

Accelerating Genome Analysis

A Primer on an Ongoing Journey

Onur Mutlu

omutlu@gmail.com

<https://people.inf.ethz.ch/omutlu>

18 June 2022

AACBB Workshop Keynote @ ISCA 2022

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](https://doi.org/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](https://doi.org/10.1109/MM.2021.3088396)



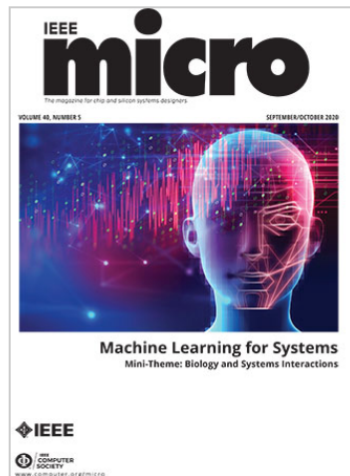
SmidgION from ONT

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

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](https://doi.org/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

◀	▶
Previous	Next
☰	Table of Contents
📄	Past Issues

A Few Overview Readings (III)

Going From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis

Mohammed Alser^{1,*}, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu^{2,*}

ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland

¹alserm@ethz.ch, ²omutlu@gmail.com

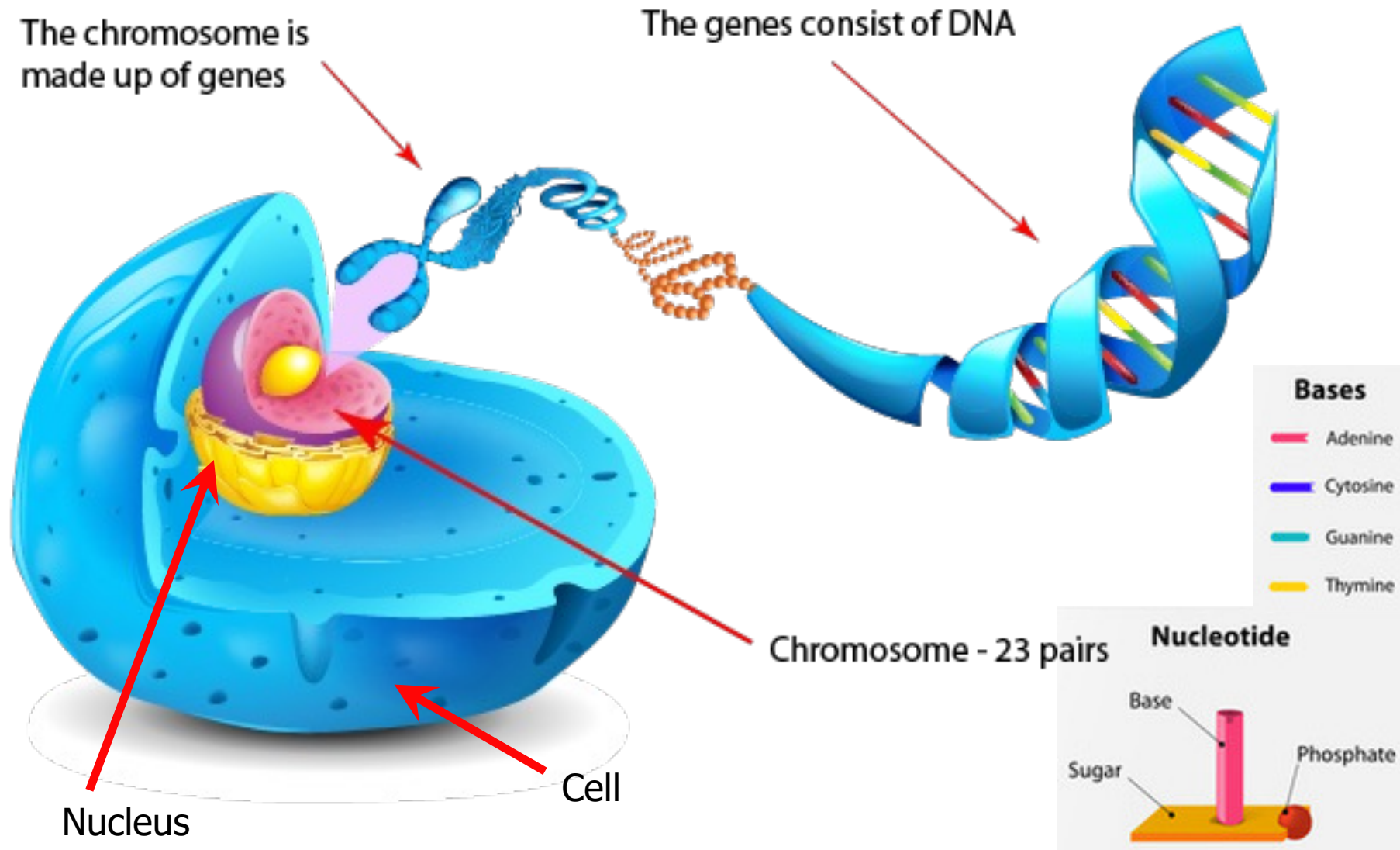
*Corresponding authors

<https://arxiv.org/pdf/2205.07957.pdf>

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

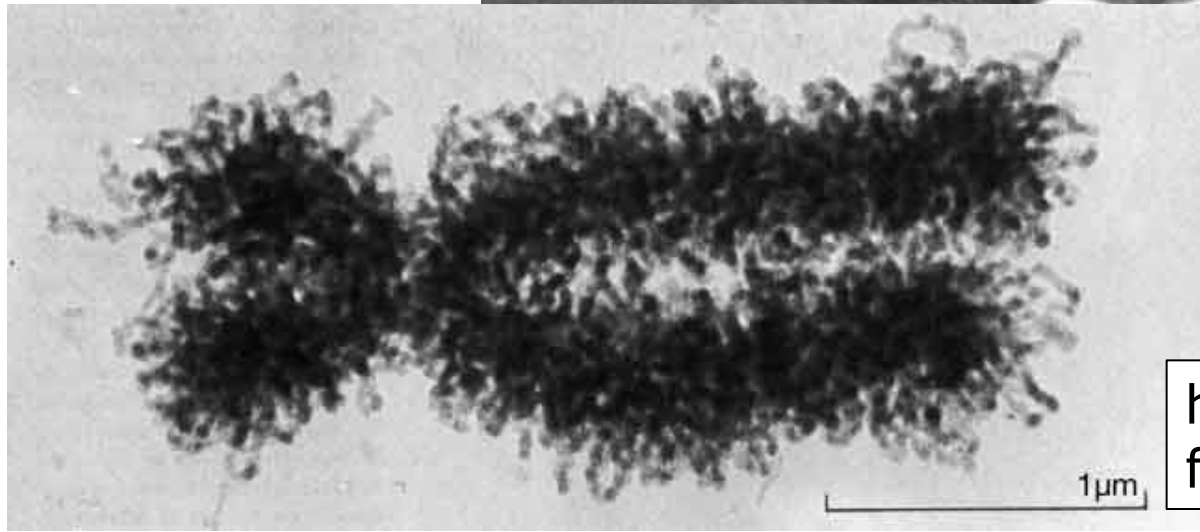
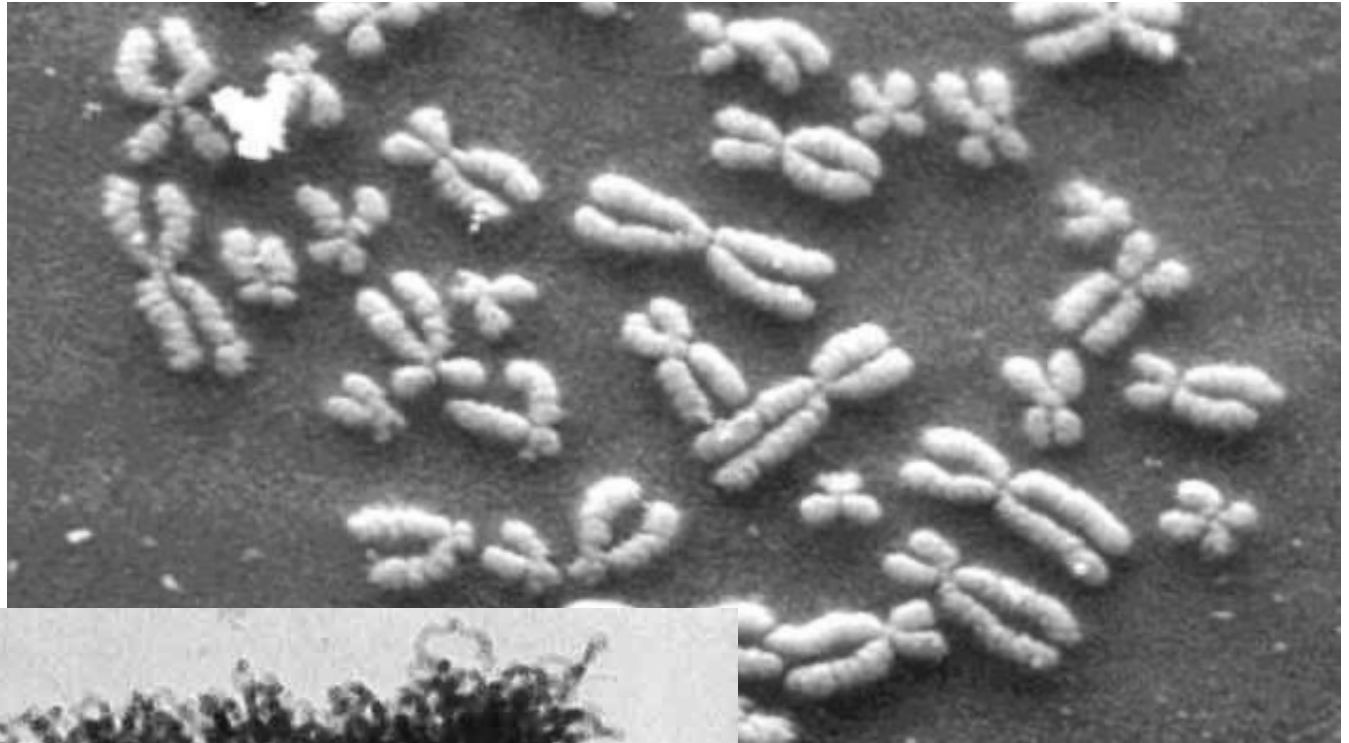
What Is a Genome Made Of?



The diagram illustrates the Central Dogma of Molecular Biology, showing the flow of genetic information from DNA to RNA to Protein. It features three large, colored arrows pointing right: a red arrow for DNA (labeled 'Genotypes'), a brown arrow for RNA, and a green arrow for Protein (labeled 'Phenotypes'). A curved arrow labeled 'Transcription' points from DNA to RNA, and another curved arrow labeled 'Translation' points from RNA to Protein. A third curved arrow labeled 'Replication' points from DNA back to DNA. Small molecular models are shown: a DNA double helix at the top left, an mRNA strand at the bottom of the RNA arrow, and a ribosome at the bottom right. On the far right, a vertical strip of four human faces is shown, representing the phenotypic expression of the genetic information.



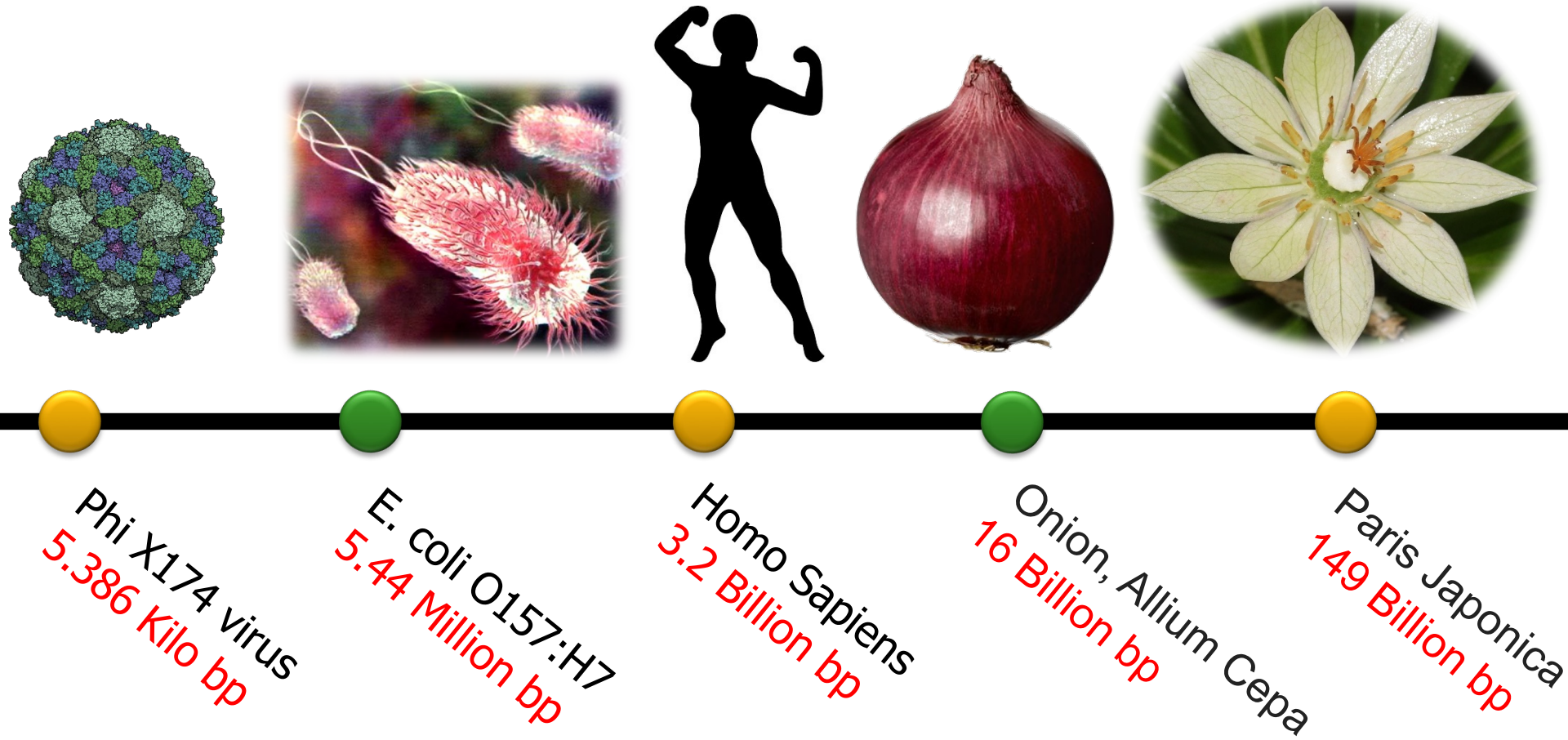
DNA Under Electron Microscope



human chromosome #12
from HeLa's cell

CCTCCTCAGTGCCACCCAGCCCCTGGCAGCTCCCAAACA
GGCTCTTATTAAACACCCTGTTCCCTGCCCCCTTGGAGTG
AGGTGTCAAGGACCTAAACTAAAAAAAAAAAAAAGAAAA
AGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAA
AAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATG
TGCTAAACAGCACTTTT**TTGACCATTAT**TTTGGATCTGAAA
GAAATCAAGAATAAATGAAGGACTTGATACATTGGAAGA
GGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAAAAAGAAA
AAGAAAAGAAAAAGA**A**TTTAAAATTTAAGTAATTCTTTGA
AAAAAACTAATTTCTAAGCTTCTT**C**ATGTCAAGGACCTAAT
GTCTGTGTTGCAGGTCTTCTTGCATTTCCCTGTCAAAGA
AAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAAACTA
ATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCAGGCC
GGCTCTTATTAAACACCCTGTTCCCTGCCCCCTTGGAGTG

How Large is a Genome?



DNA Sequencing

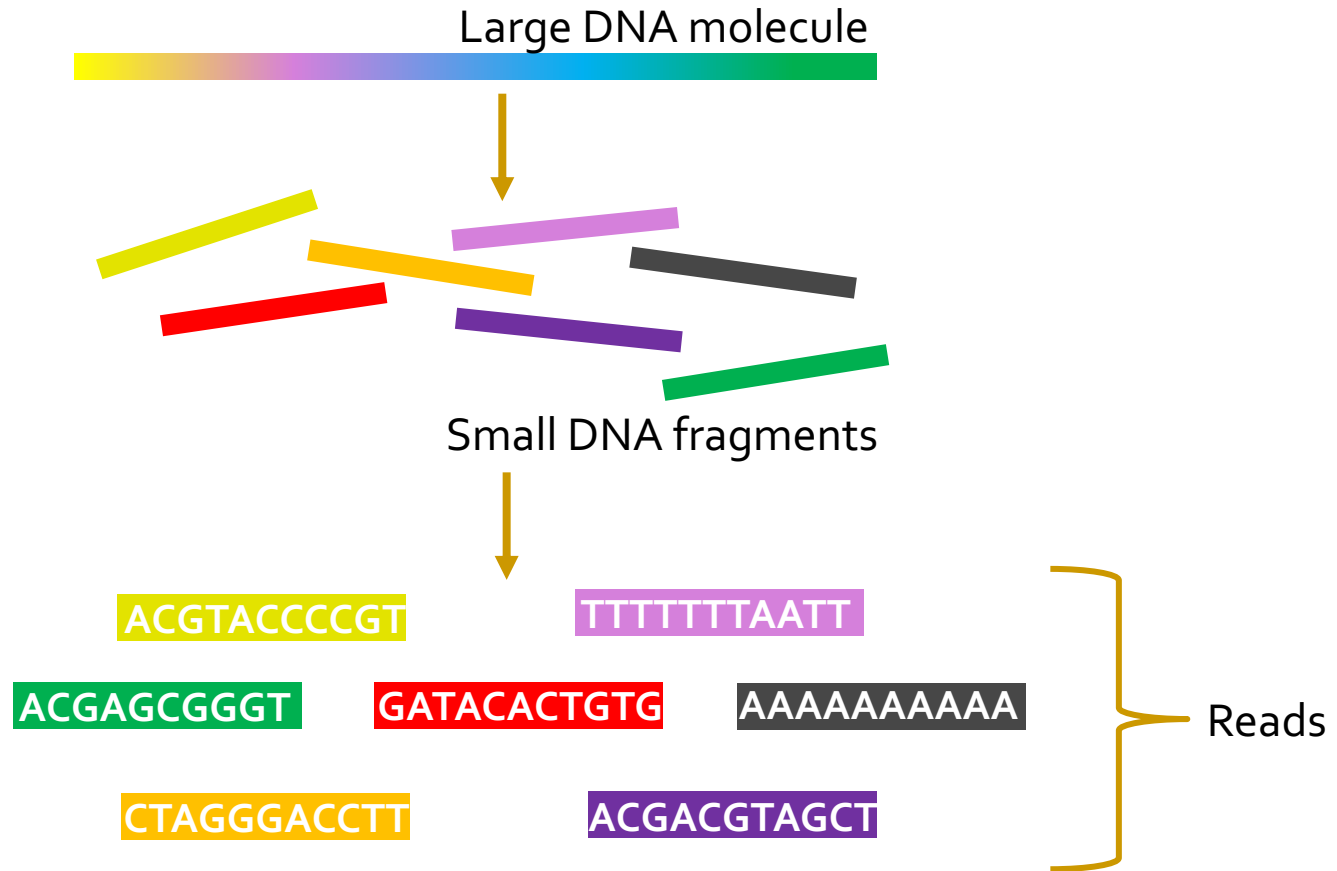
- Goal:

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

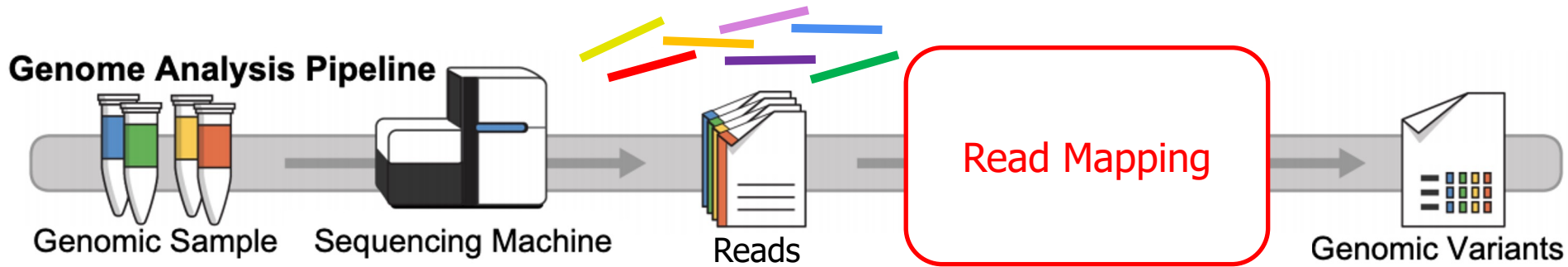
- Challenge:

- 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

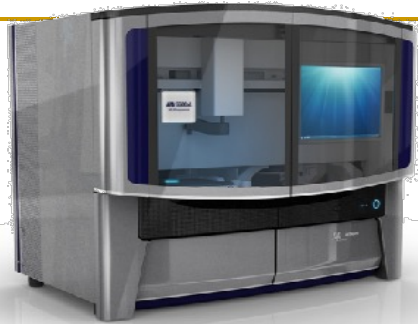
Untangling Yarn Balls & DNA Sequencing



Genome Sequencers



Roche/454



AB SOLiD



Illumina MiSeq



Complete Genomics



Illumina HiSeq2000



Pacific Biosciences RS



Oxford Nanopore MinION



Illumina NovaSeq 6000



Ion Torrent PGM



Ion Torrent Proton



Oxford Nanopore GridION

SAFARI

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

High-Throughput Sequencers



Illumina MiSeq



Pacific
Biosciences
Sequel II

Oxford
Nanopore
PromethION



Illumina NovaSeq 6000



Pacific Biosciences RS II



Oxford Nanopore MinION

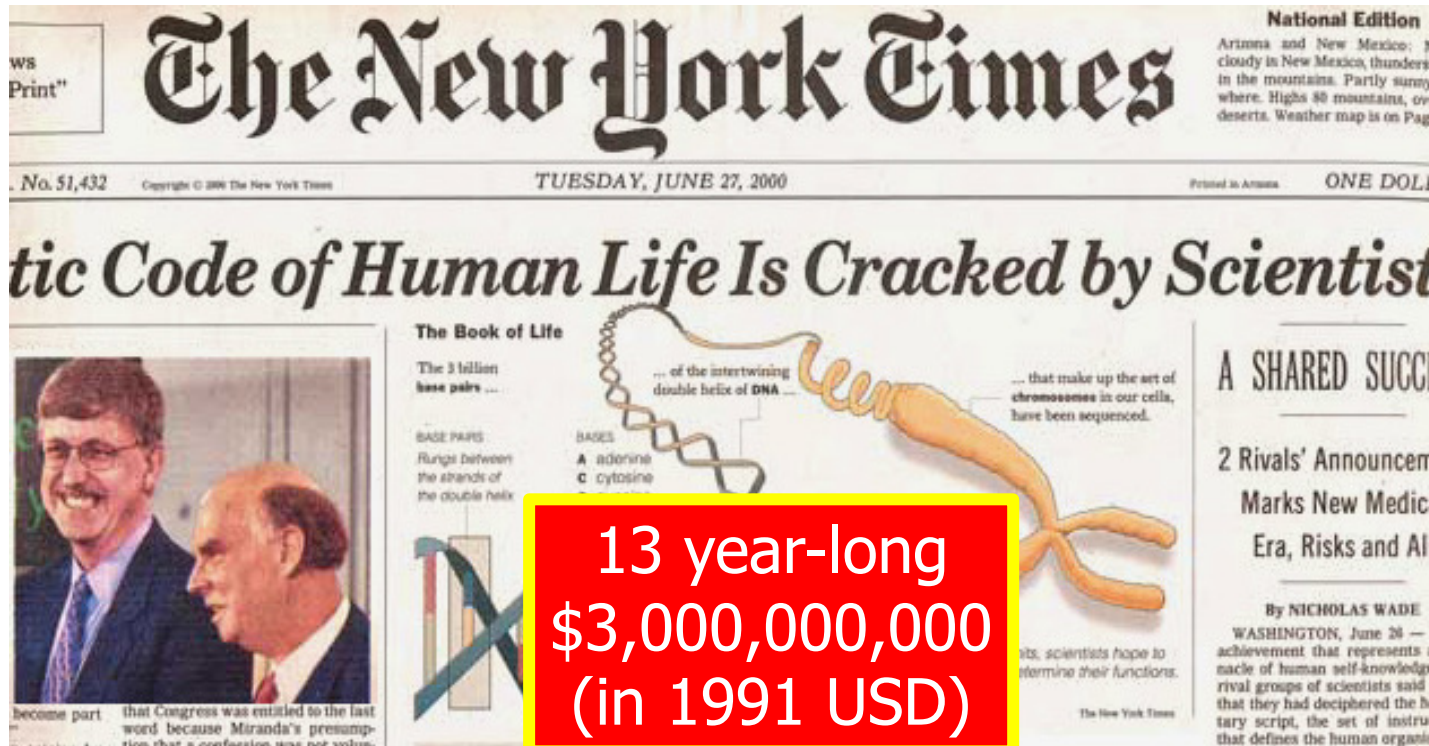


Oxford
Nanopore
SmidgION

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

The Genomic Era

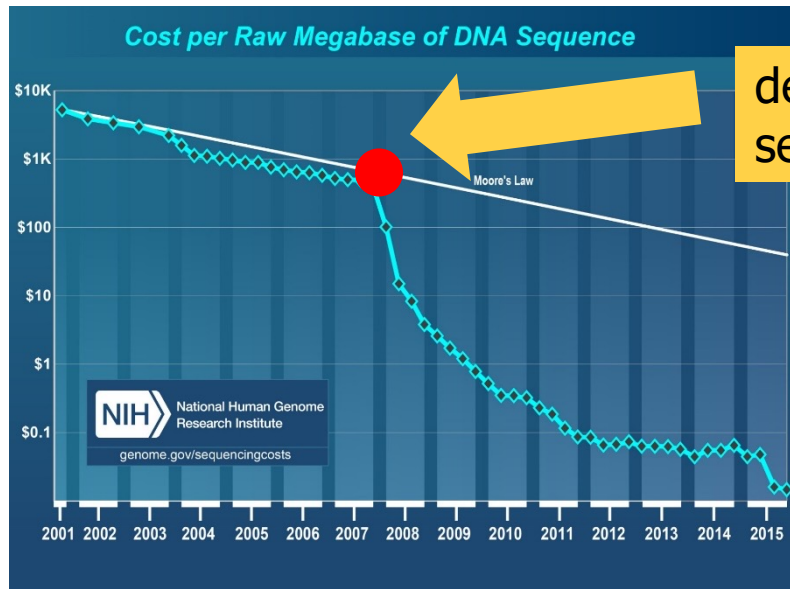
- 1990-2003: The Human Genome Project (HGP) provides a complete and accurate sequence of all **DNA base pairs** that make up the human genome and finds 20,000 to 25,000 human genes.



