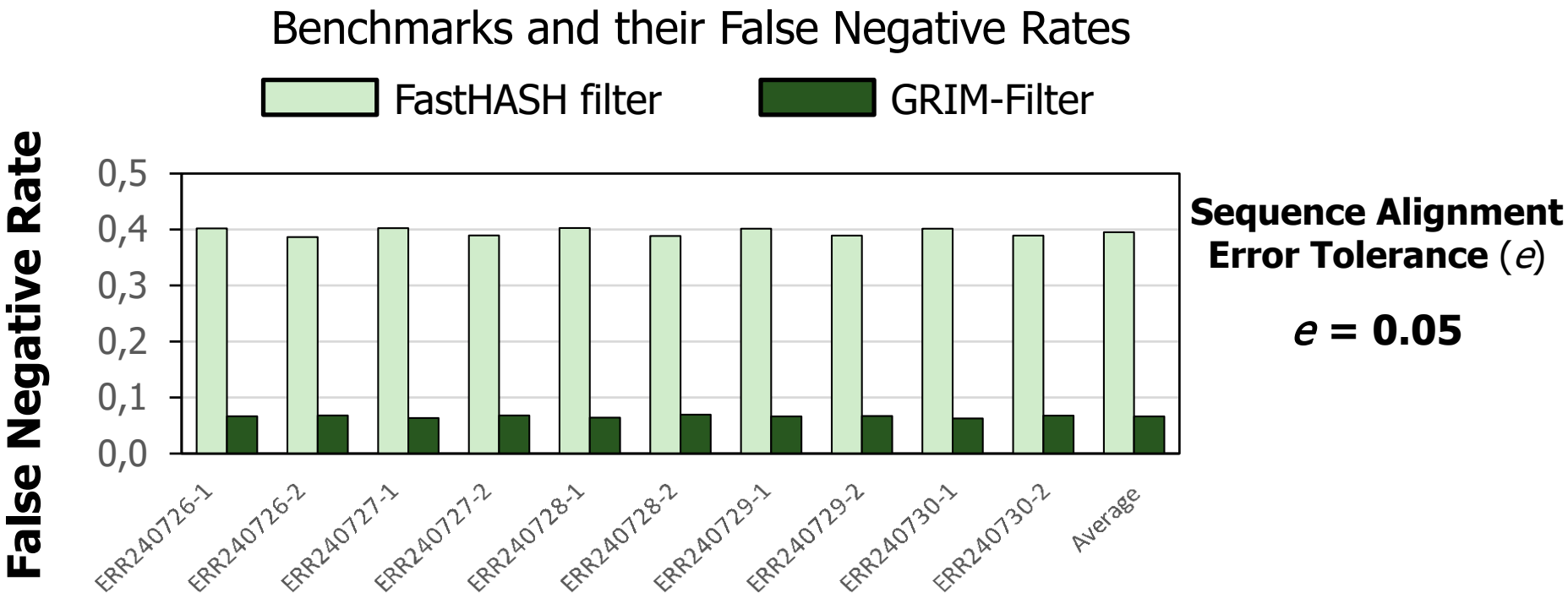


1.8x-3.7x performance benefit across real data sets

2.1x average performance benefit

GRIM-Filter gets performance due to its hardware-software co-design

GRIM-Filter False Negative Rate



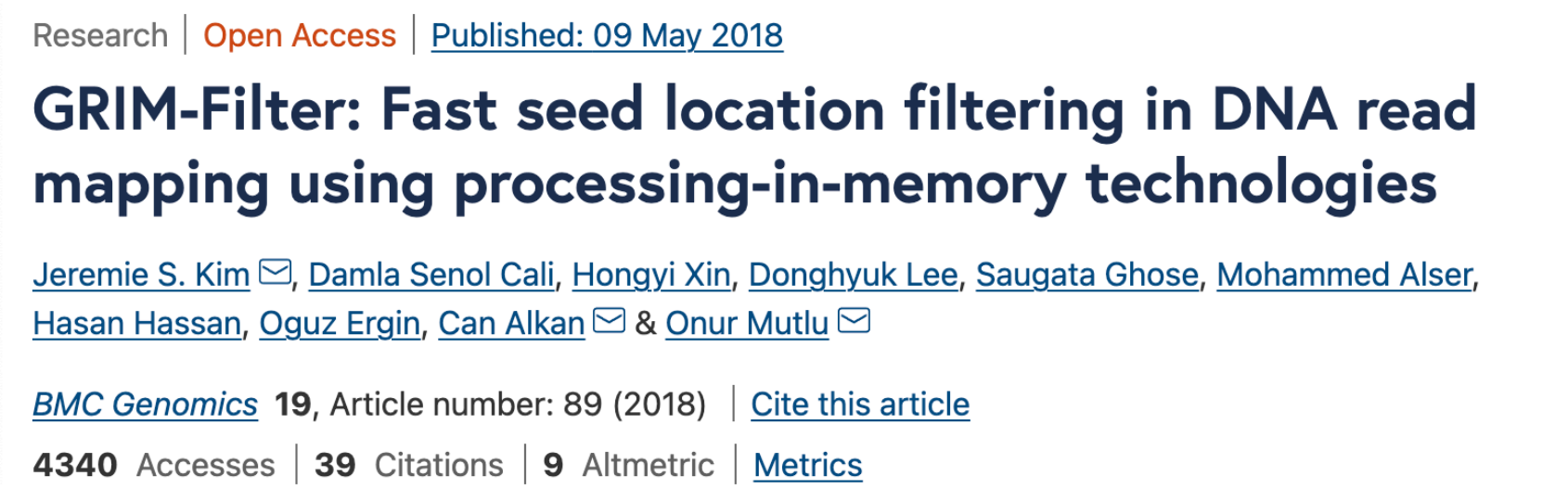
5.6x-6.4x False Negative reduction across real data sets

6.0x average reduction in False Negative Rate

GRIM-Filter utilizes more information available in the read to filter

More on GRIM-Filter

- 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.
  Proceedings of the 16th Asia Pacific Bioinformatics Conference (APBC), Yokohama, Japan, January 2018.
  [Slides (pptx) (pdf)]
  [Source Code]
  [arxiv.org Version (pdf)]
  [Talk Video at AACBB 2019]



Research | Open Access | Published: 09 May 2018

GRIM-Filter: Fast seed location filtering in DNA read mapping using processing-in-memory technologies

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

BMC Genomics **19**, Article number: 89 (2018) | Cite this article

4340 Accesses | **39** Citations | **9** Altmetric | Metrics

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

In-Storage Genome Filtering [ASPLOS 2022]

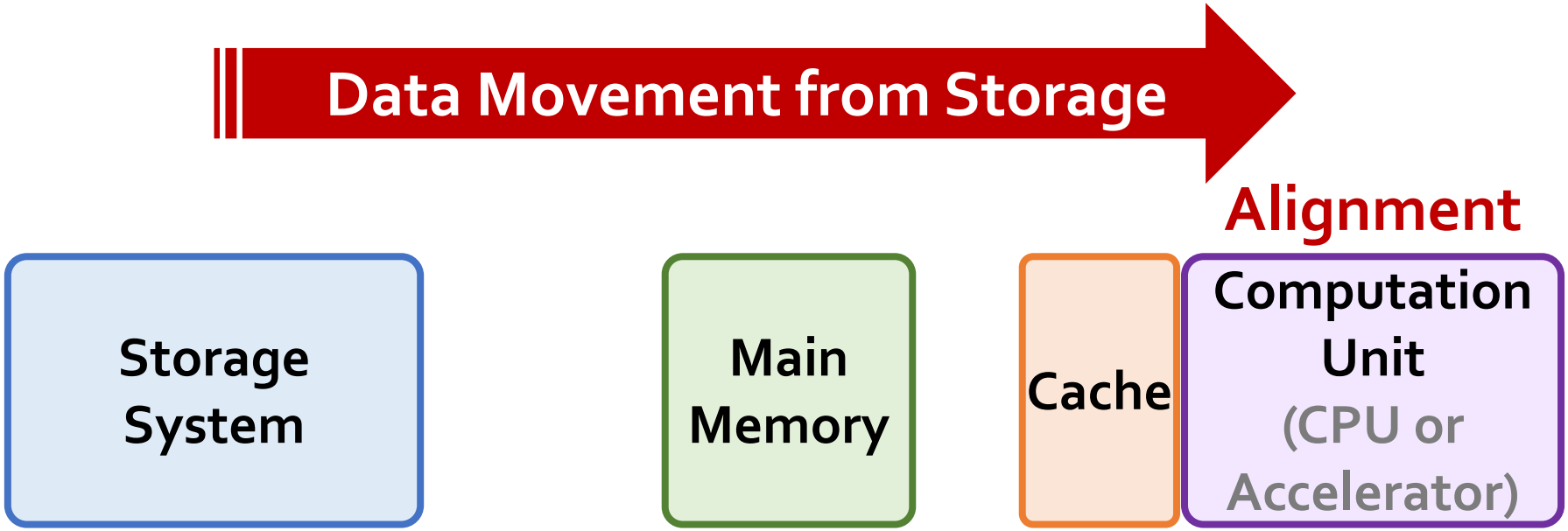
- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,
  "GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"
  Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, February-March 2022.
  [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video (90 seconds)]

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

Nika Mansouri 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

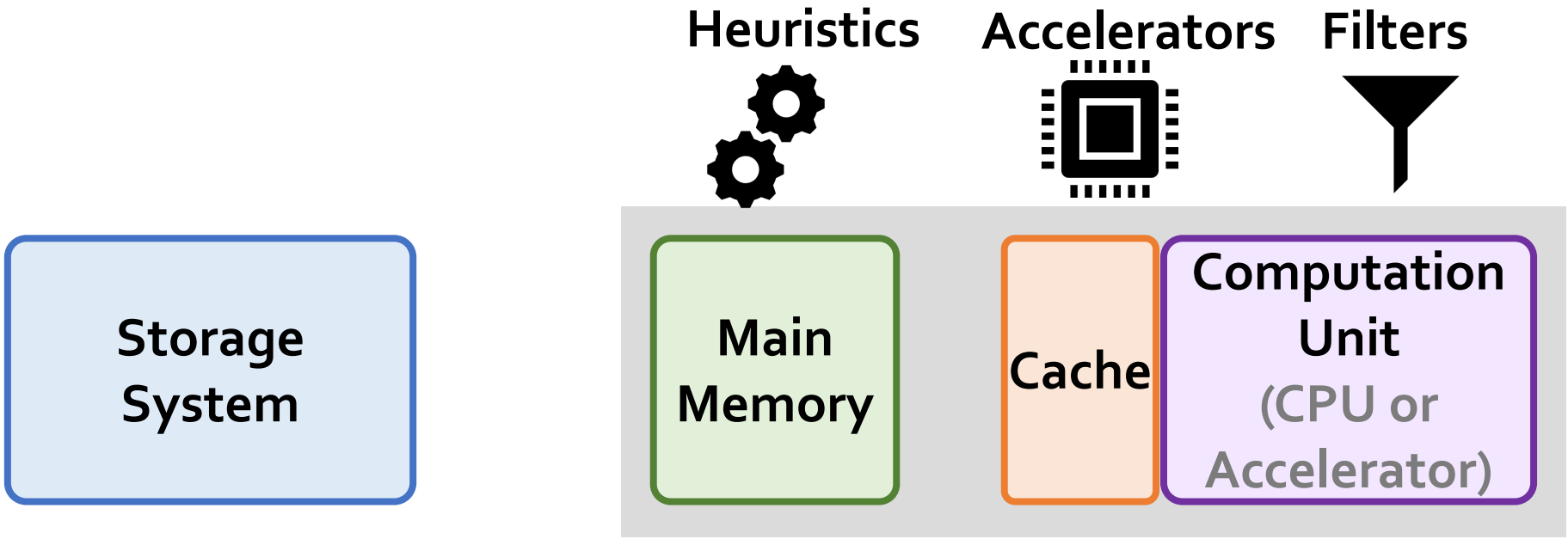
Genome Sequence Analysis



✗ Computation overhead

✗ Data movement overhead

Accelerating Genome Sequence Analysis

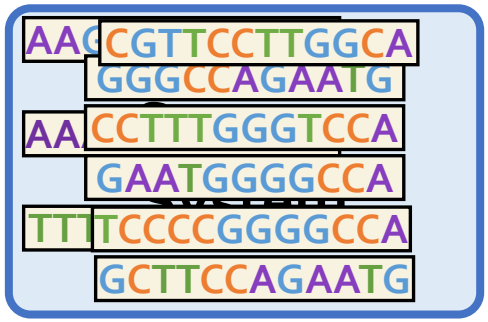


✓ Computation overhead

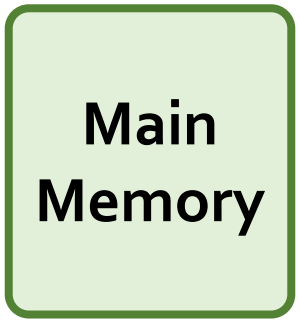
✗ Data movement overhead

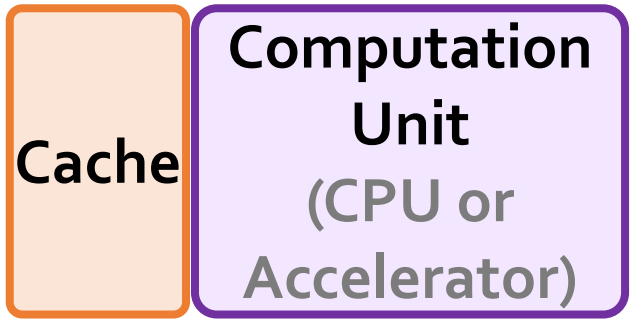
Key Idea

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

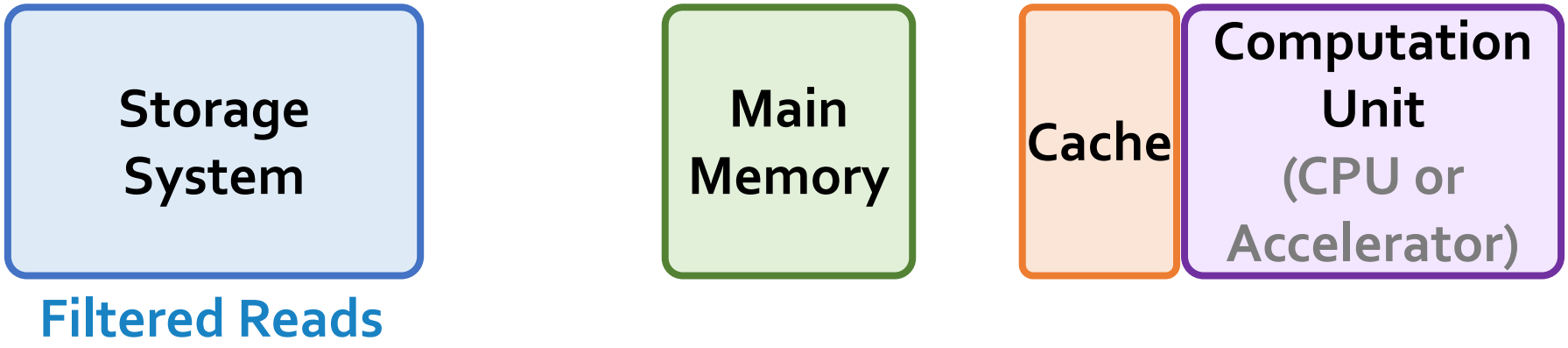
Non-matching reads

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*

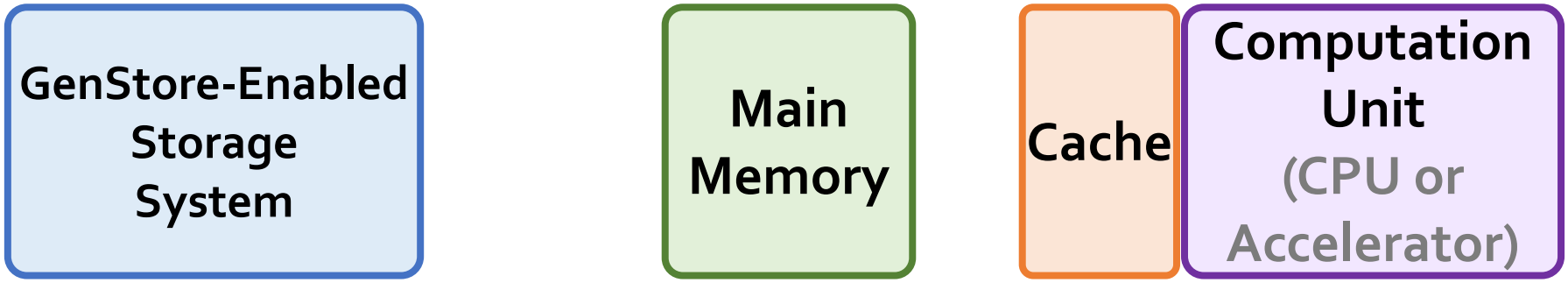


Read mapping workloads can exhibit different behavior

There are limited hardware resources in the storage system

GenStore

Filter reads that do *not* require alignment *inside the storage system*



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"
  *Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**)*, Virtual, February-March 2022.
  [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video (90 seconds)]

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

Nika Mansouri 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

PIM Review and Open Problems

A Modern Primer on Processing in Memory

Onur Mutlu[a,b], Saugata Ghose[b,c], Juan Gómez-Luna[a], Rachata Ausavarungnirun[d]

SAFARI Research Group

[a] *ETH Zürich*
[b] *Carnegie Mellon University*
[c] *University of Illinois at Urbana-Champaign*
[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"
Invited Book Chapter in Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann, Springer, to be published in 2021.

https://arxiv.org/pdf/1903.03988.pdf

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*

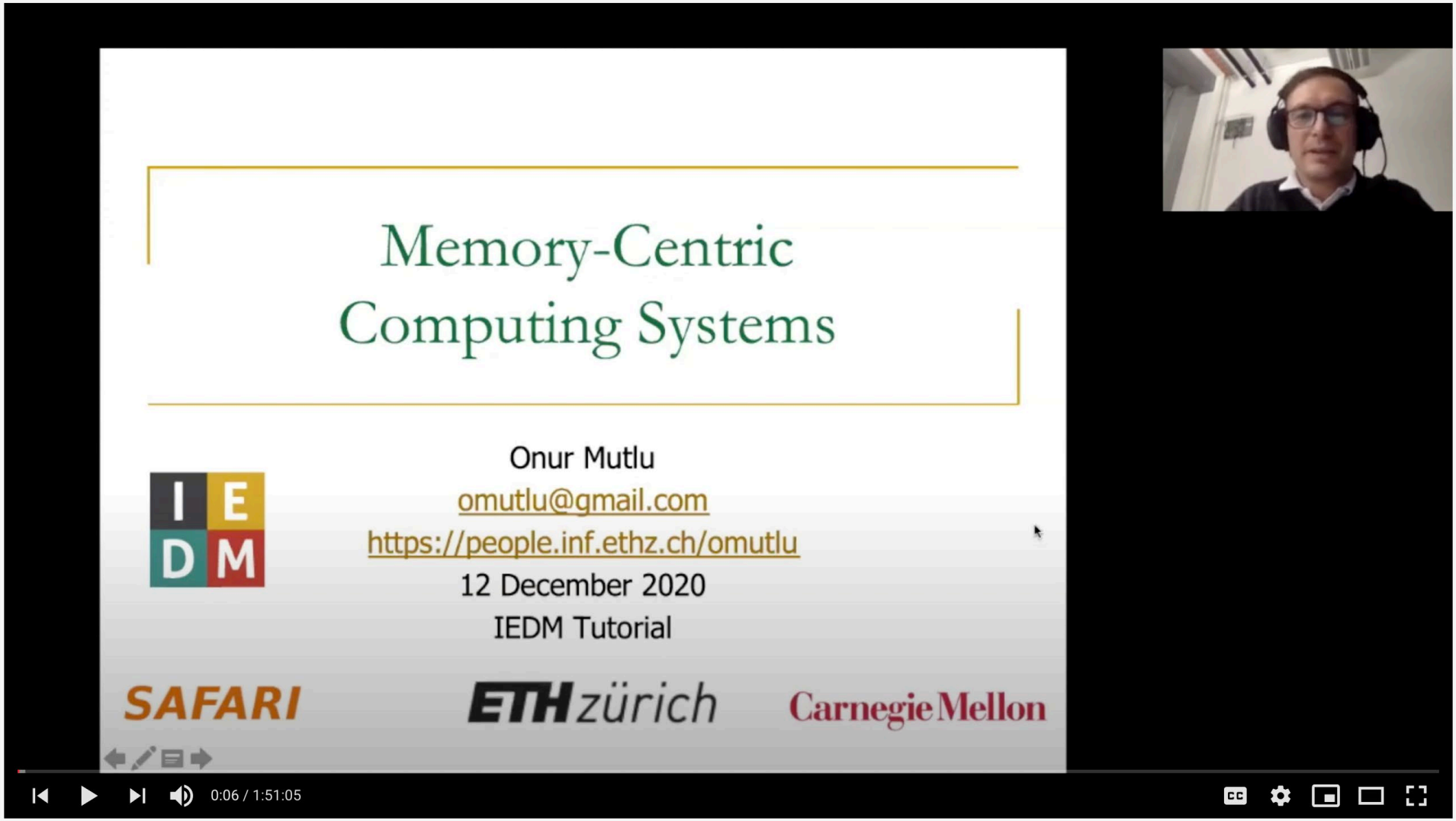
Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu,
"Processing-in-Memory: A Workload-Driven Perspective"
Invited Article in IBM Journal of Research & Development, Special Issue on Hardware for Artificial Intelligence, to appear in November 2019.
[Preliminary arXiv version]

https://arxiv.org/pdf/1907.12947.pdf

More on Processing-in-Memory

- Onur Mutlu,
  "Memory-Centric Computing Systems"
  Invited Tutorial at *66th International Electron Devices Meeting (IEDM)*, Virtual, 12 December 2020.
  [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



IEDM 2020 Tutorial: Memory-Centric Computing Systems, Onur Mutlu, 12 December 2020

1,641 views • Dec 23, 2020

48 0 SHARE SAVE

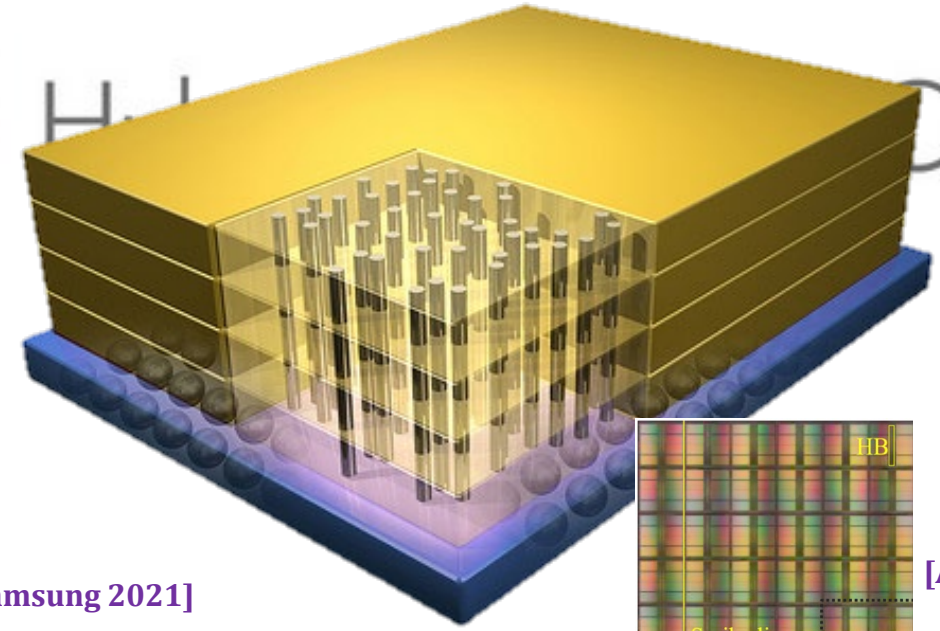
Onur Mutlu Lectures
13.9K subscribers

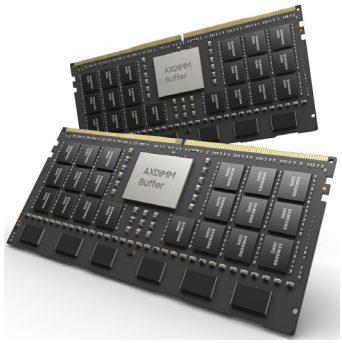
https://www.youtube.com/watch?v=H3sEaINPBOE

ANALYTICS EDIT VIDEO

https://www.youtube.com/onurmutlulectures

Processing-in-Memory Landscape Today

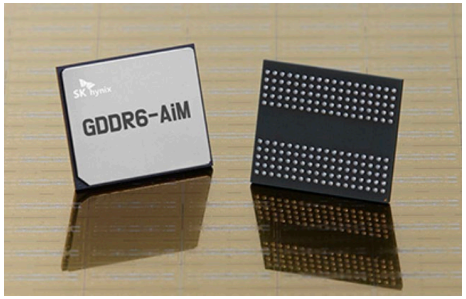




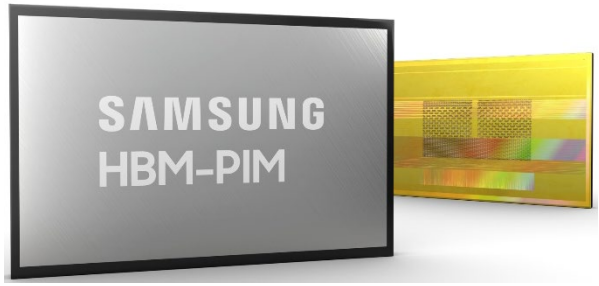
[Samsung 2021]

[Alibaba 2022]

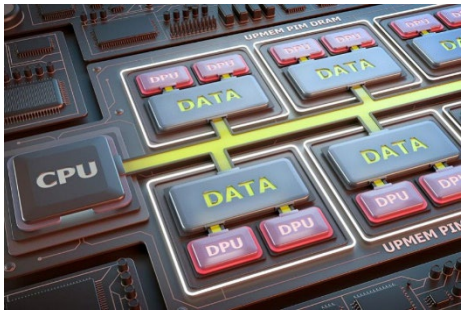
Micron
AUTOMATA PROCESSING



[SK Hynix 2022]



[Samsung 2021]



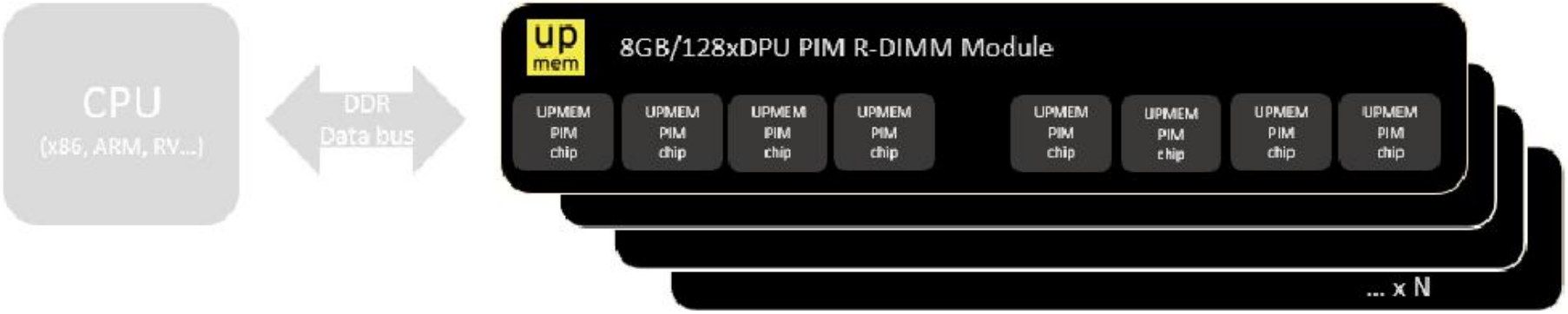
[UPMEM 2019]

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.