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

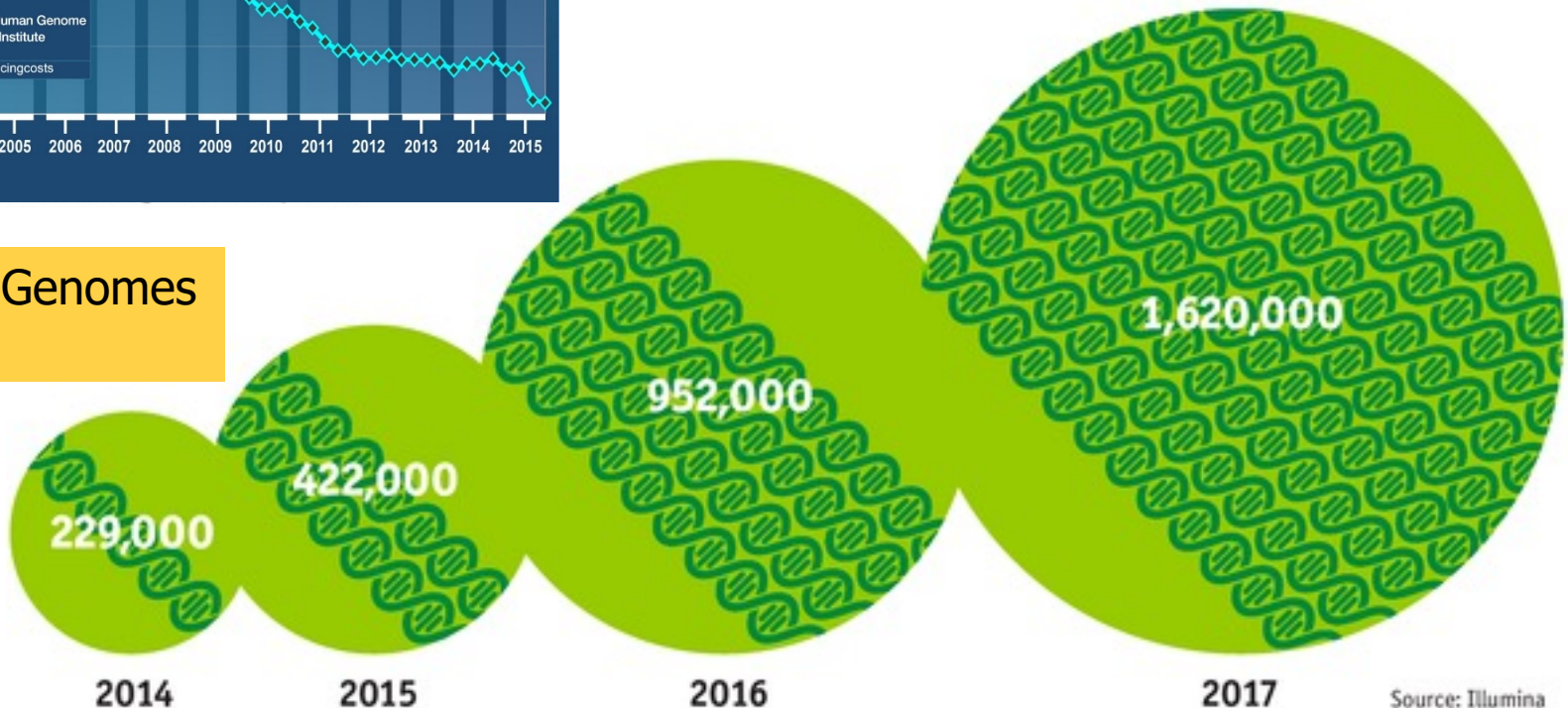


The Genomic Era (continued)



development of high-throughput sequencing (HTS) technologies

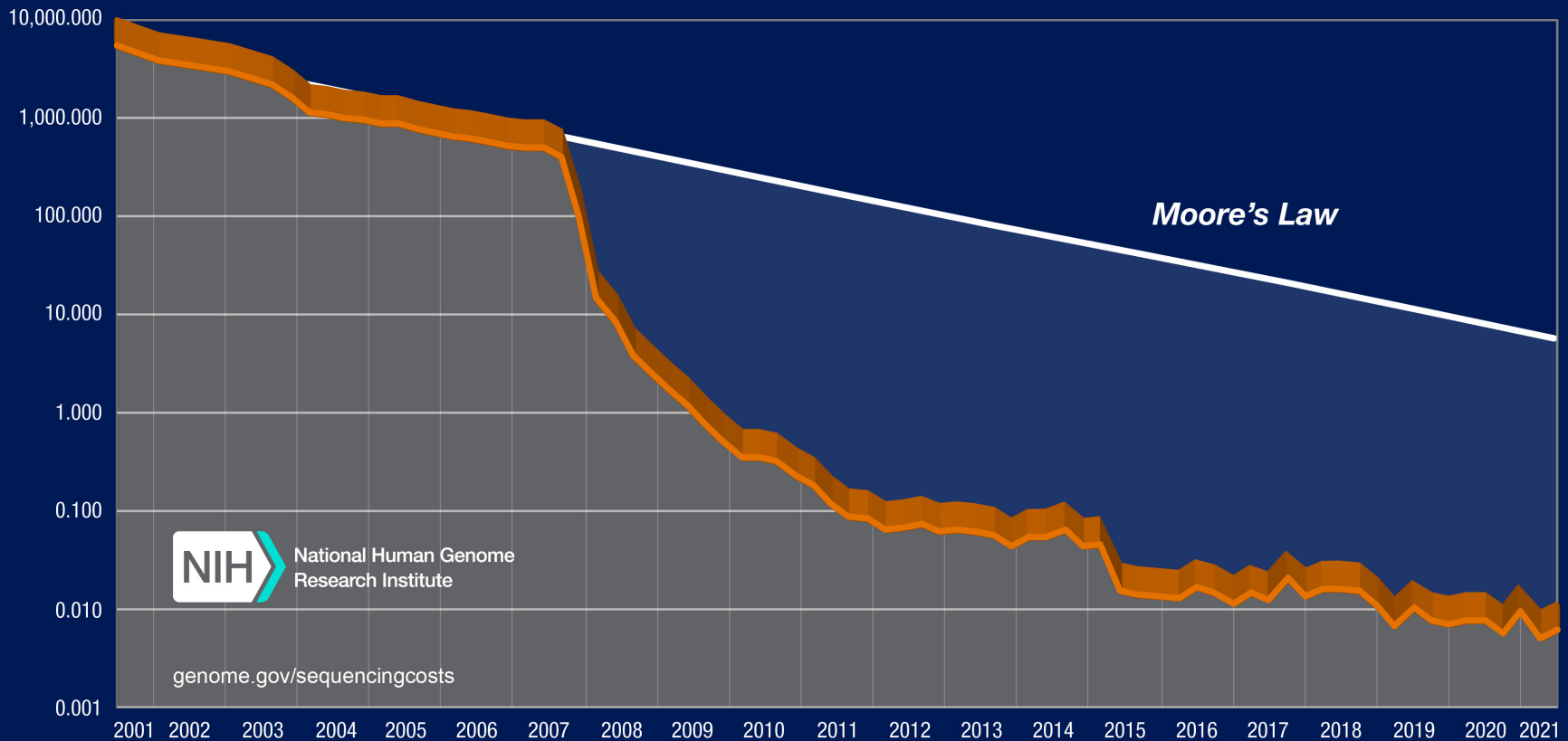
Number of Genomes Sequenced



The Economist

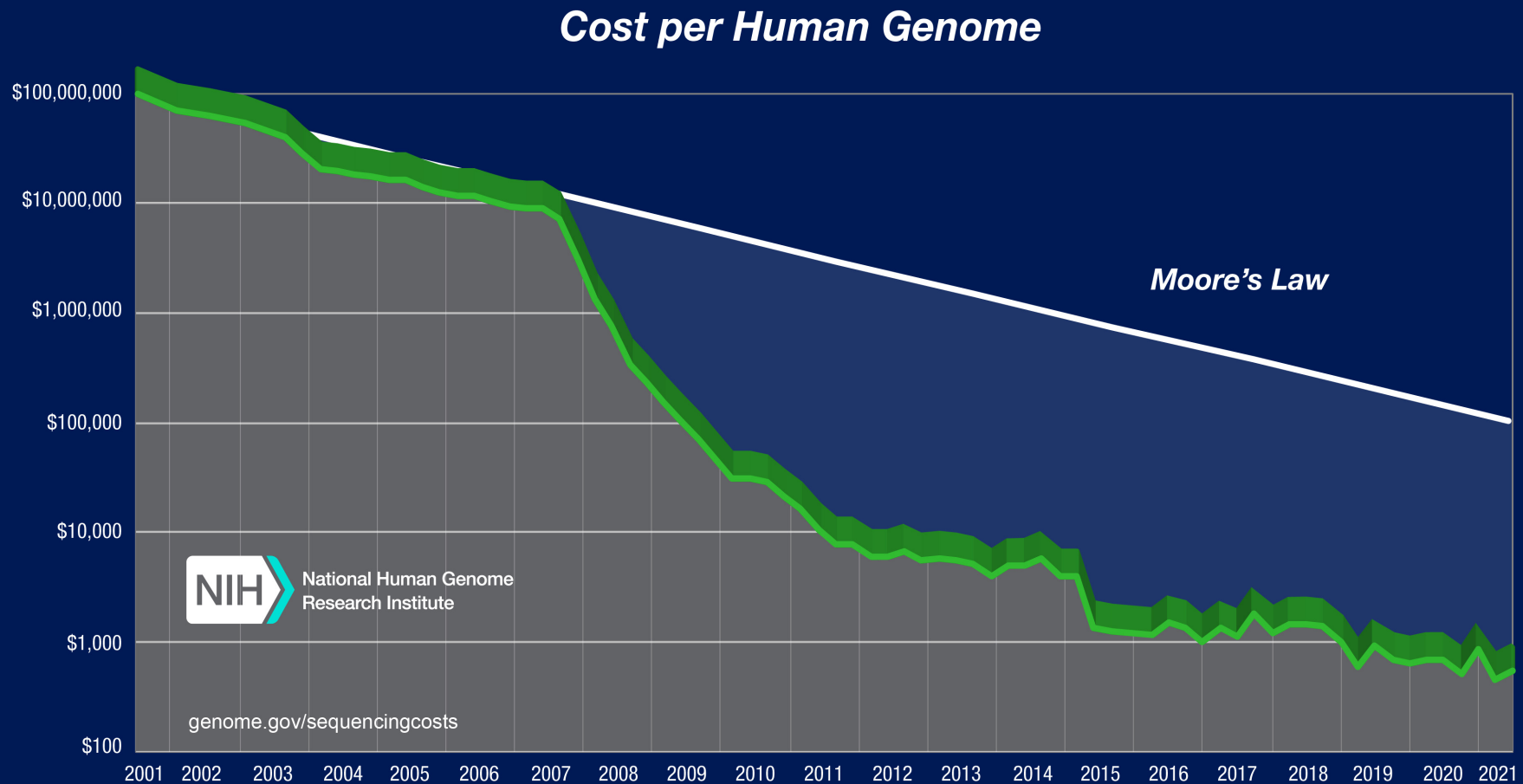
Genome Sequencing Cost Is Reducing

Cost per Raw Megabase of DNA Sequence



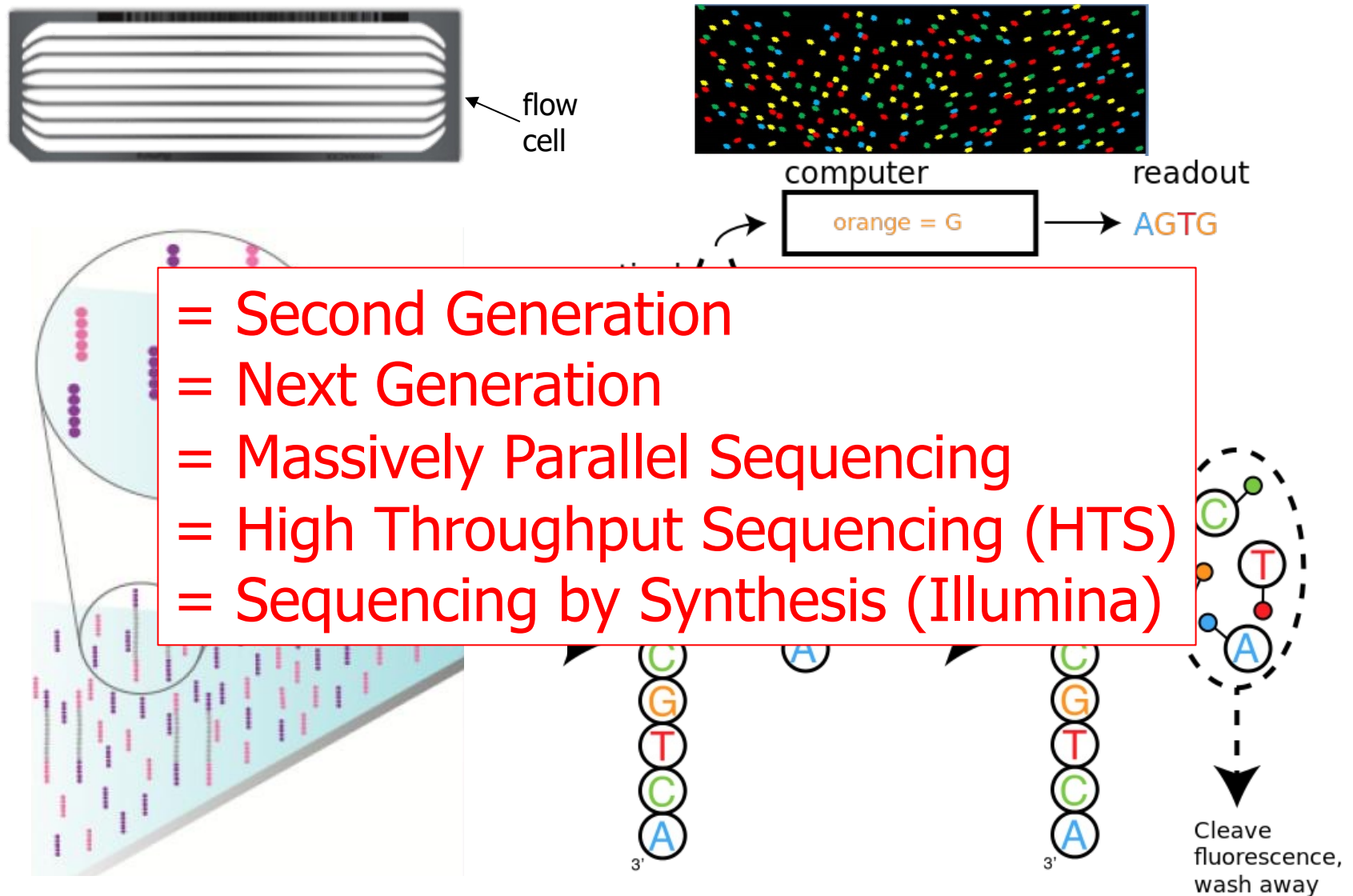
*From NIH (<https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data>)

Genome Sequencing Cost Is Reducing

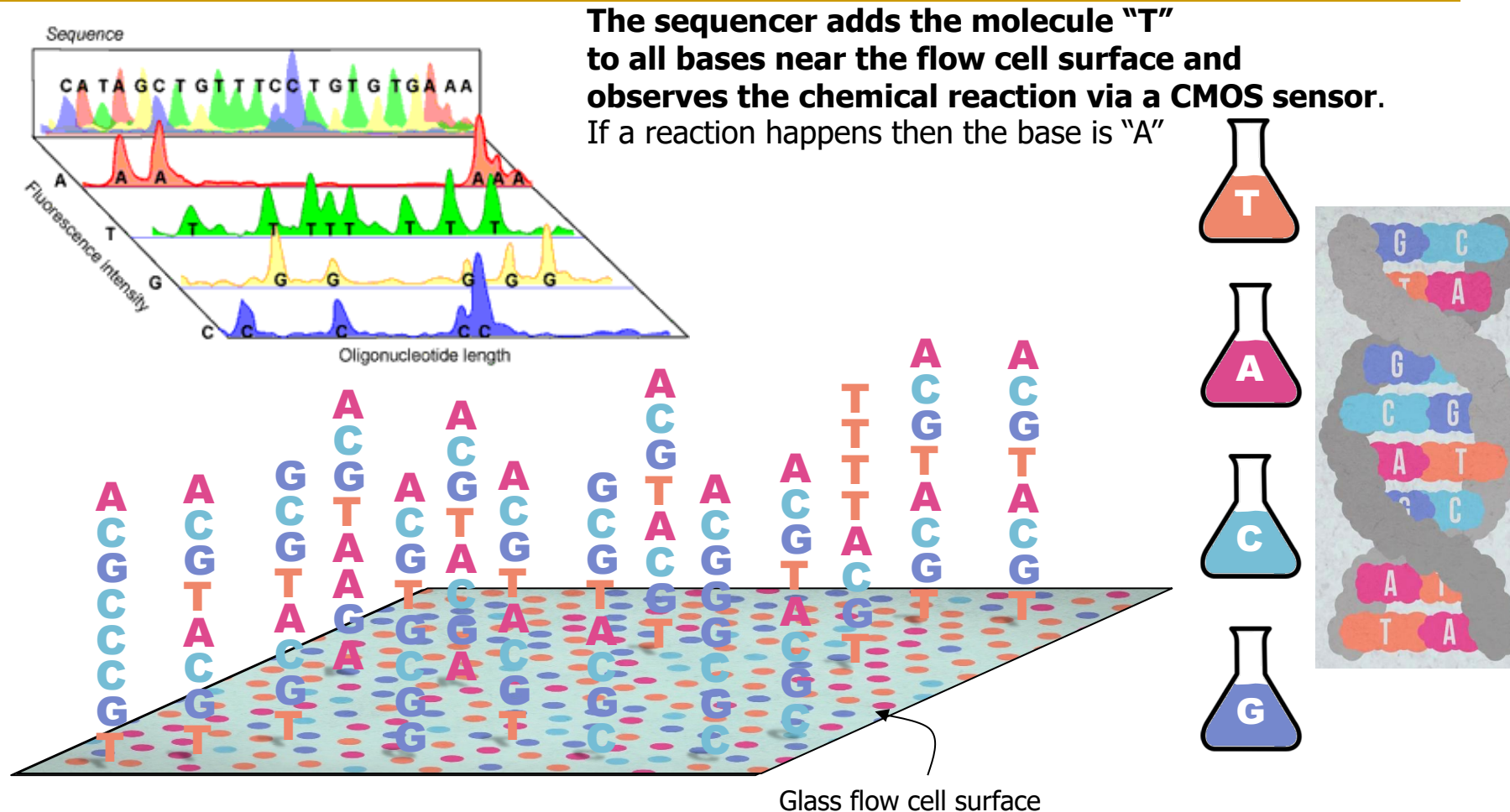


*From NIH (<https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data>)

High-Throughput Sequencing (HTS)



High-Throughput Sequencing (HTS)



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



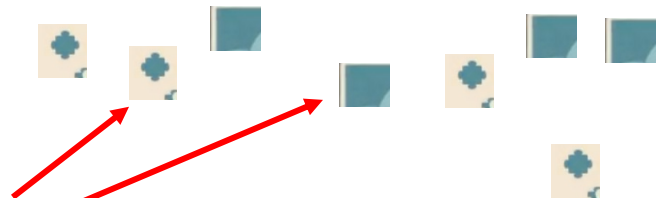
**Reference
genome**



.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** ▼

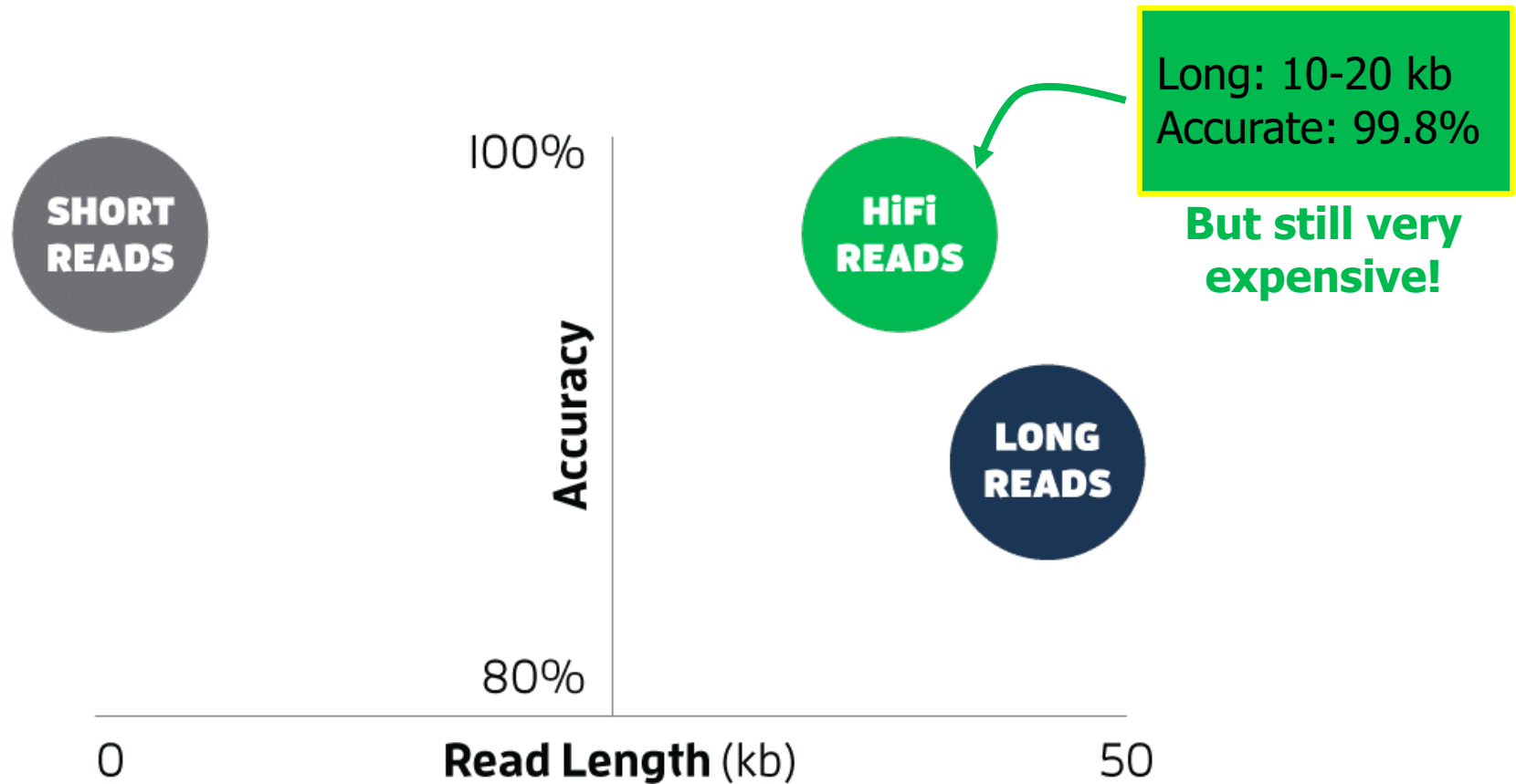


Oxford Nanopore MinION

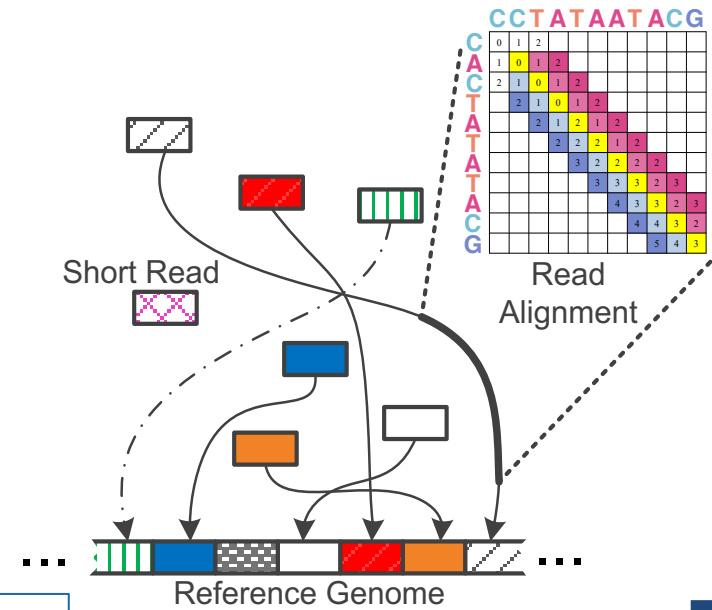
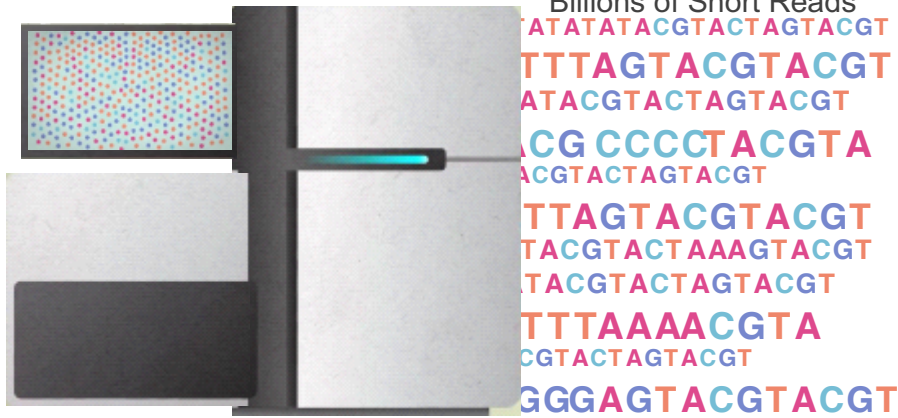
Senol Cali+, “[Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions](#),” *Briefings in Bioinformatics*, 2018.

[[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



1 Sequencing

Genome Analysis

2 Read Mapping

reference: TTTATCGCTTCCATGACGCAG

read1: ATCGCATCC

read2: TATCGCATC

read3: CATCCATGA

read4: CGCTTCCAT

read5: CCATGACGC

read6: TTCCATGAC



3 Variant Calling

4 Scientific Discovery

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 Deshpande⁵, Kodi Taraszka⁴, Huwenbo Shi^{6,7}, Pelin Icer Baykal⁸, Harry Taegyun Yang^{4,9}, Victor Xue⁴, Sergey Knyazev⁸, Benjamin D. Singer^{10,11,12}, Brunilda Balliu¹³, David Koslicki^{14,15,16}, Pavel Skums⁸, Alex Zelikovsky^{8,17}, Can Alkan^{2,18}, Onur Mutlu^{1,2,3†} and Serghei Mangul^{5*†} 

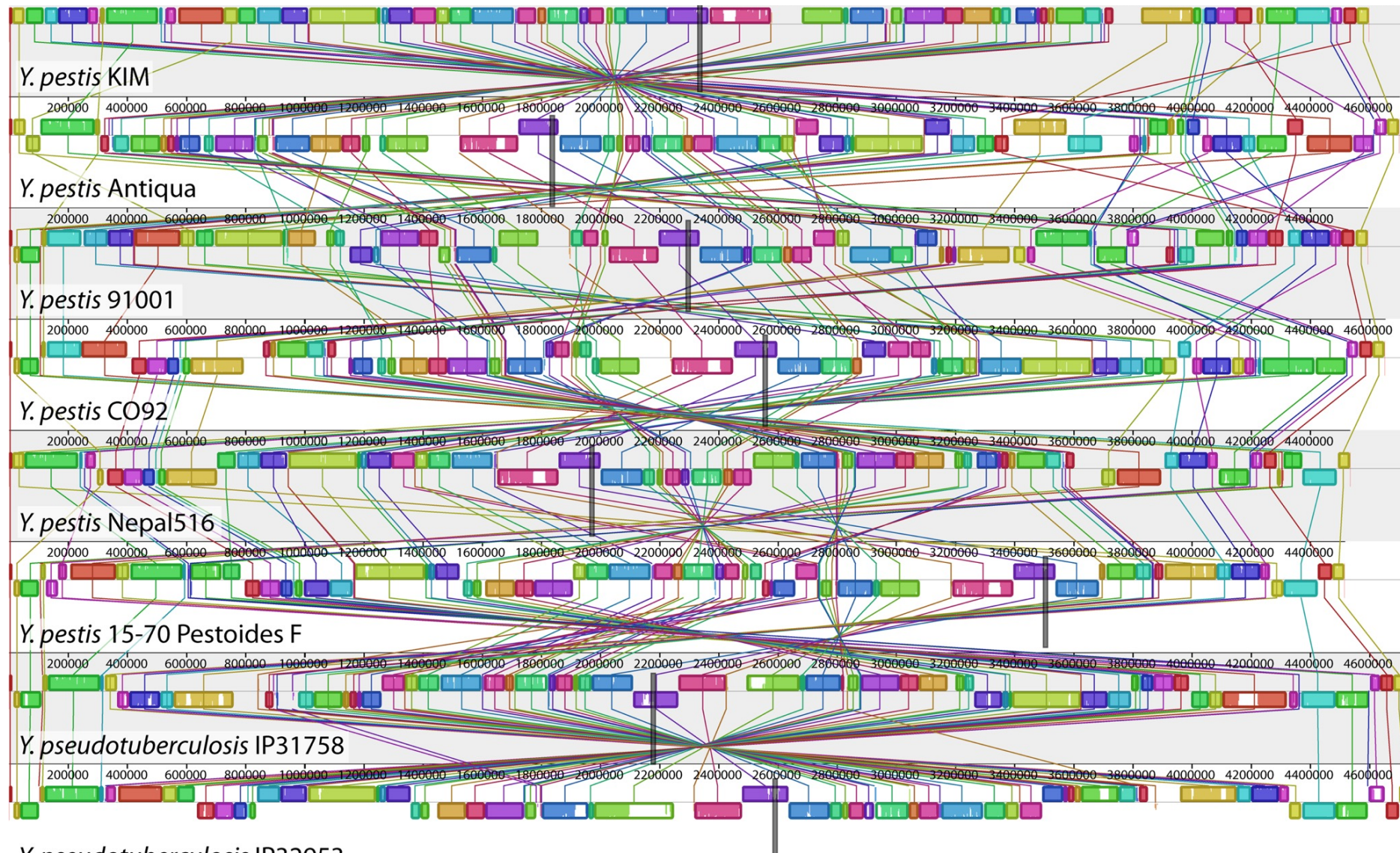
Why Do We Care?

Multiple sequence alignment

PHDHtm			-----MMMMMMMMMMMMMMMMMMMM-----	
16082665	<i>T acid</i>	10	----MASDRKSEGFQSGAGLIRYFEEEEEIKGPALDPKLVVYMGIAVAIIVEIAKIFWFP---	(55)
13541150	<i>T volc</i>	10	----MASDRKSEGFQSGAGLIRYFEEEEEIKGPALDPKLVVYIGIAVAIMVELAKIFWFP---	(55)
RFAC01077	<i>F acid</i>	13	-MTSMAKDNQNFQSGAGLIRYFNEEEIKGPAIDPKLIITYIGIAMGVIVELAKVFWFPV---	(58)
15791336	<i>H NRC1</i>	10	----MSSGQNSGGLMSSSAGLVRYFDSEDSNALQIDPRSVVAVGAFFGLVLVLLAQFFA----	(53)
RAG22196	<i>A fulg</i>	14	MAKAPKKGAKTPPLMSSSAGIMRYFEE-EKTQIKVSPKTI LAAGIVTGVLI IILNAYYGLWP-	(68)
RPO01000	<i>P abys</i>	9	-----MAKEKTTLPPTGAGLMRFFDE-DTRAIKITPKGAVALTLILIIIFEIILHVVGPRIFG	(56)
RPH01741	<i>P hori</i>	9	-----MAKEKTTLPPTGAGLMRFFDE-DTRAIKITPKGAIALVLILIIIFEIILHVVGPRIFG	(56)
AE000914	<i>M ther</i>	10	----MAKKDKKTLPPSGAGLVRYFEE-ETKGEKLTPEQVVVMSIILAVFCLVLRFSG----	(52)
RMJ09857	<i>M jann</i>	9	-----MSKREESTGLATSAGLIRYMDE-TFSKIRVKPEHVIGVTVAFVIIIEAILTYGRFL---	(53)
15920503	<i>S toko</i>	13	-MPSSKKKKSTVPLASMAGLIRYYEE-ENEKIKISPKLLIIISIMVAGVIVASILIPPP--	(58)
AE006662	<i>S solf</i>	11	-MPSSKKKKSTVPMVMAGLIRYYEE-ENEKVKISPKIVIGASLALTIIIVIVITKLF-----	(55)
RPK02491	<i>P aero</i>	12	--MARRRKYEGLNPFVAAGLIKFSSEGELEKIKLTPRAAVVISLAIIGLLIAINLLLPL--	(58)
RAP00437	<i>A pern</i>	13	-MSVRRRRRERRATPVTAAGLLSFYEE-YEGKIKISPTIVVGAAILVSAVVAABHIFLPAVP-	(59)
5803165	<i>H sapi</i>	49	-----SAGTGGMWRFYTE-DSPGLKVGPPVFLVMSLLFIASVFMLHIWGTKYTRS	(96)
13324684	<i>M musc</i>	49	-----SAGTGGMWRFYTE-DSPGLKVGPPVFLVMSLLFIAAVFMLHIWGTKYTRS	(96)
6002114	<i>D mela</i>	53	-----GAGTGGMWRFYTD-DSPGIRKVGPPVFLVMSLLFIASVFMLHIWGTKYNRS	(100)
14574310	<i>C eleg</i>	32	-----GGNNGGLWRFYTE-DSTGLKIGPPVFLVMSLVFIASVFVLHIWGTKFTRS	(81)
10697176	<i>Y lipo</i>	41	-----GGSSSTMLKLYTD-ESQGLKVDPPVVMVLSLGFIFSVVALHLLAKVSTK	(91)
6320857	<i>S cere</i>	40	-----GGSSSSILKLYTD-EANGFRVDSLVLFLSVGFIFSVIALHLLTKFTHI	(88)
6320932	<i>S cere</i>	33	-----TNSNNSILKIYSD-EATGLRVDPLVLFLAVGFIFSVVALHVISKVAGK	(82)

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

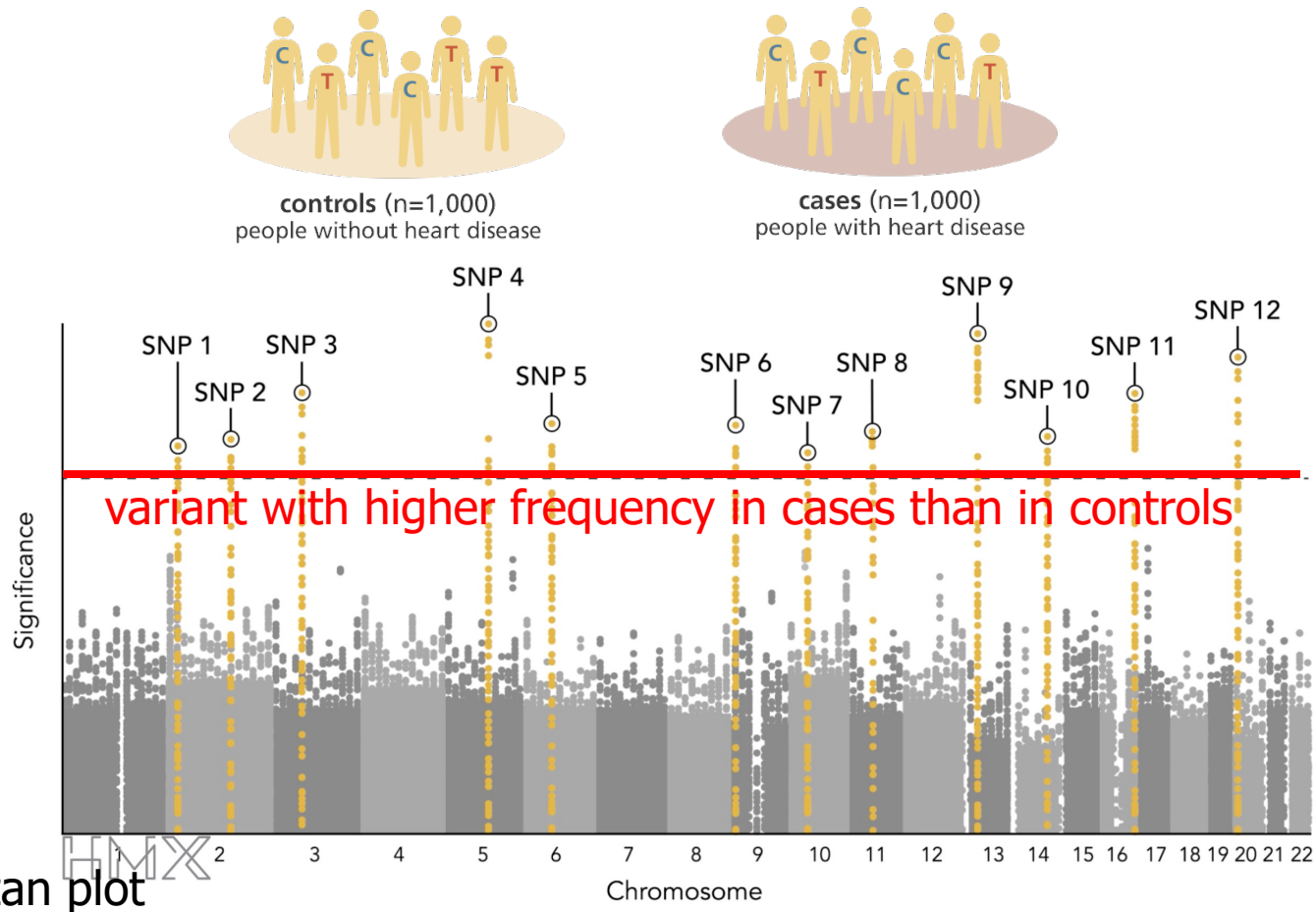
	SNP1	SNP2	Blood Pressure
Individual #1	...ACATG C CGACATTTCATA G GCC...		180
Individual #2	...ACATG C CGACATTTCATA A GCC...		175
Individual #3	...ACATG C CGACATTTCATA G GCC...		170
Individual #4	...ACATG C CGACATTTCATA A GCC...		165
Individual #5	...ACATG C CGACATTTCATA G GCC...		160
Individual #6	...ACATG C CGACATTTCATA G GCC...		145
Individual #7	...ACATG C CGACATTTCATA A GCC...		140
Individual #8	...ACATG C CGACATTTCATA A GCC...		130
Individual #9	...ACATG T CGACATTTCATA G GCC...		120
Individual #10	...ACATG T CGACATTTCATA A GCC...		120
Individual #11	...ACATG T CGACATTTCATA G GCC...		115
Individual #12	...ACATG T CGACATTTCATA A GCC...		110
Individual #13	...ACATG T CGACATTTCATA G GCC...		110
Individual #14	...ACATG T CGACATTTCATA A GCC...		110
Individual #15	...ACATG T CGACATTTCATA G GCC...		105
Individual #16	...ACATG T CGACATTTCATA A GCC...		100



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

Q Search

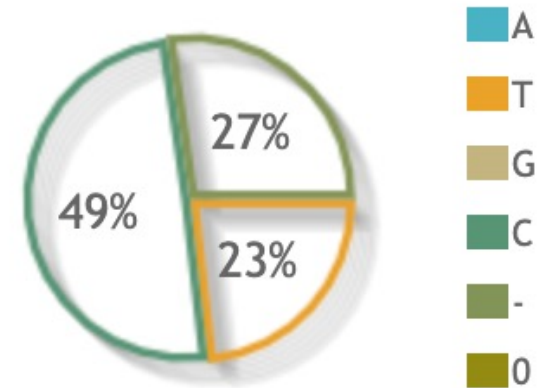
☰

SNP rs12979860

Basic Information

Name	rs12979860
Chromosome	19
Position	39248147
Weight of evidence	926

Allele Frequency



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

Much Larger Structural Variations



AUTISM

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



SCHIZOPHRENIA

McCarthy, *Nat Genet* 2009
Duplication of 593 kb



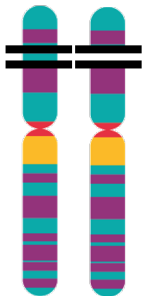
OBESITY

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. Kaplan, Richard Kronick, Matthew N. Bainbridge, Jennifer Friedman, Jeffrey J. Goepfert, 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](#)

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](#) 

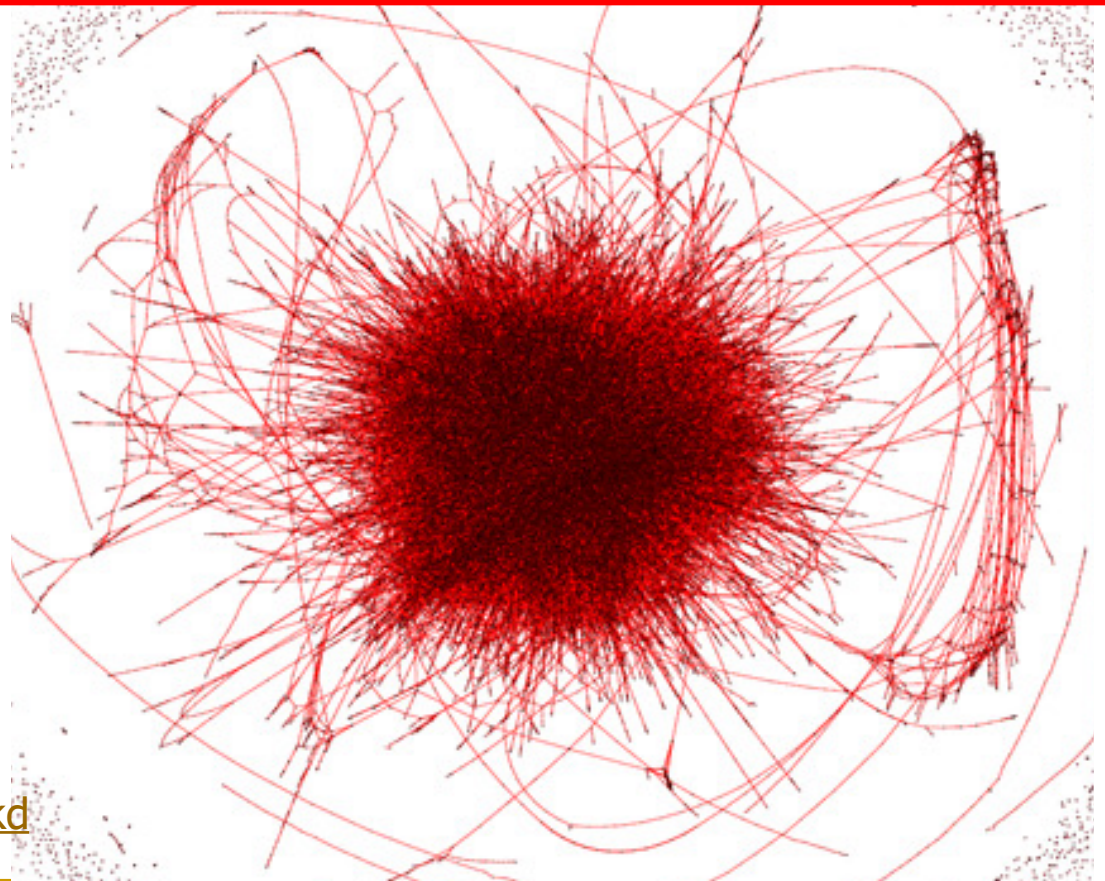
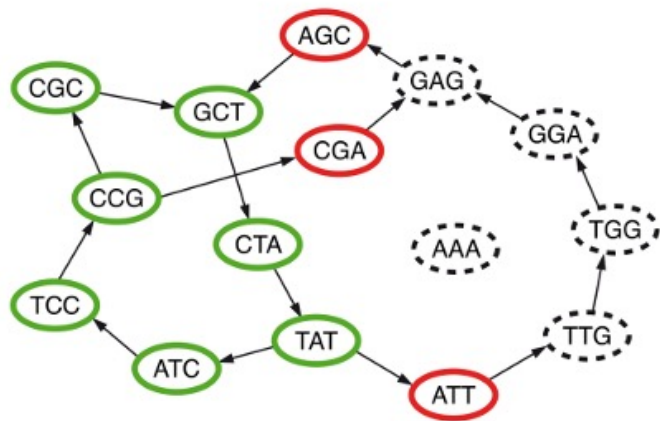
Nature Reviews Genetics **21**, 171–189(2020) | [Cite this article](#)

15k Accesses | **16** Citations | **309** Altmetric | [Metrics](#)