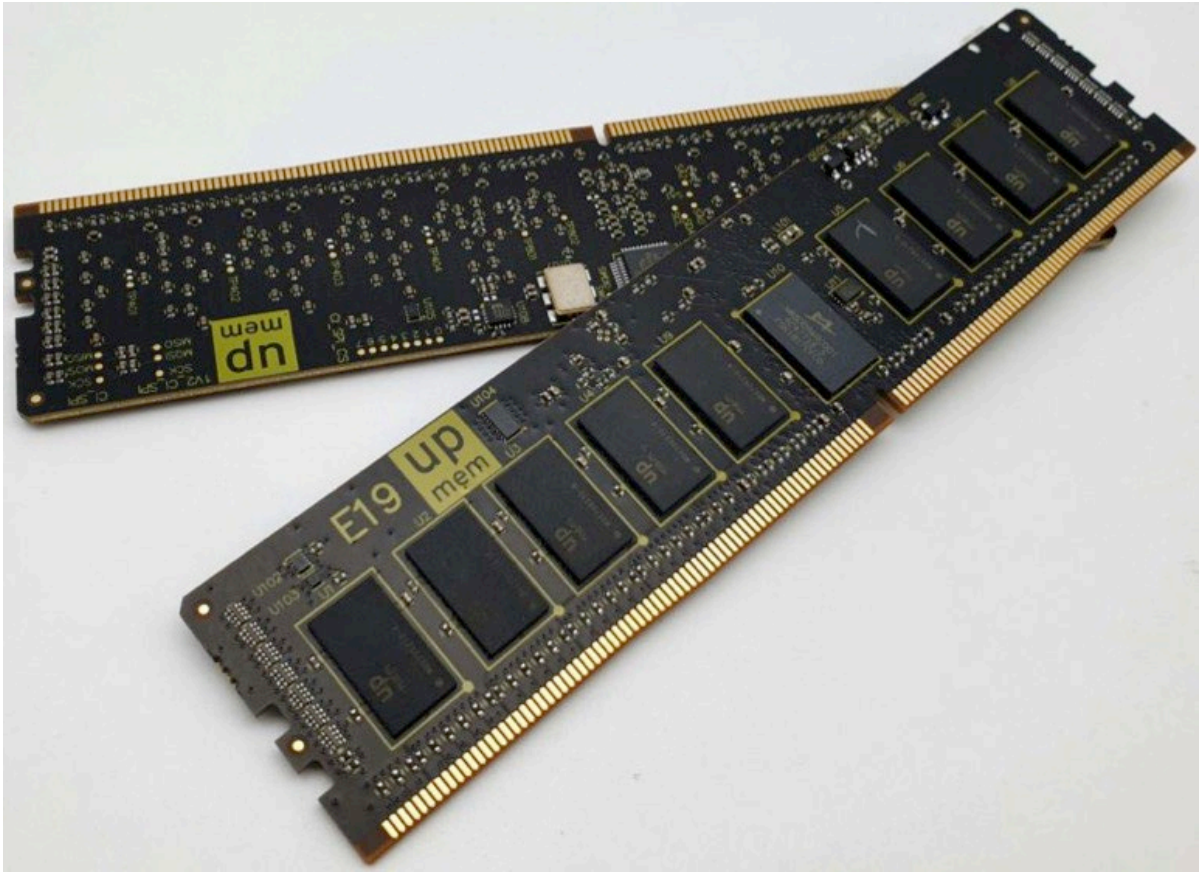
- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth



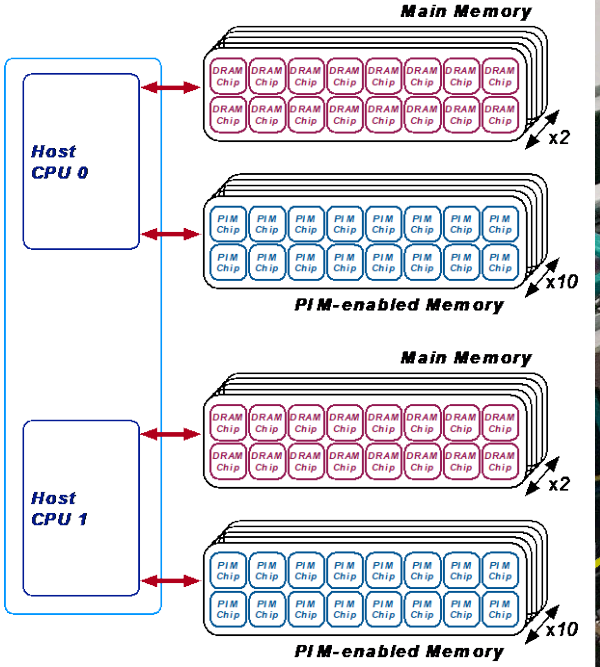
https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem
https://www.upmem.com/video-upmem-presenting-its-true-processing-in-memory-solution-hot-chips-2019/

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



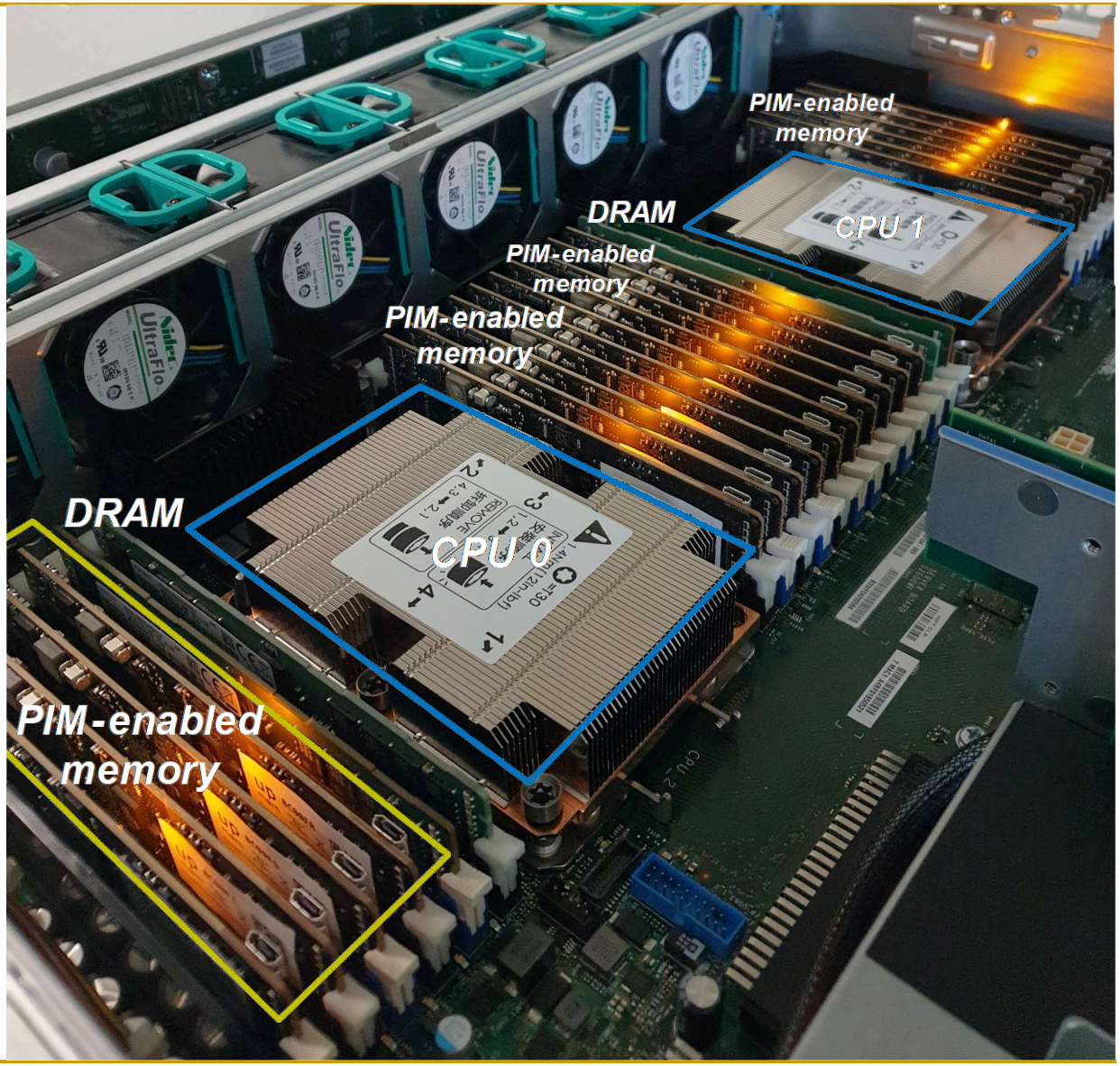
Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
IZZAT EL HAJJ, American University of Beirut, Lebanon
IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
CHRISTINA GIANNOULA, 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*).

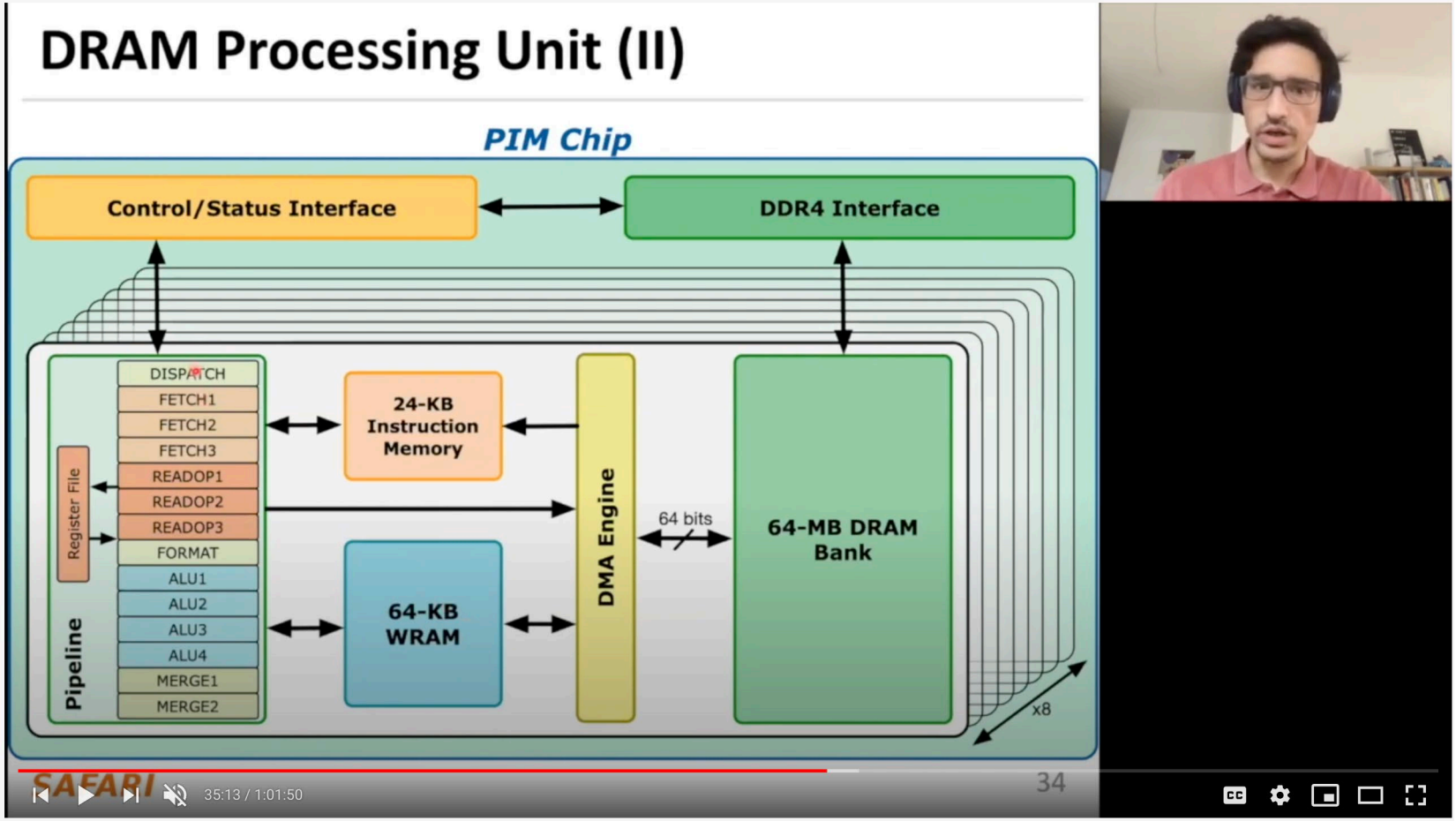
Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware 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* (*DPUs*), 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 characterization 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 *PrIM* (*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 scaling characteristics of PrIM benchmarks on the UPMEM 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 UPMEM-based PIM systems with 640 and 2,556 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.



https://arxiv.org/pdf/2105.03814.pdf

More on the UPMEM PIM System



https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=26

Experimental Analysis of the UPMEM PIM Engine

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, Switzerland and University of Malaga, Spain
CHRISTINA GIANNOULA, 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 architectures, 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* (*DPUs*), 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 characterization 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 *PrIM* (*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 scaling characteristics of PrIM benchmarks on the UPMEM 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 UPMEM-based PIM systems with 640 and 2,556 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.

https://arxiv.org/pdf/2105.03814.pdf

Upcoming TECHCON Presentation

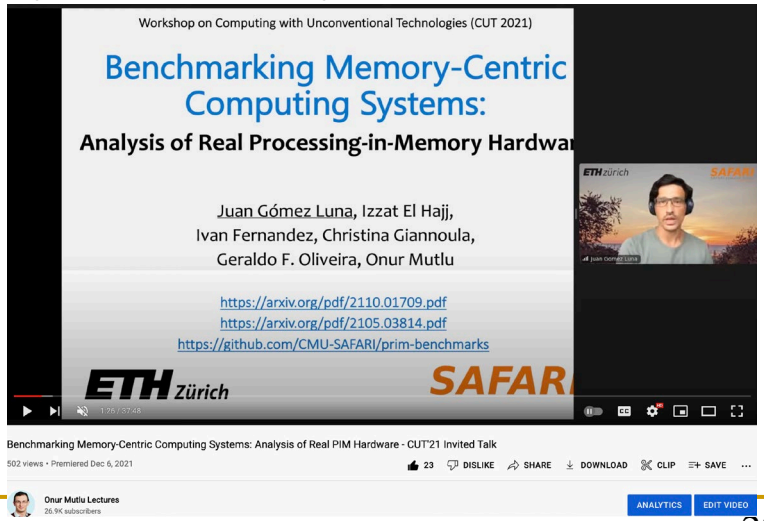
- Dr. Juan Gomez-Luna
  - Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware
  - Based on two major works
    - https://arxiv.org/pdf/2105.03814.pdf
    - https://arxiv.org/pdf/2207.07886.pdf

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 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



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

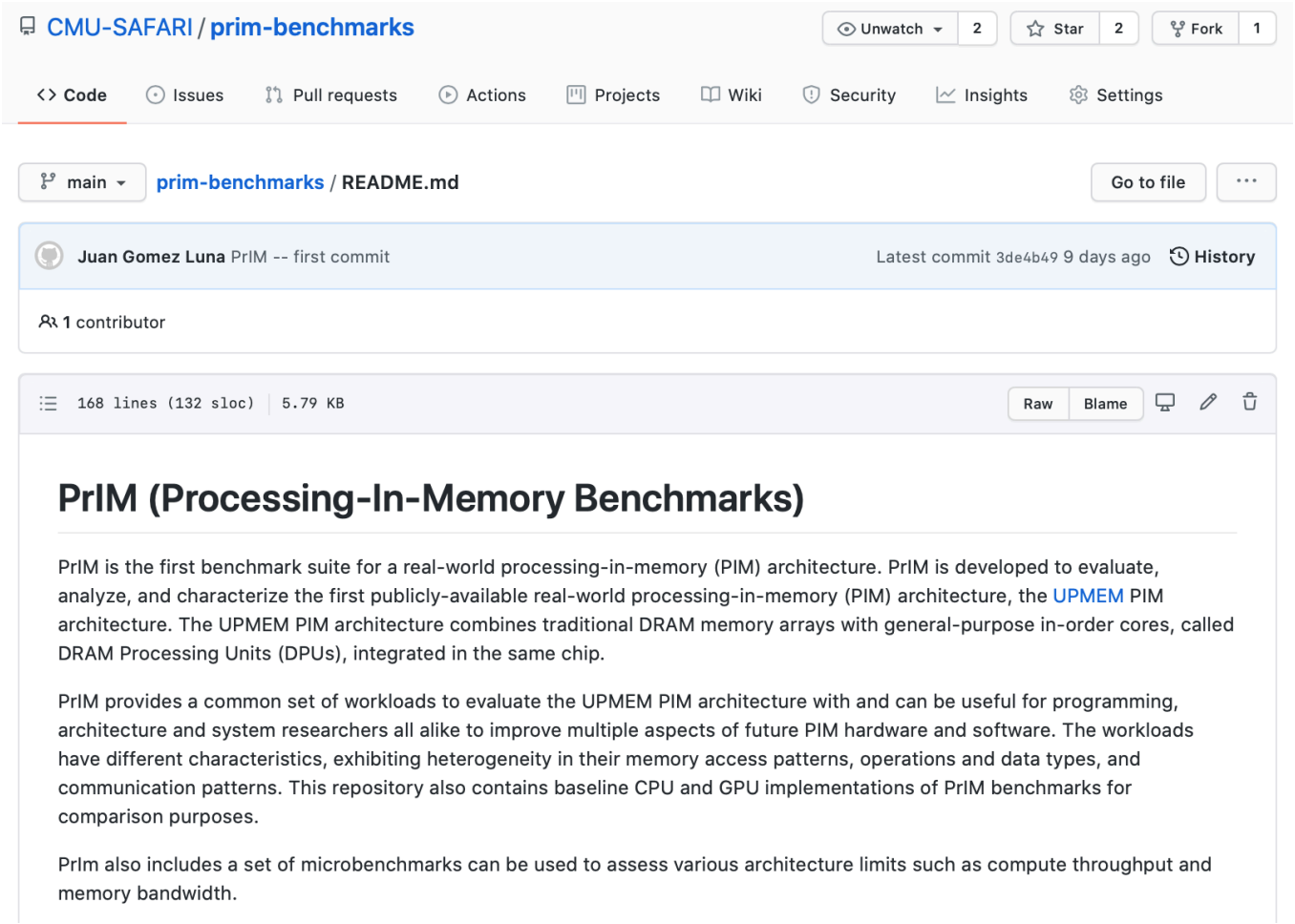
Juan Gómez-Luna, *ETH Zürich*
Izzat El Hajj, *American University of Beirut*
Ivan Fernandez, *University of Malaga*
Christina Giannoula, *National Technical University of Athens*
Geraldo F. Oliveira, *ETH Zürich*
Onur Mutlu, *ETH Zürich*

PrIM Benchmarks: Application Domains

Domain	Benchmark	Short name
Dense linear algebra	Vector Addition	VA
	Matrix-Vector Multiply	GEMV
Sparse linear algebra	Sparse Matrix-Vector Multiply	SpMV
Databases	Select	SEL
	Unique	UNI
Data analytics	Binary Search	BS
	Time Series Analysis	TS
Graph processing	Breadth-First Search	BFS
Neural networks	Multilayer Perceptron	MLP
Bioinformatics	Needleman-Wunsch	NW
Image processing	Image histogram (short)	HST-S
	Image histogram (large)	HST-L
Parallel primitives	Reduction	RED
	Prefix sum (scan-scan-add)	SCAN-SSA
	Prefix sum (reduce-scan-scan)	SCAN-RSS
	Matrix transposition	TRNS

PrIM Benchmarks are Open Source

- All microbenchmarks, benchmarks, and scripts
- https://github.com/CMU-SAFARI/prim-benchmarks



Understanding a Modern PIM Architecture

Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System

JUAN GÓMEZ-LUNA[1], IZZAT EL HAJJ[2], IVAN FERNANDEZ[1,3], CHRISTINA GIANNOULA[1,4], GERALDO F. OLIVEIRA[1], AND ONUR MUTLU[1]

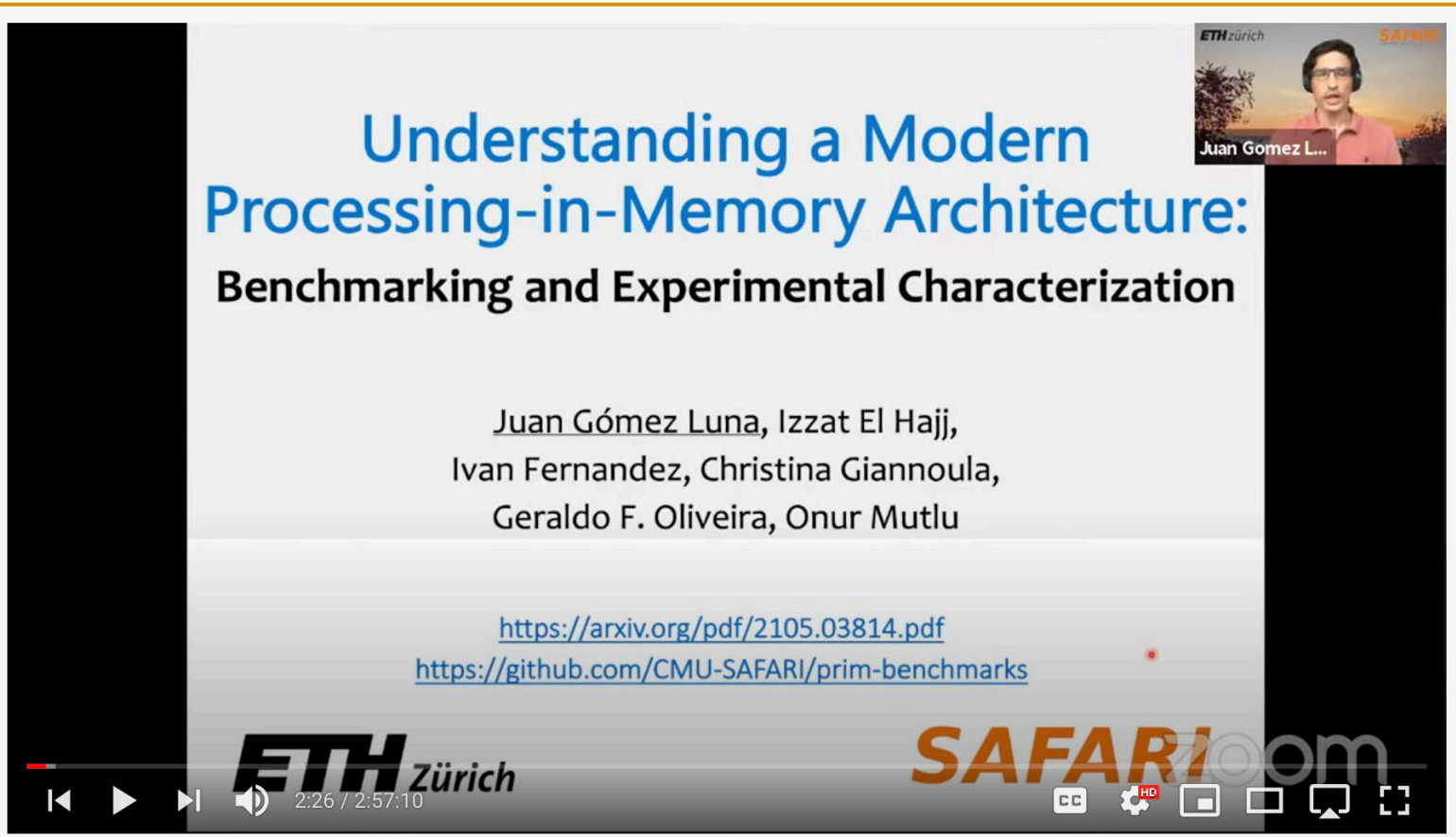
[1]ETH Zürich
[2]American University of Beirut
[3]University of Malaga
[4]National Technical University of Athens

Corresponding author: Juan Gómez-Luna (e-mail: juang@ethz.ch).

https://arxiv.org/pdf/2105.03814.pdf

https://github.com/CMU-SAFARI/prim-benchmarks

Understanding a Modern PIM Architecture



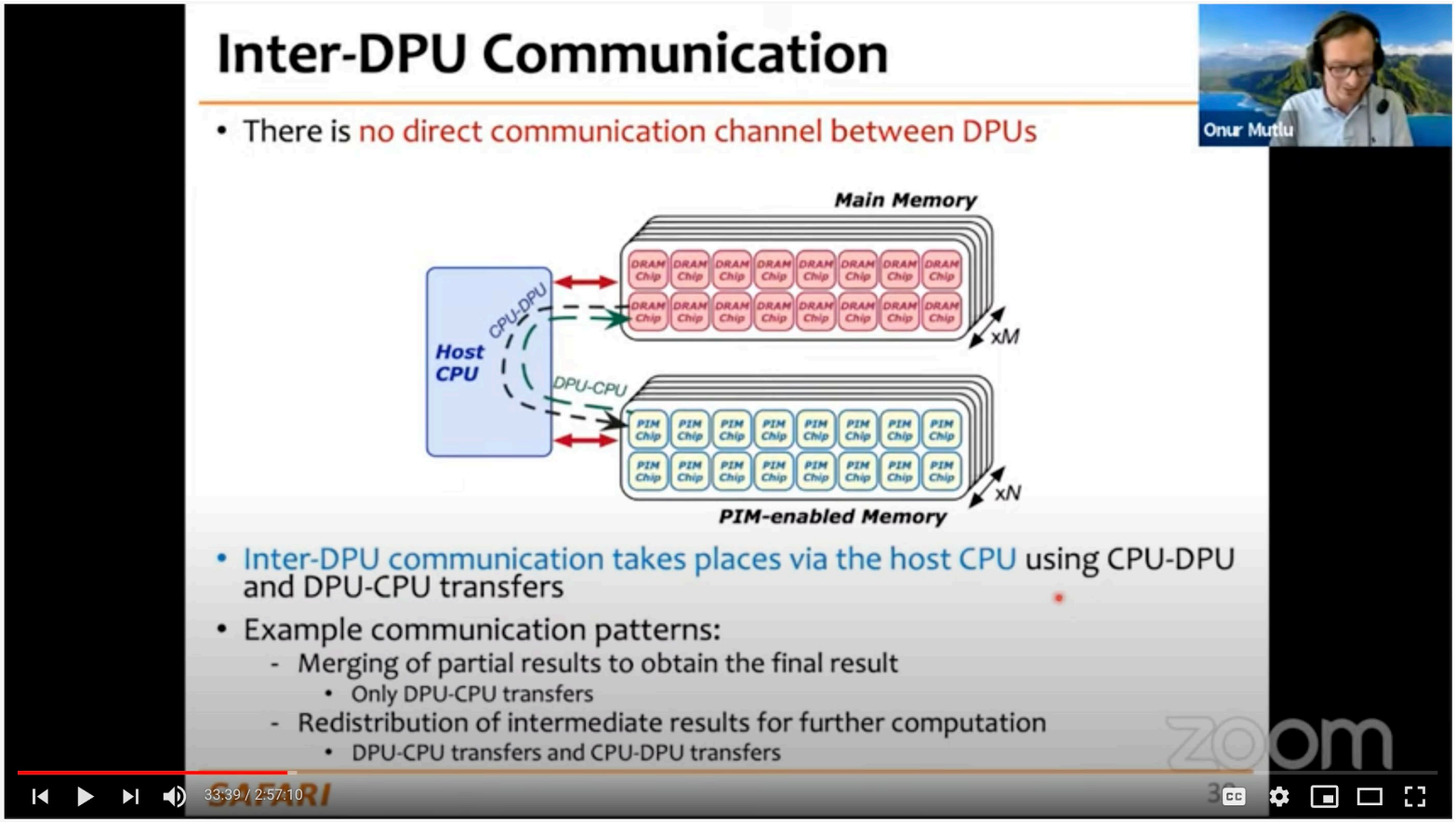
SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture

2,579 views • Streamed live on Jul 12, 2021 93 0 SHARE SAVE ...

Onur Mutlu Lectures
18.7K subscribers
SUBSCRIBED

https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9

More on Analysis of the UPMEM PIM Engine



SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture

1,868 views • Streamed live on Jul 12, 2021

81 0 SHARE SAVE ...

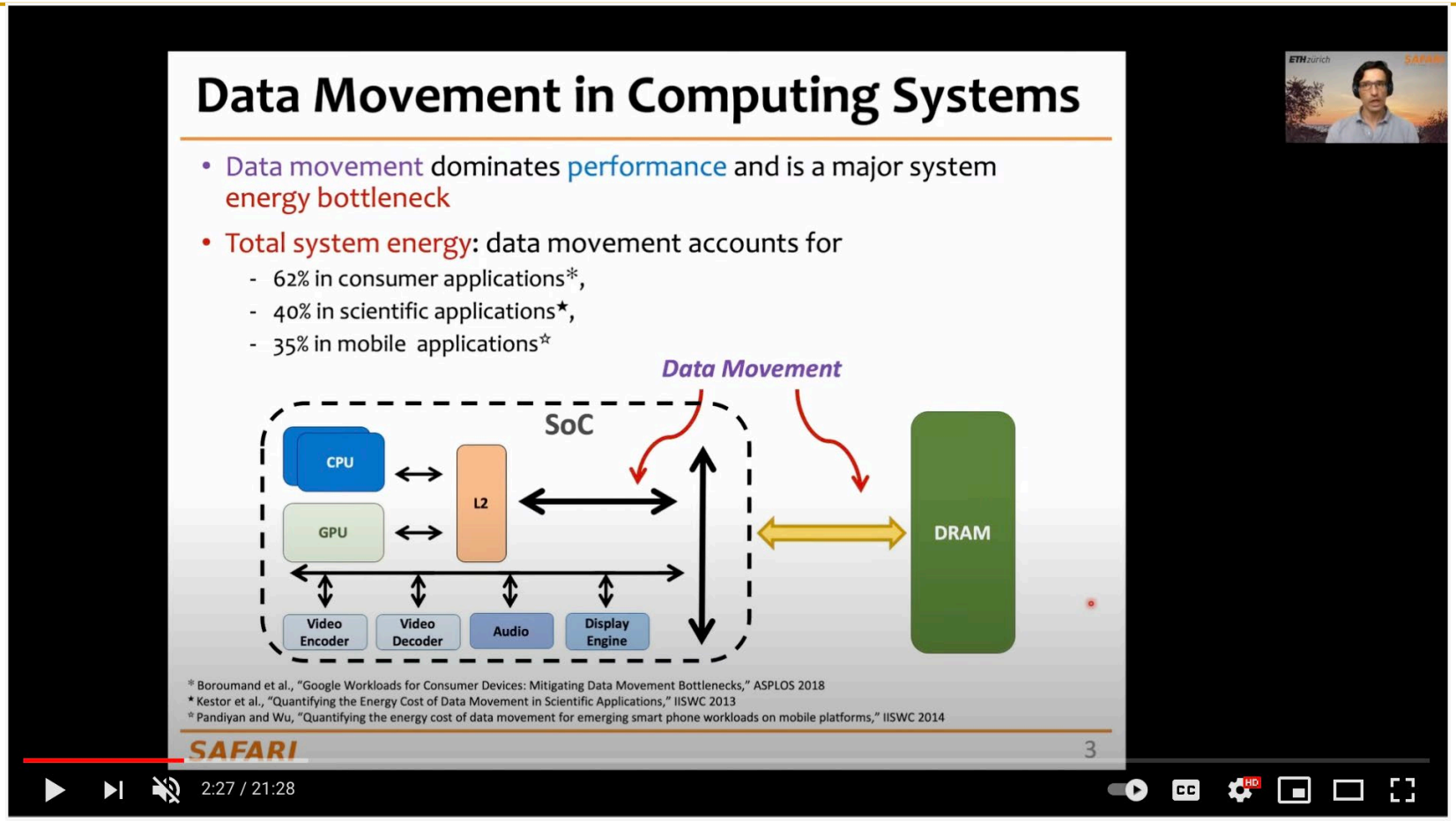
Onur Mutlu Lectures
17.6K subscribers

ANALYTICS EDIT VIDEO

Talk Title: Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization
Dr. Juan Gómez-Luna, SAFARI Research Group, D-ITET, ETH Zurich

https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9

More on Analysis of the UPMEM PIM Engine



Understanding a Modern Processing-in-Memory Arch: Benchmarking & Experimental Characterization; 21m

3,482 views • Premiered Jul 25, 2021 38 0 SHARE SAVE

Onur Mutlu Lectures
17.9K subscribers

ANALYTICS EDIT VIDEO

https://www.youtube.com/watch?v=Pp9jSU2b9oM&list=PL5Q2soXY2Zi8_VVChACnON4sfh2bJ5IrD&index=159

More on PRIM Benchmarks

- Juan Gomez-Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu,
  "Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture"
  Preprint in arXiv, 9 May 2021.
  [arXiv preprint]
  [PrIM Benchmarks Source Code]
  [Slides (pptx) (pdf)]
  [Long Talk Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]
  [SAFARI Live Seminar Slides (pptx) (pdf)]
  [SAFARI Live Seminar Video (2 hrs 57 mins)]
  [Lightning Talk Video (3 minutes)]

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*

ML Training on a Real PIM System

Machine Learning Training on a Real Processing-in-Memory System

Juan Gómez-Luna[1] Yuxin Guo[1] Sylvan Brocard[2] Julien Legriel[2]
Remy Cimadomo[2] Geraldo F. Oliveira[1] Gagandeep Singh[1] Onur Mutlu[1]
[1]ETH Zürich [2]UPMEM

An Experimental Evaluation of Machine Learning Training on a Real Processing-in-Memory System

Juan Gómez-Luna[1] Yuxin Guo[1] Sylvan Brocard[2] Julien Legriel[2]
Remy Cimadomo[2] Geraldo F. Oliveira[1] Gagandeep Singh[1] Onur Mutlu[1]
[1]ETH Zürich [2]UPMEM

Short version: https://arxiv.org/pdf/2206.06022.pdf

Long version: https://arxiv.org/pdf/2207.07886.pdf

https://www.youtube.com/watch?v=qeukNs5XI3g&t=11226s

AIM (PIM Sequence Alignment Framework)

Safaa Diab, Amir Nassereldine, Mohammed Alser, Juan Gómez-Luna, Onur Mutlu, Izzat El Hajj
"**A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems**"
arXiv, 2022
[Source code]

A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems

Safaa Diab[1], Amir Nassereldine[1], Mohammed Alser[2], Juan Gómez Luna[2], Onur Mutlu[2], Izzat El Hajj[1]

[1]*American University of Beirut, Lebanon* [2]*ETH Zürich, Switzerland*

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"
  Proceedings of the *55th International Symposium on Microarchitecture* ***(MICRO)***, Chicago, Illinois, October 2022.