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

Metagenomics, genome assembly, de novo sequencing

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

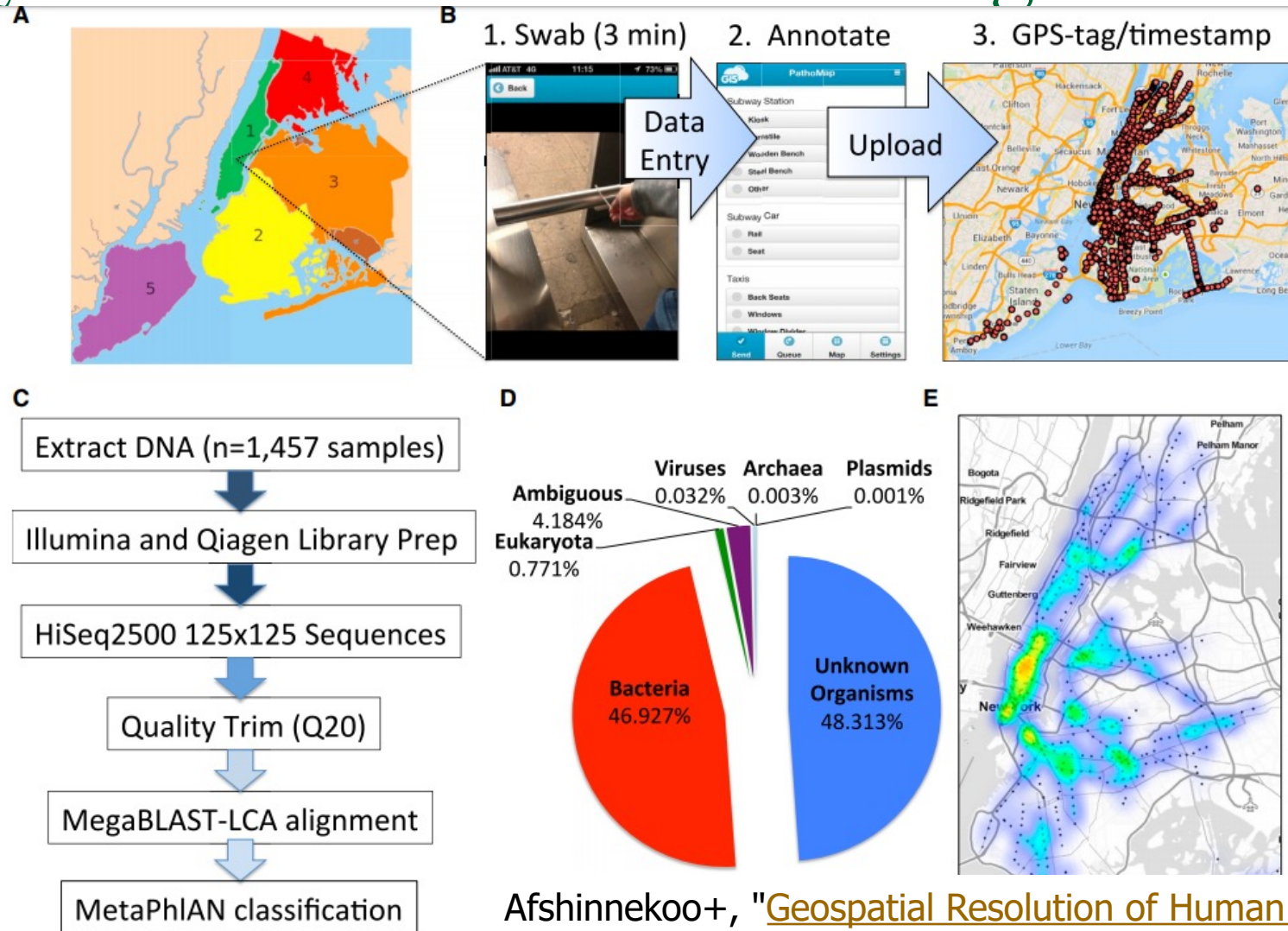


Figure 1. The Metagenome of New York City

(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

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

Global-Scale Microbiome Profiling

Cell Log in Register Su

ARTICLE | ONLINE NOW

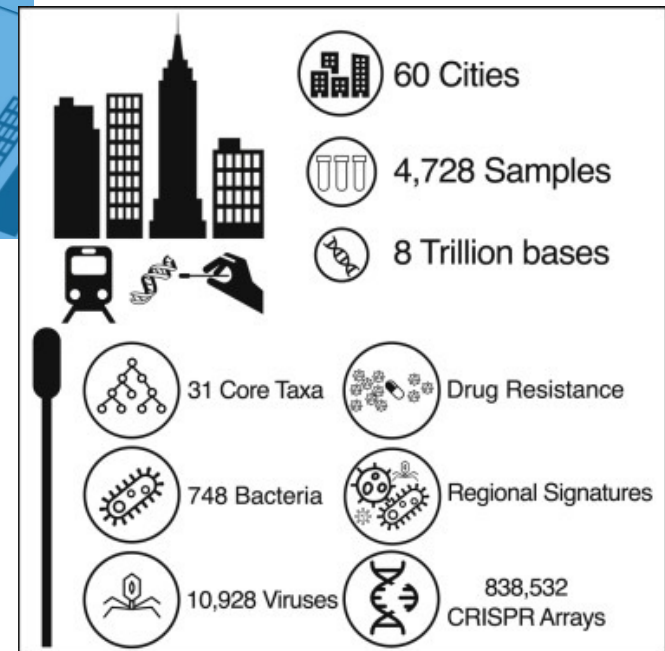
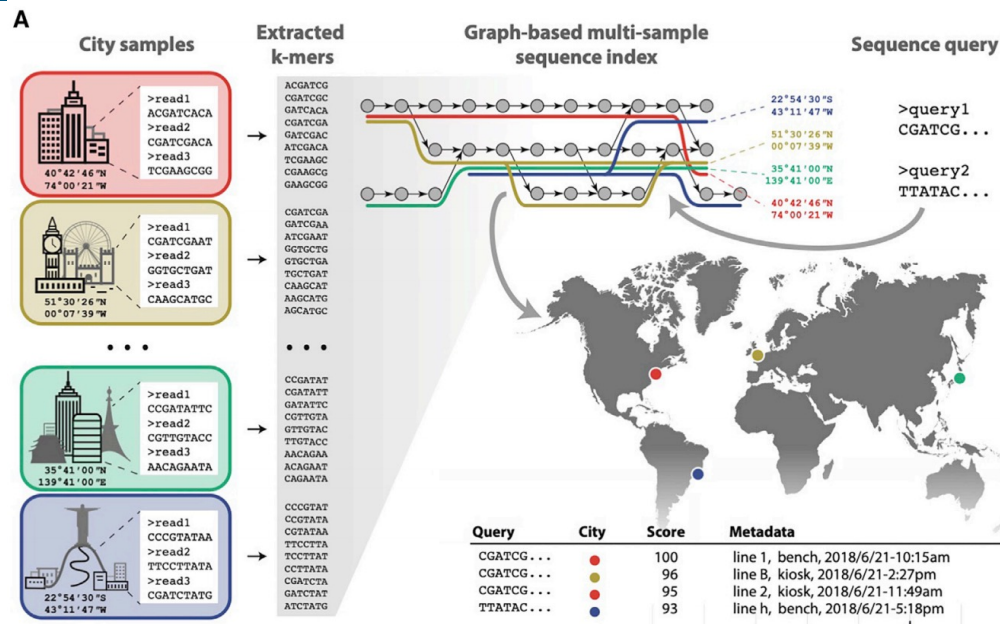
PDF [9 MB] Figures Save

A global metagenomic map of urban microbiomes and antimicrobial resistance

David Danko ⁶⁸ • Daniela Bezdán ⁶⁸ • Evan E. Afshin • ... Sibó Zhu • Christopher E. Mason ⁶⁹ 

The International MetaSUB Consortium • [Show all authors](#) • [Show footnotes](#)

Open Access • Published: May 26, 2021 • DOI: <https://doi.org/10.1016/j.cell.2021.05.002>



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.

Example: Scalable SARS-CoV-2 Testing



THE PREPRINT SERVER FOR HEALTH SCIENCES



HOME | ABOUT

Search

[Comments \(1\)](#)

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

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



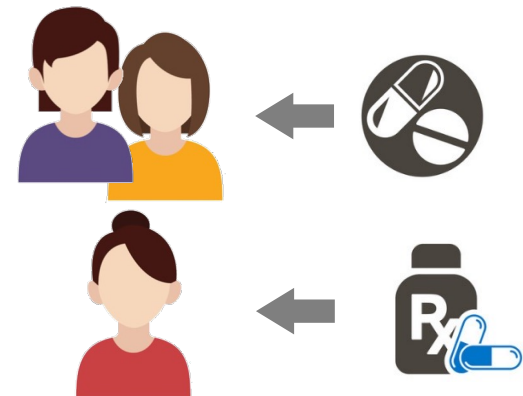
Understanding **genetic variations**,
species, **evolution**, ...



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



Rapid surveillance of **disease outbreaks**



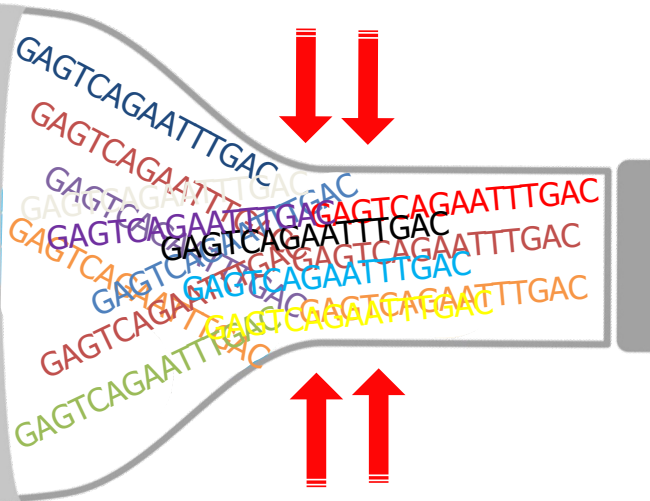
Developing **personalized medicine**

One Problem

[illegible]

Read Mapping 2

300 M
bases/min



2 M

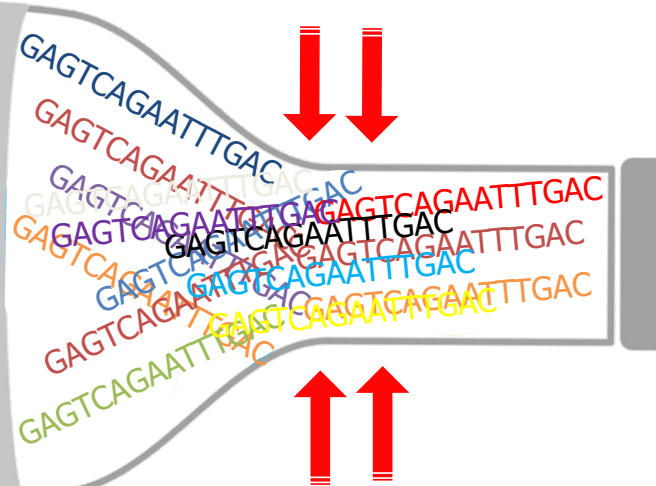
bases/min

(0.6%)

The Read Mapping Bottleneck

300 Million
bases/minute

Read Sequencing **



2 Million
bases/minute

Read Mapping *

150x slower

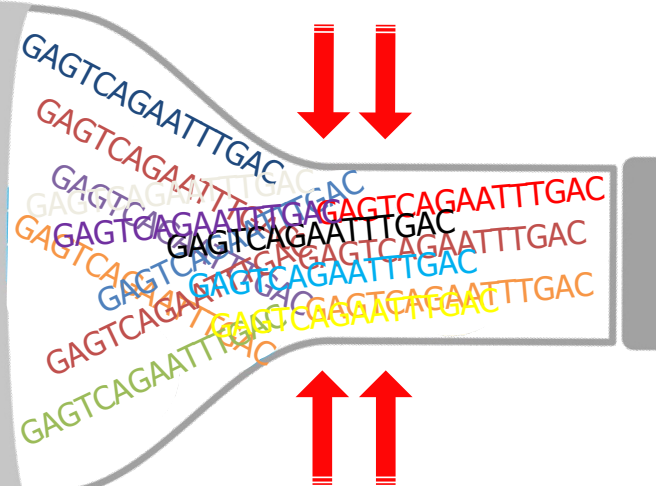
* BWA-MEM

** HiSeqX10, MinION

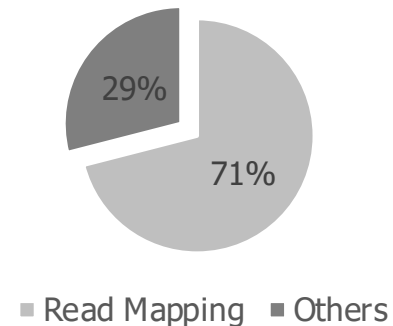
The Read Mapping Bottleneck

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

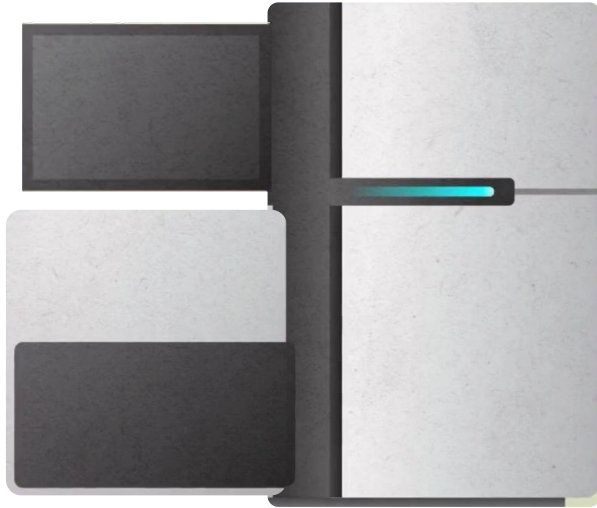
Illumina NovaSeq 6000



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

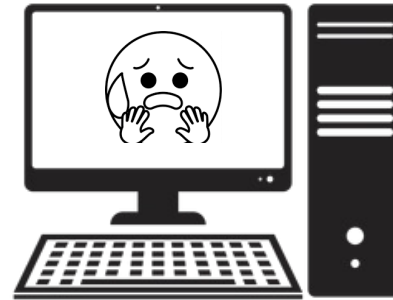
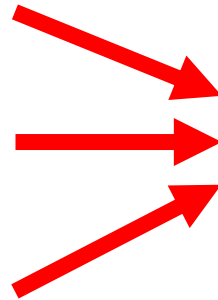


Lack of Specialized Compute Capability



Special-Purpose Machine
for **Sequencing**

FAST

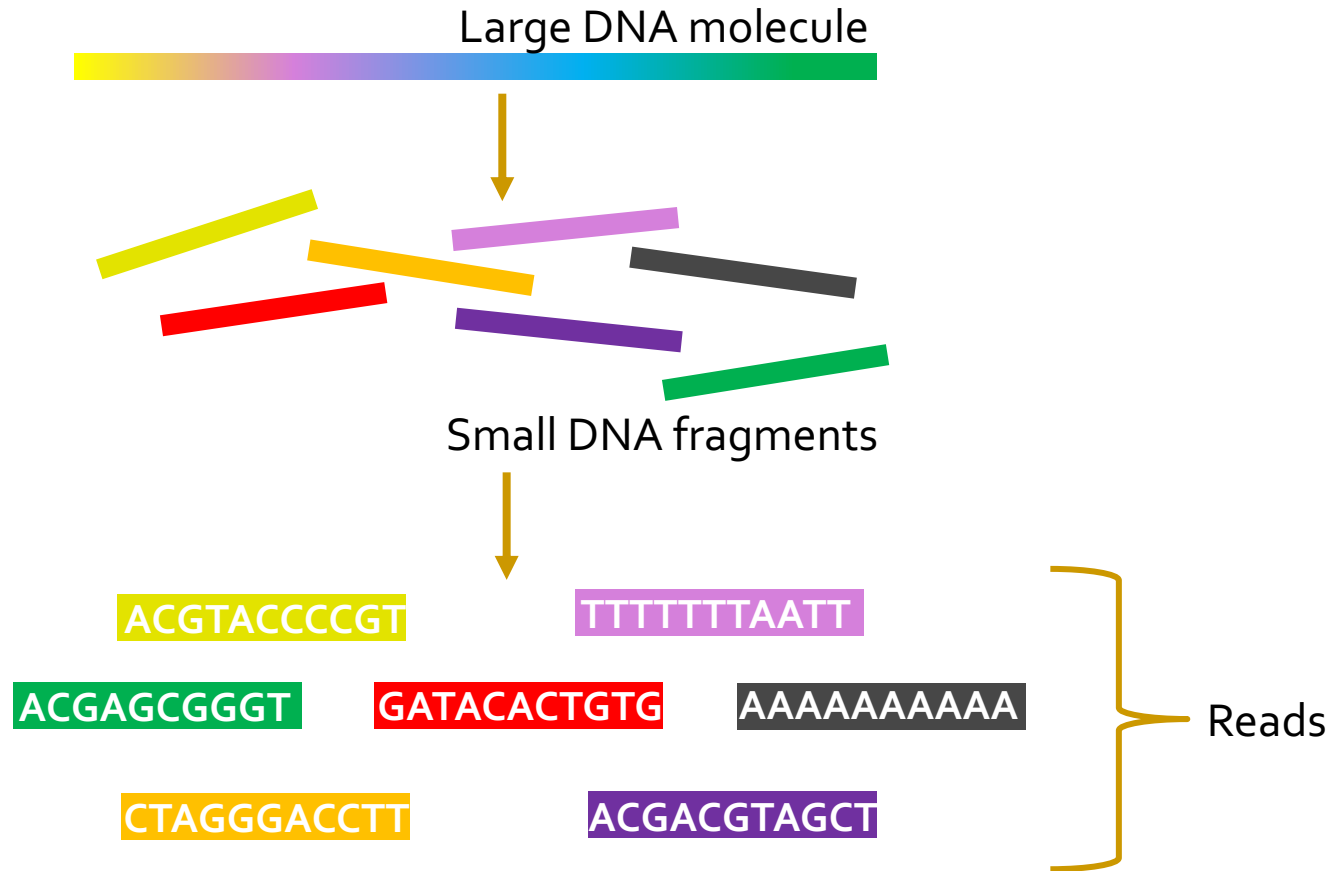


General-Purpose Machine
for **Analysis**

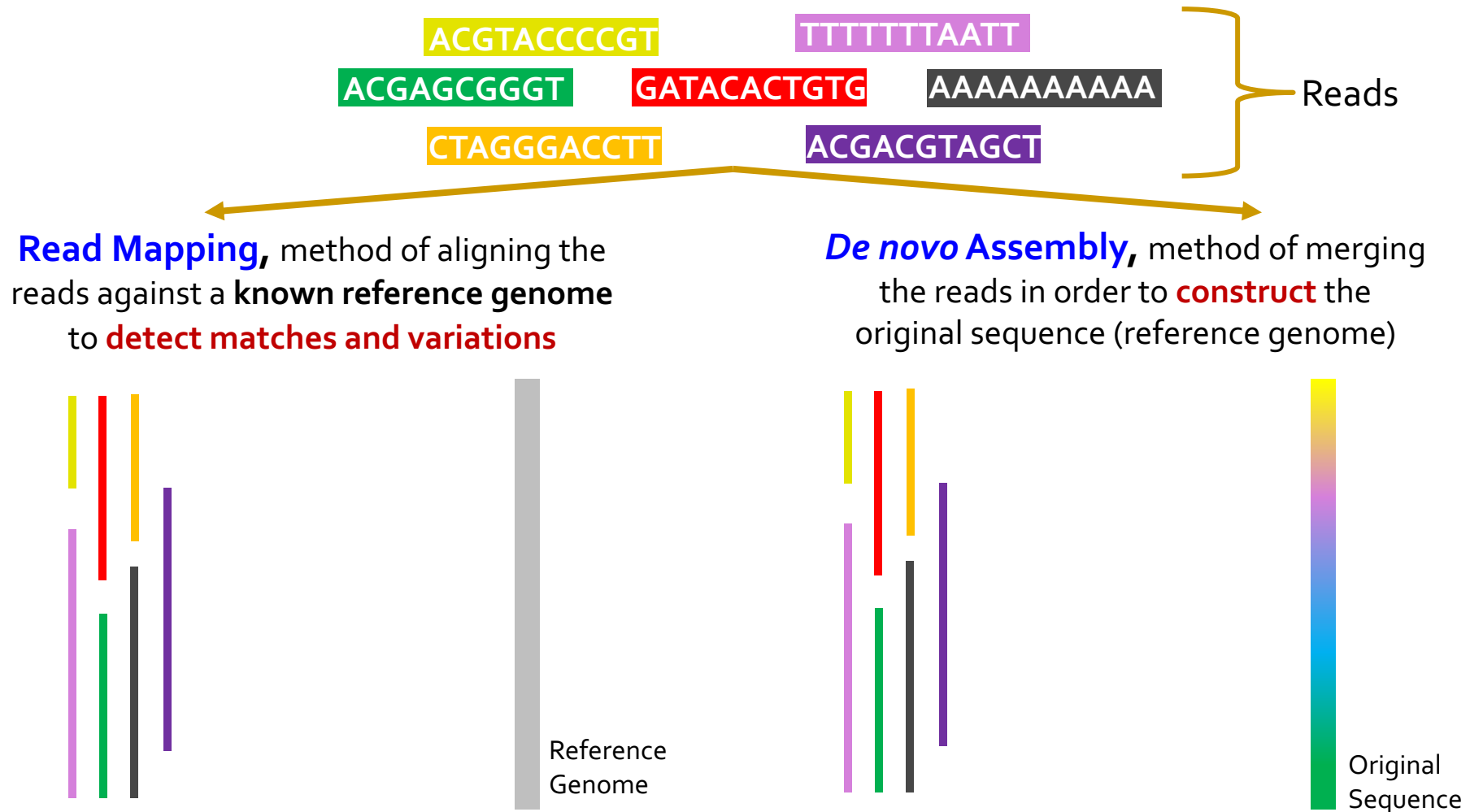
SLOW

**Need to construct
the entire genome
from many sequenced reads**

Genome Sequencing

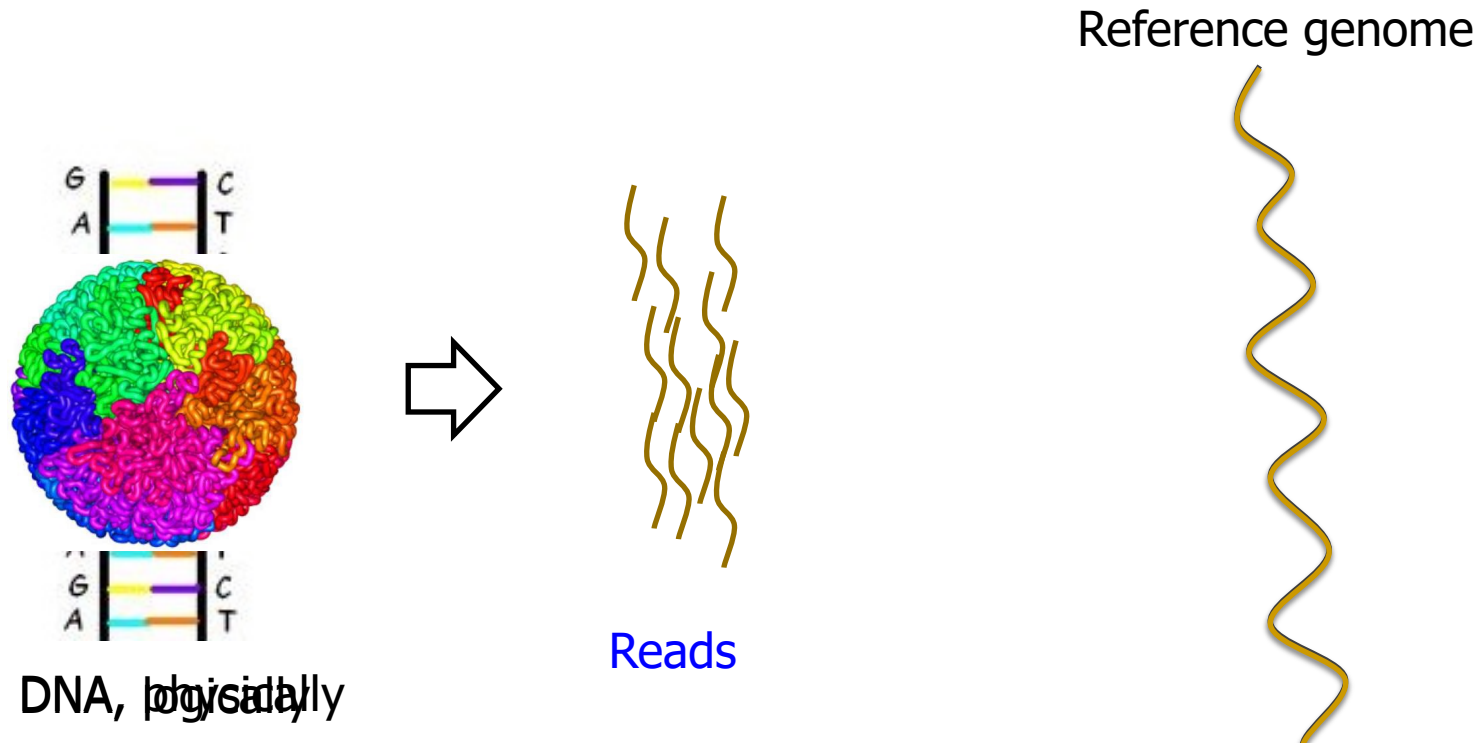


Genome Sequence Analysis



Read Mapping

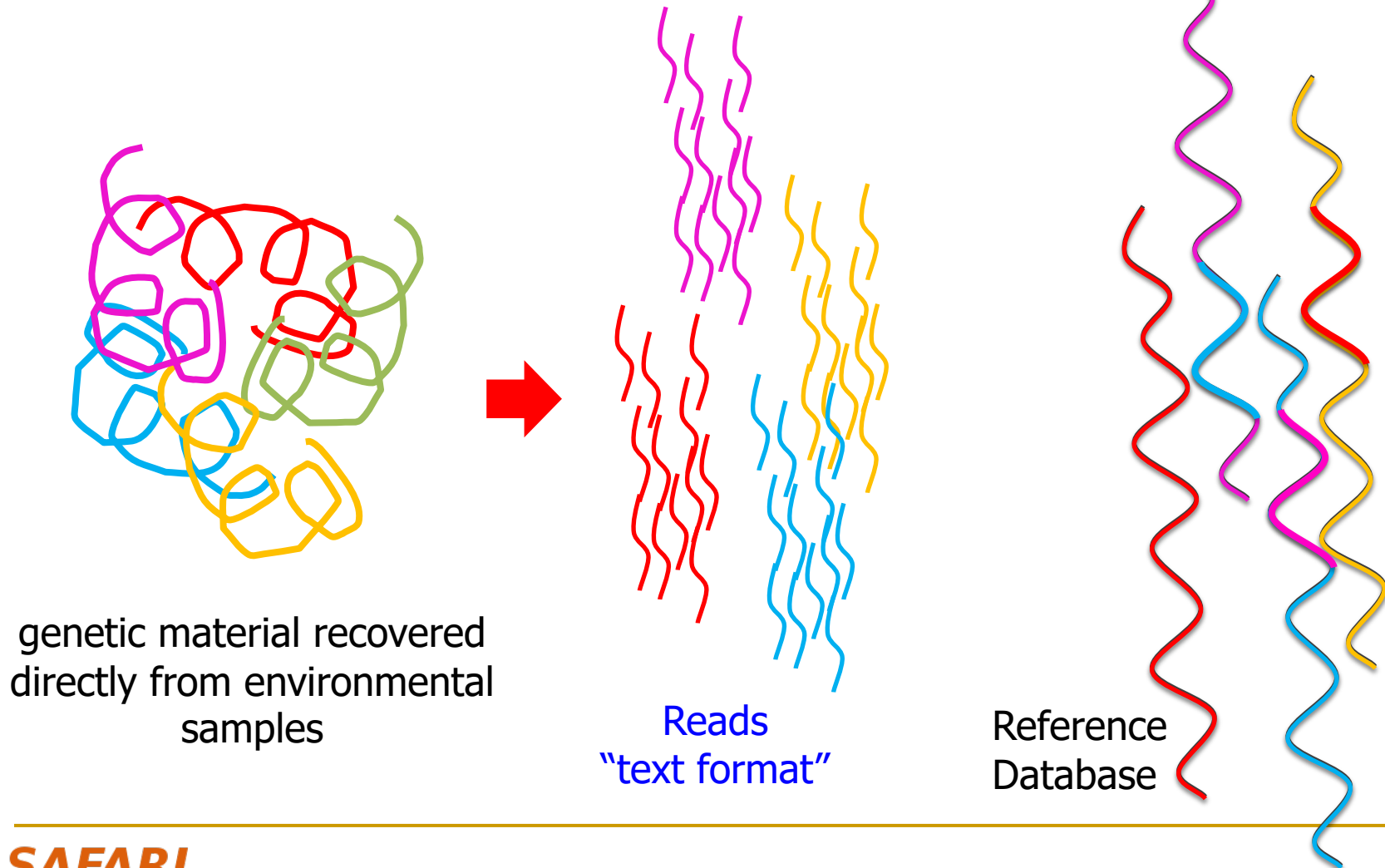
- Map many short DNA fragments (**reads**) to a known reference genome with some differences allowed