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

Newer Genome Sequencing Technologies

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** ▾

Oxford Nanopore MinION

Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions**," Briefings in Bioinformatics, 2018.
[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 MansouriGhiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,
  "SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
  *Proceedings of the 49th International Symposium on Computer Architecture (**ISCA**)*, New York, June 2022.
  [arXiv version]

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

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

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

New Applications: Ref Genome Updates

RESEARCH

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

Jeremie S. Kim[1], Can Firtina[1], Meryem Banu Cavlak[2], Damla Senol Cali[3], Nastaran Hajinazar[1,4], Mohammed Alser[1], Can Alkan[2] and Onur Mutlu[1,2,3*]

https://people.inf.ethz.ch/omutlu/pub/AirLift_genome-remapper_arxiv21.pdf

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"
  Preprint in arXiv and bioRxiv, 2021.
  [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[1†], Can Firtina[1†], Meryem Banu Cavlak[2], Damla Senol Cali[3], Nastaran Hajinazar[1,4], Mohammed Alser[1], Can Alkan[2] and Onur Mutlu[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*

Newer Genome Sequencing Technologies

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** ▾

Oxford Nanopore MinION

Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions**," Briefings in Bioinformatics, 2018.
[Open arxiv.org version] [Slides (pptx) (pdf)] [Talk Video at AACBB 2019]

Recall: High-Throughput Sequencing

- Massively parallel sequencing technology
  - Illumina, Roche 454, Ion Torrent, SOLID…
- Small DNA fragments are first amplified and then sequenced in parallel, leading to
  - High throughput
  - High speed
  - Low cost
  - Short reads
    - Amplification step limits the read length since too short or too long fragments are not amplified well.
- Sequencing is done by either reading optical signals as each base is added, or by detecting hydrogen ions instead of light, leading to:
  - Low error rates (relatively)
  - Reads lack information about their order and which part of genome they are originated from

Nanopore Sequencing Technology

- **Nanopore sequencing** is an emerging and a promising single-molecule DNA sequencing technology

- First nanopore sequencing device, **MinION**, made commercially available by **Oxford Nanopore Technologies** (ONT) in **May 2014.**
  - Inexpensive
  - Long read length (> 882K bp)
  - Portable: Pocket-sized
  - Produces data in real-time

Nanopore Sequencing Technology

an emerging and a promising
ncing technology
read length → Longer read length

- First nanopore sequencing device, **MinION**, made commercially available by **Oxford Nanopore Technologies** (ONT) in **May 2014.**
  - Inexpensive
  - Long read length (> 882K bp)
  - Portable: Pocket-sized
  - Produces data in real-time

Oxford Nanopore Sequencers

MinION Mk1B

MinION Mk1C

GridION Mk1

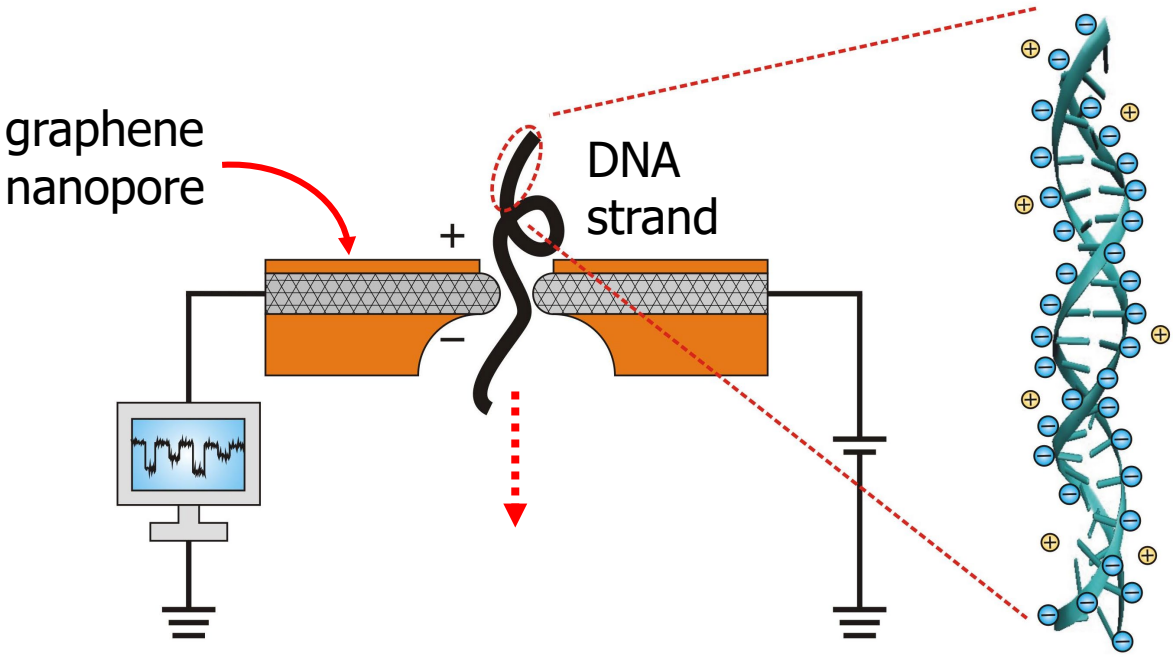
PromethION 24/48

	MinION Mk1B	MinION Mk1C	GridION Mk1	PromethION 24	PromethION 48
Read length	> 2Mb	> 2Mb	> 2Mb	> 2Mb	> 2Mb
Yield per flow cell	50 Gb	50 Gb	50 Gb	220 Gb	220 Gb
Number of flow cells per device	1	1	5	24	48
Yield per device	<50 Gb	<50 Gb	<250 Gb	<5.2 Tb	<10.5 Tb
Starting price	$1,000	$4,990	$49,995	$195,455	$327,455

Illumina Sequencers

Run time	9.5–19 hrs	4–24 hrs	4–55 hrs	12–30 hrs	24-48 hrs	13-44 hrs
Max. reads per run	4 million	25 million	25 million	400 million	1 billion	20 billion
Max. read length	2 × 150 bp	2 × 150 bp	2 × 300 bp	2 × 150 bp	2 × 150 bp	2 x 250
Max. output	1.2 Gb	7.5 Gb	15 Gb	120 Gb	300 Gb	6000 Gb
Estimated price	$19,900	$49,500	$128,000	$275,000	$335,000	$985,000

How Does Nanopore Sequencing Work?



- **Nanopore** is a nano-scale hole (<20nm).
- In nanopore sequencers, an **ionic current** passes through the nanopores
- When the DNA strand passes through the nanopore, the sequencer measures the the **change in current**
- This change is used to identify the bases in the strand with the help of **different electrochemical structures** of the different bases

Figure is adapted from: https://phys.org/news/2013-12-gene-sequencing-future.html

Advantages of Nanopore Sequencing

Nanopores:

- Do *not* require any labeling of the DNA or nucleotide for detection during sequencing
- Rely on the electronic or chemical structure of the different nucleotides for identification
- Allow sequencing very long reads, and
- Provide portability, low cost, and high throughput.

Challenges of Nanopore Sequencing

- One major drawback: high error rates
- Nanopore sequence analysis tools have a critical role to:
  - overcome high error rates
  - take better advantage of the technology
- Faster tools are critically needed to:
  - Take better advantage of the real-time data production capability of nanopore sequencing
  - Enable fast, real-time data analysis

Nanopore Genome Assembly Pipeline

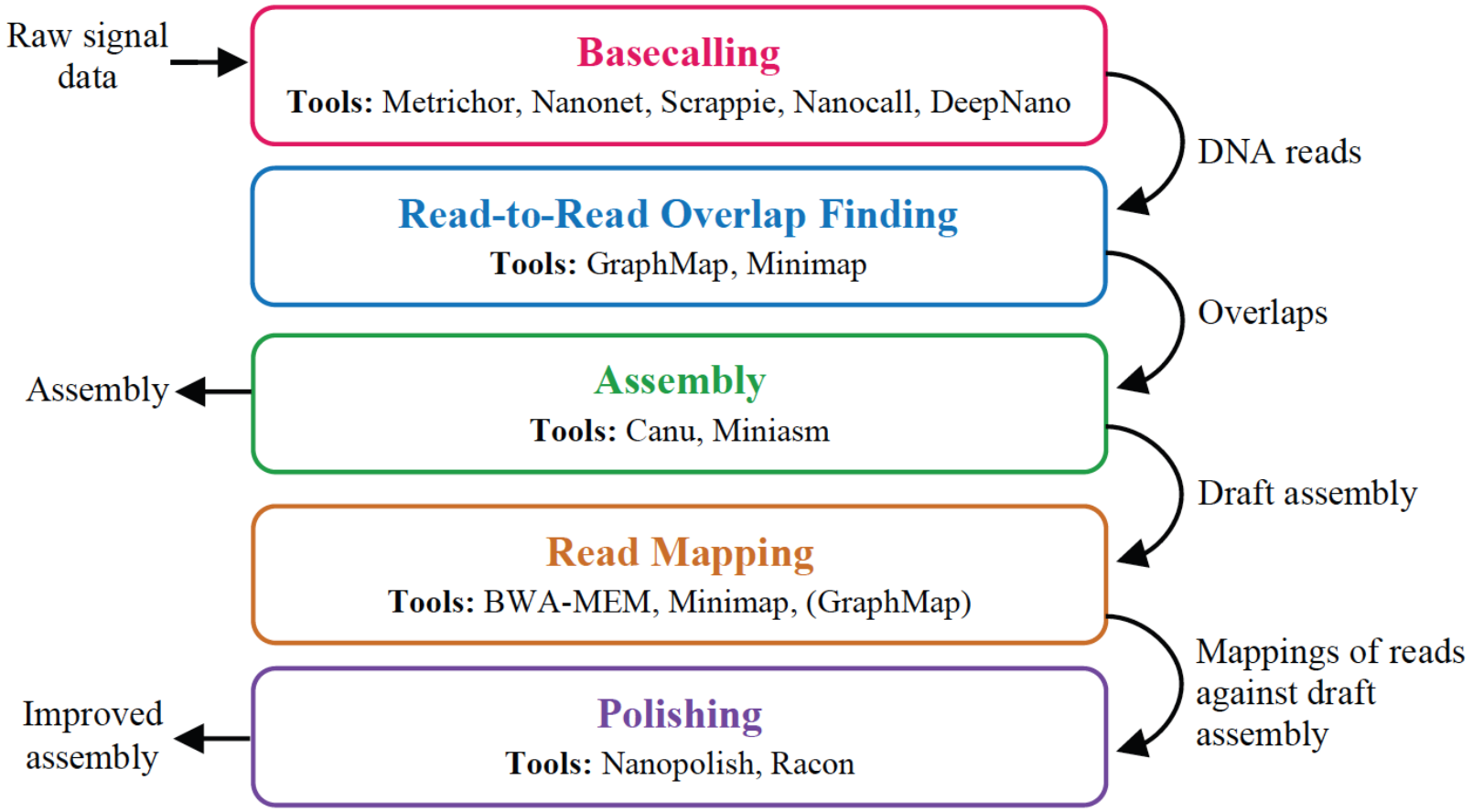


Figure 1. The analyzed genome assembly pipeline using nanopore sequence data, with its five steps and the associated tools for each step.

Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly**" Briefings in Bioinformatics, 2018.

Nanopore Genome Assembly Tools (I)

Table 12. Accuracy analysis results for the full pipeline with a focus on the last two steps.

						Number of Bases	Number of Contigs	Identity (%)	Coverage (%)	Number of Mismatches	Number of Indels
1	Metrichor	+ —	+ Canu	+ BWA-MEM	+ Nanopolish	4,683,072	1	99.48	99.93	8,198	15,581
2	Metrichor	+ Minimap	+ Miniasm	+ BWA-MEM	+ Nanopolish	4,540,352	1	92.33	96.31	162,884	182,965
3	Metrichor	+ GraphMap	+ Miniasm	+ BWA-MEM	+ Nanopolish	4,637,916	2	92.38	95.80	159,206	180,603
4	Metrichor	+ —	+ Canu	+ BWA-MEM	+ Racon	4,650,502	1	98.46	100.00	18,036	51,842
5	Metrichor	+ —	+ Canu	+ Minimap	+ Racon	4,648,710	1	98.45	100.00	17,906	52,168
6	Metrichor	+ Minimap	+ Miniasm	+ BWA-MEM	+ Racon	4,598,267	1	97.70	99.91	24,014	82,906
7	Metrichor	+ Minimap	+ Miniasm	+ Minimap	+ Racon	4,600,109	1	97.78	100.00	23,339	79,721
8	Nanonet	+ —	+ Canu	+ BWA-MEM	+ Racon	4,622,285	1	98.48	100.00	16,872	52,509
9	Nanonet	+ —	+ Canu	+ Minimap	+ Racon	4,620,597	1	98.49	100.00	16,874	52,232
10	Nanonet	+ Minimap	+ Miniasm	+ BWA-MEM	+ Racon	4,593,402	1	98.01	99.97	20,322	72,284
11	Nanonet	+ Minimap	+ Miniasm	+ Minimap	+ Racon	4,592,907	1	98.04	100.00	20,170	70,705
12	Scrappie	+ —	+ Canu	+ BWA-MEM	+ Racon	4,673,871	1	98.40	99.98	13,583	60,612
13	Scrappie	+ —	+ Canu	+ Minimap	+ Racon	4,673,606	1	98.40	99.98	13,798	60,423
14	Scrappie	+ Minimap	+ Miniasm	+ BWA-MEM	+ Racon	5,157,041	8	97.87	99.80	18,085	78,492
15	Scrappie	+ Minimap	+ Miniasm	+ Minimap	+ Racon	5,156,375	8	97.87	99.94	17,922	77,807
16	Nanocall	+ —	+ Canu	+ BWA-MEM	+ Racon	1,383,851	86	93.49	28.82	19,057	65,244
17	Nanocall	+ —	+ Canu	+ Minimap	+ Racon	1,367,834	86	94.43	28.74	15,610	55,275
18	Nanocall	+ Minimap	+ Miniasm	+ BWA-MEM	+ Racon	4,707,961	5	90.75	97.11	91,502	347,005
19	Nanocall	+ Minimap	+ Miniasm	+ Minimap	+ Racon	4,673,069	5	92.23	97.10	72,646	291,918
20	DeepNano	+ —	+ Canu	+ BWA-MEM	+ Racon	7,429,290	106	96.46	99.24	27,811	102,682
21	DeepNano	+ —	+ Canu	+ Minimap	+ Racon	7,404,454	106	96.03	99.21	34,023	110,640
22	DeepNano	+ Minimap	+ Miniasm	+ BWA-MEM	+ Racon	4,566,253	1	96.76	99.86	25,791	125,386
23	DeepNano	+ Minimap	+ Miniasm	+ Minimap	+ Racon	4,571,810	1	96.90	99.97	24,994	119,519

Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly**" Briefings in Bioinformatics, 2018.

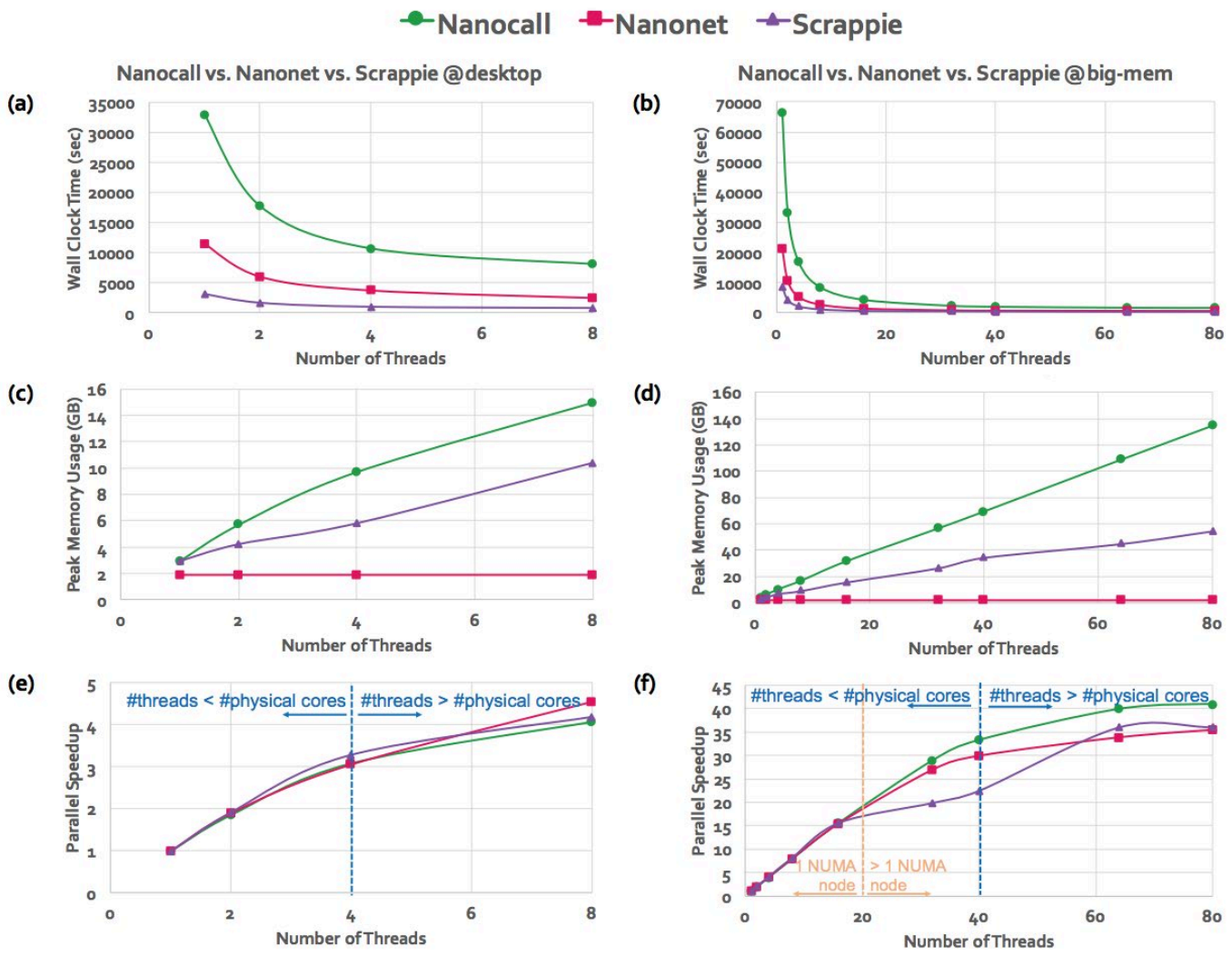
Nanopore Genome Assembly Tools (II)

Table 13. Performance analysis results for the full pipeline with a focus on the last two steps.

									Step 4: Read Mapper			Step 5: Polisher		
									Wall Clock Time (h:m:s)	CPU Time (h:m:s)	Memory Usage (GB)	Wall Clock Time (h:m:s)	CPU Time (h:m:s)	Memory Usage (GB)
1	Metrichor	+	—	+	Canu	+	BWA-MEM	+	Nanopolish	24:43	15:47:21	5.26	5:51:00	191:18:52	13.38
2	Metrichor	+	Minimap	+	Miniasm	+	BWA-MEM	+	Nanopolish	12:33	7:50:54	3.75	122:52:00	4458:36:10	31.36
3	Metrichor	+	GraphMap	+	Miniasm	+	BWA-MEM	+	Nanopolish	12:47	7:57:58	3.60	129:46:00	4799:03:51	31.31
4	Metrichor	+	—	+	Canu	+	BWA-MEM	+	Racon	24:20	15:43:40	6.60	14:44	9:09:22	8.11
5	Metrichor	+	—	+	Canu	+	Minimap	+	Racon	3	1:35	0.26	15:12	9:45:33	14.55
6	Metrichor	+	Minimap	+	Miniasm	+	BWA-MEM	+	Racon	12:10	7:48:10	5.19	15:43	9:33:39	9.98
7	Metrichor	+	Minimap	+	Miniasm	+	Minimap	+	Racon	3	1:24	0.26	20:28	8:57:40	18.24
8	Nanonet	+	—	+	Canu	+	BWA-MEM	+	Racon	9:08	5:53:18	4.84	6:33	4:02:10	4.47
9	Nanonet	+	—	+	Canu	+	Minimap	+	Racon	2	54	0.26	6:45	4:17:26	7.93
10	Nanonet	+	Minimap	+	Miniasm	+	BWA-MEM	+	Racon	4:40	2:58:02	3.88	7:08	4:19:30	5.35
11	Nanonet	+	Minimap	+	Miniasm	+	Minimap	+	Racon	2	46	0.26	7:01	4:18:48	9.53
12	Scrappie	+	—	+	Canu	+	BWA-MEM	+	Racon	33:41	21:11:06	8.66	13:32	8:24:44	7.58
13	Scrappie	+	—	+	Canu	+	Minimap	+	Racon	3	1:39	0.27	18:45	7:43:17	13.20
14	Scrappie	+	Minimap	+	Miniasm	+	BWA-MEM	+	Racon	22:41	14:31:00	6.08	14:37	8:53:59	9.50
15	Scrappie	+	Minimap	+	Miniasm	+	Minimap	+	Racon	3	1:27	0.27	15:10	9:02:45	12.72
16	Nanocall	+	—	+	Canu	+	BWA-MEM	+	Racon	4:52	3:01:15	3.80	11:07	3:26:52	5.63
17	Nanocall	+	—	+	Canu	+	Minimap	+	Racon	3	1:16	0.22	7:28	2:50:35	3.62
18	Nanocall	+	Minimap	+	Miniasm	+	BWA-MEM	+	Racon	16:06	10:27:20	5.06	18:56	11:32:45	11.47
19	Nanocall	+	Minimap	+	Miniasm	+	Minimap	+	Racon	4	1:18	0.26	11:49	7:08:59	10.98
20	DeepNano	+	—	+	Canu	+	BWA-MEM	+	Racon	17:36	11:30:20	4.43	12:48	7:13:04	8.88
21	DeepNano	+	—	+	Canu	+	Minimap	+	Racon	3	1:24	0.28	11:39	6:55:01	3.73
22	DeepNano	+	Minimap	+	Miniasm	+	BWA-MEM	+	Racon	8:15	5:22:29	4.11	14:16	8:34:32	10.30
23	DeepNano	+	Minimap	+	Miniasm	+	Minimap	+	Racon	3	1:10	0.26	12:29	7:55:32	17.11

Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly**" Briefings in Bioinformatics, 2018.

Nanopore Genome Assembly Tools (III)



Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly**" to appear in Briefings in Bioinformatics, 2018.

More on Nanopore Sequencing & Tools

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** ▾

BiB arXiv

Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions**," Briefings in Bioinformatics, 2018.
[Preliminary arxiv.org version]

Why Do We Care? An Example from 2020

200 Oxford Nanopore sequencers have left UK for China, to support rapid, near-sample coronavirus sequencing for outbreak surveillance

Fri 31st January 2020

Following extensive support of, and collaboration with, public health professionals in China, Oxford Nanopore has shipped an additional 200 MinION sequencers and related consumables to China. These will be used to support the ongoing surveillance of the current coronavirus outbreak, adding to a large number of the devices already installed in the country.

Each MinION sequencer is approximately the size of a stapler, and can provide rapid sequence information about the coronavirus.

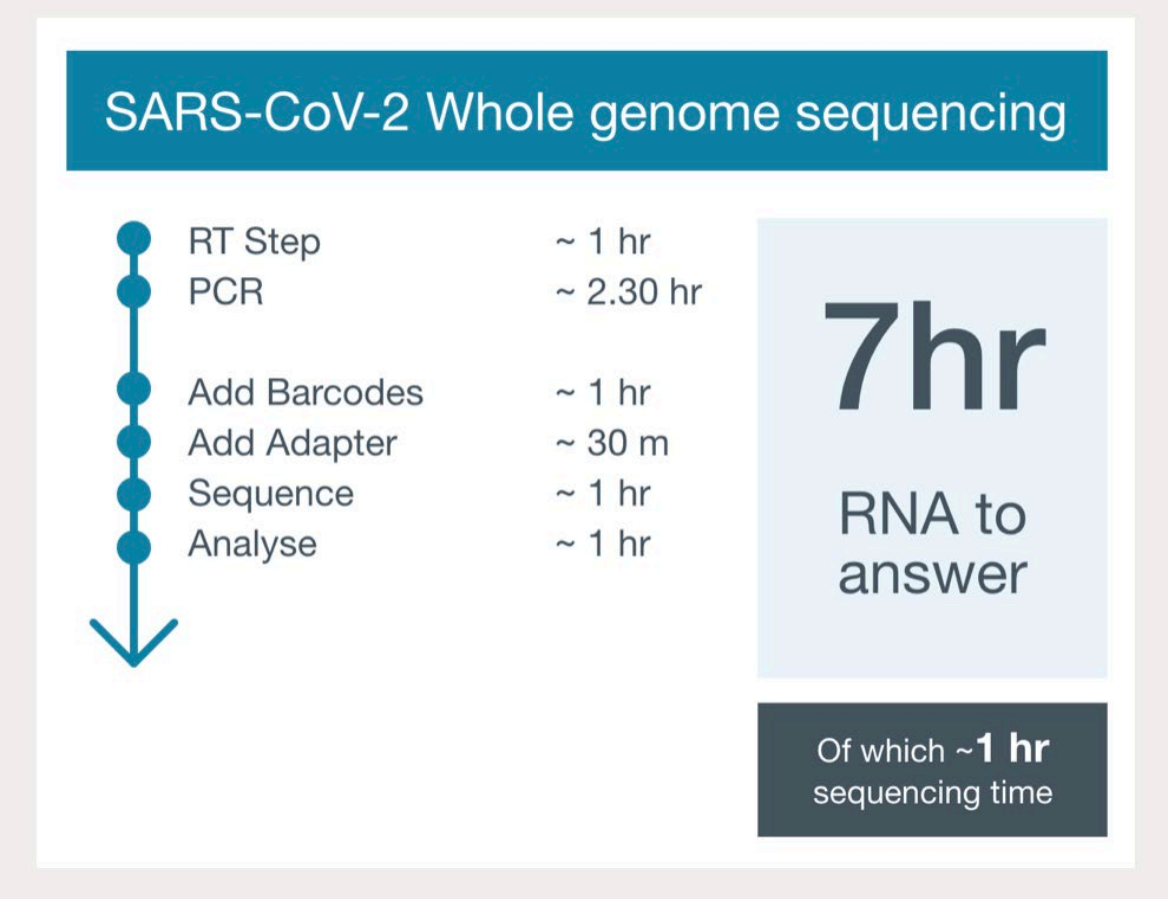
700Kg of Oxford Nanopore sequencers and consumables are on their way for use by Chinese scientists in understanding the current coronavirus outbreak.

Source: https://nanoporetech.com/about-us/news/200-oxford-nanopore-sequencers-have-left-uk-china-support-rapid-near-sample

Sequencing of COVID-19

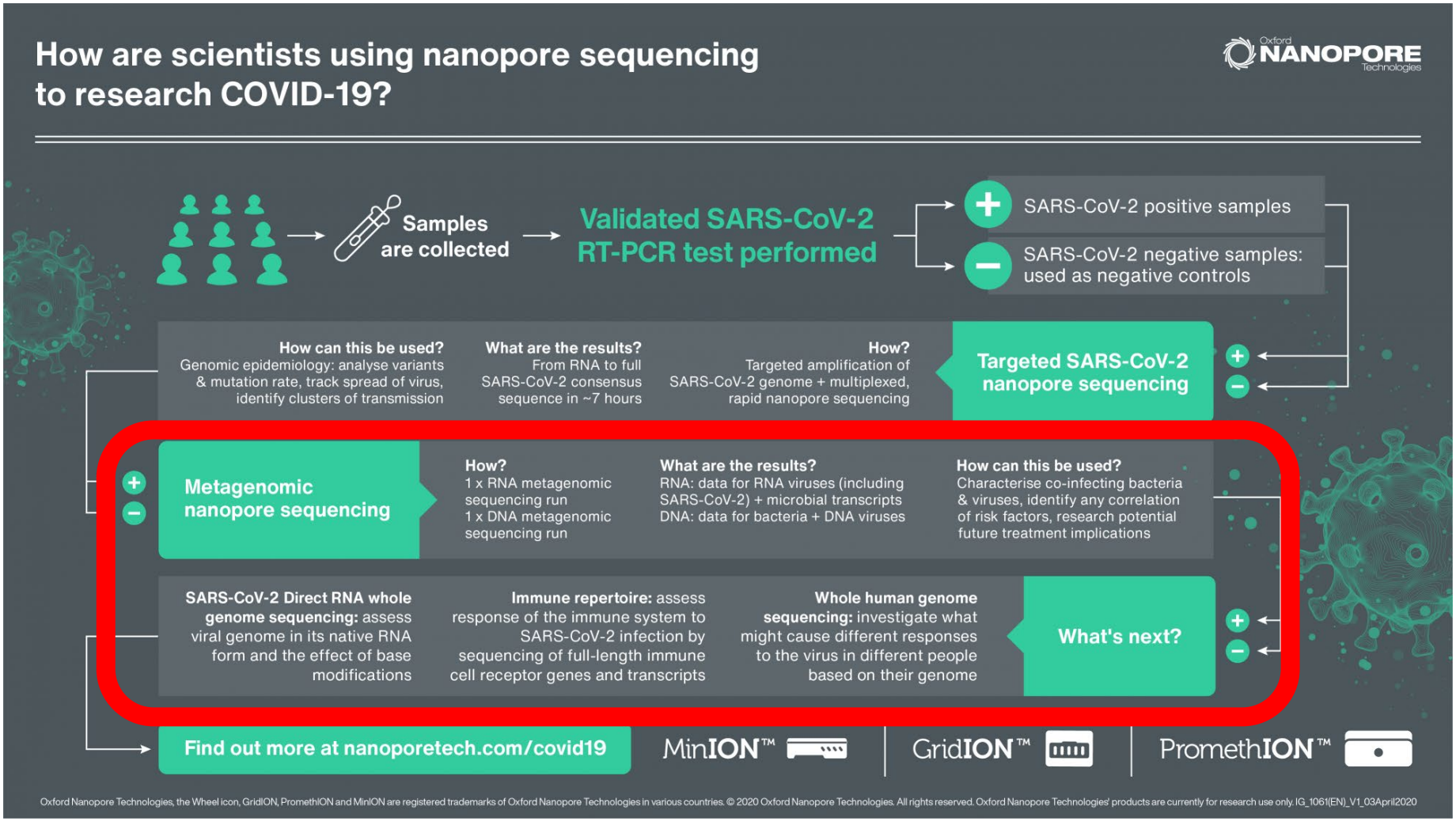
- **Whole genome sequencing (WGS) and sequence data analysis are important**
  - To detect the virus from a human sample such as saliva, Bronchoalveolar fluid etc.
  - To understand the sources and modes of transmission of the virus
  - To discover the genomic characteristics of the virus, and compare with better-known viruses (e.g., 02-03 SARS epidemic)
  - To design and evaluate the diagnostic tests and deep-dive studies
- **Two key areas of COVID-19 genomic research**
  - To sequence the genome of the virus itself, COVID-19, in order to track the mutations in the virus.
  - To explore the genes of infected patients. This analysis can be used to understand why some people get more severe symptoms than others, as well as, help with the development of new treatments in the future.