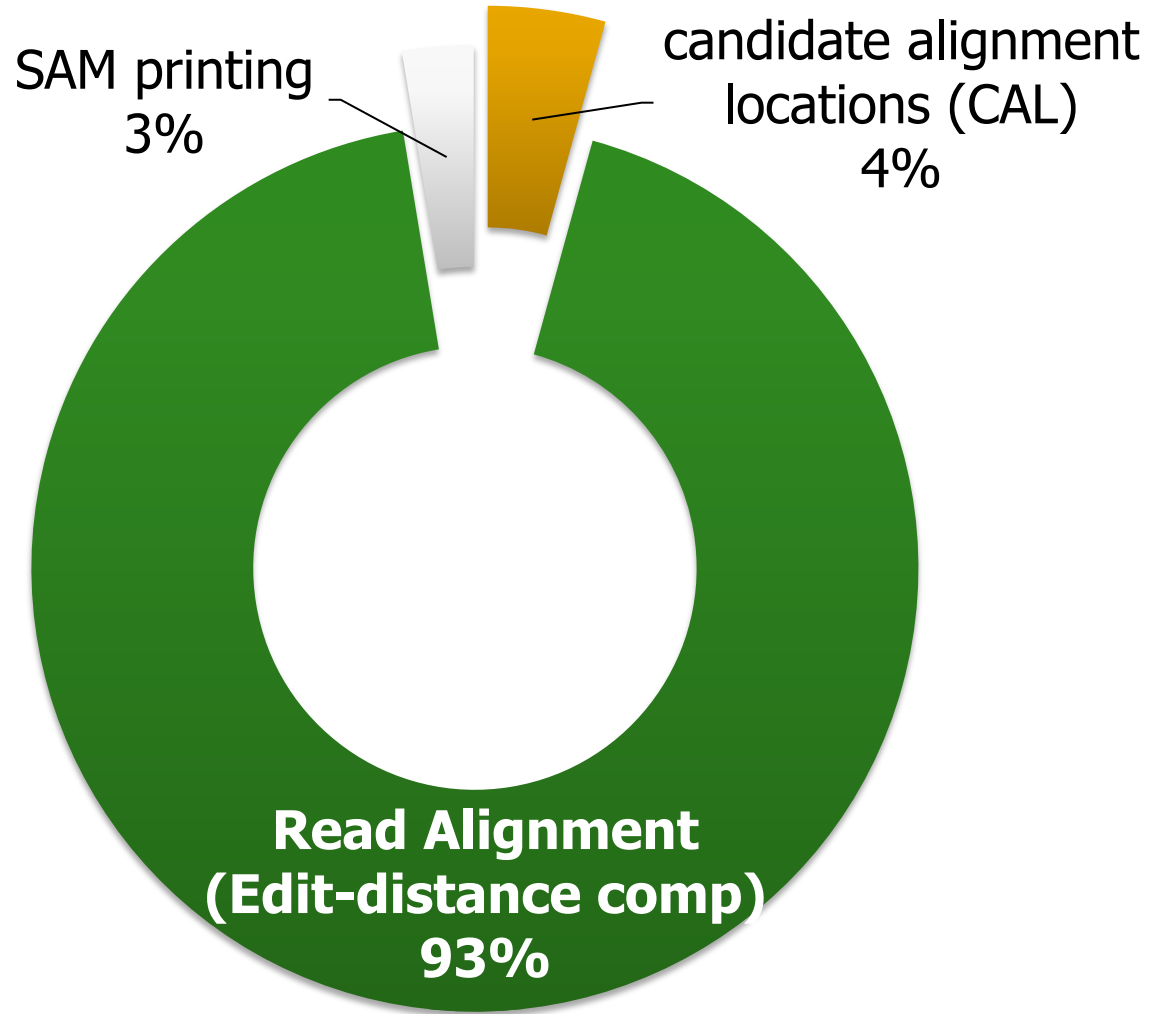
Mapping short reads to reference genome is challenging (billions of 50-300 base pair reads)

Read Mapping for Metagenomic Analysis

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



Read Mapping Execution Time (Old Times)



Matching Each Read to Reference Genome

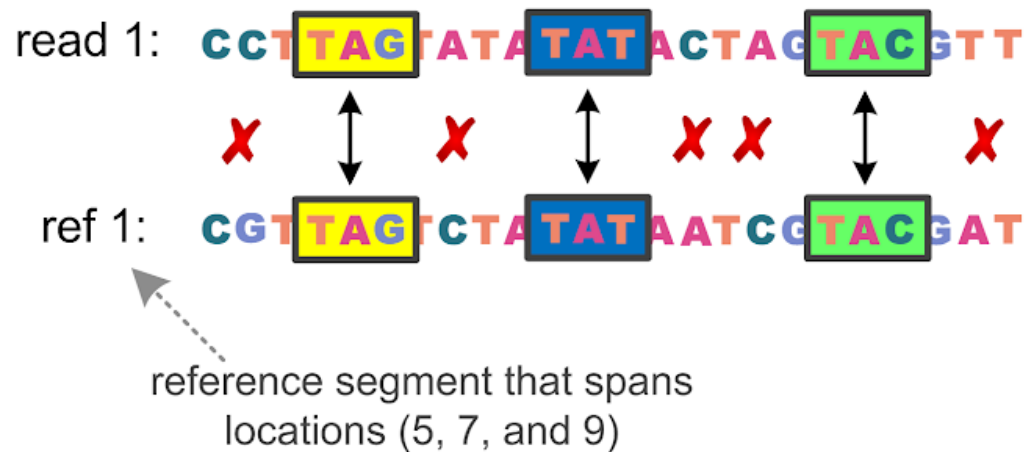
Reference Genome .FASTA file:

```
>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCC[T]TCATTGACATTTAAACTCTGGGGCAGG[T]GAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGGAACCGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACCGCC[T]CCCCGGCCCGGCTCGGGGCCCGCGGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCCGCCCAAGTGGCCCCGGGGCTTGATTTTTGCTTTTAAAAG
GAGGCATACAAAGATGGAAGCGAGTTACTGAGGGAGGGATAGGAAGGGGGGTGGAGGAGGGACTTGTCCTT
TCCCGAGTGT[T]CAAAAGTAGCA[T]CTCCTA[T]TCCAGTCCGGCCCT
GAGCTGGGAGTAGGGGGCGGGAGTCTGCTGCTGCTGTCTGCTAAAGCCACTCGCGACCGCGAAAAATGCA
GGAGGTGGGGACGCACTTTGCATCCAGACCTCCTCTGCATCGCAGTTC[T]CGCTTGGGAAAG
TCCGTACCCGCGCCT[T]AAAGACACCCTGCCGCGGGTCGGGCGAGGTGCAGCAGAAGTTTCCC
GCGGTTGCAAAGTGCAGATGGCTGGACCGCAACAAAGTCTAGAGATGGGGTTCGTTTCTCAGAAAGACGC
```

Sequenced Reads .FASTQ file:

```
@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
T[T]AATAAATCT[T]TTAGATN[T]NNNNNNNTAG
+HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
efcfffffcfeefffcfffffdddf`feed]`_]_Ba_^__[YBBBBBBBBBBRTT
```

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	-

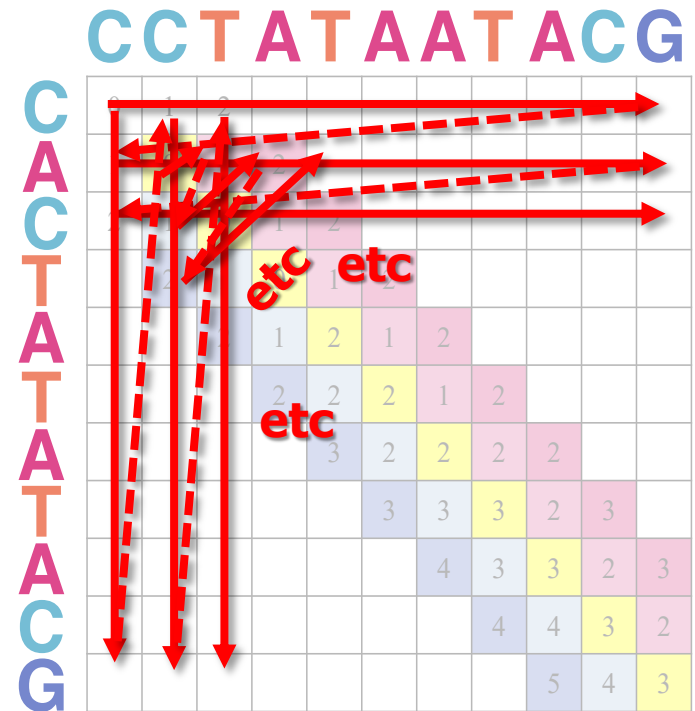
match
deletion
insertion
mismatch

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



Read Alignment

Example: Dynamic Programming Table

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											

immediate left,
upper left,
upper entries of its own



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

NETHERLANDS x S

- [NETHERLANDS x SW

C NETHERLANDS x SWI

NETERLANDS x SWIT

NETHERLANDS x SWITZ

NETHERLANDS x SWITZE

NETHERLANDS x SWITZER

NETHERLANDS x SWITZERL

- E NETHERLANDS x SWITZERLA

€ NETHERLANDS x SWITZERLAN

C 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	3	4	5	6	7	8	9	10	11		
I	3	3	4	5	6	7	8	9	10	11			
T	4	4	5	6	7	8	9	10	11				
Z	5	5	6	7	8	9	10	11					
E	6	6	7	8	9	10	11						
R	7	7	8	9	10	11							
L	8	8	9	10	11								
A	9	9	10	11									
N	10	10	11										
D	11	11											

Computational Cost is Mathematically Proven

arXiv.org > cs > arXiv:1412.0348

Search...

Help | Advanced

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.

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

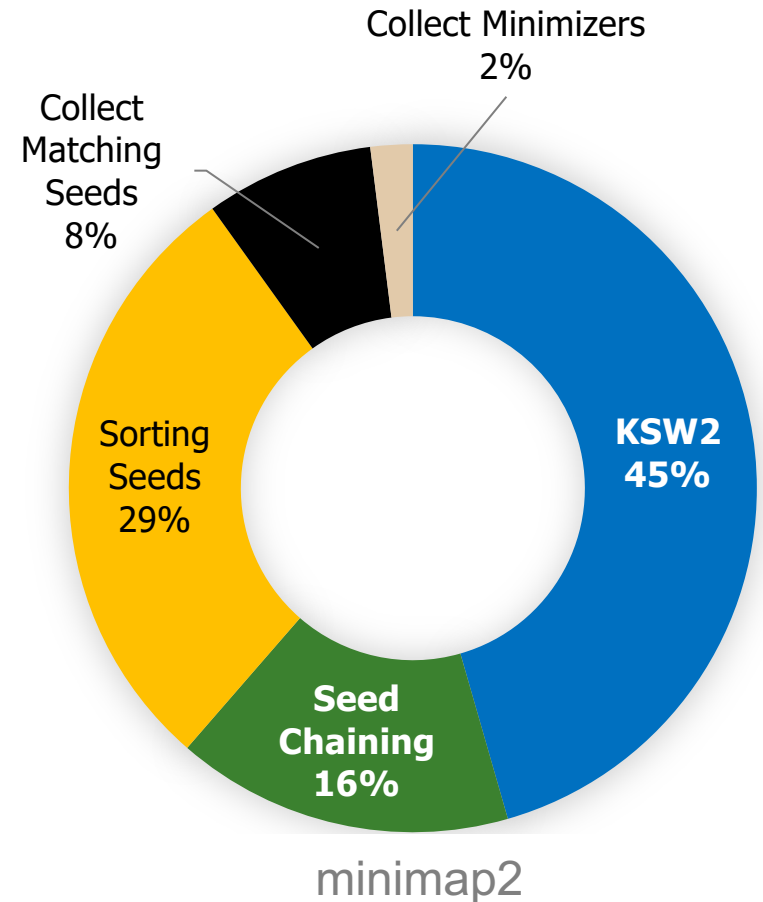
Technology dictates algorithms: recent developments in read alignment



Mohammed Alser^{1,2,3†}, Jeremy Rotman^{4†}, Dhrithi Deshpande⁵, Kodi Taraszka⁴, Huwenbo Shi^{6,7}, Pelin Icer Baykal⁸, Harry Taegyun Yang^{4,9}, Victor Xue⁴, Sergey Knyazev⁸, Benjamin D. Singer^{10,11,12}, Brunilda Balliu¹³, David Koslicki^{14,15,16}, Pavel Skums⁸, Alex Zelikovsky^{8,17}, Can Alkan^{2,18}, Onur Mutlu^{1,2,3†} and Serghei Mangul^{5*†} 

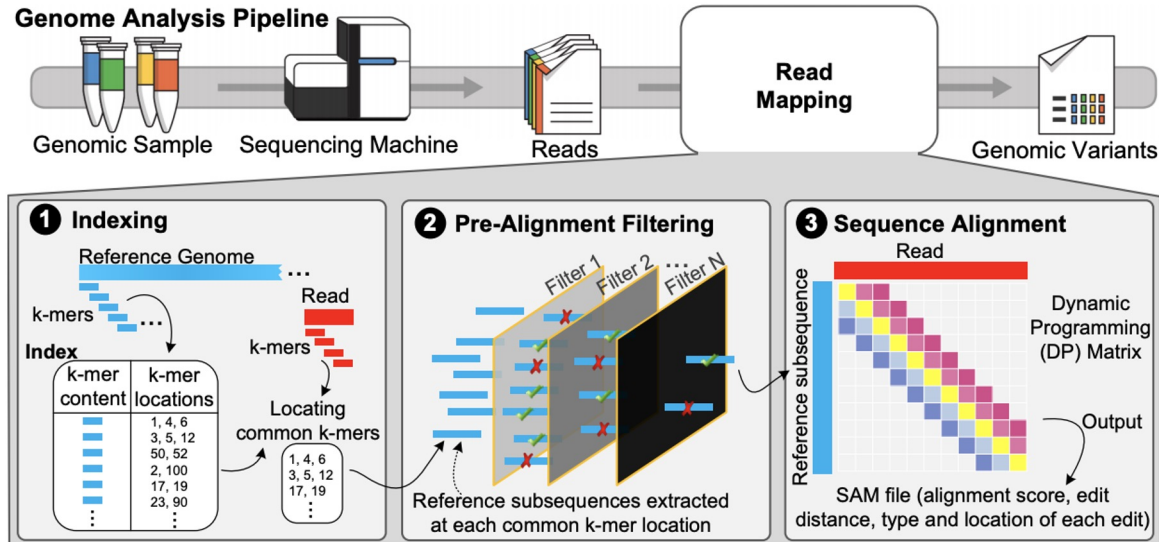
Read Mapping Execution Time (Modern)

>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

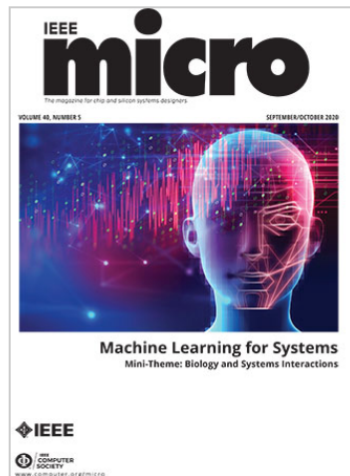
Alser+, "[Accelerating Genome Analysis: A Primer on an Ongoing Journey](#)", IEEE Micro, 2020.

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 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

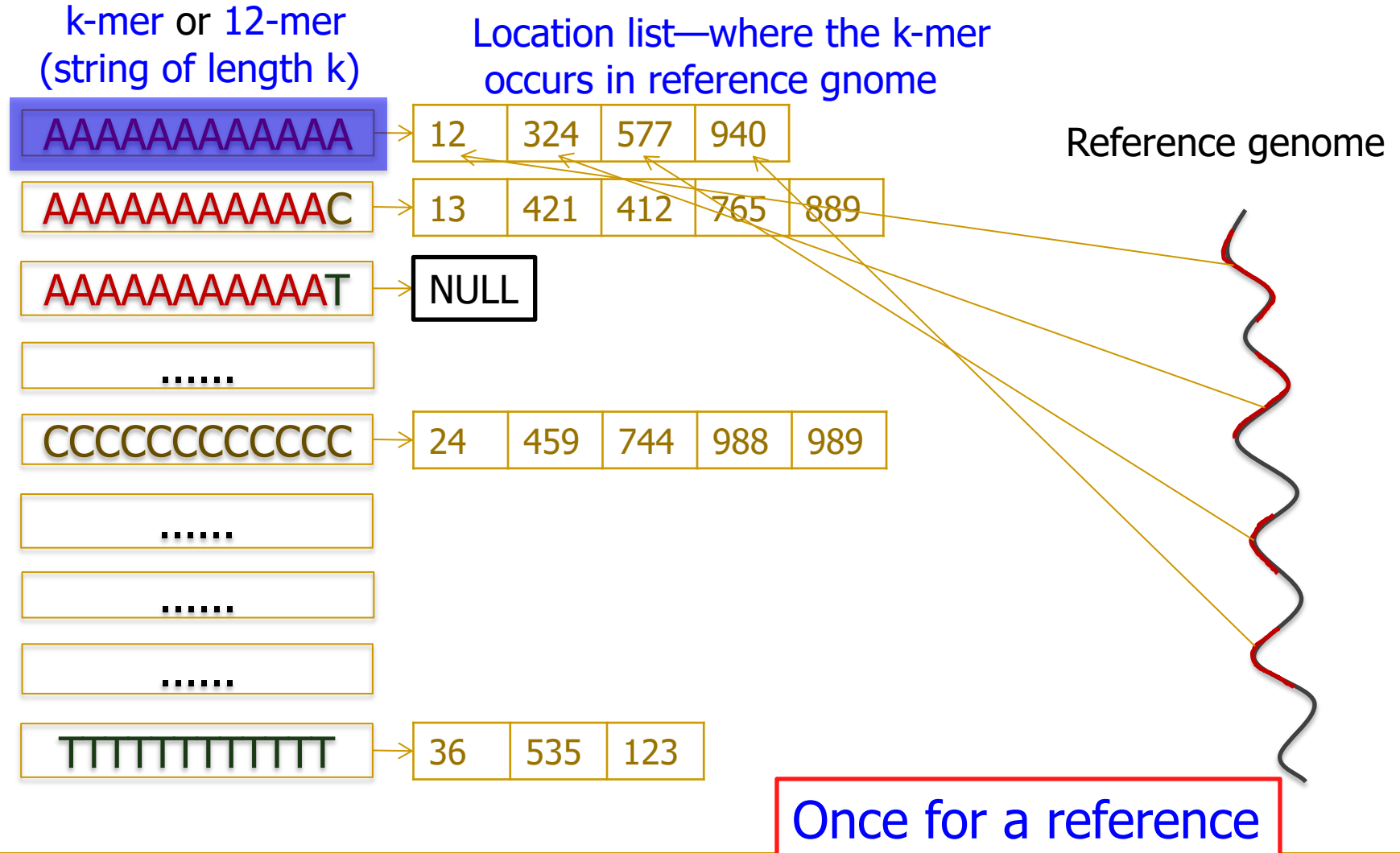
Read Mapping Algorithms: Two Styles

- 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

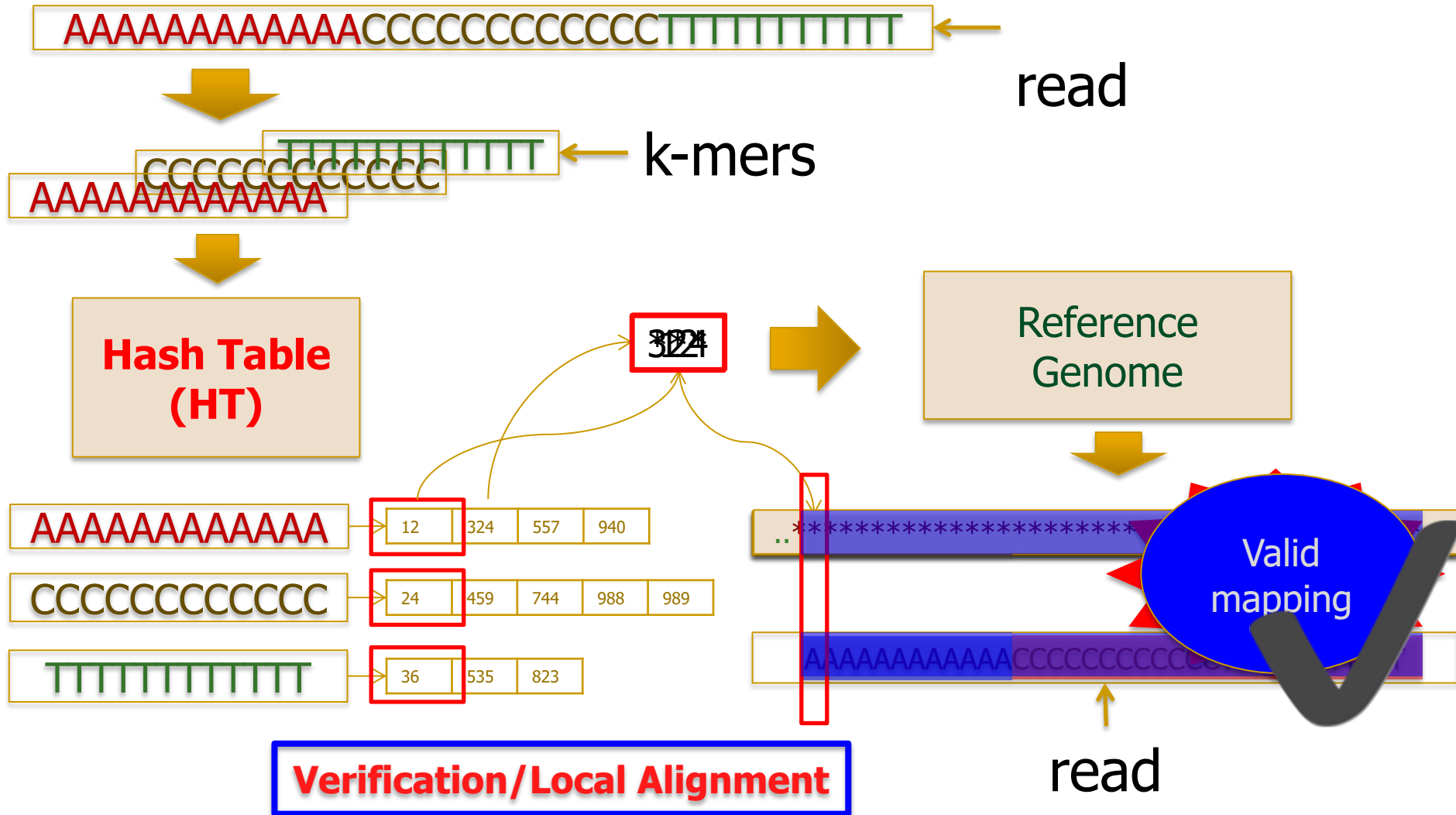
Hash Table-Based Mappers [Alkan+ Nature Gen'09]