COVID-19 Nanopore Sequencing (I)



- From ONT (https://nanoporetech.com/covid-19/overview)

COVID-19 Nanopore Sequencing (II)



- From ONT (https://nanoporetech.com/covid-19/overview)

A Bright Future for Intelligent Genome Analysis

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

MinION from ONT

Accelerating Genome Analysis: A Primer on an Ongoing Journey
Sept.-Oct. 2020, pp. 65-75, vol. 40
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
July-Aug. 2021, pp. 39-48, vol. 41
DOI Bookmark: 10.1109/MM.2021.3088396

SmidgION from ONT

Agenda

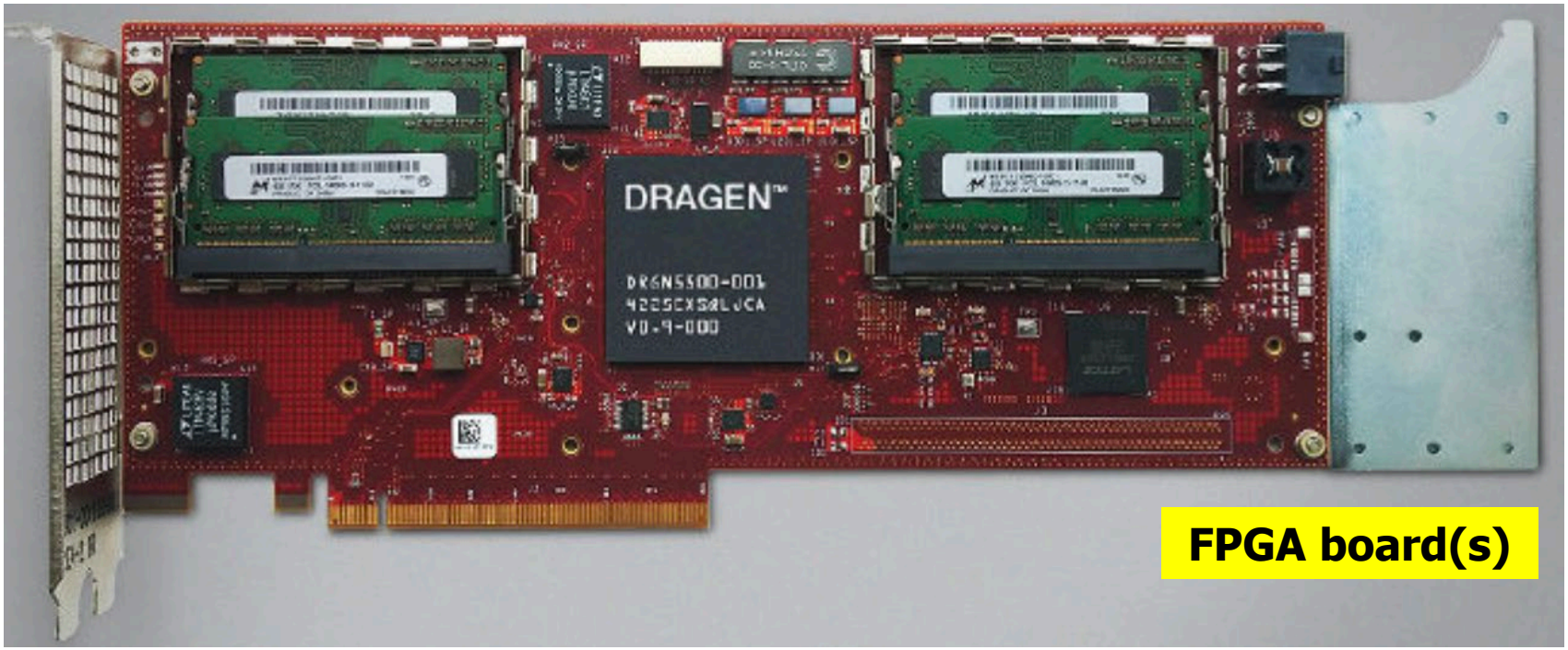
- The Problem: DNA Read Mapping
  - State-of-the-art Read Mapper Design
- Algorithmic Acceleration
  - Exploiting Structure of the Genome
  - Exploiting SIMD Instructions
- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory & Storage
- Future Opportunities: New Technologies & Applications

Conclusion

Things Are Happening In Industry

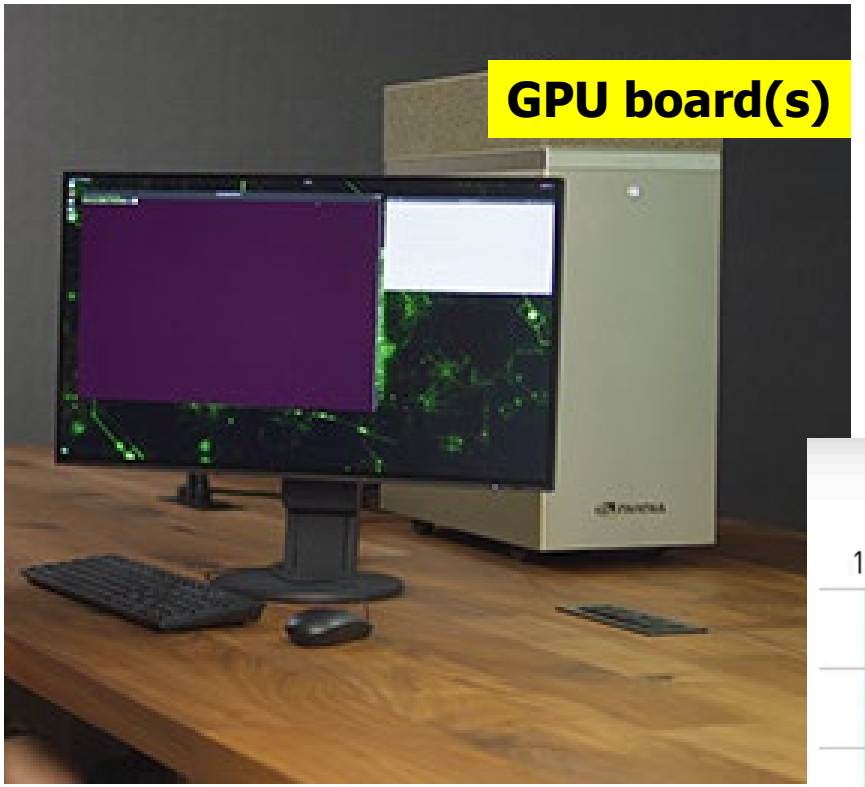
Illumina DRAGEN Bio-IT Platform (2018)

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

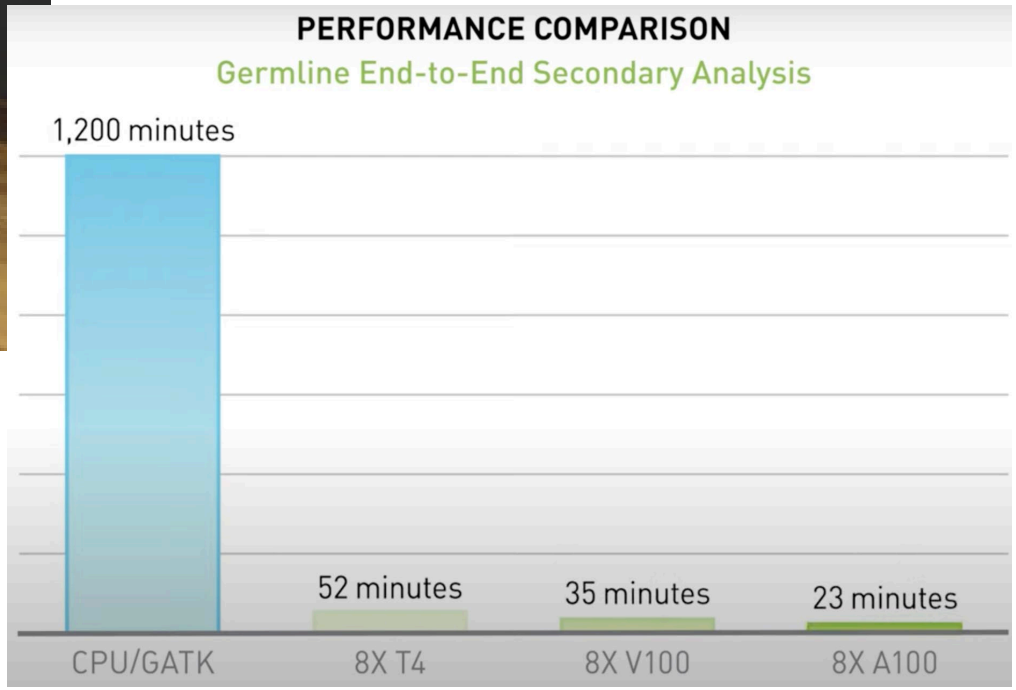


emea.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html
emea.illumina.com/company/news-center/press-releases/2018/2349147.html

NVIDIA Clara Parabricks (2020)



A University of Michigan startup in 2018 joined NVIDIA in 2020



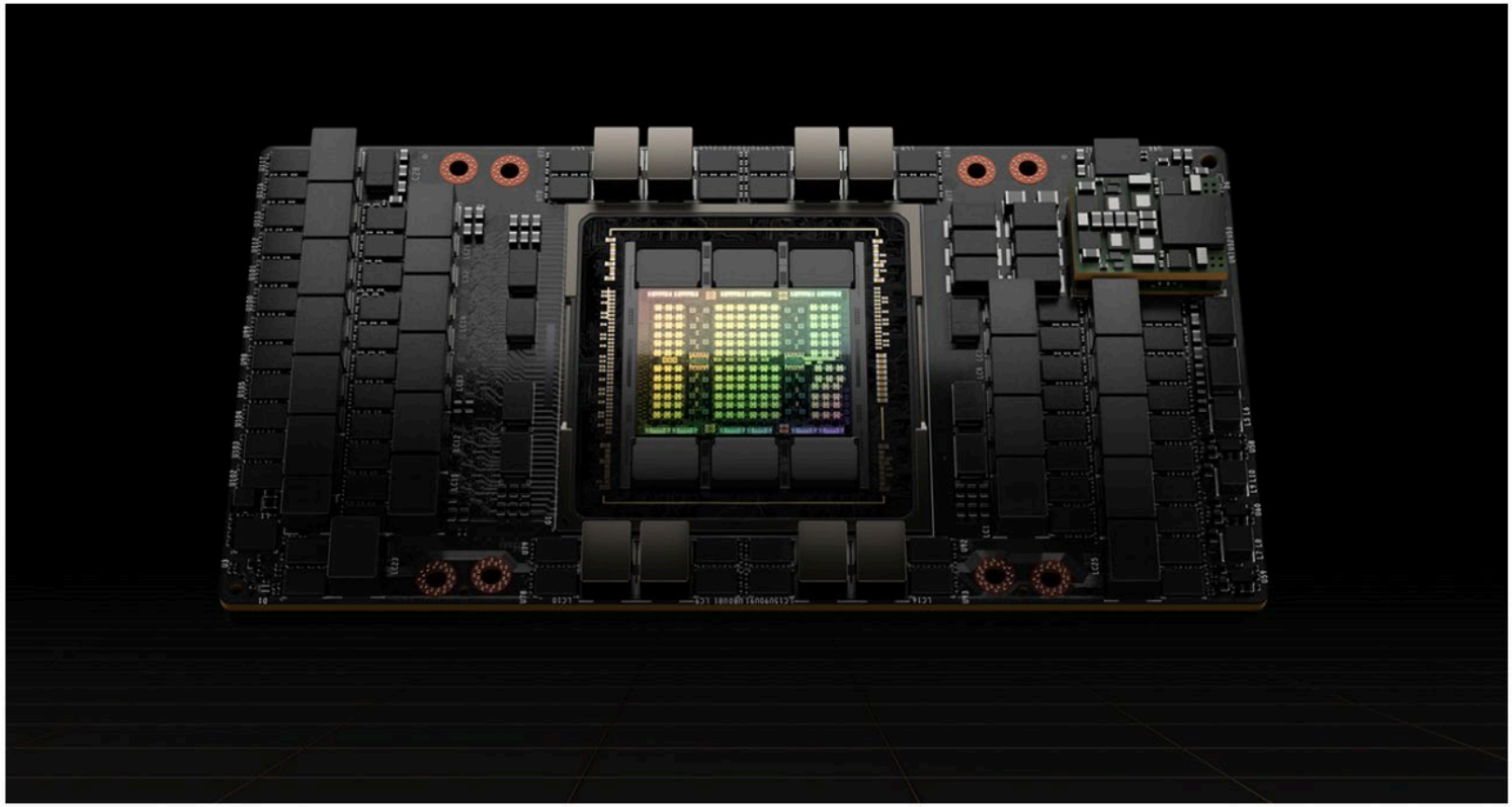
https://developer.nvidia.com/clara-parabricks

NVIDIA Hopper DPX Instructions (2022)

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/

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" IEEE Micro, August 2020.

MinION from ONT

Accelerating Genome Analysis: A Primer on an Ongoing Journey
Sept.-Oct. 2020, pp. 65-75, vol. 40
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
July-Aug. 2021, pp. 39-48, vol. 41
DOI Bookmark: 10.1109/MM.2021.3088396

SmidgION from ONT

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"
  IEEE Micro (*IEEE MICRO*), Vol. 40, No. 5, pages 65-75, September/October 2020.
  [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

PIM Review and Open Problems

A Modern Primer on Processing in Memory

Onur Mutlu[a,b], Saugata Ghose[b,c], Juan Gómez-Luna[a], Rachata Ausavarungnirun[d]

SAFARI Research Group

[a] *ETH Zürich*
[b] *Carnegie Mellon University*
[c] *University of Illinois at Urbana-Champaign*
[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"
Invited Book Chapter in Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann, Springer, to be published in 2021.

https://arxiv.org/pdf/2012.03112.pdf

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*

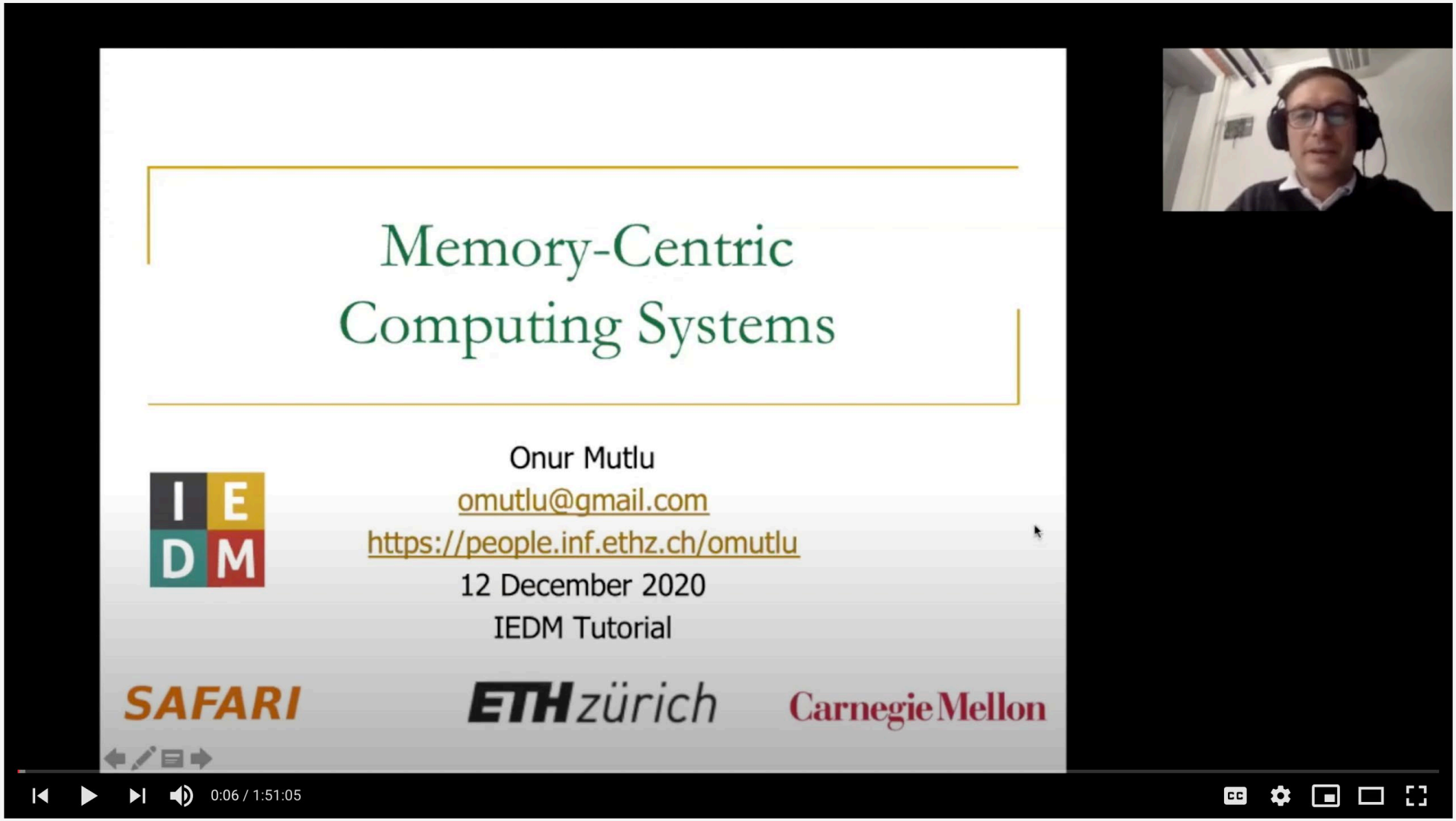
Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu,
"Processing-in-Memory: A Workload-Driven Perspective"
Invited Article in IBM Journal of Research & Development, Special Issue on Hardware for Artificial Intelligence, to appear in November 2019.
[Preliminary arXiv version]

https://arxiv.org/pdf/1907.12947.pdf

More on Memory-Centric System Design

- Onur Mutlu,
  "Memory-Centric Computing Systems"
  Invited Tutorial at *66th International Electron Devices Meeting (IEDM)*, Virtual, 12 December 2020.
  [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



IEDM 2020 Tutorial: Memory-Centric Computing Systems, Onur Mutlu, 12 December 2020

1,641 views • Dec 23, 2020

48 0 SHARE SAVE

Onur Mutlu Lectures
13.9K subscribers

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

ANALYTICS EDIT VIDEO

https://www.youtube.com/onurmutlulectures

Special Research Sessions & Courses

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

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

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"
IEEE Micro, 2021.
[Source Code]

Home / Magazines / IEEE Micro / 2021.04

IEEE Micro

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

July-Aug. 2021, pp. 39-48, vol. 41
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

https://arxiv.org/pdf/2106.06433.pdf

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]

ELSEVIER

COMPUTATIONAL ANDSTRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

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 *

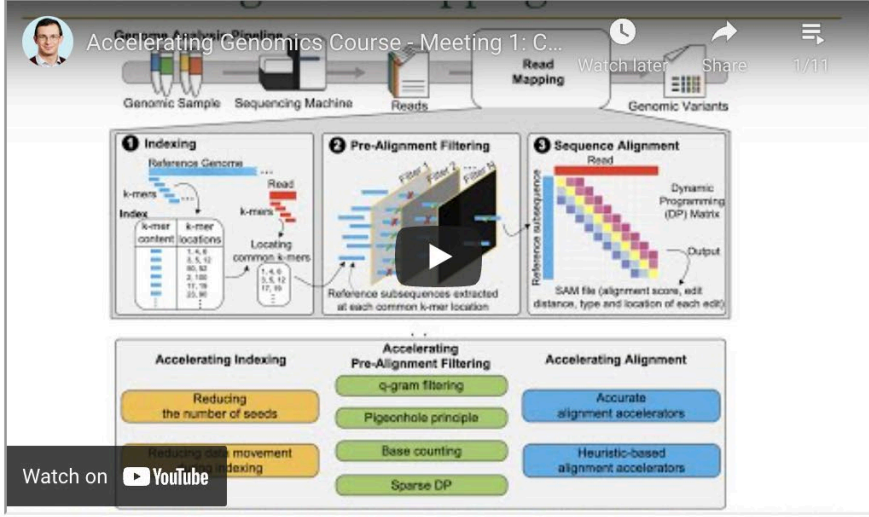
ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland

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
- Computer Architecture, Fall 2020, Lecture 8
  - **Intelligent Genome Analysis** (ETH Zürich, Fall 2020)
  - 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)
  - 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

Genomics (Spring 2022)

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics
- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18
- Project course
  - Taken by Bachelor's/Master's students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings

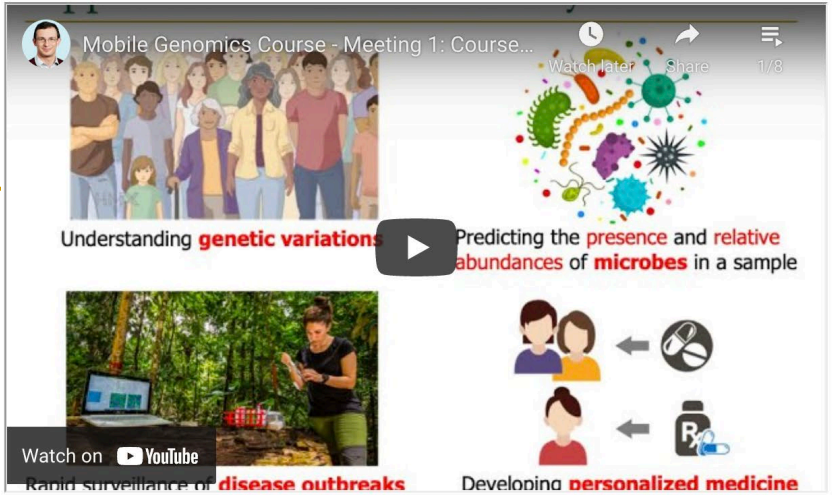


Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	11.3 Fri.	YouTube Live	M1: P&S Accelerating Genomics Course Introduction & Project Proposals (PDF) (PPT)	Required Materials Recommended Materials	
W2	18.3 Fri.	YouTube Live	M2: Introduction to Sequencing (PDF) (PPT)		
W3	25.3 Fri.	YouTube Premiere	M3: Read Mapping (PDF) (PPT)		
W4	01.04 Fri.	YouTube Premiere	M4: GateKeeper (PDF) (PPT)		
W5	08.04 Fri.	YouTube Premiere	M5: MAGNET & Shouji (PDF) (PPT)		
W6	15.4 Fri.	YouTube Premiere	M6: SneakySnake (PDF) (PPT)		
W7	29.4 Fri.	YouTube Premiere	M7: GenStore (PDF) (PPT)		
W8	06.05 Fri.	YouTube Premiere	M8: GRIM-Filter (PDF) (PPT)		
W9	13.05 Fri.	YouTube Premiere	M9: Genome Assembly (PDF) (PPT)		
W10	20.05 Fri.	YouTube Live	M10: Genomic Data Sharing Under Differential Privacy (PDF) (PPT)		
W11	10.06 Fri.	YouTube Premiere	M11: Accelerating Genome Sequence Analysis (PDF) (PPT)		

Genomics (Fall 2021)

- **Fall 2021 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2021/doku.php?id=bioinformatics
- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=MnogTeMjY8k&list=PL5Q2soXY2Zi8sngH-TrNZnDhDkPq55J9J
- Project course
  - Taken by Bachelor's/Master's students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings



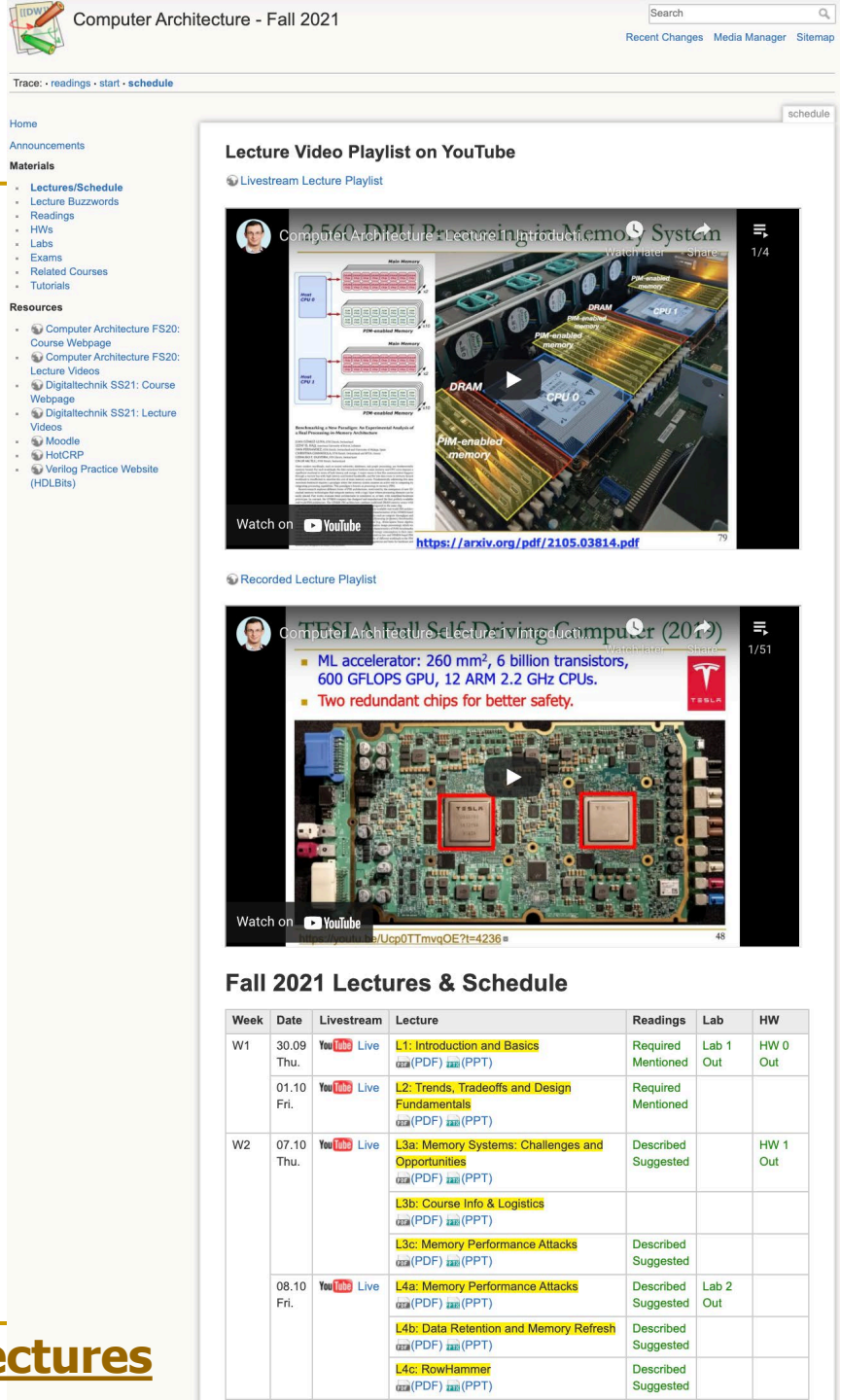
Fall 2021 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	5.10 Tue.	YouTube Live	M1: P&S Accelerating Genomics Course Introduction & Project Proposals (PDF) (PPT) YouTube Video	Required Materials Recommended Materials	
W2	20.10 Wed.	YouTube Live	M2: Introduction to Sequencing (PDF) (PPT)		
W3	27.10 Wed.	YouTube Live	M3: Read Mapping (PDF) (PPT)		
W4	3.11 Wed.	YouTube Live	M4: GateKeeper (PDF) (PPT)		
W5	10.11 Wed.	YouTube Live	M5: MAGNET & Shouji (PDF) (PPT)		
W6	17.11 Wed.		M6.1: SneakySnake (PDF) (PPT) Video		
			M6.2: GRIM-Filter (PDF) (PPT) YouTube Video		
W7	24.11 Wed.		M7: GenASM (PDF) (PPT) YouTube Video		
W8	01.12 Wed.	YouTube Live	M8: Genome Assembly (PDF) (PPT)		
W9	13.12 Mon.	YouTube Live	M9: GRIM-Filter (PDF) (PPT)		
W10	15.12 Wed.	YouTube Live	M10: Genomic Data Sharing Under Differential Privacy (PDF) (PPT)		

Comp Arch (Fall'21)

- **Fall 2021 Edition:**
  - https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule
- **Fall 2020 Edition:**
  - https://safari.ethz.ch/architecture/fall2020/doku.php?id=schedule
- **Youtube Livestream (2021):**
  - 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
- 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

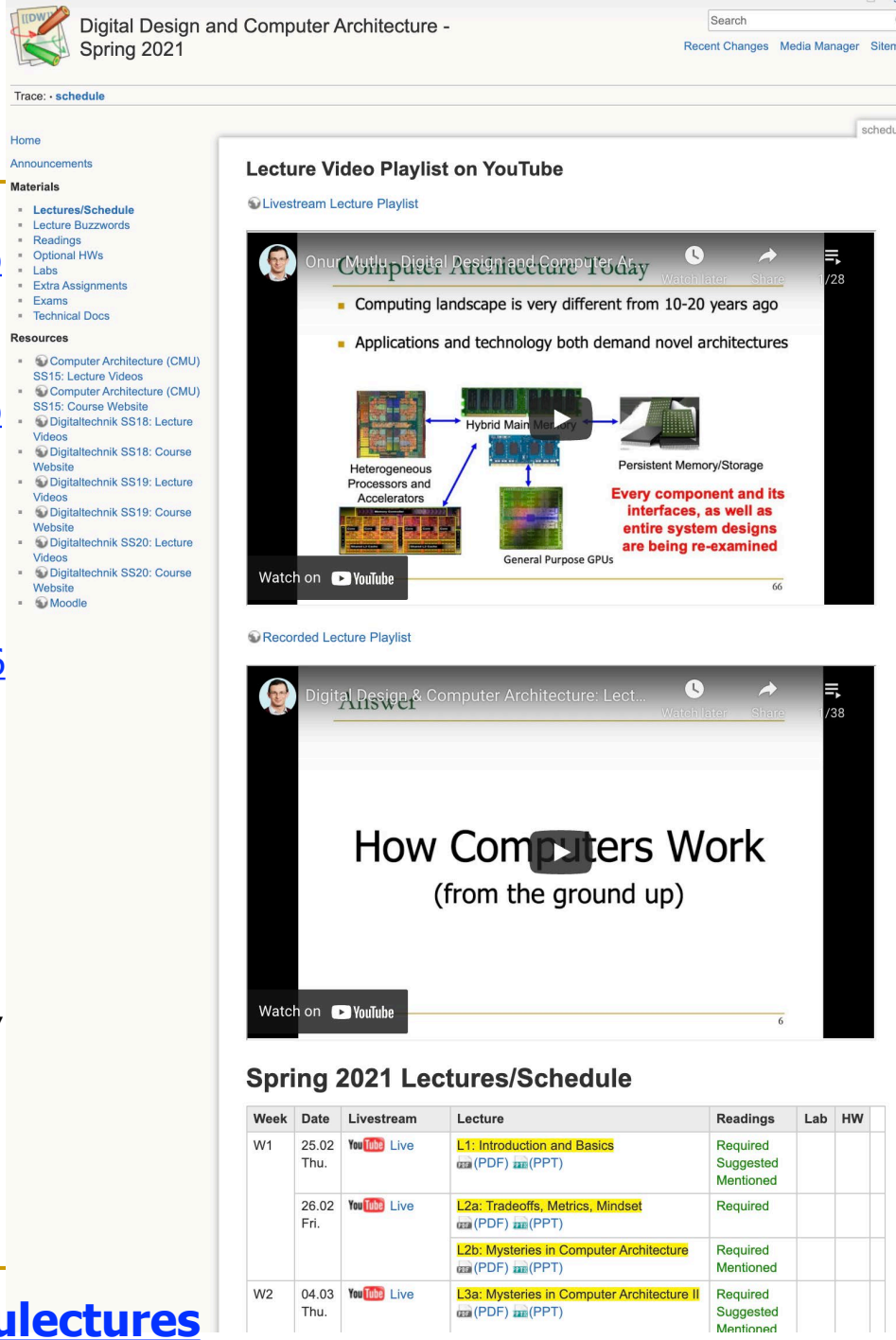
https://www.youtube.com/onurmutlulectures



DDCA (Spring 2022)

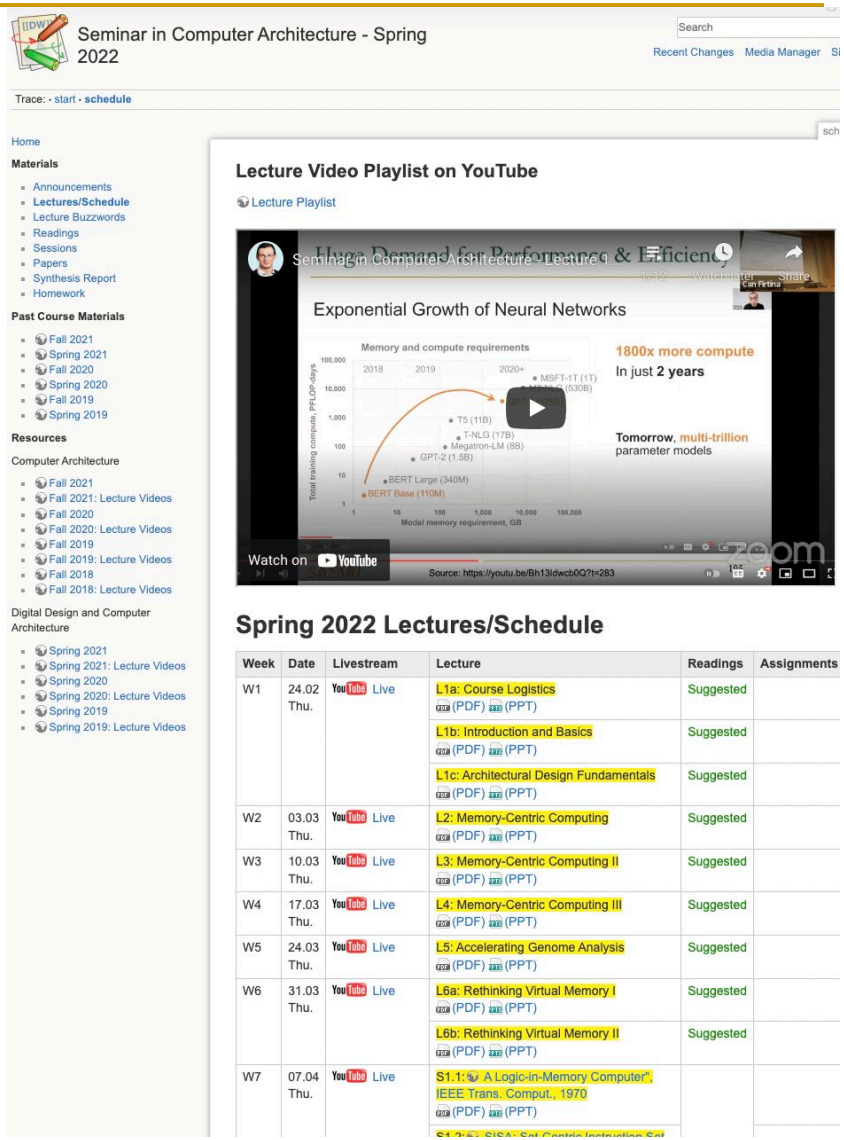
- **Spring 2022 Edition:**
  - https://safari.ethz.ch/digitaltechnik/spring2022/doku.php?id=schedule
- **Spring 2021 Edition:**
  - https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=schedule
- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=cpXdE3HwvK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6
- **Youtube Livestream (Spring 2021):**
  - https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LlN
- 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



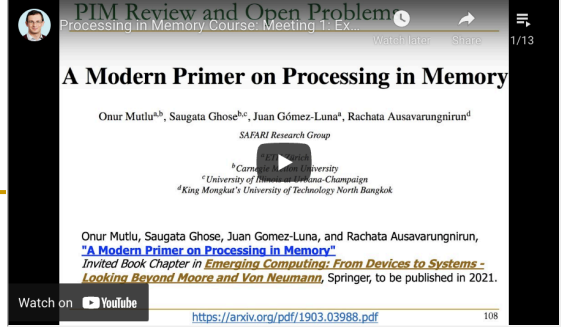
Seminar in Comp Arch (Spring & Fall)

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/architecture_seminar/spring2022/doku.php?id=schedule
- **Fall 2021 Edition:**
  - 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
- **Youtube Livestream (Fall 2021):**
  - 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=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX
- Project course
  - Taken by Bachelor's/Master's students
  - Processing-in-Memory lectures
  - Hands-on research exploration
  - Many research readings

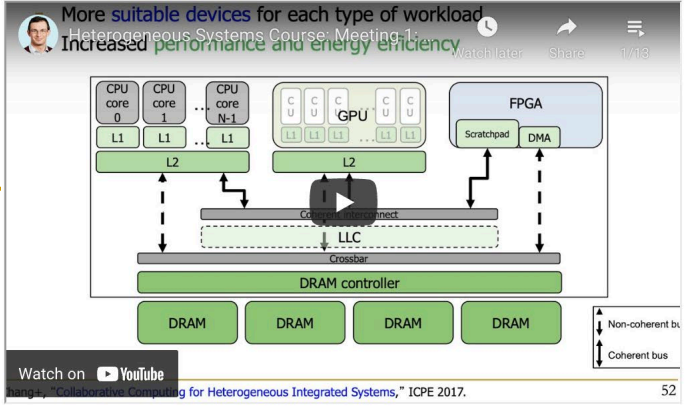


Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	10.03 Thu.	Live	M1: P&S PIM Course Presentation (PDF) (PPT)	Required Materials Recommended Materials	HW 0 Out
W2	15.03 Tue.		Hands-on Project Proposals		
	17.03 Thu.	Premiere	M2: Real-world PIM: UPMEM PIM (PDF) (PPT)		
W3	24.03 Thu.	Live	M3: Real-world PIM: Microbenchmarking of UPMEM PIM (PDF) (PPT)		
W4	31.03 Thu.	Live	M4: Real-world PIM: Samsung HBM-PIM (PDF) (PPT)		
W5	07.04 Thu.	Live	M5: How to Evaluate Data Movement Bottlenecks (PDF) (PPT)		
W6	14.04 Thu.	Live	M6: Real-world PIM: SK Hynix AiM (PDF) (PPT)		
W7	21.04 Thu.	Premiere	M7: Programming PIM Architectures (PDF) (PPT)		
W8	28.04 Thu.	Premiere	M8: Benchmarking and Workload Suitability on PIM (PDF) (PPT)		
W9	05.05 Thu.	Premiere	M9: Real-world PIM: Samsung AxDIMM (PDF) (PPT)		
W10	12.05 Thu.	Premiere	M10: Real-world PIM: Alibaba HB-PNM (PDF) (PPT)		
W11	19.05 Thu.	Live	M11: SpMV on a Real PIM Architecture (PDF) (PPT)		
W12	26.05 Thu.	Live	M12: End-to-End Framework for Processing-using-Memory (PDF) (PPT)		
W13	02.06 Thu.	Live	M13: Bit-Serial SIMD Processing using DRAM (PDF) (PPT)		
W14	09.06 Thu.	Live	M14: Analyzing and Mitigating ML Inference Bottlenecks (PDF) (PPT)		
W15	15.06 Thu.	Live	M15: In-Memory HTAP Databases with HW/SW Co-design (PDF) (PPT)		
W16	23.06 Thu.	Live	M16: In-Storage Processing for Genome Analysis (PDF) (PPT)		
W17	18.07 Mon.	Premiere	M17: How to Enable the Adoption of PIM? (PDF) (PPT)		
W18	09.08 Tue.	Premiere	SS1: ISVLSI 2022 Special Session on PIM (PDF & PPT)		

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=oFO5fTrgFIY&list=PL5Q2soXY2Zi9XrgXR38IM_FTjmY6h7Gzm
- Project course
  - Taken by Bachelor's/Master's students
  - GPU and Parallelism lectures
  - Hands-on research exploration
  - Many research readings

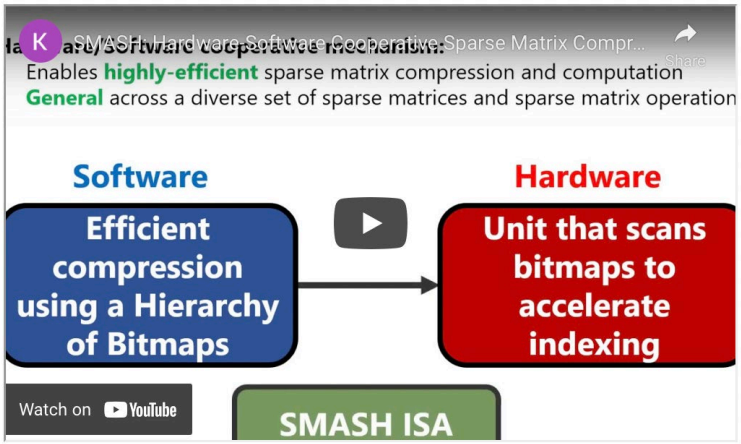


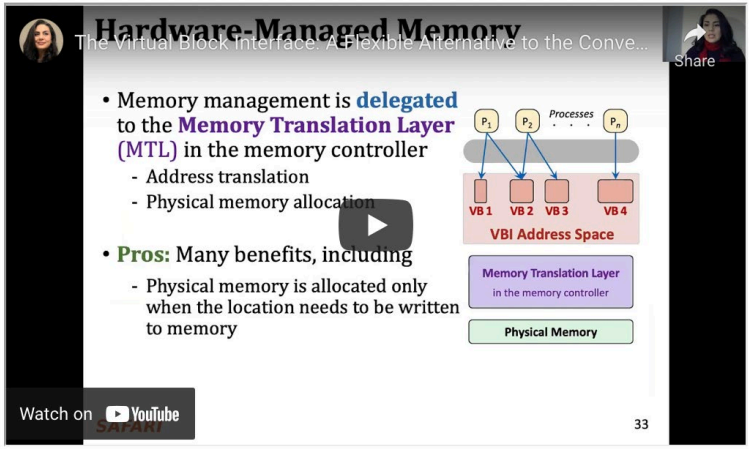
Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	15.03 Tue.	Premiere	M1: P&S Course Presentation (PDF) (PPT)	Required Materials Recommended Materials	HW 0 Out
W2	22.03 Tue.	Premiere	M2: SIMD Processing and GPUs (PDF) (PPT)		
W3	29.03 Tue.	Premiere	M3: GPU Software Hierarchy (PDF) (PPT)		
W4	05.04 Tue.	Premiere	M4: GPU Memory Hierarchy (PDF) (PPT)		
W5	12.04 Tue.	Premiere	M5: GPU Performance Considerations (PDF) (PPT)		
W6	19.04 Tue.	Premiere	M6: Parallel Patterns: Reduction (PDF) (PPT)		
W7	26.04 Tue.	Premiere	M7: Parallel Patterns: Histogram (PDF) (PPT)		
W8	03.05 Tue.	Premiere	M8: Parallel Patterns: Convolution (PDF) (PPT)		
W9	10.05 Tue.	Premiere	M9: Parallel Patterns: Prefix Sum (Scan) (PDF) (PPT)		
W10	17.05 Tue.	Premiere	M10: Parallel Patterns: Sparse Matrices (PDF) (PPT)		
W11	24.05 Tue.	Premiere	M11: Parallel Patterns: Graph Search (PDF) (PPT)		
W12	01.06 Wed.	Premiere	M12: Parallel Patterns: Merge Sort (PDF) (PPT)		
W13	07.06 Tue.	Premiere	M13: Dynamic Parallelism (PDF) (PPT)		
W14	15.06 Wed.	Premiere	M14: Collaborative Computing (PDF) (PPT)		
W15	24.06 Fri.	Premiere	M15: GPU Acceleration of Genome Sequence Alignment (PDF) (PPT)		
W16	14.07 Thu.	Premiere	M16: Accelerating Agent-based Simulations (PDF) (ODP)		

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





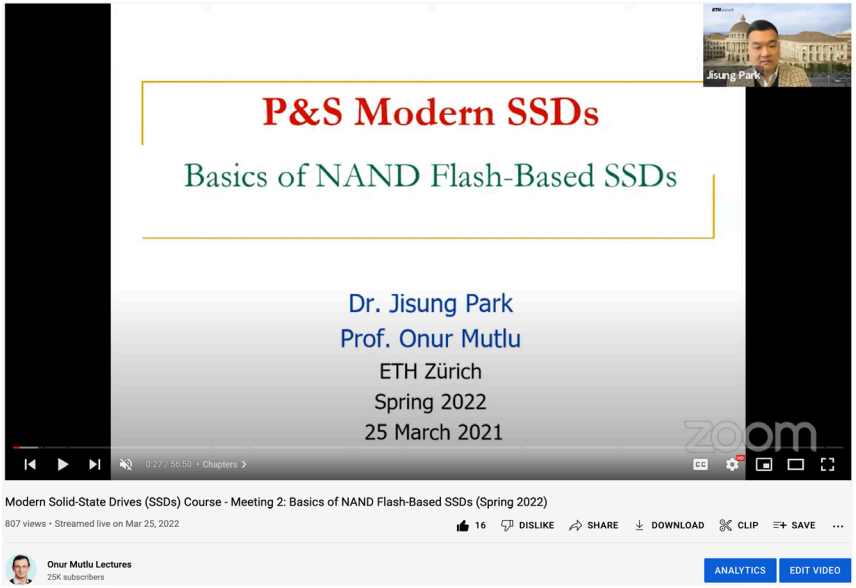
Edit

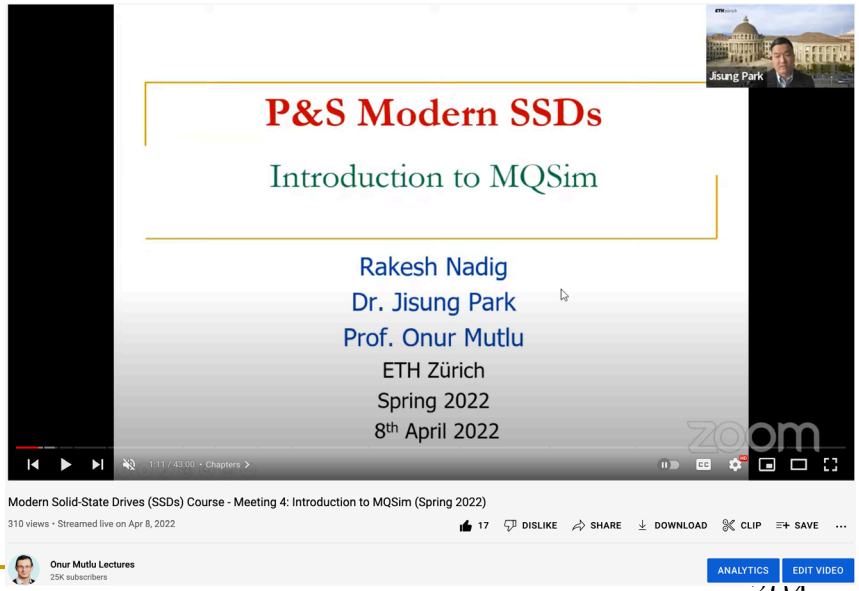
2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Materials	Assignments
W0	16.03	YouTube Live	Intro to HW/SW Co-Design (PPTX) (PDF)	Required	HW 0 Out
W1	23.03		Project selection	Required	
W2	30.03	YouTube Live	Virtual Memory (I) (PPTX) (PDF)		
W3	13.04	YouTube Live	Virtual Memory (II) (PPTX) (PDF)		

SSD Course (Spring 2022)

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=modern_ssds
- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=_q4rm71DsY4&list=PL5Q2soXY2Zi8vabcse1kL22DEcgMl2RAq
- 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

SAFARI Newsletter April 2020 Edition

- https://safari.ethz.ch/safari-newsletter-april-2020/

SAFARI
SAFARI Research Group

View in your browser

Think Big, Aim High

Dear SAFARI friends,

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

SAFARI Newsletter January 2021 Edition

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

SAFARI
SAFARI Research Group

Newsletter
January 2021

Think Big, Aim High, and Have a Wonderful 2021!

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/

SAFARI
SAFARI Research Group

Think Big, Aim High

ETH zürich

View in your browser
December 2021

Referenced Papers, Talks, Artifacts

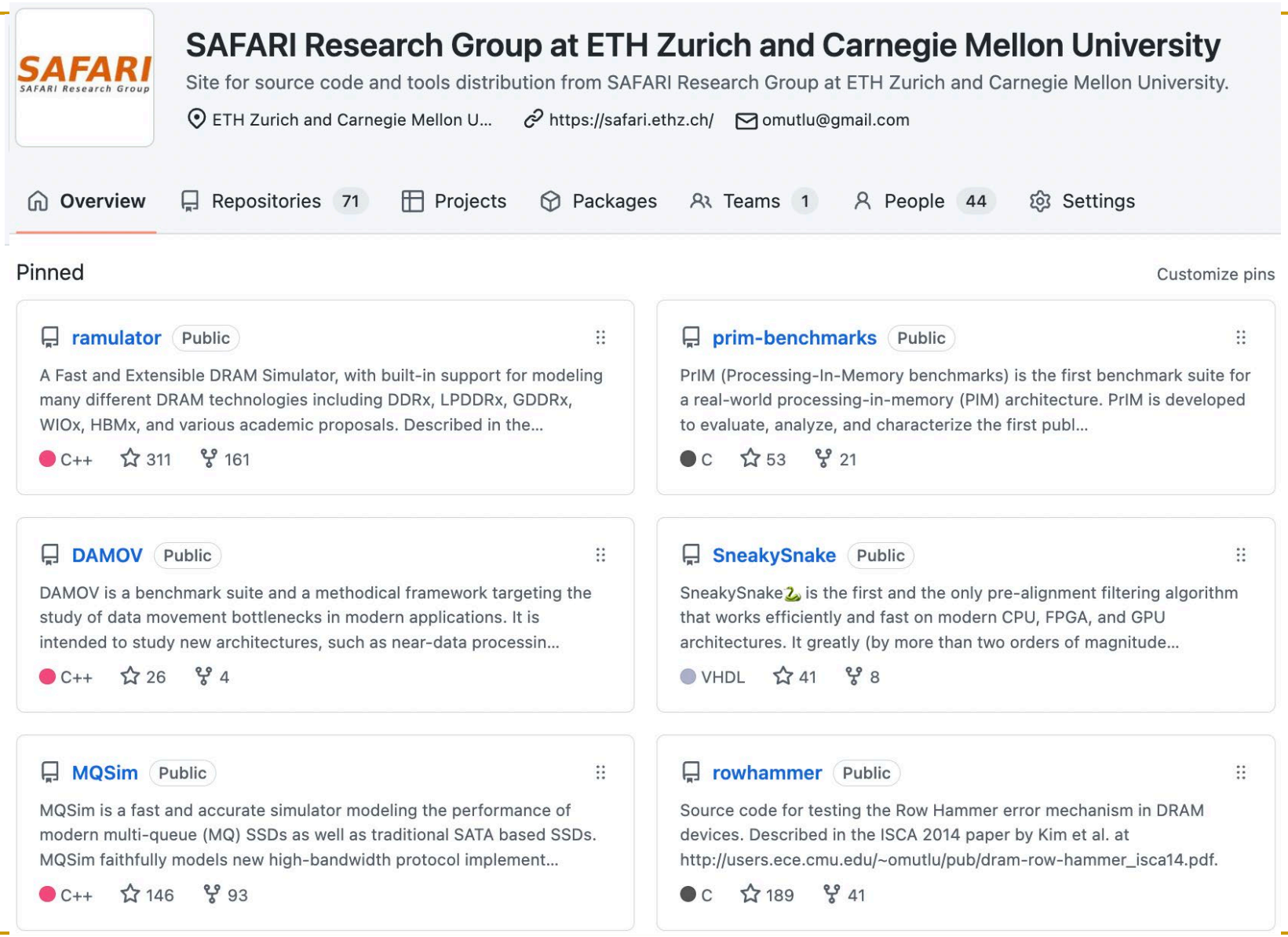
- All are available at

  https://people.inf.ethz.ch/omutlu/projects.htm

  https://www.youtube.com/onurmutlulectures

  https://github.com/CMU-SAFARI/

Open Source Tools: SAFARI GitHub



https://github.com/CMU-SAFARI/

Accelerating Genome Analysis

A Primer on an Ongoing Journey

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
6 September 2022
Barcelona Supercomputing Center

SAFARI ETH zürich Carnegie Mellon

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"
  Proceedings of the *55th International Symposium on Microarchitecture (MICRO)*, Chicago, Illinois, October 2022.

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*

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"
  Preprint in *arXiv*, 2021.
  [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-Ghiasi[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

- Can Firtina, Kamlesh Pillai, Gurpreet S. Kalsi, Bharathwaj Suresh, Damla Senol Cali, Jeremie S. Kim, Taha Shahroodi, Meryem Banu Cavlak, Joel Lindegger, Mohammed Alser, Juan Gómez-Luna, Sreenivas Subramoney, and Onur Mutlu, "ApHMM: A Profile Hidden Markov Model Acceleration Framework for Genome Analysis"
  Preprint in *arXiv*, 2022.
  [Source Code]

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

Can Firtina[1] Kamlesh Pillai[2] Gurpreet S. Kalsi[2] Bharathwaj Suresh[2] Damla Senol Cali[3]
Jeremie S. Kim[1] Taha Shahroodi[4] Meryem Banu Cavlak[1] Joel Lindegger[1] Mohammed Alser[1]
Juan Gómez Luna[1] Sreenivas Subramoney[2] Onur Mutlu[1]
[1]*ETH Zurich* [2]*Intel Labs* [3]*Bionano Genomics* [4]*TU 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"
  Preprint in arXiv and bioRxiv, 2021.
  [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[1†], Can Firtina[1†], Meryem Banu Cavlak[2], Damla Senol Cali[3], Nastaran Hajinazar[1,4], Mohammed Alser[1], Can Alkan[2] and Onur Mutlu[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*

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]

frontiers | Frontiers in Public Health

ORIGINAL RESEARCH
published: 17 June 2022
doi: 10.3389/fpubh.2022.877621

COVIDHunter: COVID-19 Pandemic Wave Prediction and Mitigation via Seasonality Aware Modeling

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

Department of Information Technology and Electrical Engineering (D-ITET), ETH Zurich, Zurich, Switzerland

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

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

Mohammed Alser[1], Sharon Waymost[2], Ram Ayyala[3,4], Brendan Lawlor[5], Richard J. Abdill[6], Neha Rajkumar[7], Nathan LaPierre[2], Jaqueline Brito[4], André M. Ribeiro-dos-Santos[8], Can Firtina[1], Nour Almadhoun[1], Varuni Sarwal[2], Eleazar Eskin[2,9,10], Qiyang Hu[11], Derek Strong[12], Byoung-Do (BD) Kim[12], Malak S. Abedalthagafi[13,14,15*], Onur Mutlu[1,*], Serghei Mangul[4,*]

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 SHAHROODI[1], MAHDI ZAHEDI[1], CAN FIRTINA[2], MOHAMMED ALSER[2], STEPHAN WONG[1], (Senior Member, IEEE), ONUR MUTLU[2], (Fellow, IEEE), AND SAID HAMDIOUI[1], (Senior Member, IEEE)
[1]Q&CE Department, EEMCS Faculty, Delft University of Technology (TU Delft), 2628 CD Delft, The Netherlands
[2]SAFARI Research Group, D-ITET, ETH Zürich, 8092 Zürich, Switzerland

AIM (PIM Sequence Alignment Framework)

Safaa Diab, Amir Nassereldine, Mohammed Alser, Juan Gómez-Luna, Onur Mutlu, Izzat El Hajj
"A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems"
arXiv, 2022
[Source code]

A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems

Safaa Diab[1], Amir Nassereldine[1], Mohammed Alser[2], Juan Gómez Luna[2], Onur Mutlu[2], Izzat El Hajj[1]

[1]*American University of Beirut, Lebanon* [2]*ETH 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]

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Original paper

OXFORD

Genome analysis

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

Joël Lindegger [1,*], Damla Senol Cali [2], Mohammed Alser [1], Juan Gómez-Luna [1], Nika Mansouri Ghiasi [1] and Onur Mutlu [1,*]

[1]Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland and
[2]Bionano 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]

journal homepage: www.elsevier.com/locate/csbj

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"
  Preprint in *arXiv*, 2022.

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

Safaa Diab[1], Amir Nassereldine[1], Mohammed Alser[2], Juan Gómez Luna[2], Onur Mutlu[2], Izzat El Hajj[1]
[1]*American University of Beirut, Lebanon* [2]*ETH Zürich, Switzerland*

Backup Slides for Further Info

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=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=12
- Computer Architecture, Fall 2020, Lecture 7
  - **Near-Data Processing** (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=j2GIigqn1Qw&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=13
- Computer Architecture, Fall 2020, Lecture 11a
  - **Memory Controllers** (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=TeG773OgiMQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-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=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=25

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=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-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=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-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=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=41

Genome Analysis

NO machine can read the ***entire*** content of a genome

>CCTCCTCAGTGCCACCCAGCCCACTGGCAGCTCCCAAACAGGCTCTTATTAAAACACCCTGTTCCCTGCCCCTTGGAGTGAGGTGTCAAG
GACCTAAACTAAAAAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTT
CATGTCAAGGACCTAATGTGCTAAACAGCACTTTTTTGACCATTATTTTGGATCTGAAAGAAATCAAGAATAAATGAAGGACTTGATACATTG
GAAGAGGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAA
ACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCTGTGTTGCAGGTCTTCTTGCATTTCCCTGTCAAAAGAAAAAGAATTTAAAATTT
AAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCAGGCCAAGAGTTGCAAAAAAAAAAAAAGAAAAA
GAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTAGCCAGAATGG
TTGTGGGATGGGAGCCTCTGTGGACCGACCAGGTAGCTCTCTTTTCCACACTGTAGTCTCAAAGCTTCTTCATGTGGTTTCTCTGAGTGAAA
AAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTTTTCATGTCAAGGACC
TAATGTAGCTATACTGAACGTTATCTAGGGGAAAGATTGAAGGGGAGCTCTAAGGTCAACACACCACCACTTCCCAGAAAGCTTCTTCA.....

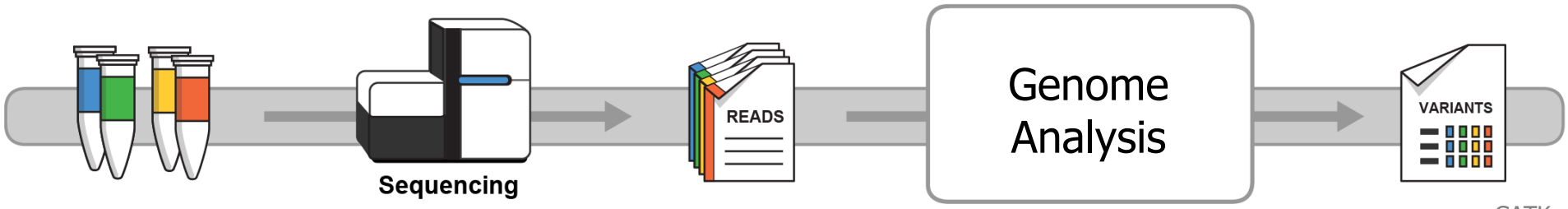
Genome Analysis

NO machine can read the ***entire*** content of a genome

Why?