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 *a//* mappings → sensitive
- + Can tolerate up to *e* errors

nature
genetics

<http://mrfast.sourceforge.net/>

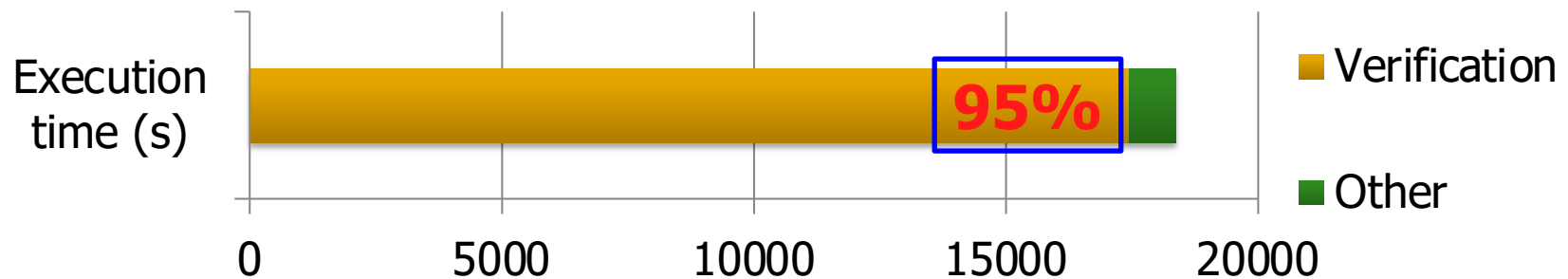
Personalized copy number and segmental duplication maps using next-generation sequencing

Can Alkan^{1,2}, Jeffrey M Kidd¹, Tomas Marques-Bonet^{1,3}, Gozde Aksay¹, Francesca Antonacci¹, Fereydoun Hormozdiari⁴, Jacob O Kitzman¹, Carl Baker¹, Maika Malig¹, Onur Mutlu⁵, S Cenk Sahinalp⁴, Richard A Gibbs⁶ & 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)



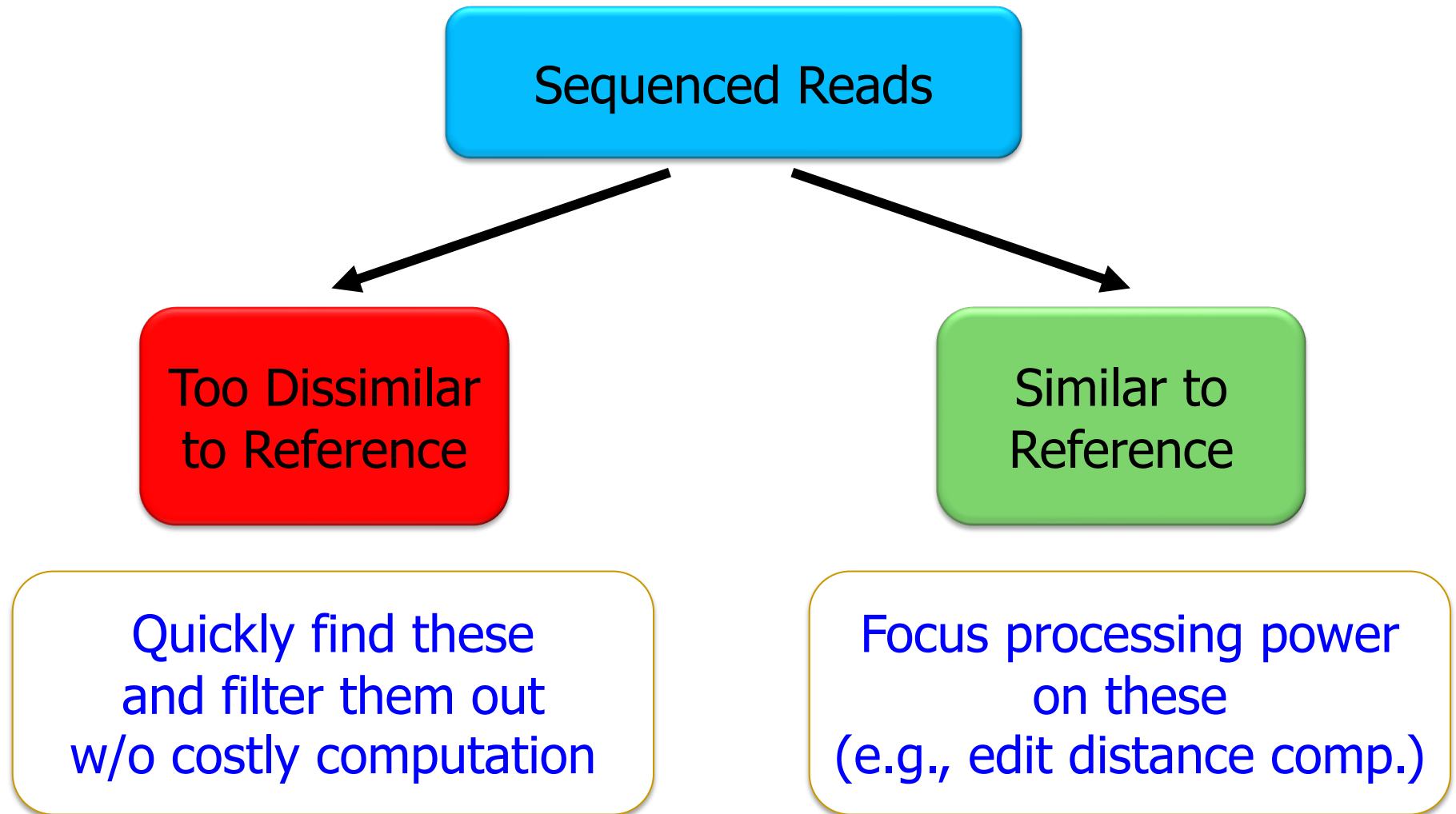
- **Goal: Speed up the mapper by reducing the cost of verification**

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, Zülal Bingöl, 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>



PROCEEDINGS

Open Access

Accelerating read mapping with FastHASH

Hongyi Xin¹, Donghyuk Lee¹, Farhad Hormozdiari², Samihan Yedkar¹, Onur Mutlu^{1*}, Can Alkan^{3*}

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

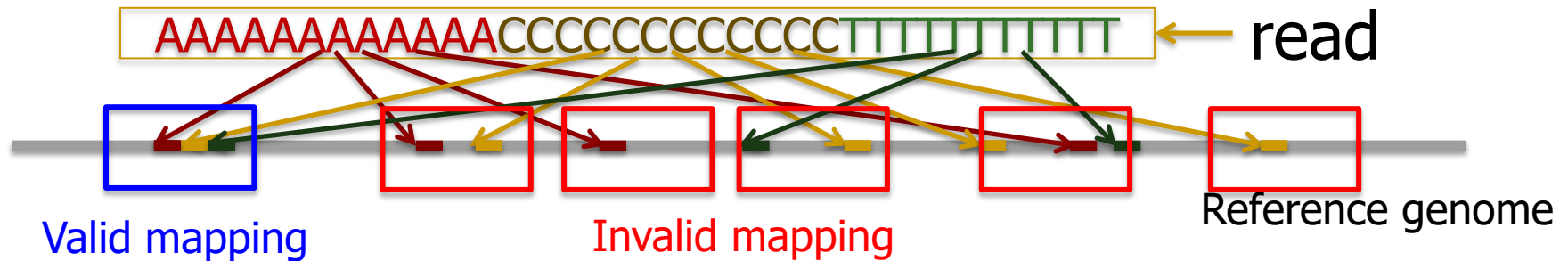
- Some k-mers are **cheaper** to verify than others because they have shorter location lists (they occur less frequently in the reference genome)
 - Mapper needs to examine only $e+1$ k-mers' locations to tolerate e errors
- Read mapper can choose the cheapest $e+1$ k-mers and verify their locations

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

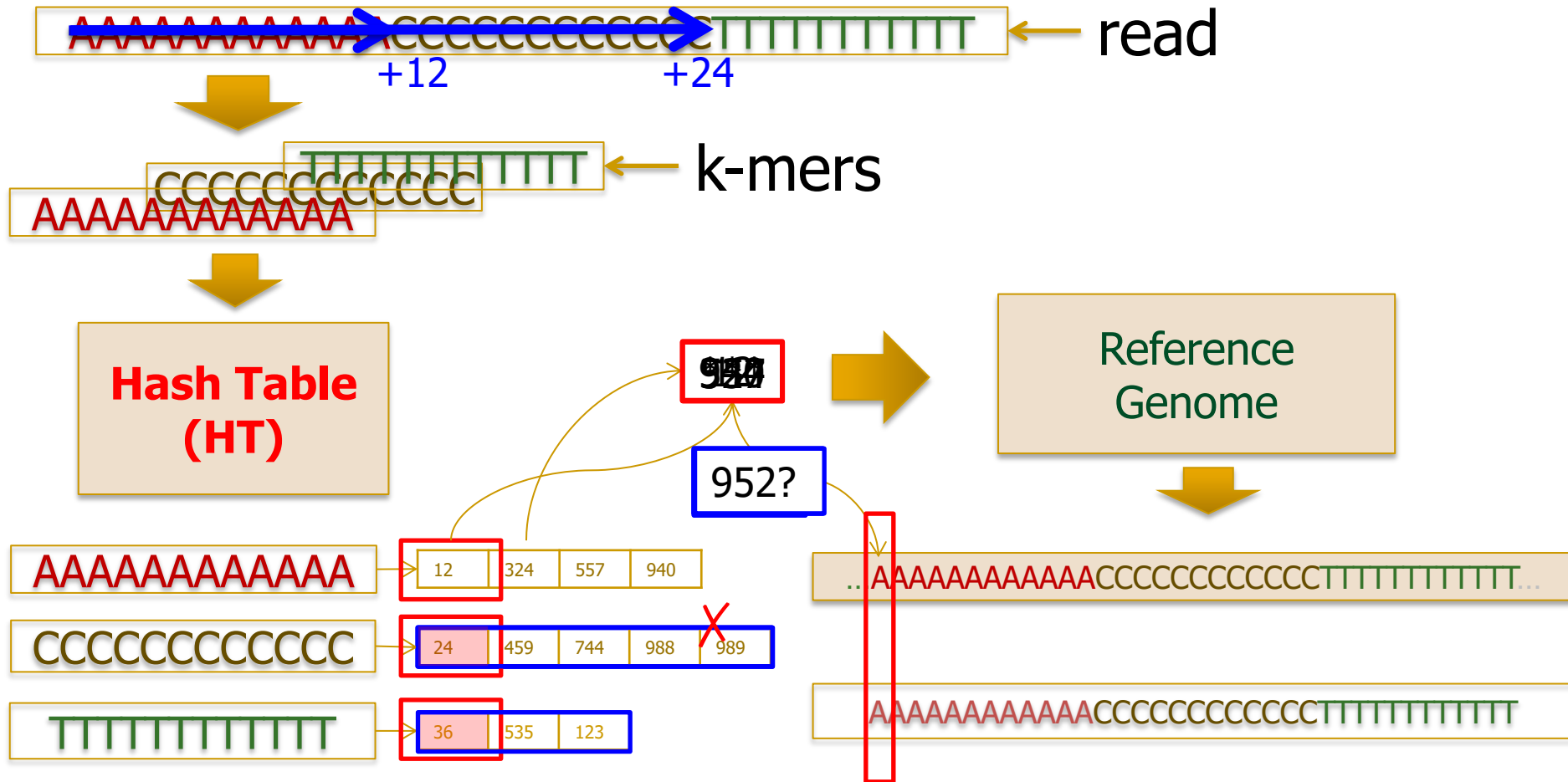
Adjacency Filtering (AF)

- **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
 - If more than e k-mers fail $\rightarrow$ there must be more than e errors $\rightarrow$ 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

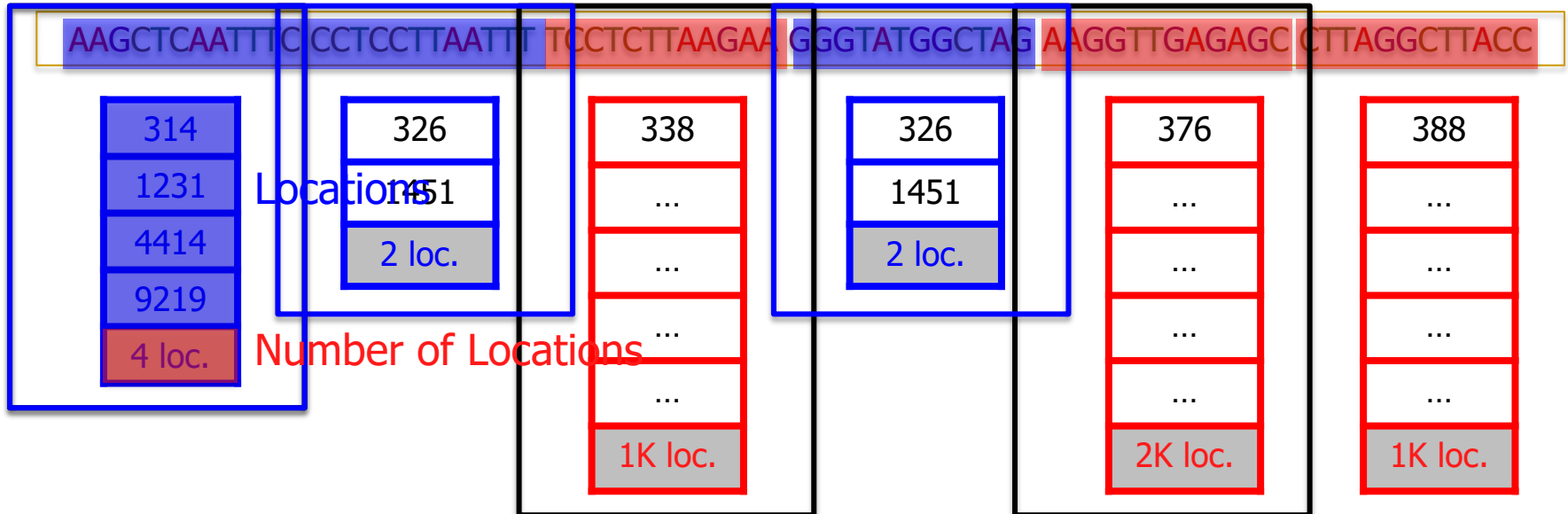
Cheap K-mer Selection (CKS)

- **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)

read



Expensive 3 k-mers

Previous work needs to verify:

3004 locations

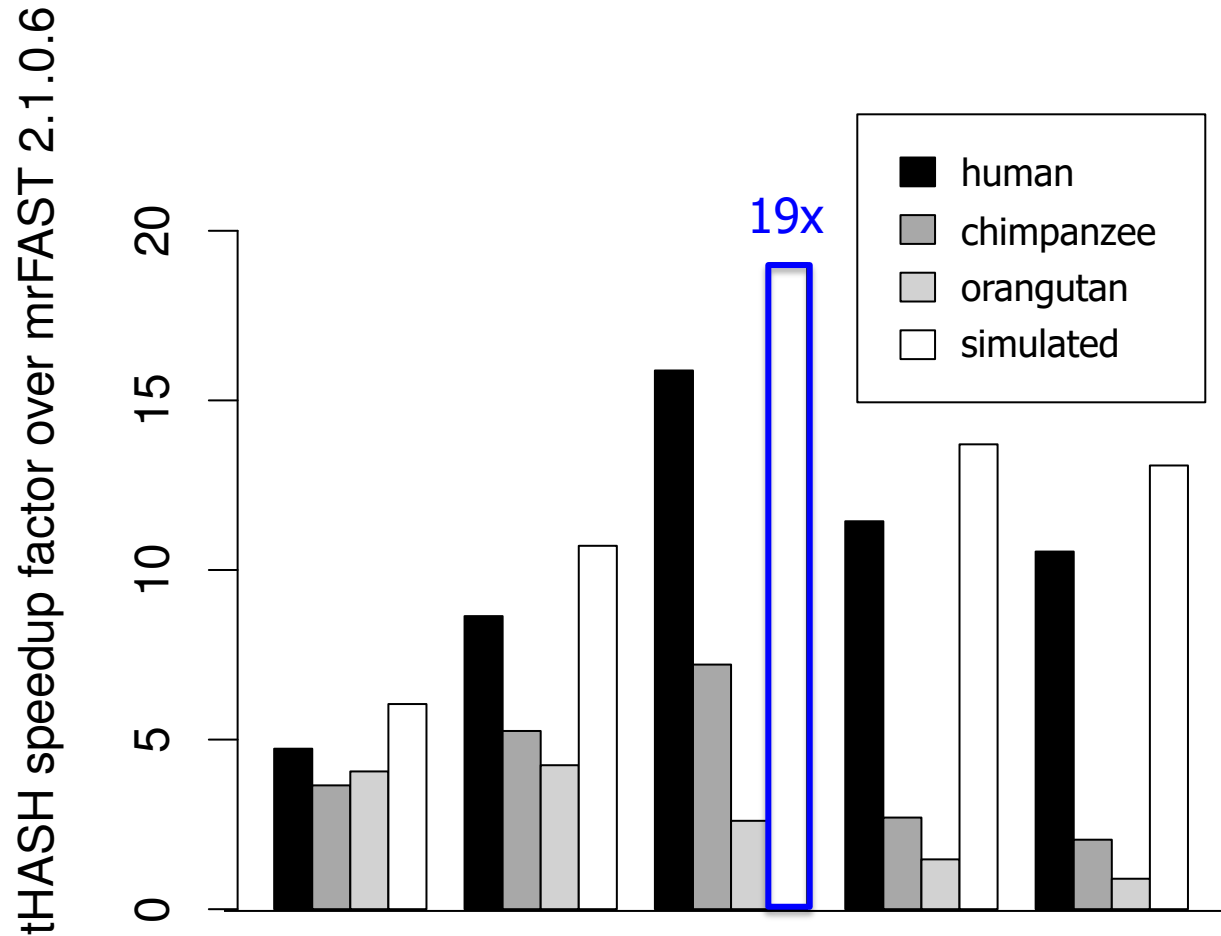
FastHASH verifies only:

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

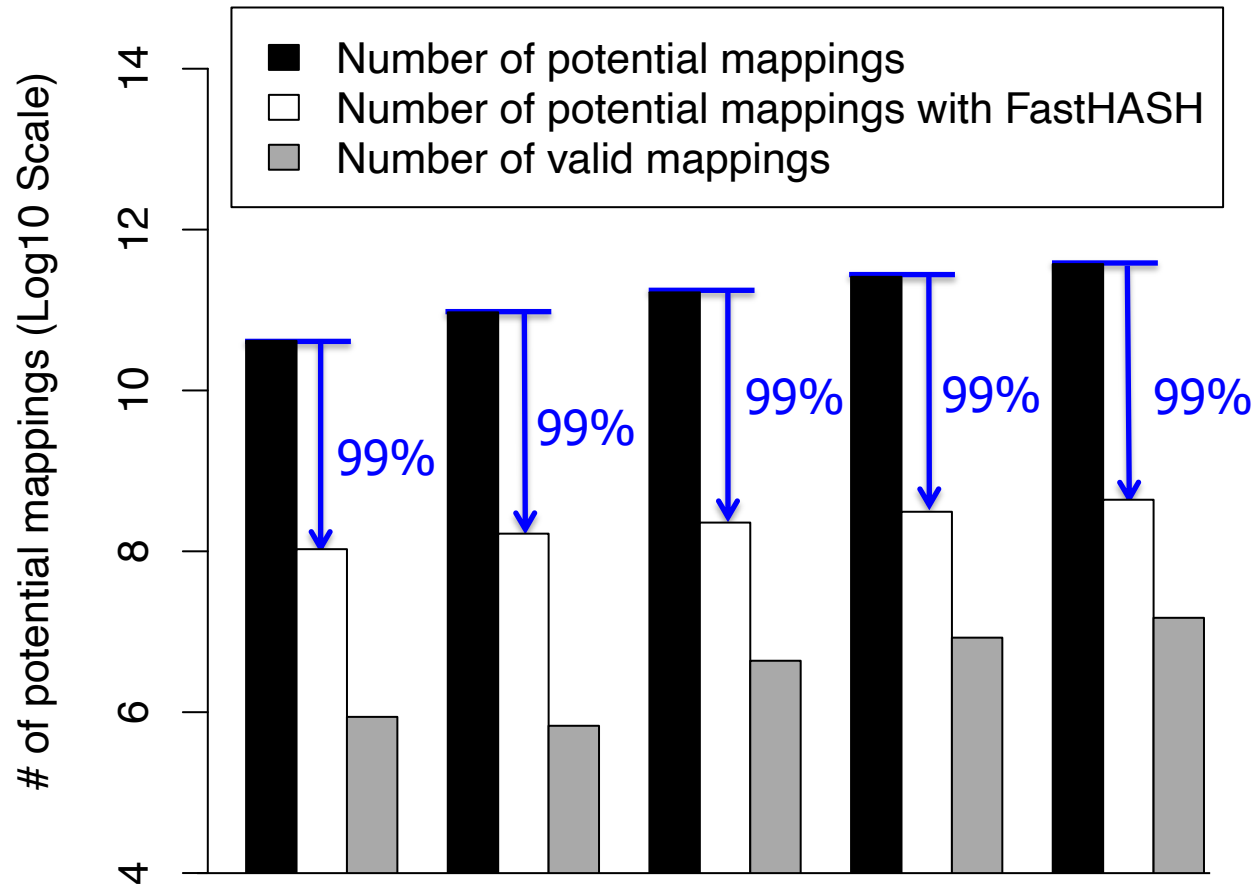
FastHASH Speedup: Entire Read Mapper



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>



PROCEEDINGS

Open Access

Accelerating read mapping with FastHASH

Hongyi Xin¹, Donghyuk Lee¹, Farhad Hormozdiari², Samihan Yedkar¹, Onur Mutlu^{1*}, Can Alkan^{3*}

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

OXFORD

Sequence analysis

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

Hongyi Xin^{1,*}, John Greth², John Emmons², Gennady Pekhimenko¹,
Carl Kingsford³, 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:

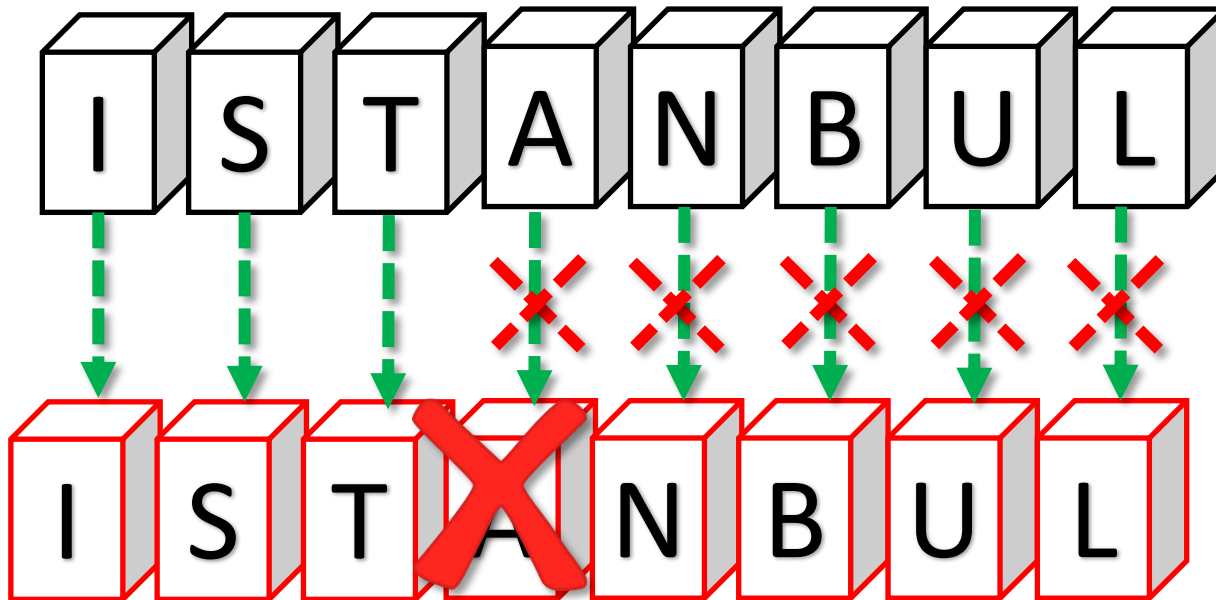
- SHD is 3x faster than SeqAn (the best implementation of Gene Myers’ bit-vector algorithm), with only a 7% false positive rate
 - The **fastest CPU-based filtering (pre-alignment) mechanism**
-