>CCT…CAAG
GACC…TCTT
CATGT…CATTG
GAAG…AAAA
ACTA…ATTT
AAGT…AAAA
GAAA…ATGG
TTGTC…GAAA
AAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTTTTCATGTCAAGGACC
TAATGTAGCTATACTGAACGTTATCTAGGGGAAAGATTGAAGGGGAGCTCTAAGGTCAACACACCACCACTTCCCAGAAAGCTTCTTCA……

Genome Sequencer is a Chopper



GATK

CCCCCCTATATATACGTACTAGTACGT
ACGACTTTAGTACGTACGT
TATATATACGTACTAGTACGT
ACGTACG CCCCTACGTA
TATATATACGTACTAGTACGT
ACGACTTTAGTACGTACGT
TATATATACGTACTAAAGTACGT
TATATATACGTACTAGTACGT
ACG TTTTTAAAACGTA
TATATATACGTACTAGTACGT
ACGAC GGGGAGTACGTACGT

1x10^{12} bases*

44 hours*

<1000 $

* NovaSeq 6000

Oxford Nanopore Sequencers

MinION Mk1B

MinION Mk1C

GridION Mk1

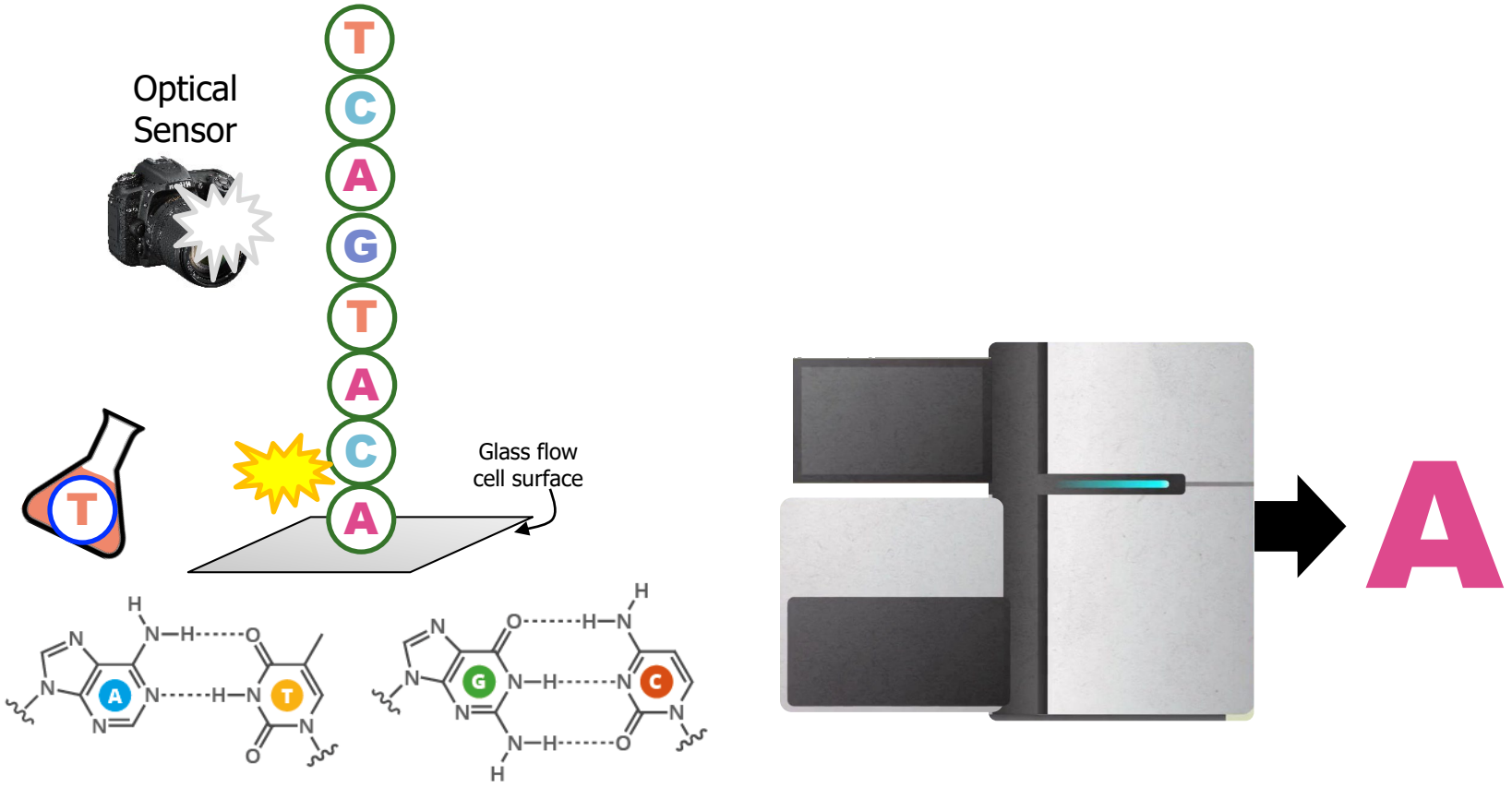
PromethION 24/48

	MinION Mk1B	MinION Mk1C	GridION Mk1	PromethION 24	PromethION 48
Read length	> 2Mb	> 2Mb	> 2Mb	> 2Mb	> 2Mb
Yield per flow cell	50 Gb	50 Gb	50 Gb	220 Gb	220 Gb
Number of flow cells per device	1	1	5	24	48
Yield per device	<50 Gb	<50 Gb	<250 Gb	<5.2 Tb	<10.5 Tb
Starting price	$1,000	$4,990	$49,995	$195,455	$327,455

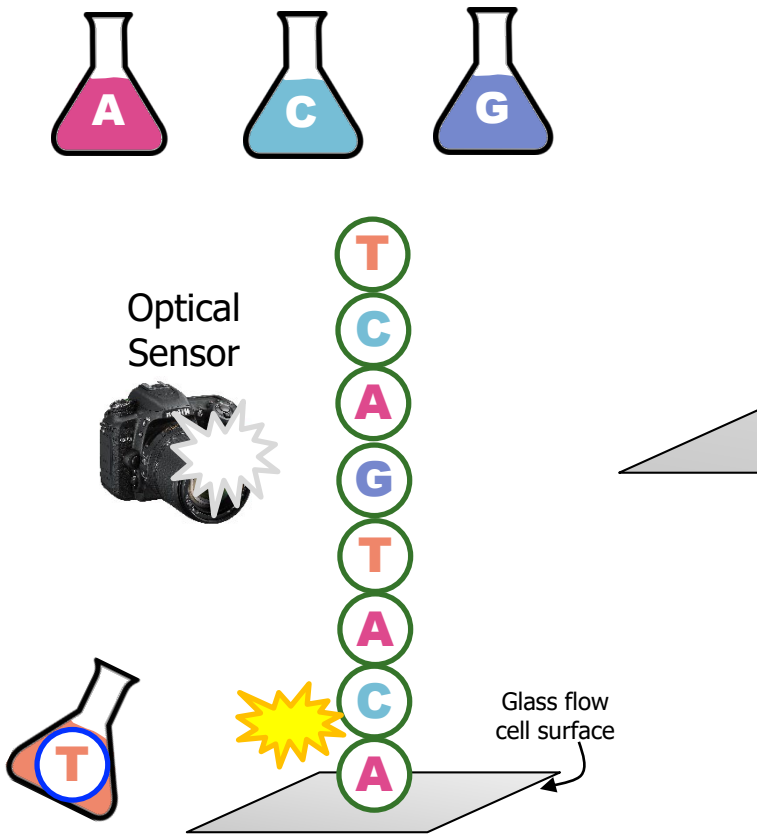
Illumina Sequencers

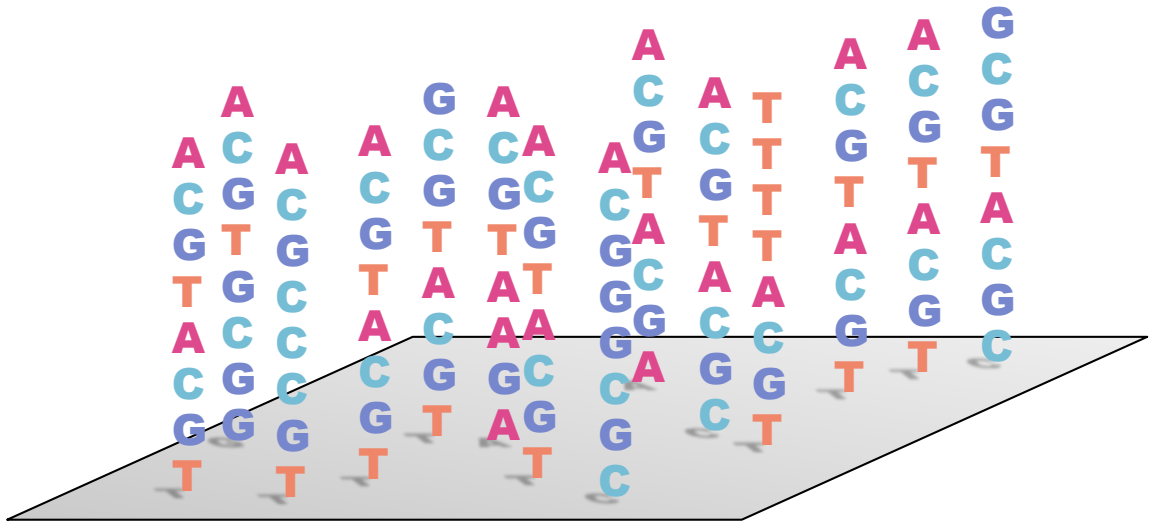
Run time	9.5–19 hrs	4–24 hrs	4–55 hrs	12–30 hrs	24-48 hrs	13-44 hrs
Max. reads per run	4 million	25 million	25 million	400 million	1 billion	20 billion
Max. read length	2 × 150 bp	2 × 150 bp	2 × 300 bp	2 × 150 bp	2 × 150 bp	2 x 250
Max. output	1.2 Gb	7.5 Gb	15 Gb	120 Gb	300 Gb	6000 Gb
Estimated price	$19,900	$49,500	$128,000	$275,000	$335,000	$985,000

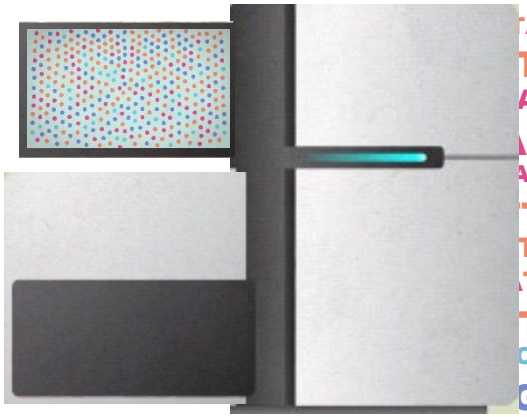
How Does Illumina Machine Work?



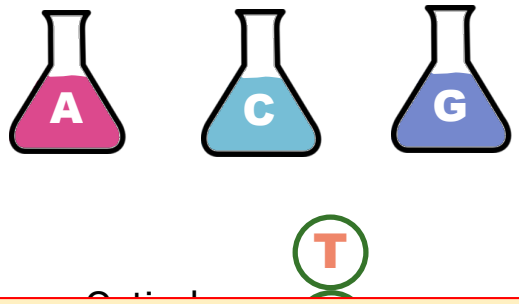
How Does Illumina Machine Work?





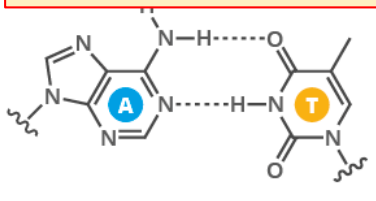


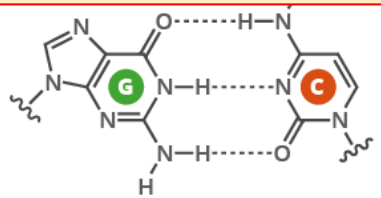
How Does Illumina Machine Work?



Check Illumina virtual tour:

https://emea.illumina.com/systems/sequencing-platforms/iseq/tour.html

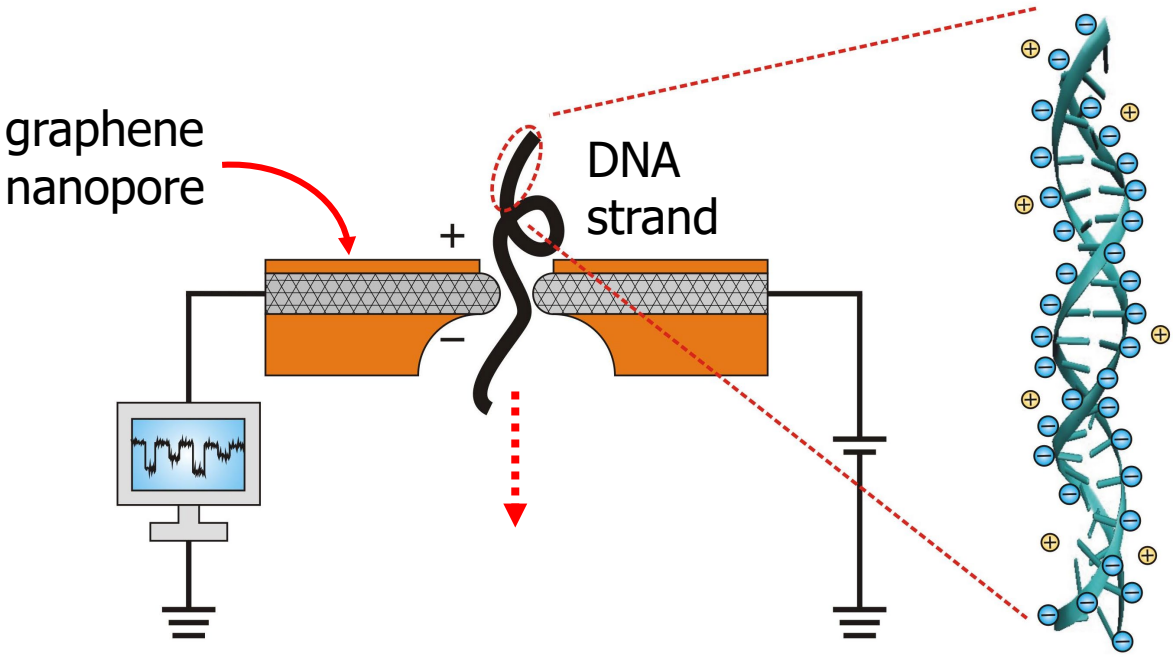




TTTAAAACGTA
CGTACTAGTACGT
GGGAGTACGTACGT

DNA fragment = Read

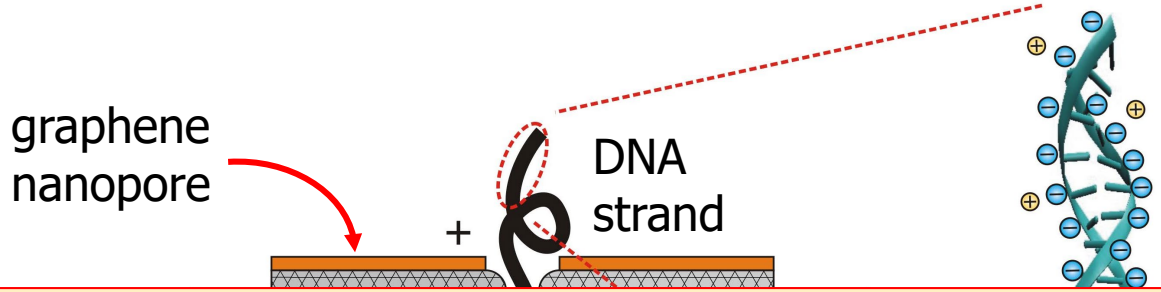
How Does Nanopore Machine Work?



- **Nanopore** is a nano-scale hole (<20nm).
- In nanopore sequencers, an **ionic current** passes through the nanopores
- When the DNA strand passes through the nanopore, the sequencer measures the the **change in current**
- This change is used to identify the bases in the strand with the help of **different electrochemical structures** of the different bases

Figure is adapted from: https://phys.org/news/2013-12-gene-sequencing-future.html

How Does Nanopore Machine Work?



Check Nanopore virtual tour:

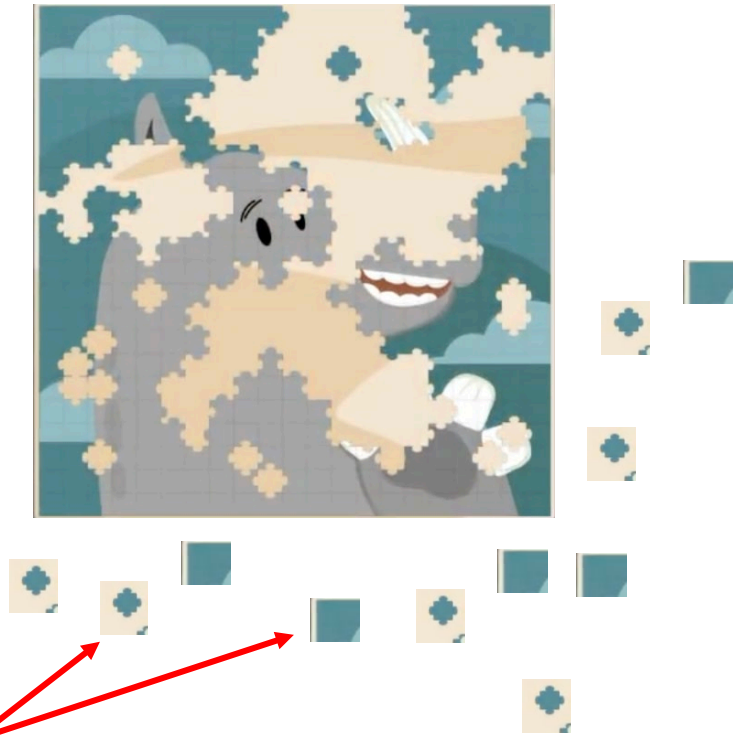
https://nanoporetech.com/resource-centre/minion-video

- measures the the **change in current**
- This change is used to identify the bases in the strand with the help of **different electrochemical structures** of the different bases

Figure is adapted from: https://phys.org/news/2013-12-gene-sequencing-future.html

Solving the Puzzle

Reference genome



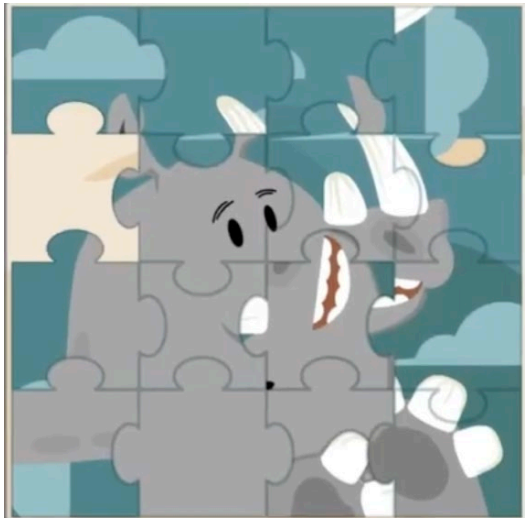
Reads

https://www.pacb.com/smrt-science/smrt-sequencing/hifi-reads-for-highly-accurate-long-read-sequencing/

HTS Sequencing Output

Small pieces of a puzzle
short reads (Illumina)

Large pieces of a puzzle
long reads (ONT & PacBio)

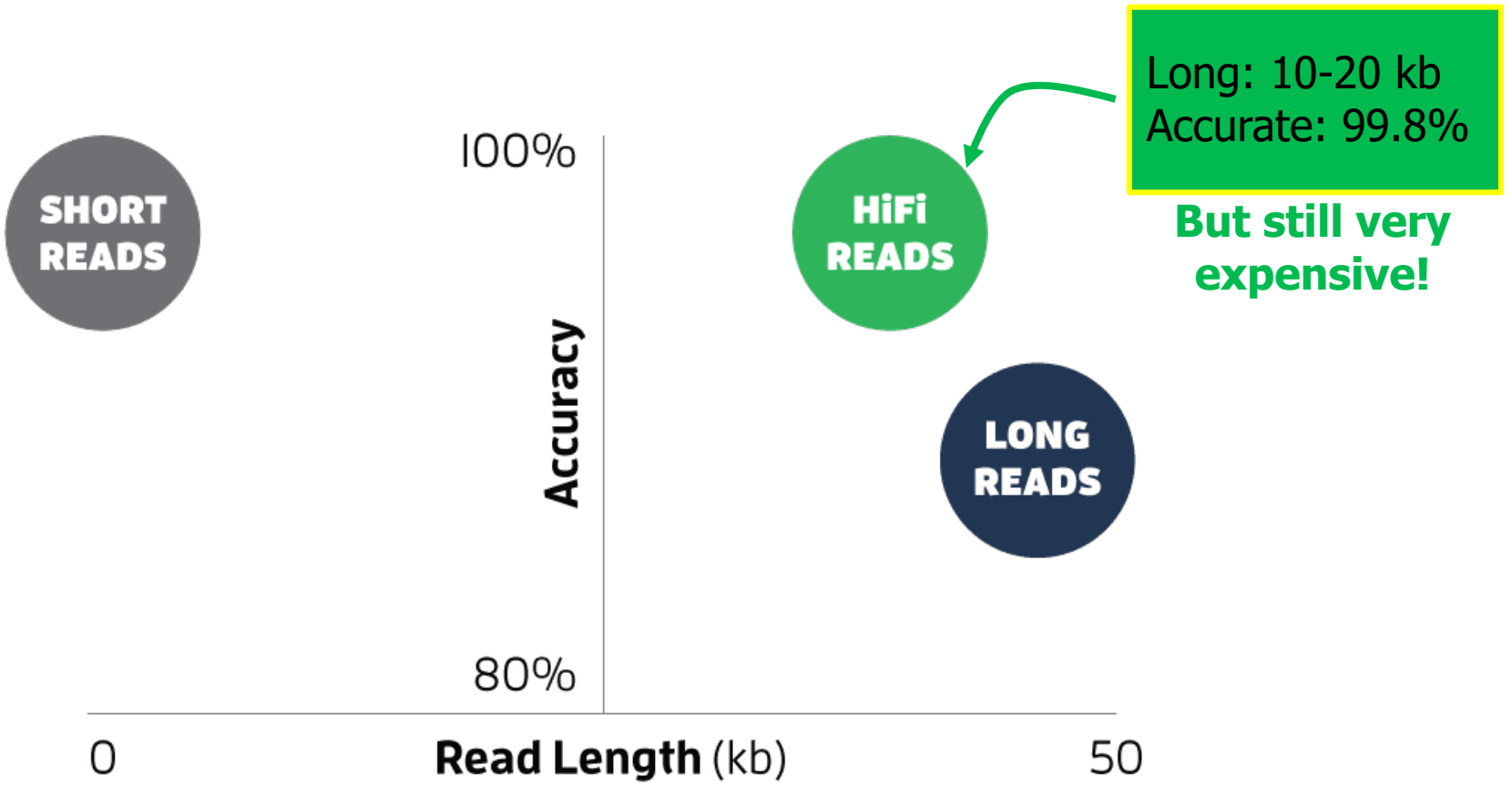


Which sequencing technology is the best?

- 100-300 bp
- low error rate (~0.1%)
- 500-2M bp
- high error rate (~15%)

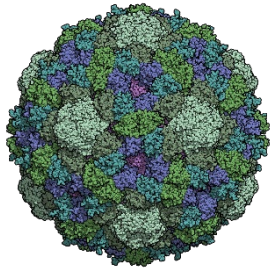
https://www.pacb.com/smrt-science/smrt-sequencing/hifi-reads-for-highly-accurate-long-read-sequencing/

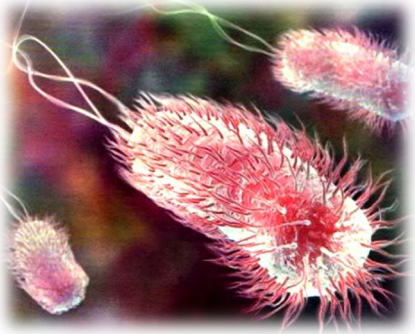
HiFi Reads (PacBio)

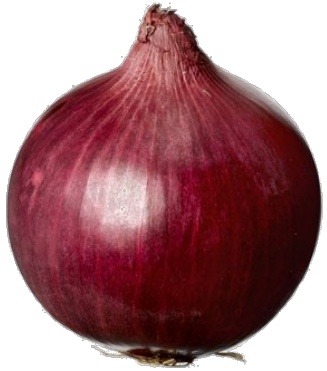


Wenger+, "Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome", *Nature Biotechnology*, 2019

How Long is DNA?







Phi X174 virus
5.386 Killo 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

Cracking the 1st Human Genome Sequence

- **1990-2003:** The Human Genome Project (HGP) provides a complete and accurate sequence of all **DNA base pairs** that make up the human genome and finds 20,000 to 25,000 human genes.