Hamming Distance ($\Sigma \oplus$)

3 matches

5 mismatches

Edit = 1 Deletion

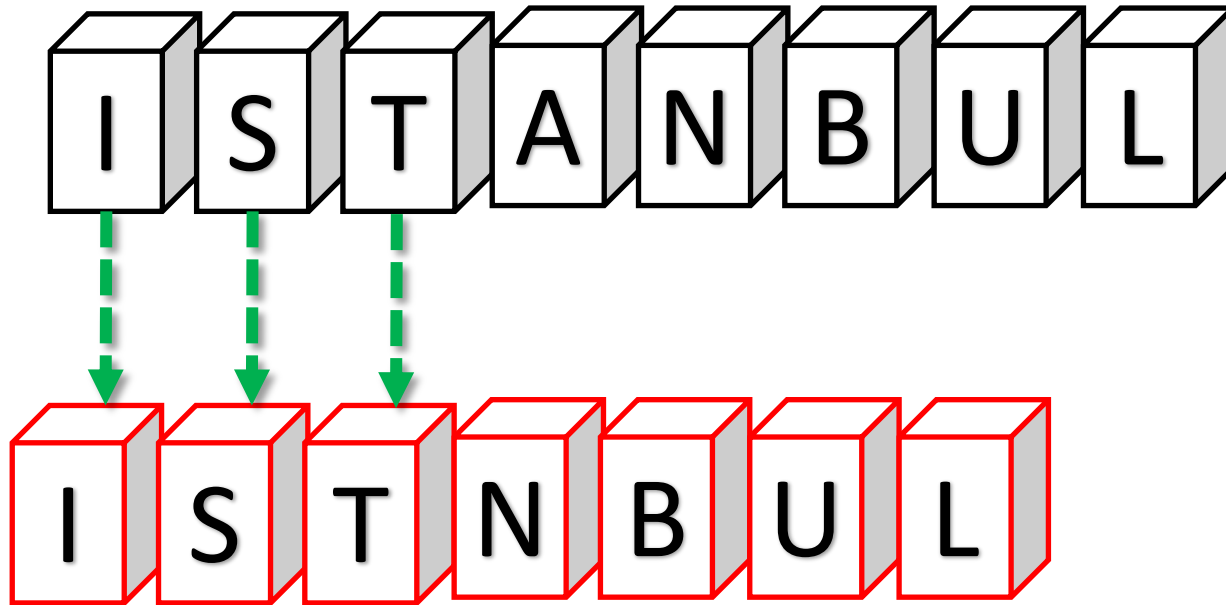


To cancel the effect of a deletion, we need to shift in the *right* direction

Insight: Shifting a String Helps Similarity Search

3 matches

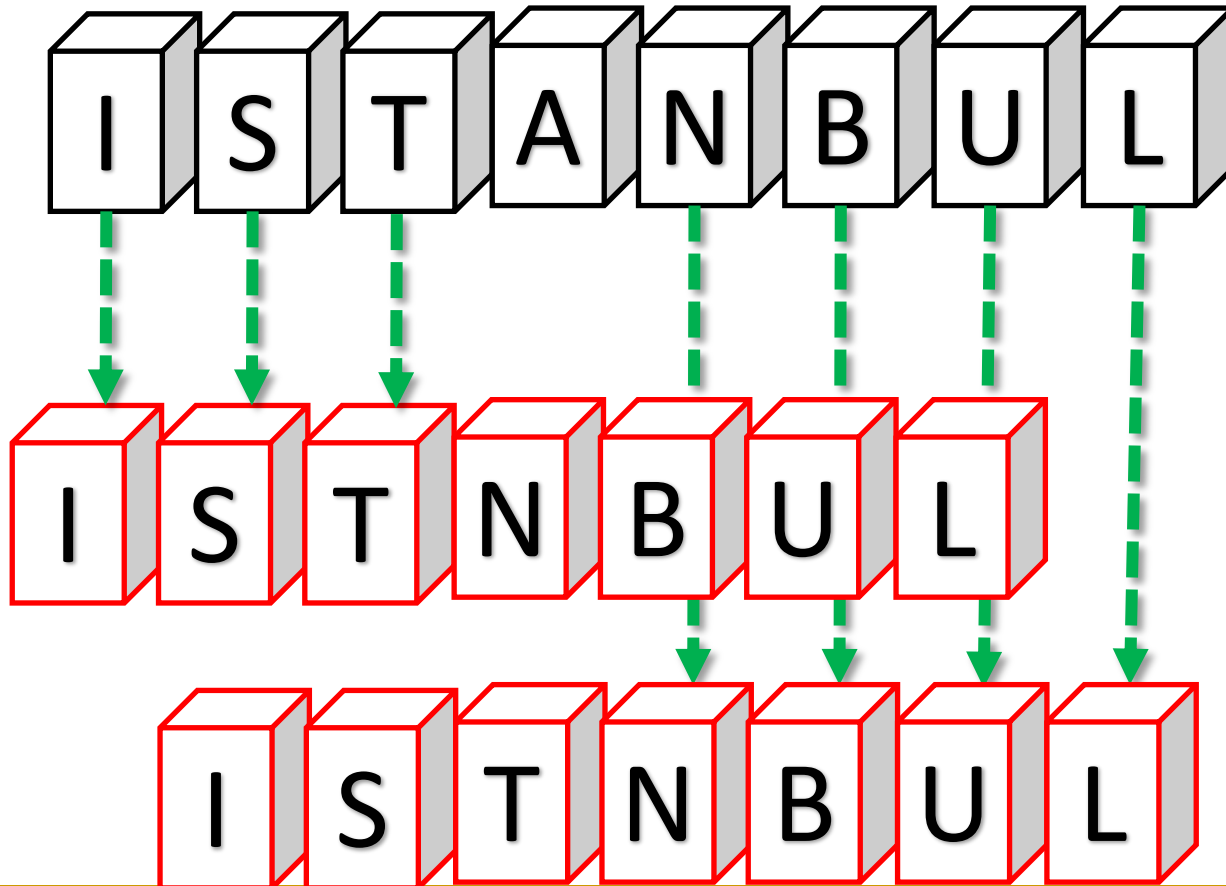
5 mismatches



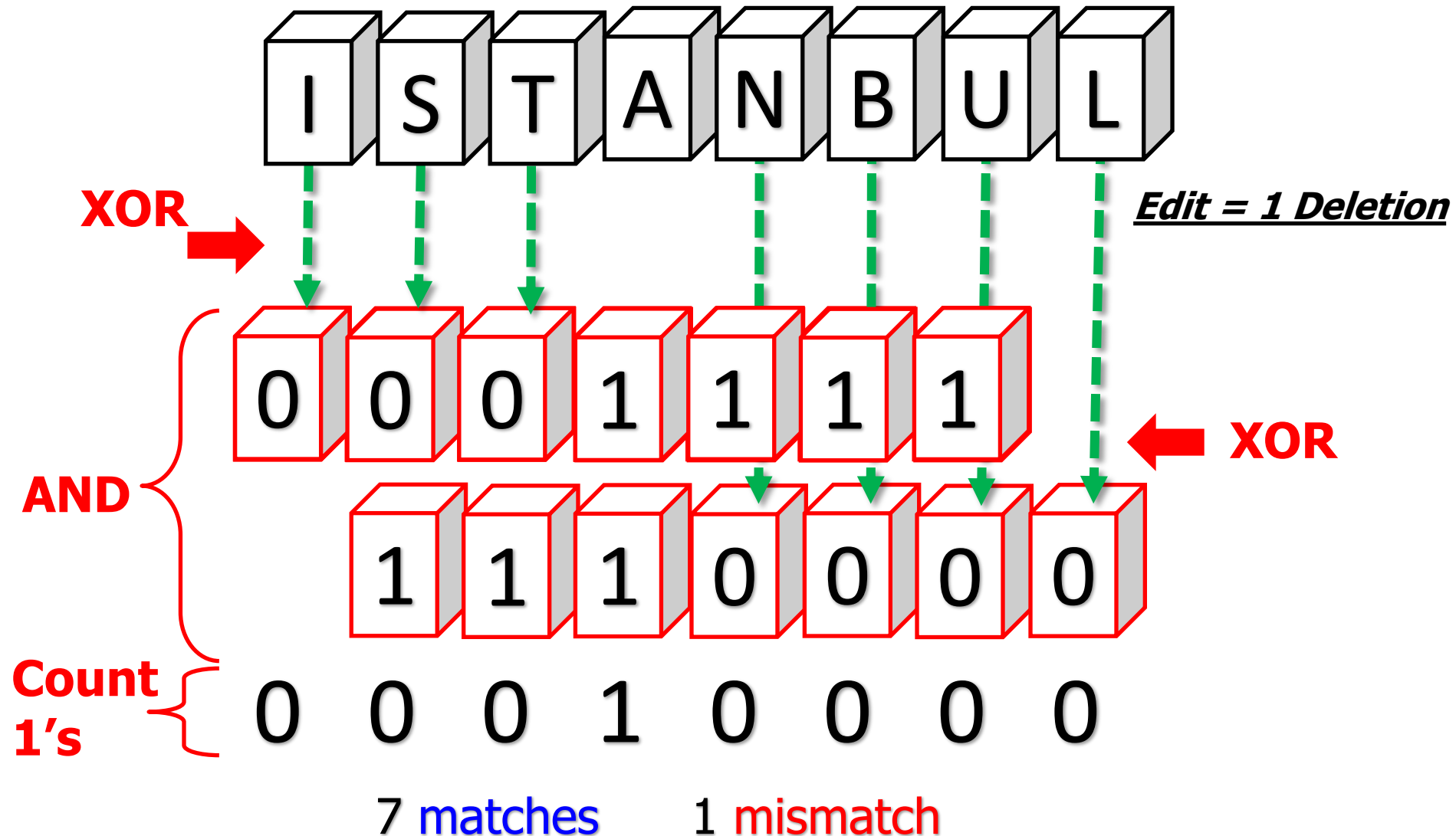
To cancel the effect of the deletion, we need to shift in the *right* direction

Insight: Shifting a String Helps Similarity Search

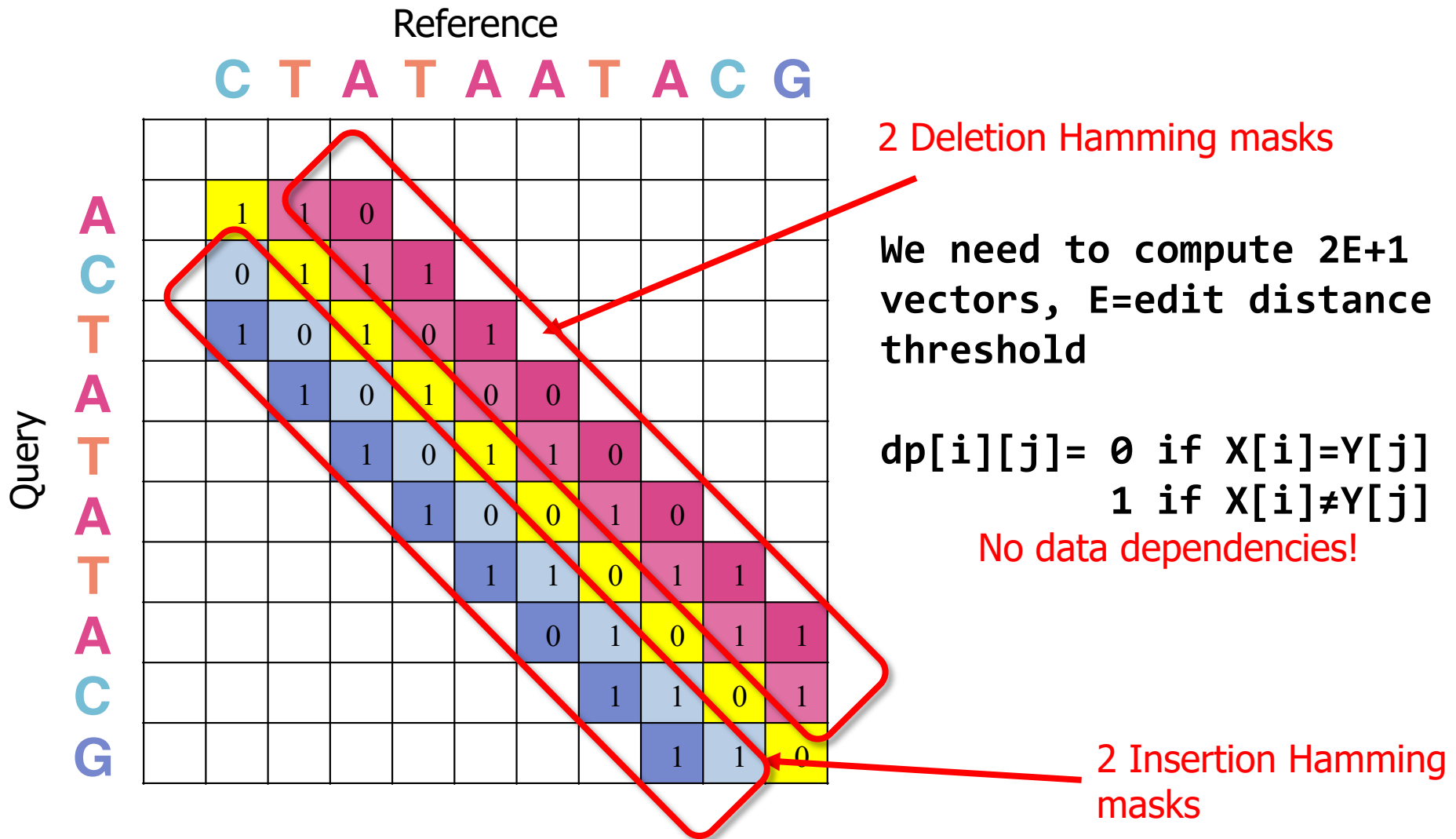
7 matches 1 mismatch



Shifted Hamming Distance



Highly Parallel Matrix Computation



Key Idea of SHD Filtering

Generate $2E+1$ masks

Amend random zeros:
101 → 111 & 1001 → 1111

AND all masks,
ACCEPT iff number of '1' $\leq$ Threshold

Query :GAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAACATTGTTGGGCCGGA

Reference :GAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGGCCCG

Hamming Mask : 0000000000010000000000001111111011110001110110101101111111100010000111011011010010101

[illegible]

2-Deletion Mask : 000000000101101110011111111111111011110001110110101101111111111000100100111101101001010

3-Deletion Mask :1111111111101110110011011101110111000100100111111111111111001011001101011011101101111

```

-Insertion Mask :1111111111101111101111110111101100010010011111111111111110010110011000 010111101110111110

```

2-Insertion Mask : 0000001001111100111111111100100011010101001101011111111111110111001 11 11000111101100

3-Insertion Mask :1111111101110110011000111111111010110111111001100101110111111110110111010111001000

--- Masks after amendment ---

Hamming Mask : 00000000001000000000001111111111110001111111101111111111110001000001111111111111111

[illegible]

2-Deletion Mask :00000000111111111111111111111111111111000111111111111111111111110001000111111111111111110

[illegible]

```
-Insertion Mask :1111111111111111111111111111111110001111111111111111111111111000111111111111111110
```

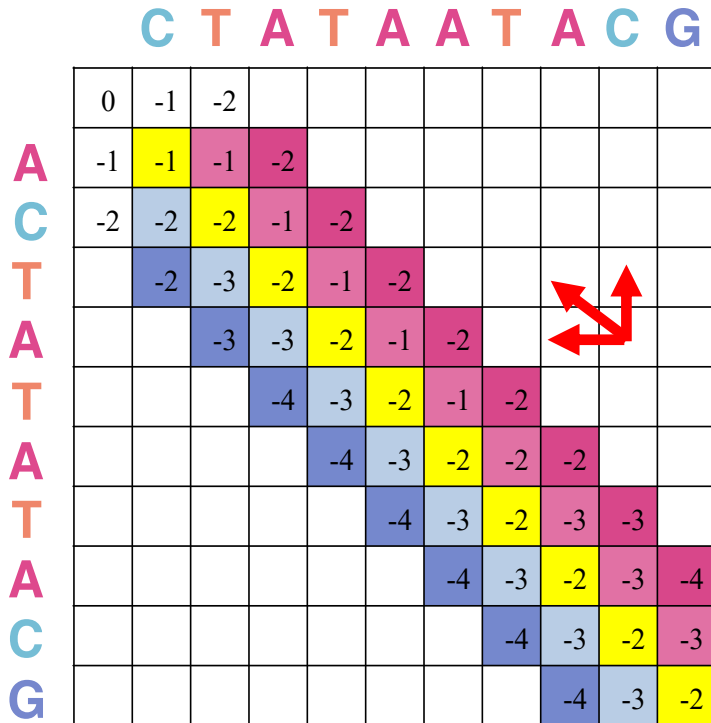
[illegible][illegible]

AND Mask :

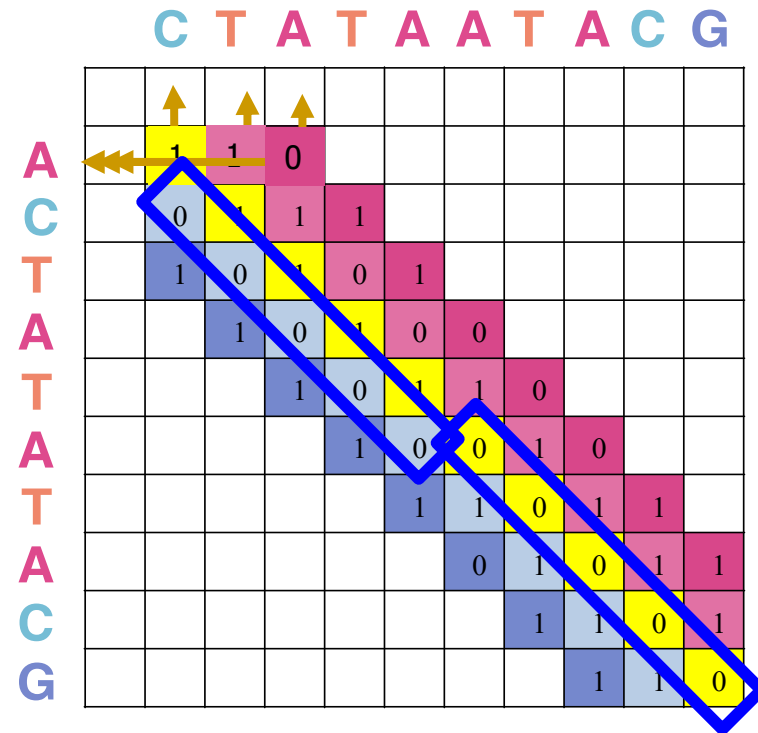
Needleman-Wunsch Alignment : 

Alignment vs. Pre-alignment (Filtering)

Needleman-Wunsch



Neighborhood Map



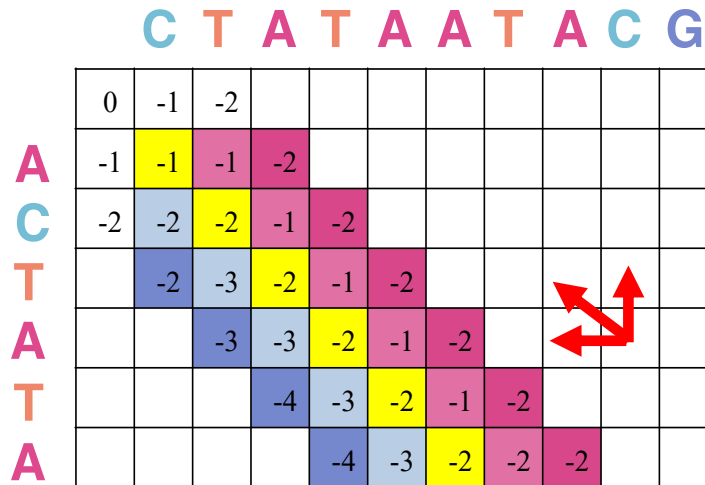
Our goal is to track the diagonally consecutive matches in the neighborhood map

pre-computed cells!

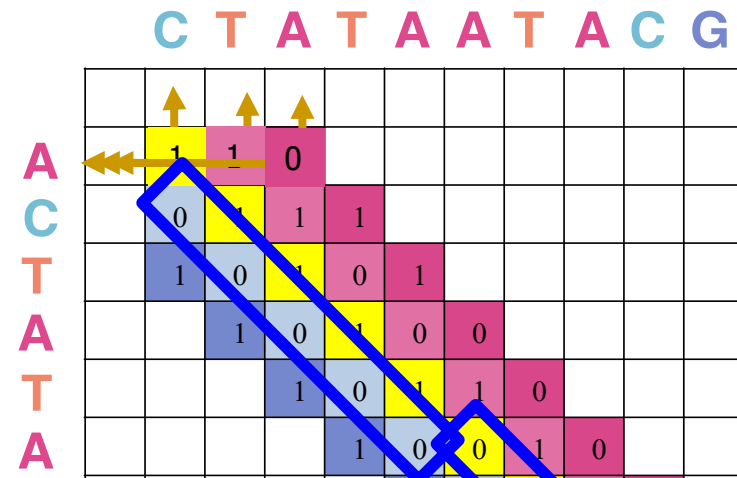
No data dependencies!