3.2×10^9 bases

13 years

$>3 \times 10^9$ $

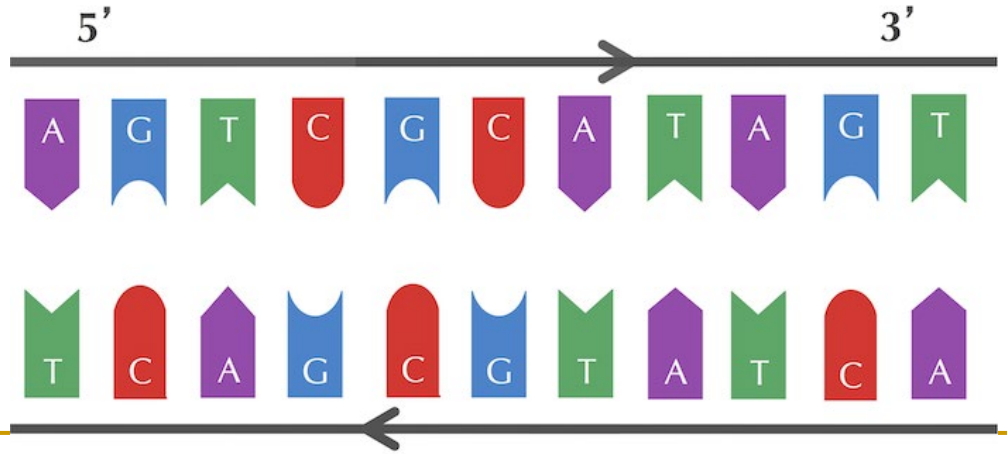
Obtaining the Human Reference Genome

- **GRCh38.p13**
- Description: Genome Reference Consortium Human Build 38 patch release 13 (GRCh38.p13)
- Organism name: Homo sapiens (human)
- Date: 2019/02/28
- 3,099,706,404 bases
- Compressed .fna file (964.9 MB)
- https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39

```
>NC_000001.11 Homo sapiens chromosome 1, GRCh38.p13 Primary Assembly
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

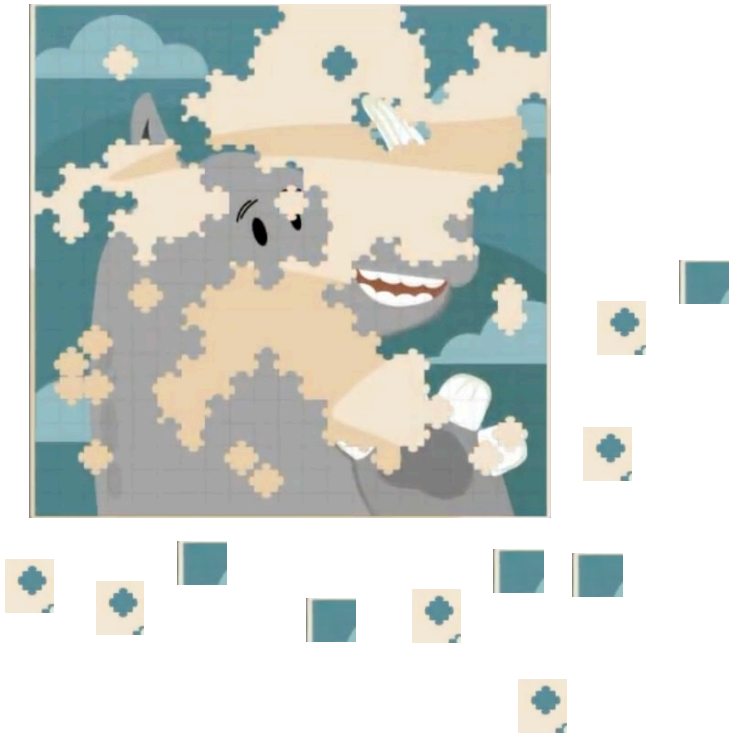
Challenges in Read Mapping

- Need to find many mappings of each read
- Need to tolerate variances/sequencing errors in each read
- Need to map each read very fast (i.e., performance is important, life critical in some cases)
- Need to map reads to both forward and reverse strands



https://www.bioinformaticsalgorithms.org/bioinformatics-chapter-1

Revisiting the Puzzle



https://www.pacb.com/smrt-science/smrt-sequencing/hifi-reads-for-highly-accurate-long-read-sequencing/

Reference Genome Bias

nature genetics

Letter | Open Access | Published: 19 November 2018

Assembly of a pan-genome from deep sequencing of 910 humans of African descent

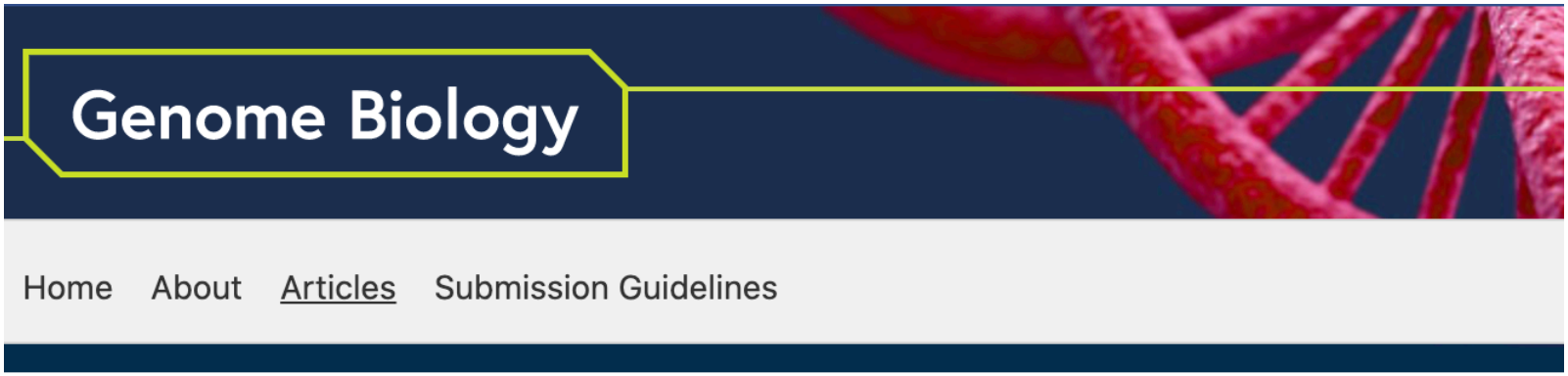
Rachel M. Sherman, Juliet Forman, [...] Steven L. Salzberg

Nature Genetics **51**, 30–35(2019) | Cite this article

"African pan-genome contains ~10% more DNA bases than the current human reference genome"

Sherman+, "Assembly of a pan-genome from deep sequencing of 910 humans of African descent" *Nature genetics*, 2019.

Time to Change the Reference Genome



Opinion | Open Access | Published: 09 August 2019

Is it time to change the reference genome?

Sara Ballouz, Alexander Dobin & Jesse A. Gillis

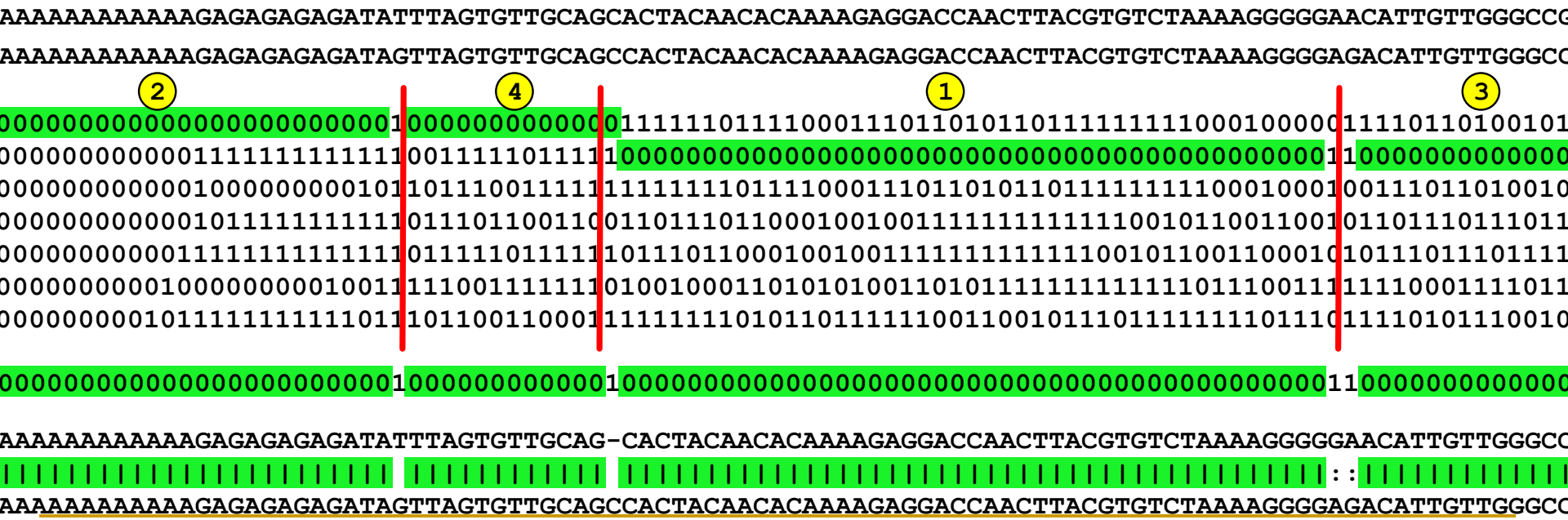
Genome Biology **20**, Article number: 159 (2019) | Cite this article

12k Accesses | **11** Citations | **45** Altmetric | Metrics

"Switching to a consensus reference would offer important advantages over the continued use of the current reference with few disadvantages"

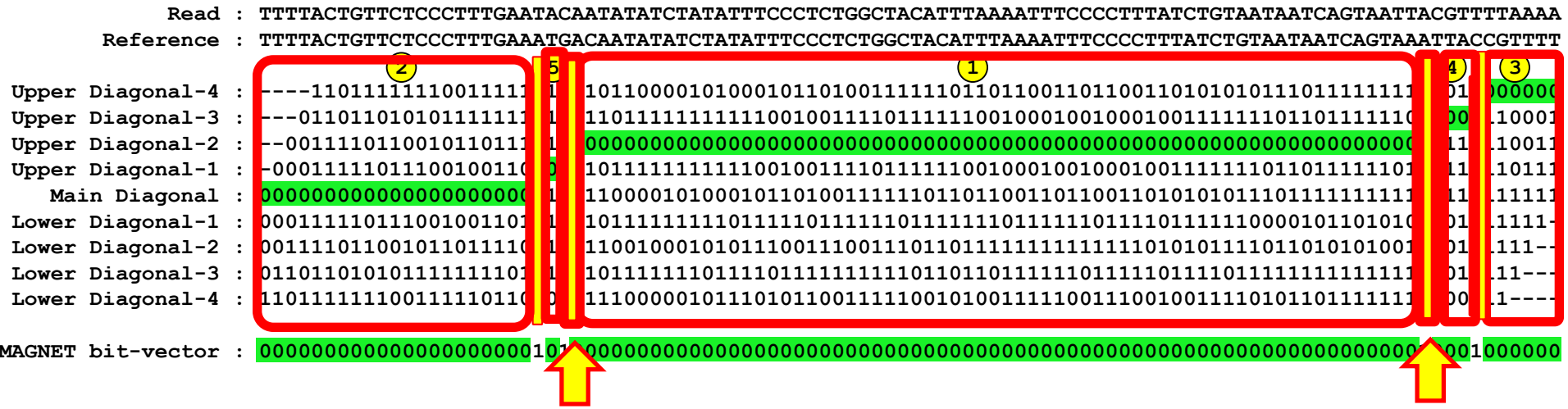
MAGNET (AACBB 2018, TIR 2017)

- Key observation: the use of **AND operation** to check if a zero (match) exists in a column introduces filtering inaccuracy.
- Key Idea: count the **consecutive zeros** in each mask and select the longest in a divide-and-conquer approach.
- **MAGNET** is **17x to 105x more accurate** than GateKeeper and SHD.



MAGNET Walkthrough

Build Neighborhood Map → Track the Diagonally Consecutive Matches → ACCEPT iff number of '1' ≤ Threshold



Find the longest segment of consecutive zeros

Exclude the errors from the search space

Divide the problem into two subproblems and repeat

"MAGNET: understanding and improving the accuracy of genome pre-alignment filtering", *arXiv preprint* 2017

What if we got a new version of the reference genome?

AirLift

- **Key observation:** Reference genomes are updated frequently. Repeating *read mapping is a computationally expensive workload*.

- **Key idea:** Update the mapping results of only affected reads depending on how a region in the old reference relates to another region in the new reference.

- **Key results:**
  - reduces number of reads that needs to be re-mapped to new reference by up to 99%
  - reduces overall runtime to re-map reads by 6.94x, 208x, and 16.4x for large (human), medium (C. elegans), and small (yeast) reference genomes

Clustering the Reference Genome Regions

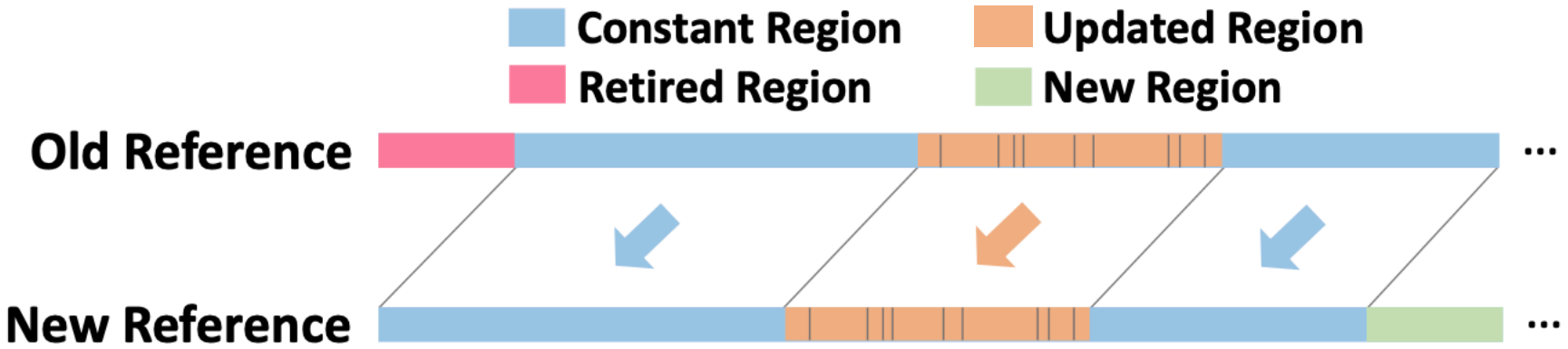


Fig. 2. Reference Genome Regions.

More Details on AirLift

Quantitative Biology > Genomics

[Submitted on 18 Dec 2019]

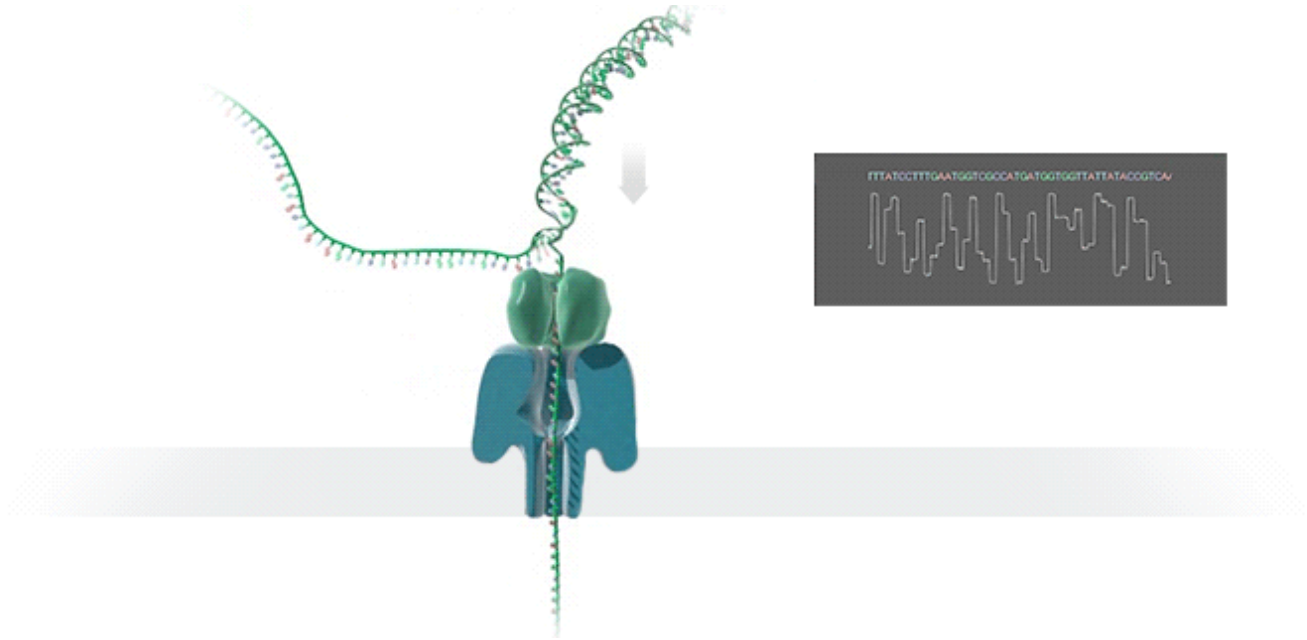
AirLift: A Fast and Comprehensive Technique for Translating Alignments between Reference Genomes

Jeremie S. Kim, Can Firtina, Damla Senol Cali, Mohammed Alser, Nastaran Hajinazar, Can Alkan, Onur Mutlu

GitHub: https://github.com/CMU-SAFARI/AirLift

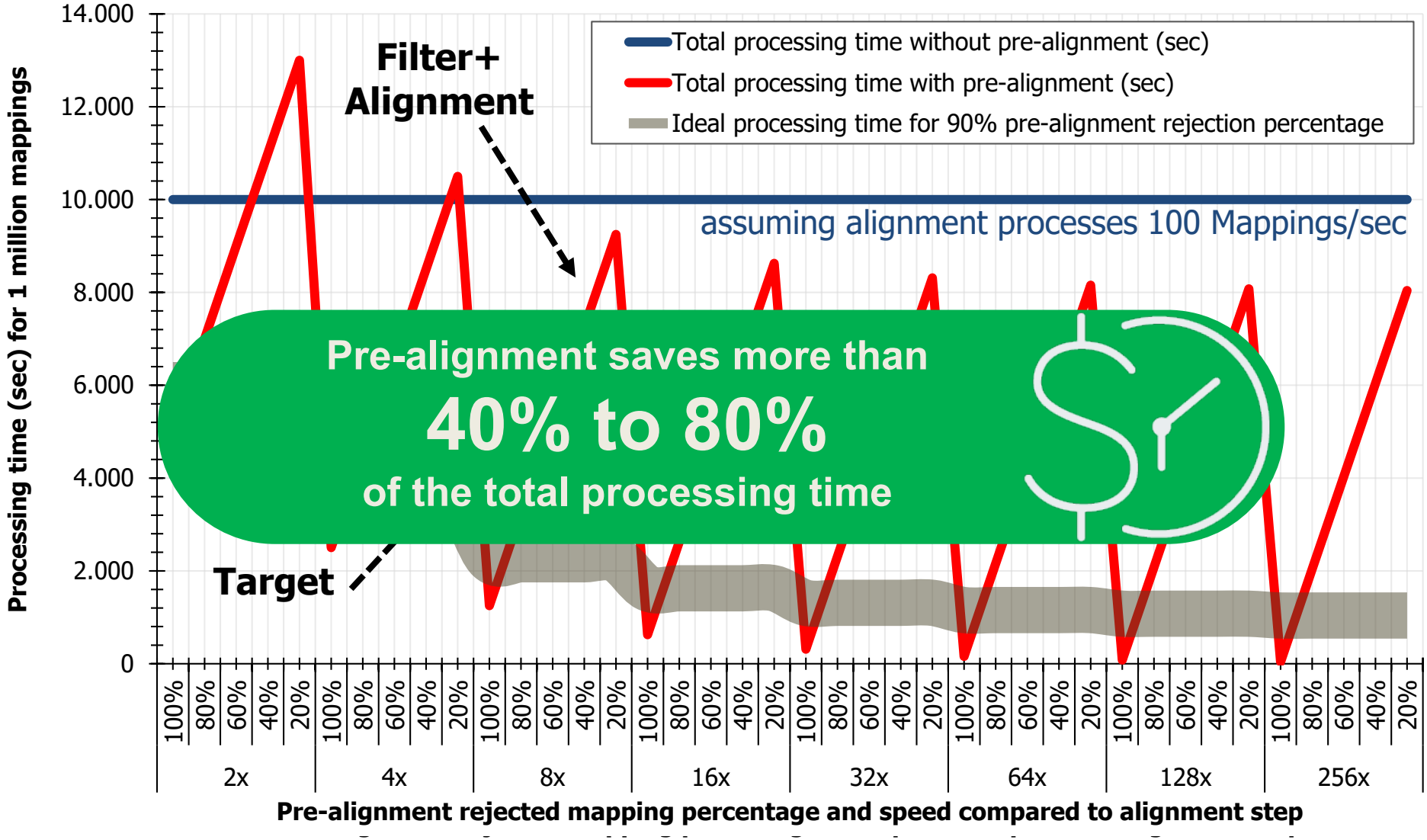
Kim+, "AirLift: A Fast and Comprehensive Technique for Translating Alignments between Reference Genomes", arXiv, 2020

Nanopore Sequencing



- **Nanopore** is a nano-scale hole
- In nanopore sequencers, an **ionic current** passes through the nanopores
- When the DNA strand passes through the nanopore, the sequencer measures the the **change in current**
- This change is used to identify the bases in the strand with the help of **different electrochemical structures** of the different bases

The Effect of Pre-Alignment (Theoretically)



Aside: In-Memory Graph Processing

- Large graphs are everywhere (circa 2015)

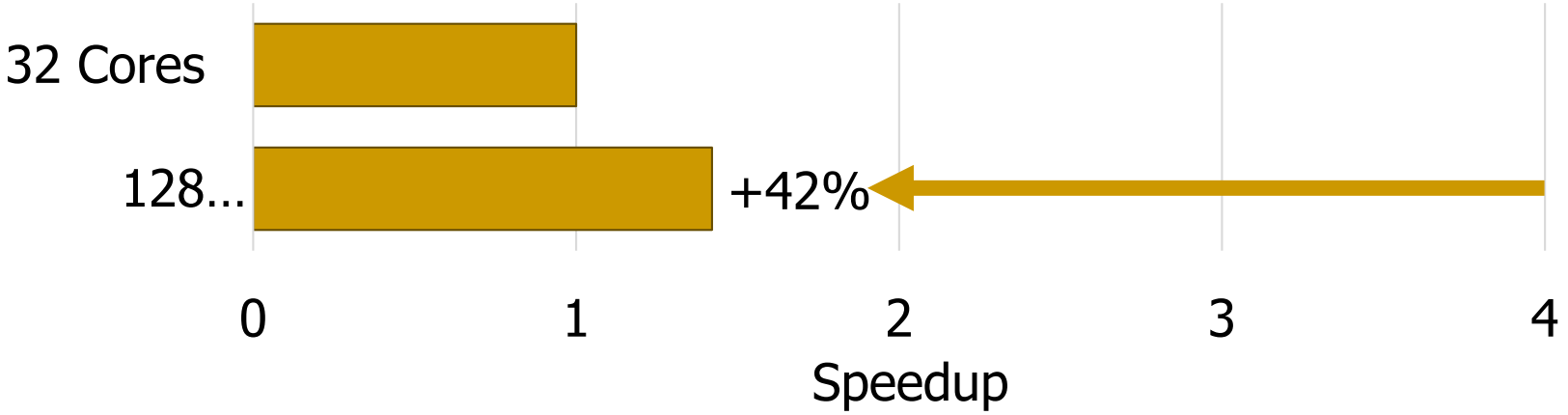
36 Million
Wikipedia Pages

1.4 Billion
Facebook Users

300 Million
Twitter Users

30 Billion
Instagram Photos

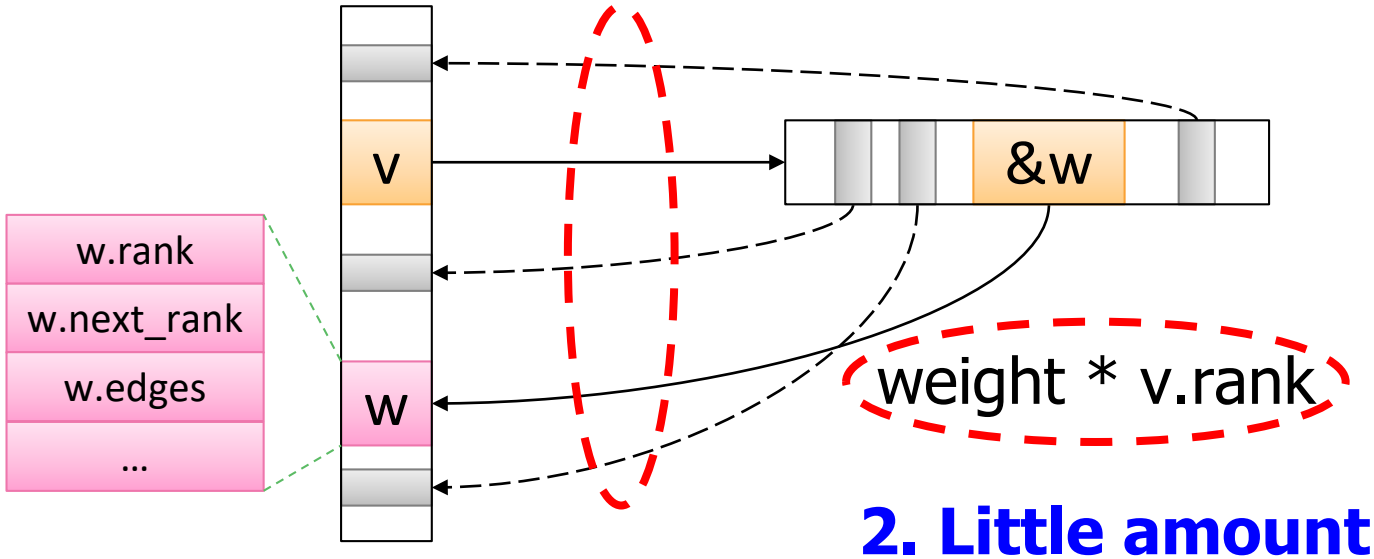
- Scalable large-scale graph processing is challenging



Key Bottlenecks in Graph Processing

```
for (v: graph.vertices) {
    for (w: v.successors) {
        w.next_rank += weight * v.rank;
    }
}
```

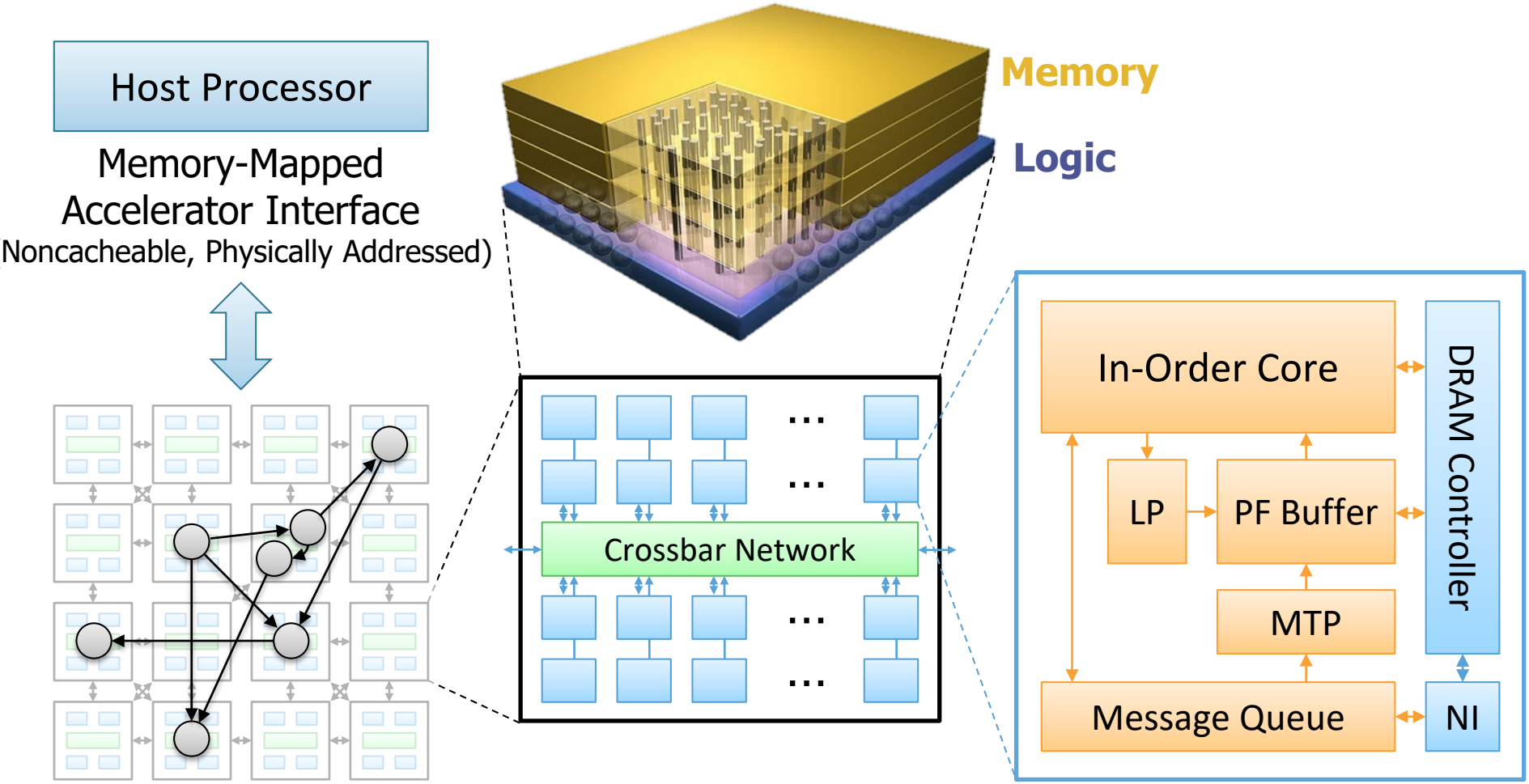
1. Frequent random memory accesses



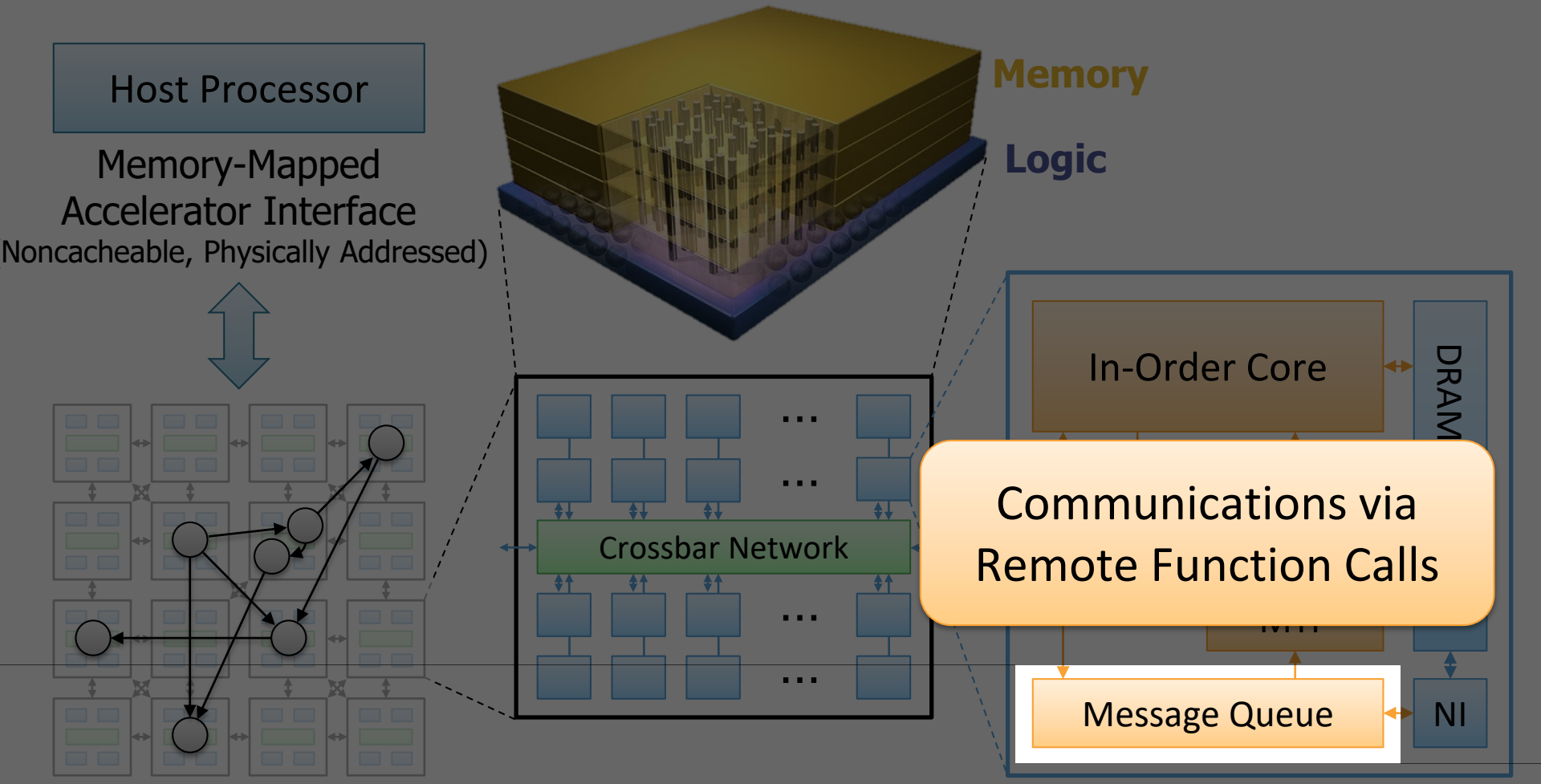
2. Little amount of computation

Tesseract System for Graph Processing

Interconnected set of 3D-stacked memory+logic chips with simple cores

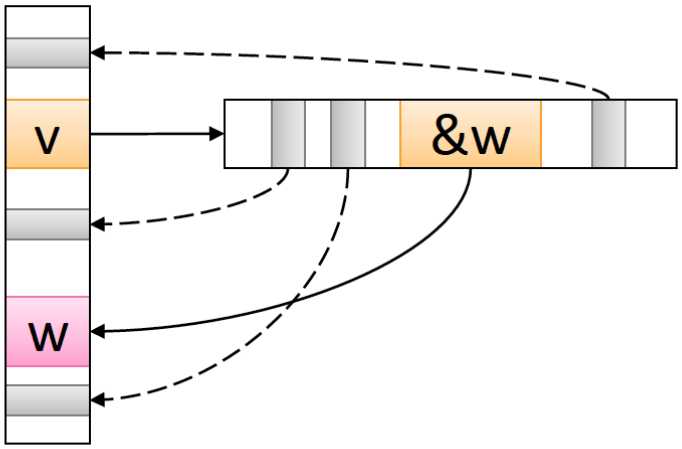


Tesseract System for Graph Processing



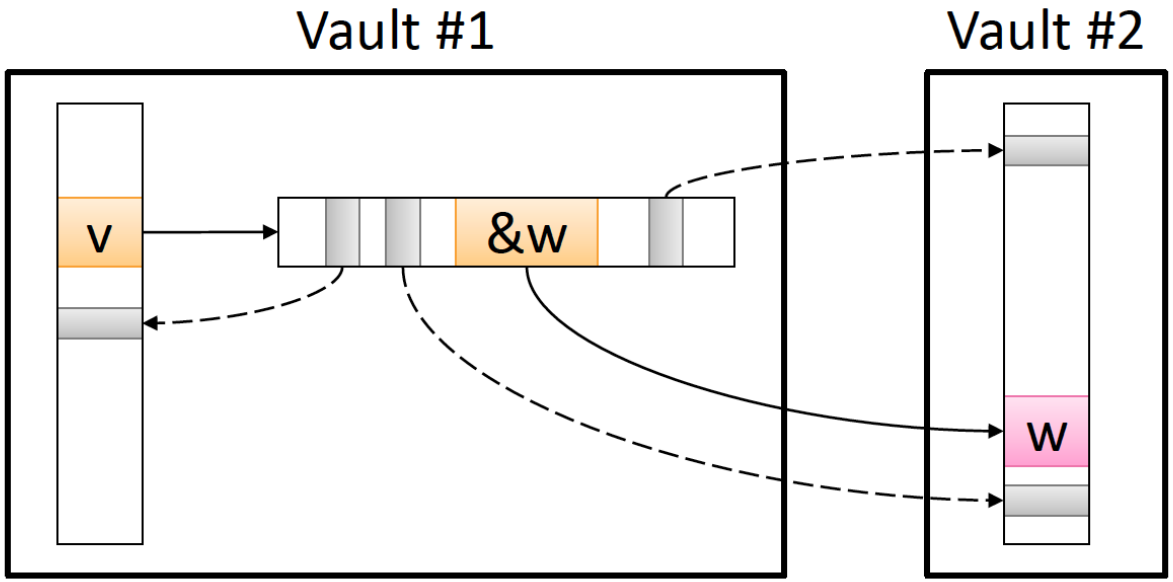
Communications In Tesseract (I)

```
for (v: graph.vertices) {
    for (w: v.successors) {
        w.next_rank += weight * v.rank;
    }
}
```



Communications In Tesseract (II)

```
for (v: graph.vertices) {
    for (w: v.successors) {
        w.next_rank += weight * v.rank;
    }
}
```

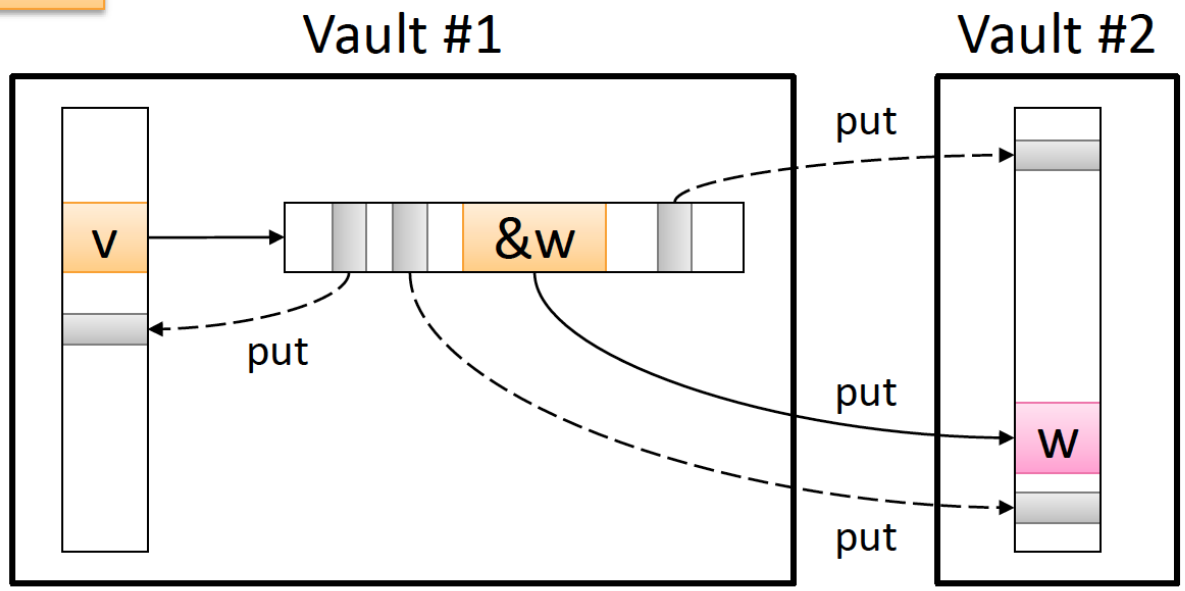


Communications In Tesseract (III)

```
for (v: graph.vertices) {
   for (w: v.successors) {
      put(w.id, function() { w.next_rank += weight * v.rank; });
   }
}
barrier();
```

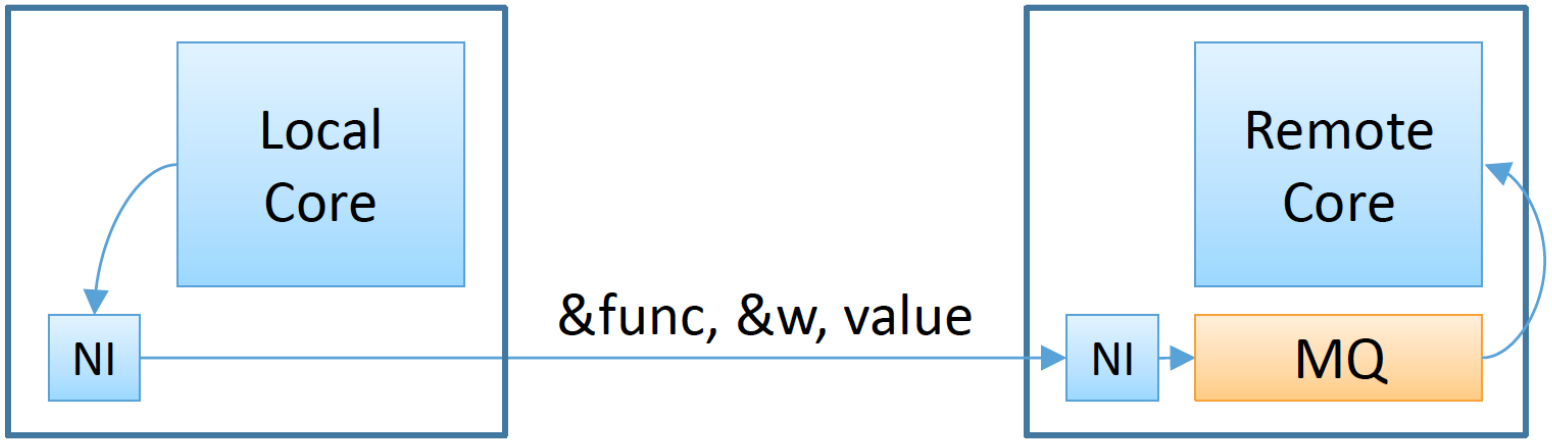
Non-blocking Remote Function Call

Can be **delayed** until the nearest barrier



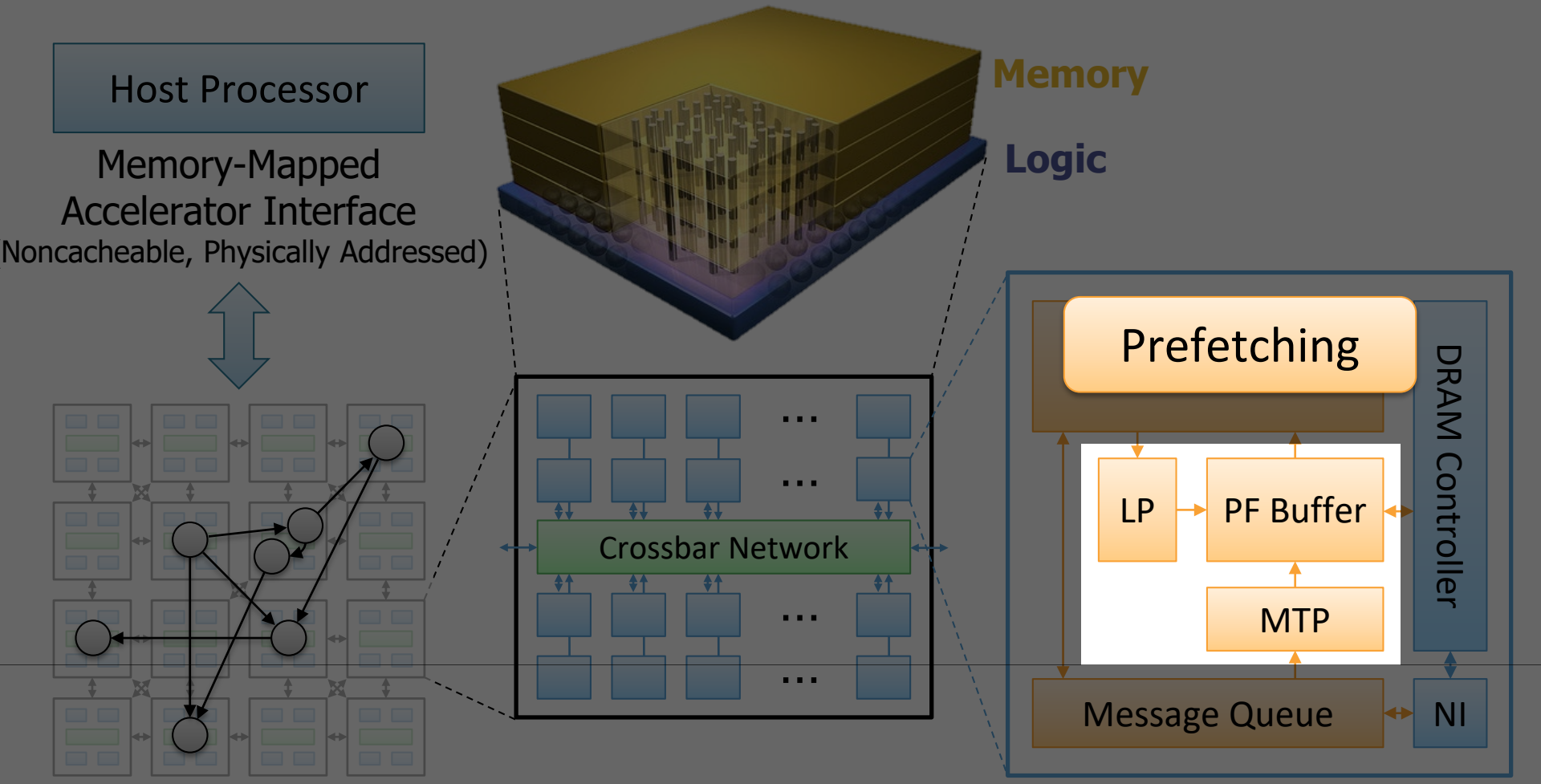
Remote Function Call (Non-Blocking)

1. Send function address & args to the remote core
2. Store the incoming message to the message queue
3. Flush the message queue when it is full or a synchronization barrier is reached

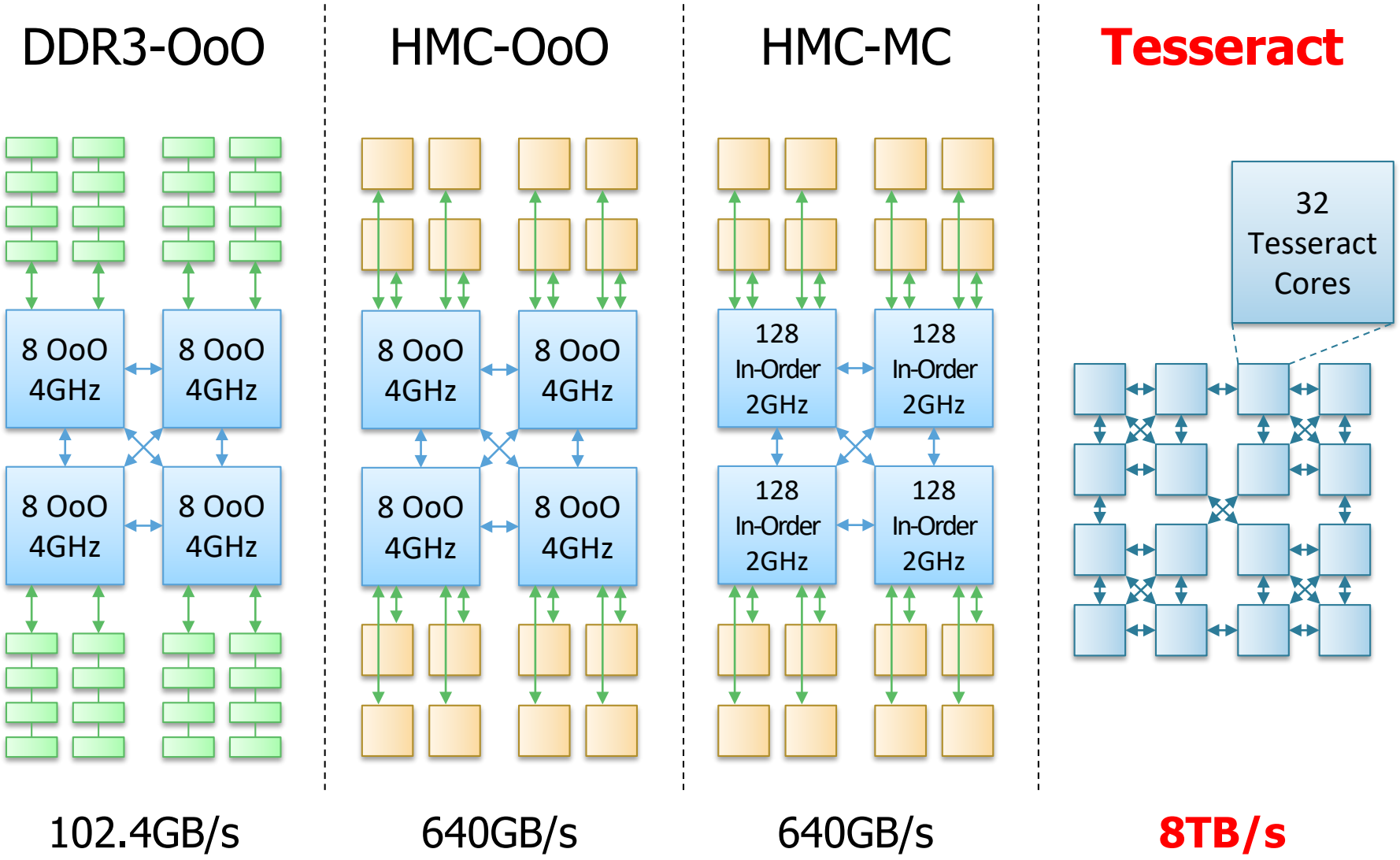


put(w.id, function() { w.next_rank += value; })

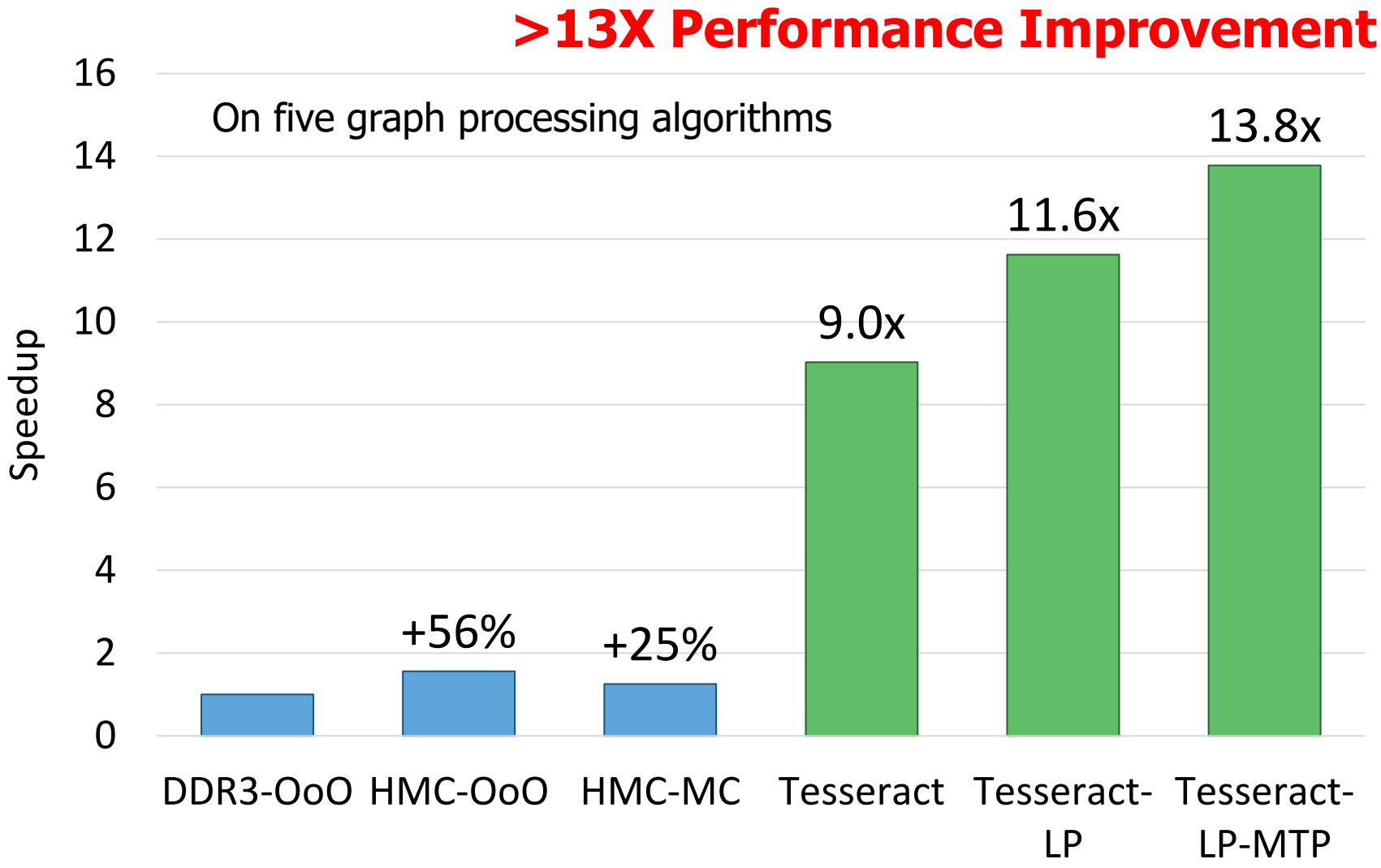
Tesseract System for Graph Processing



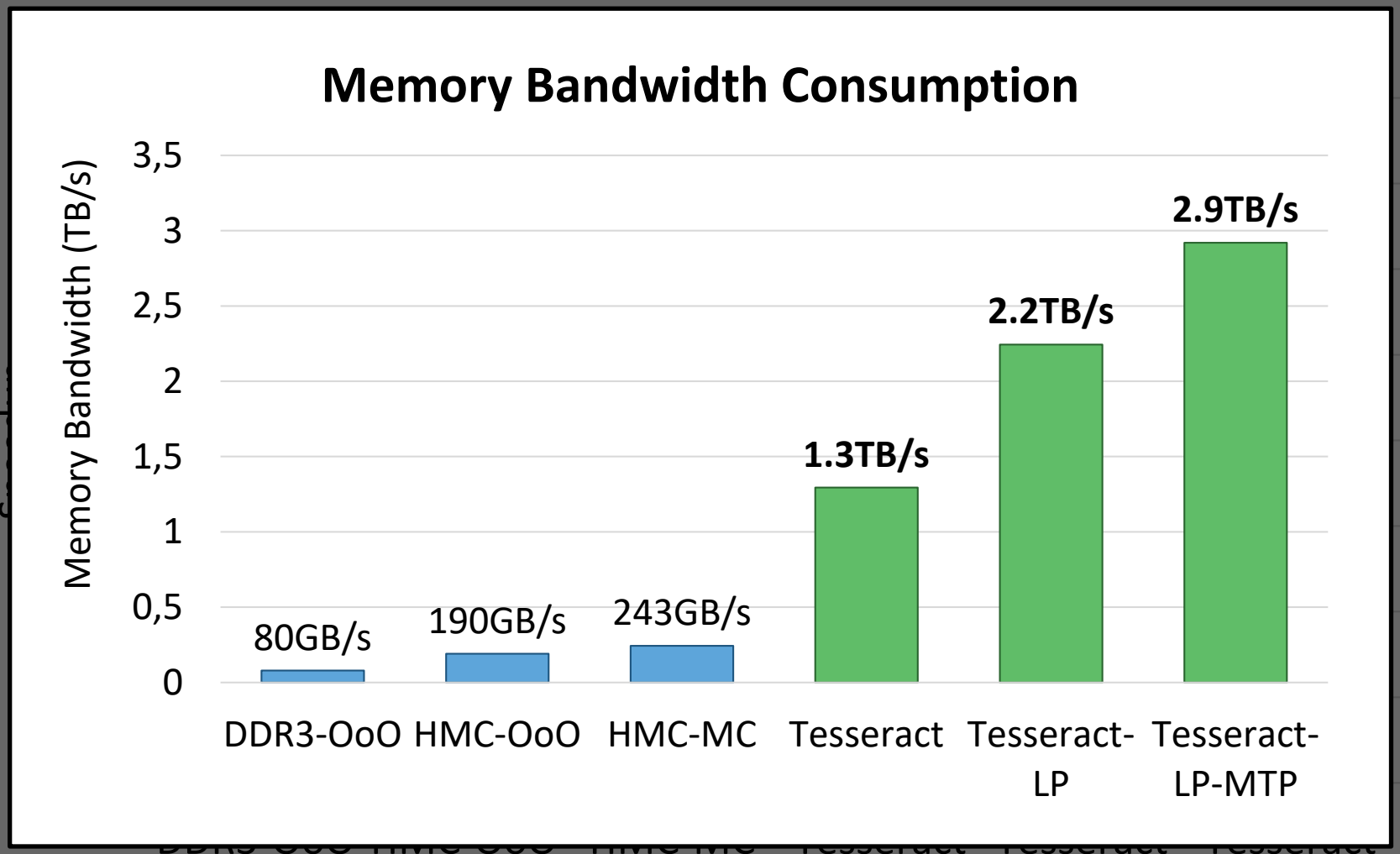
Evaluated Systems



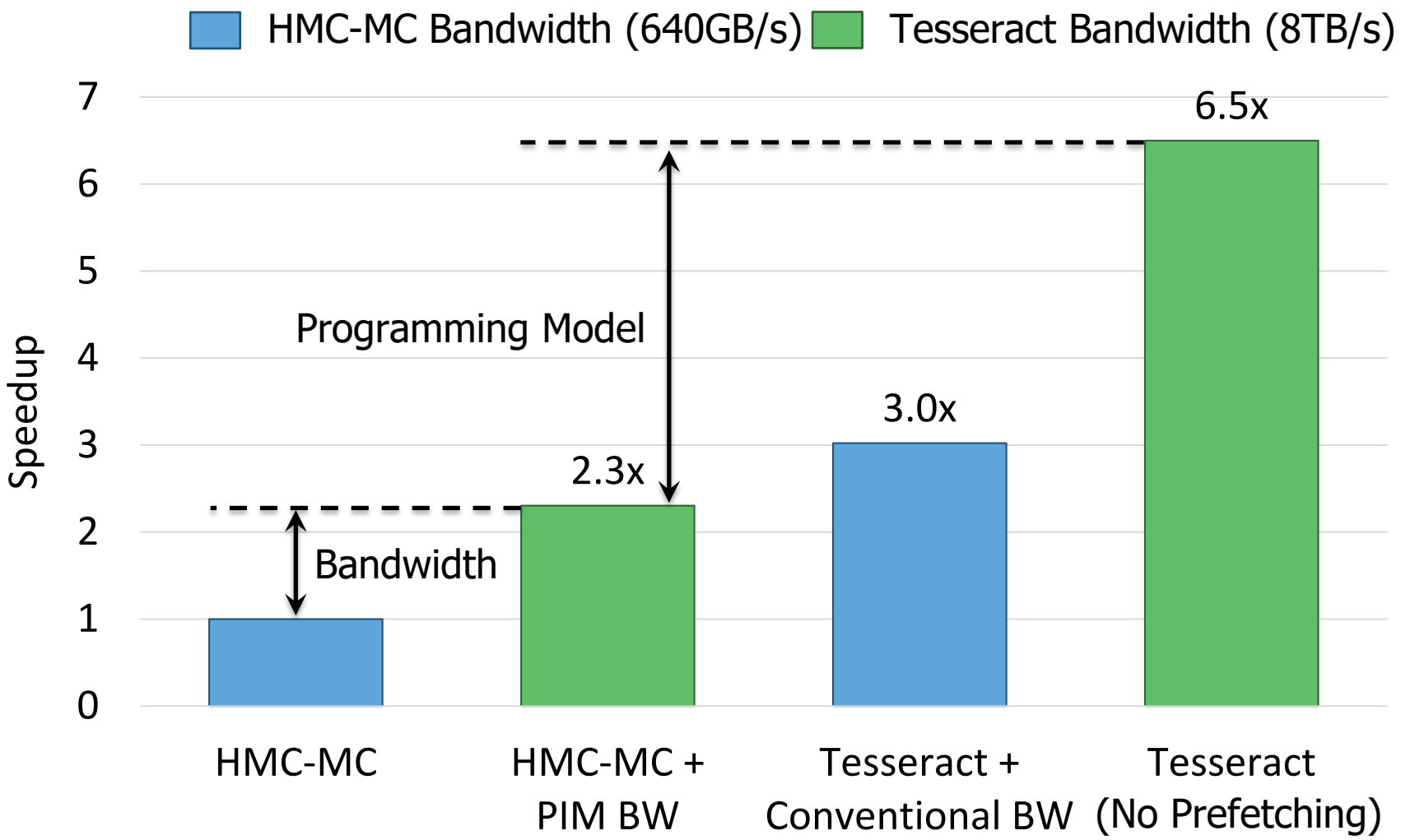
Tesseract Graph Processing Performance



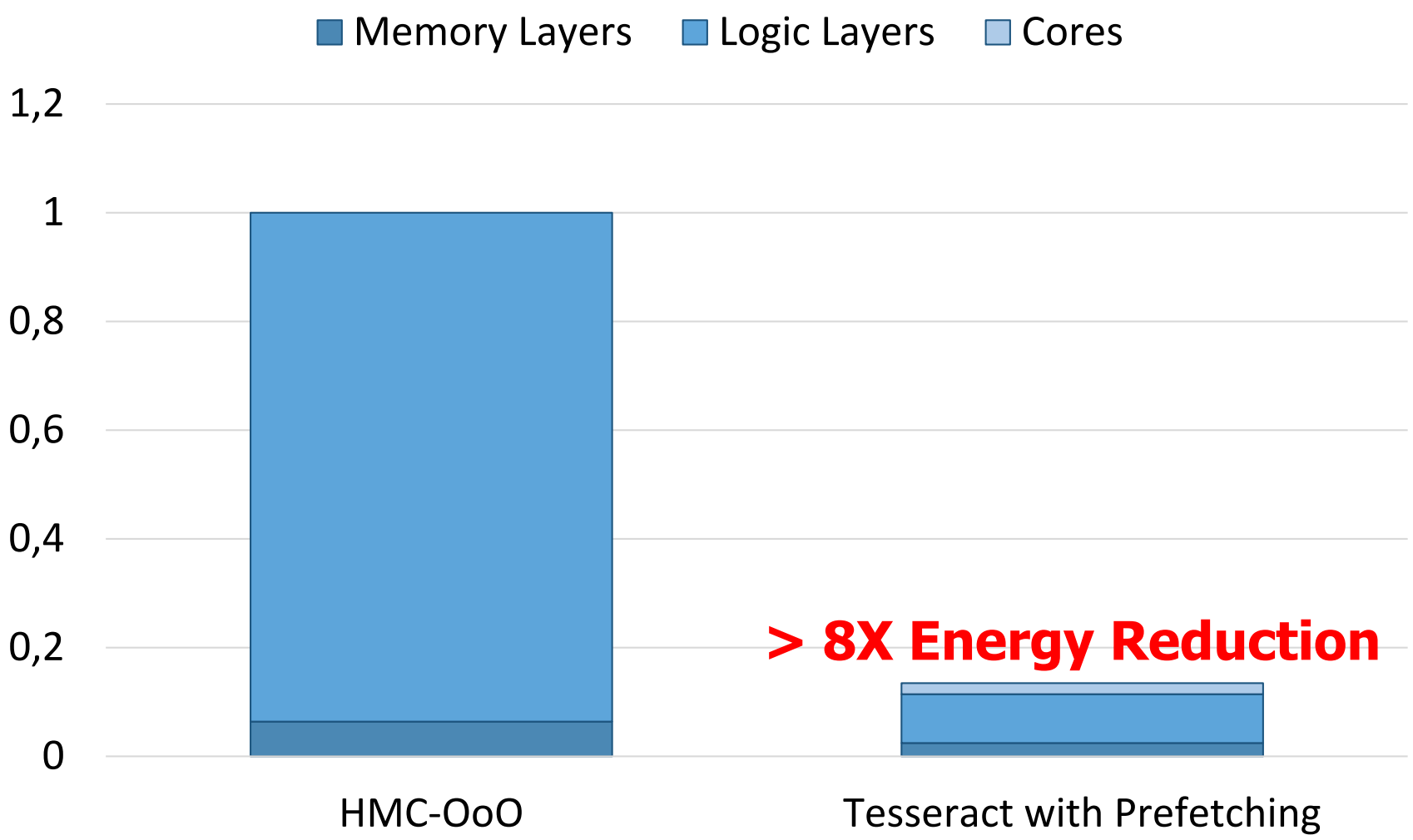
Tesseract Graph Processing Performance



Effect of Bandwidth & Programming Model

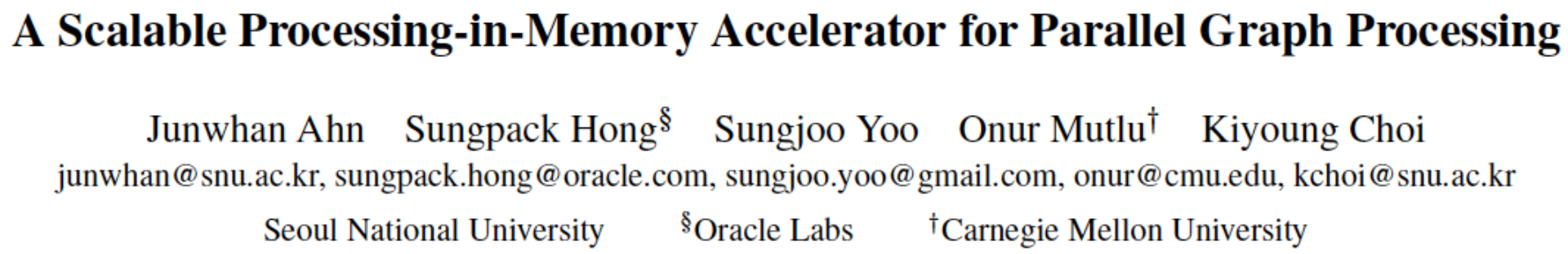


Tesseract Graph Processing System Energy



More on Tesseract

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi,
  "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"
  *Proceedings of the 42nd International Symposium on Computer Architecture (**ISCA**)*, Portland, OR, June 2015.
  [Slides (pdf)] [Lightning Session Slides (pdf)]



A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing

Junwhan Ahn Sungpack Hong[§] Sungjoo Yoo Onur Mutlu[†] Kiyoung Choi
junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr
Seoul National University [§]Oracle Labs [†]Carnegie Mellon University

End of Backup Slides