Alignment Matrix vs. Neighborhood Map

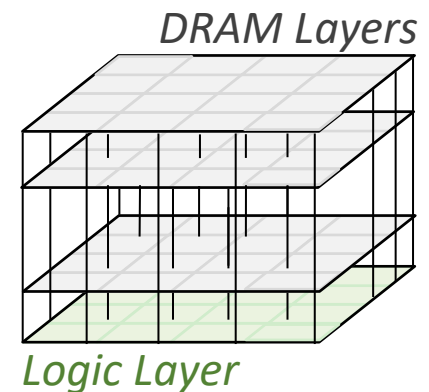
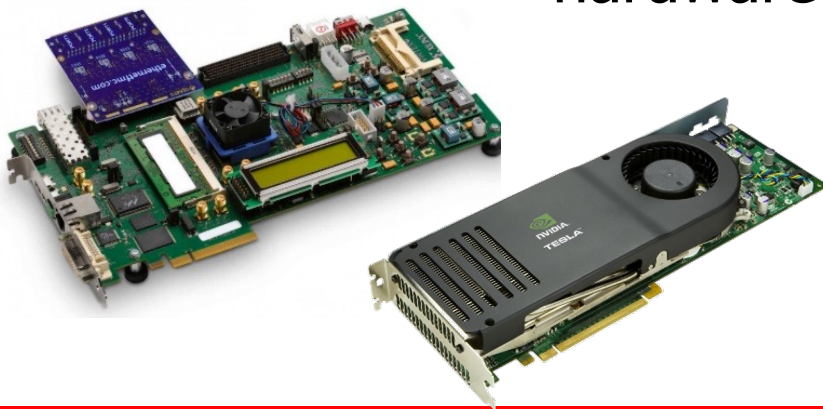
Needleman-Wunsch



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

OXFORD

Sequence analysis

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

Hongyi Xin^{1,*}, John Greth², John Emmons², Gennady Pekhimenko¹,
Carl Kingsford³, 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]

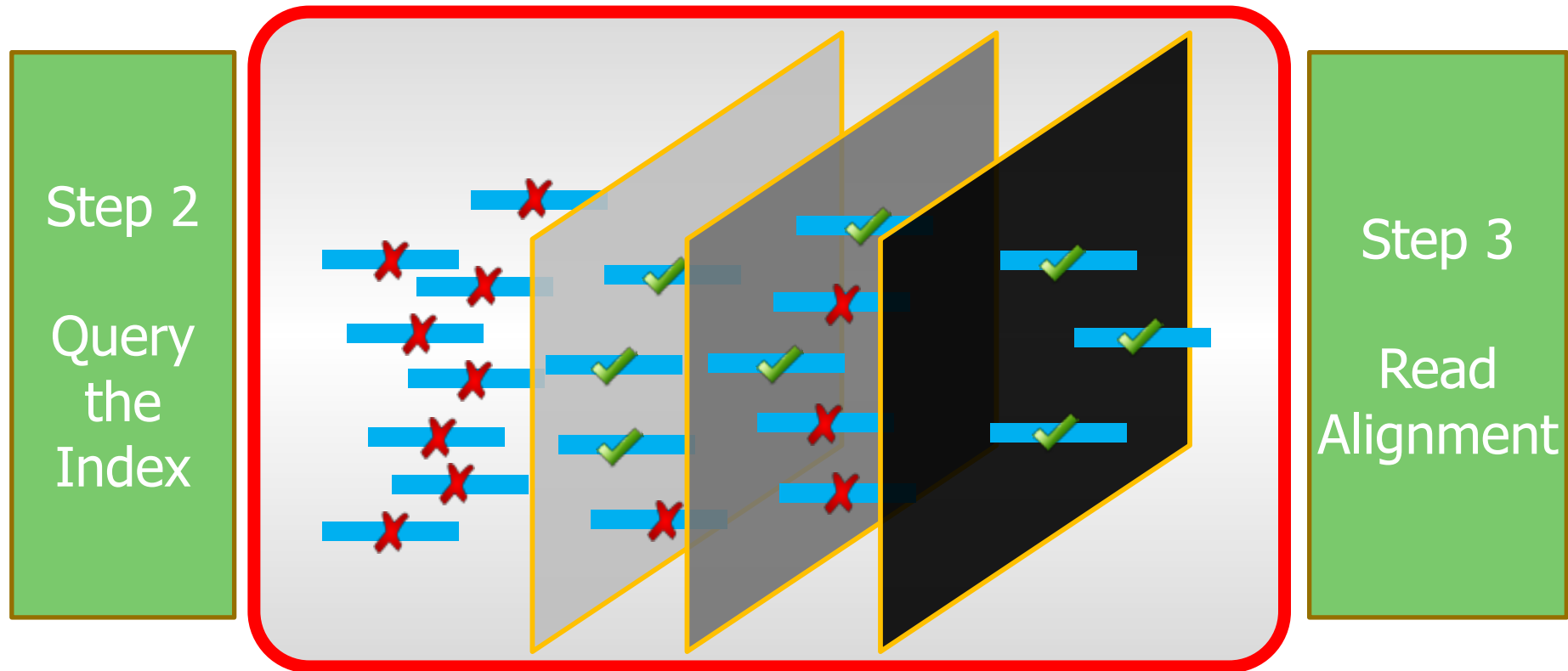
Location Filtering (Pre-alignment)

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

■ M
f
t
+
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



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

Location Filtering Example

Read Sequence (100 bp)



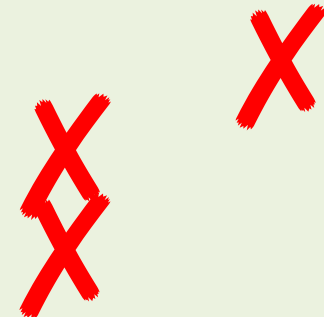
Mismatch. False
Accept

Hash Table

37 140
894 1203
1564

Reference Genome

Filter



Alignment vs. Pre-alignment (Filtering)

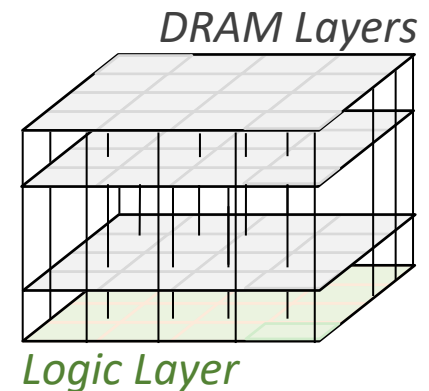
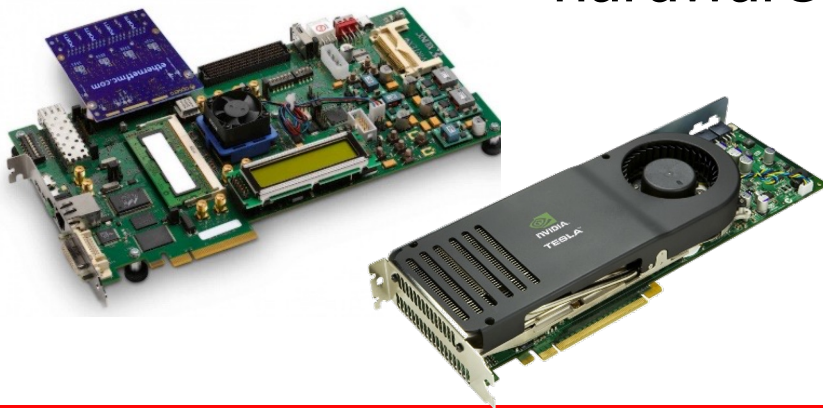
Needleman-Wunsch

	C	T	A	T	A	A	T	A	C	G
A	0	1	2							
C	1	0	1	2						
T	2	1	0	1	2					
A		2	1	0	1	2				
T			2	1	2	1	2			
A				2	2	2	1	2		
A					3	2	2	2	2	

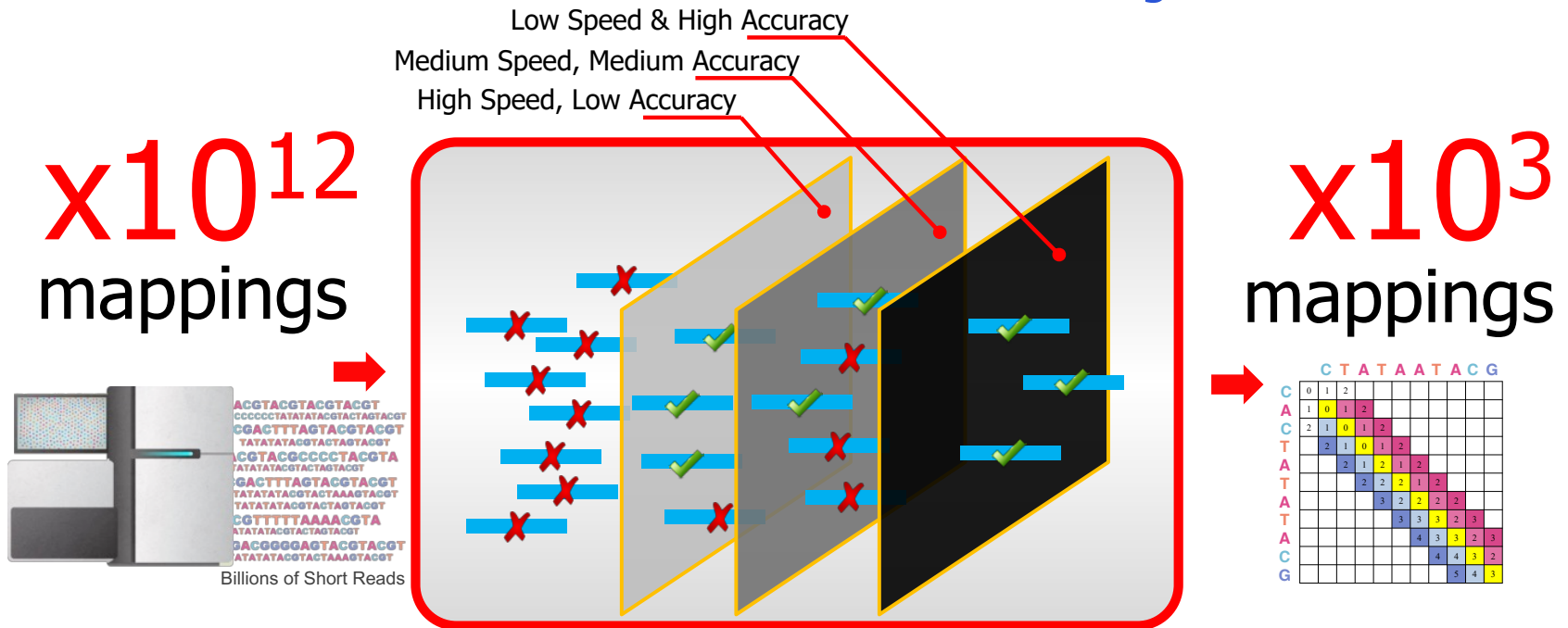
SHD

	C	T	A	T	A	A	T	A	C	G
A		1	1	0						
C		0	1	1	1					
T		1	0	1	0	1				
A			1	0	1	0	0			
T				1	0	1	1	0		
A					1	0	0	1	0	
A										

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' $\leq$ Threshold

Query :GAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGGAACATTGTTGGGCCGGA

Reference :GAGAGAGATAGTTAGTGTTCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGGCCCG

Hamming Mask : 000000000001000000000000111111101111000111011010110111111110001000001111011010010101

[illegible]

2-Deletion Mask : 000000000101101110011111111111101111000111011010110111111111000100010011101101001010

3-Deletion Mask :11111111111011101100110111011101110001001001111111111111100101100110010110111011101111

```

-Insertion Mask :1111111111101111101111101111101100010010011111111111111110010110011000101011101110111110

```

2-Insertion Mask : 000000100111110011111111001000110101010011010111111111111101110011111111000111101100

3-Insertion Mask :111111110111011001100011111111010110111111001100101110111111110111011111010111001000

--- Masks after amendment ---

Hamming Mask : 00000000001000000000001111111111110001111111101111111111110001000001111111111111111

[illegible][illegible][illegible]

```
L-Insertion Mask :11111111111111111111111111111111100011111111111111111111111111100011111111111111110
```

[illegible][illegible]

AND Mask :

Needleman-Wunsch Alignment : 

GateKeeper Walkthrough (cont'd)

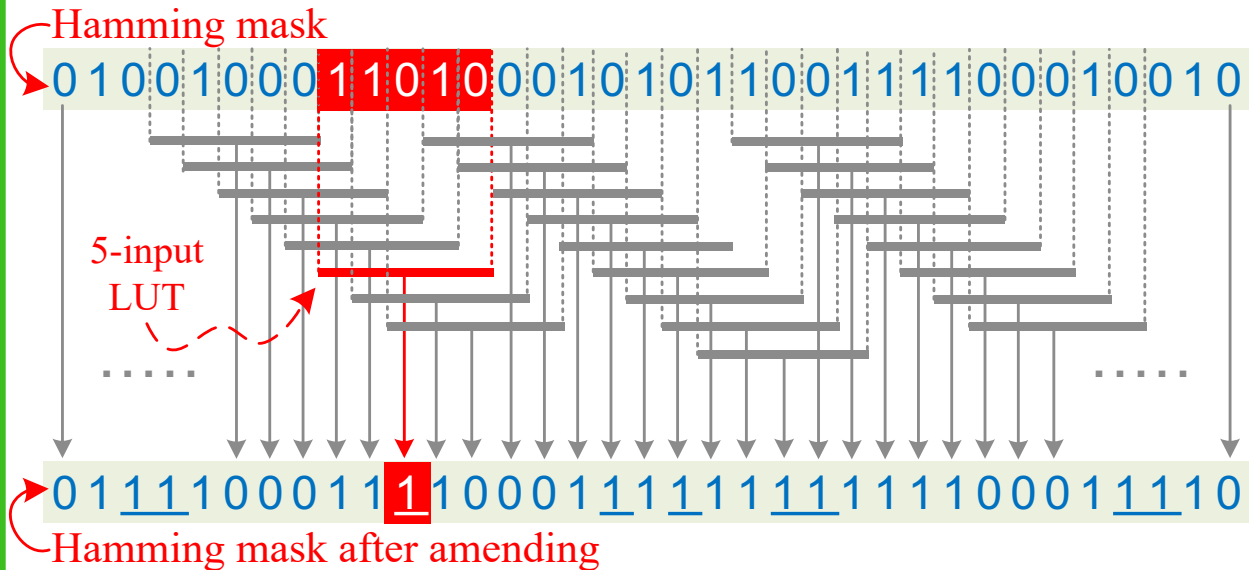
Generate $2E+1$ masks

Amend random zeros:
101 $\rightarrow$ 111 & 1001 $\rightarrow$ 1111

AND all masks,
ACCEPT iff number of '1' $\leq$ Threshold

- E right-shift registers (length=ReadLength)
- E left-shift registers (length=ReadLength)
- $(2E+1) * (\text{ReadLength})$ 2-XOR operations.

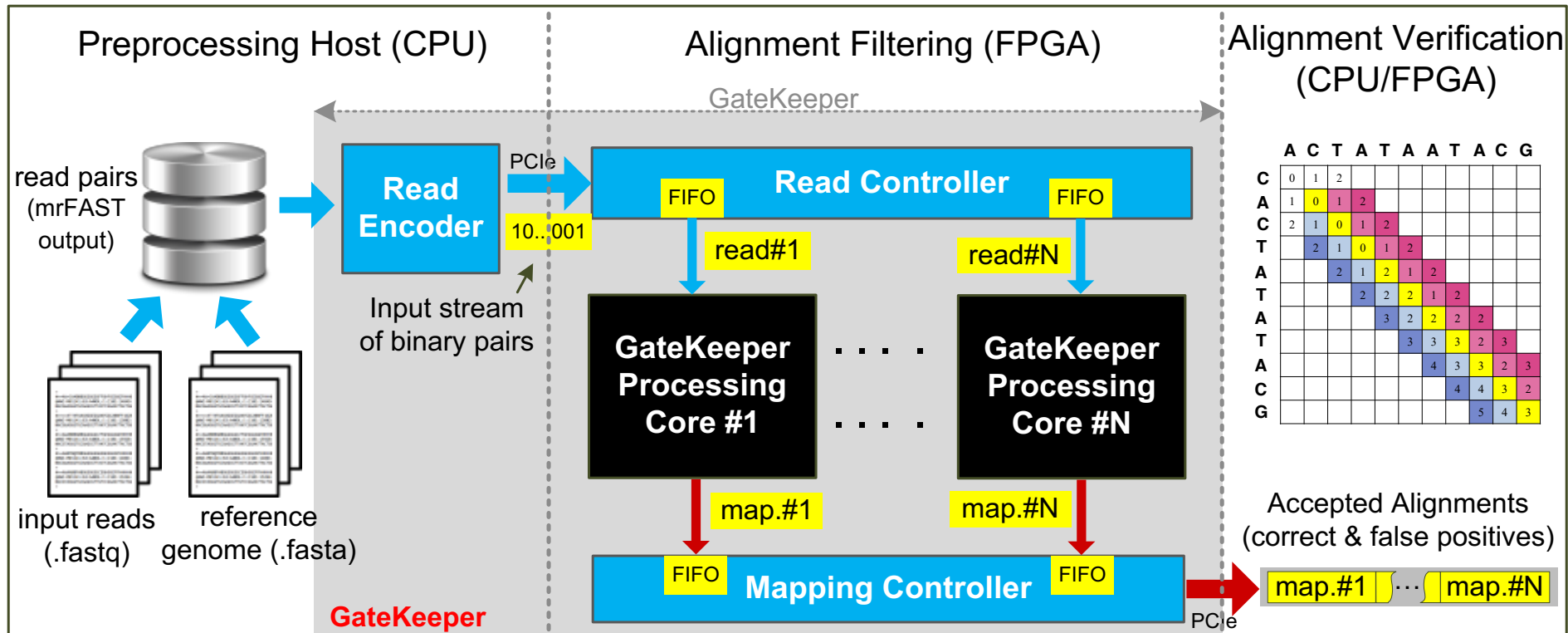
- $(2E) * (\text{ReadLength})$ 2-AND operations.
- $(\text{ReadLength}/4)$ 5-input LUT.
- $\log_2 \text{ReadLength}$ -bit counter.



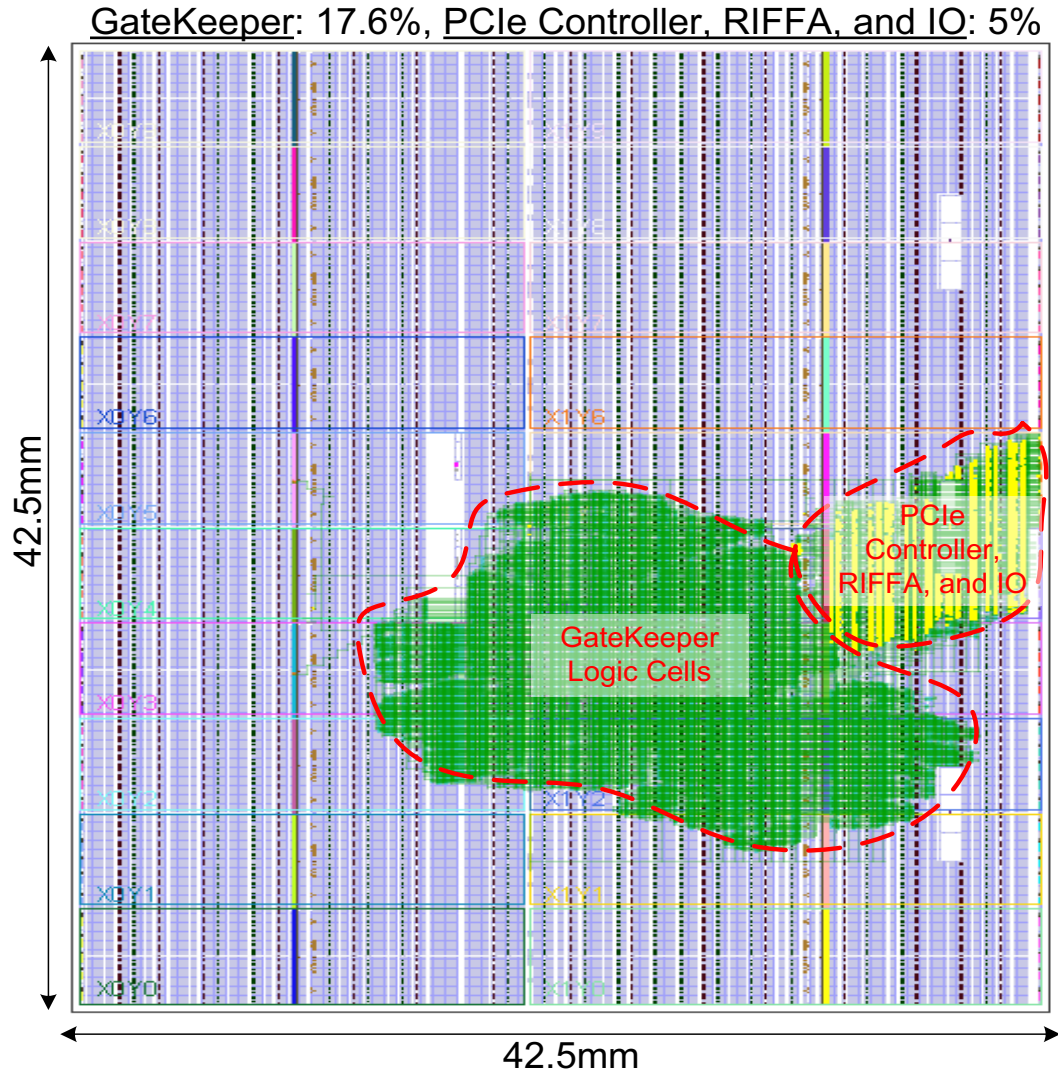
- $(2E+1) * (\text{ReadLength})$ 5-input LUT.

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 vs. SHD

GateKeeper

- 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.

SHD

- Intel SIMD
- Single-core (sequential)
- Examines a single mapping @ ~ 2 MHz
- 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

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

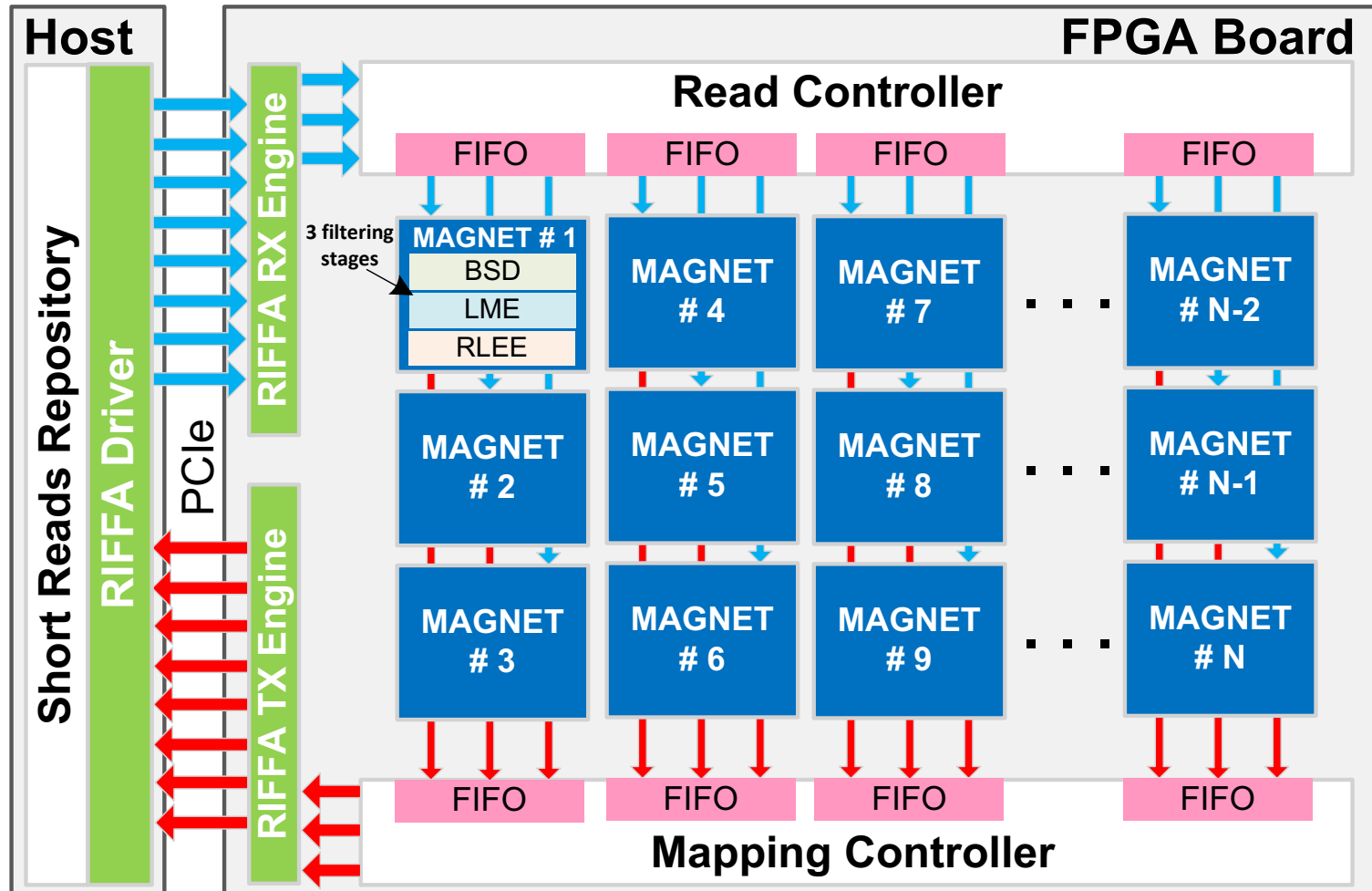
Mohammed Alser ✉, Hasan Hassan, Hongyi Xin, Oğuz Ergin, Onur Mutlu ✉, Can Alkan ✉

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]



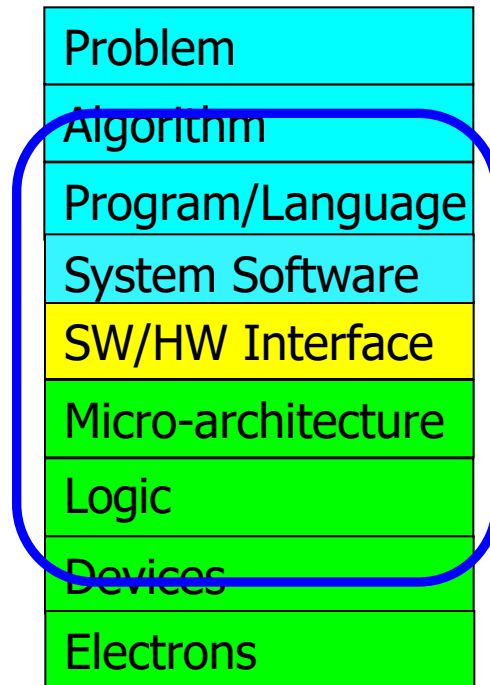
Can We Do Better?

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 Hassan¹, Akash Kumar², Onur Mutlu^{1,3,*}
and Can Alkan^{3,*}**

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

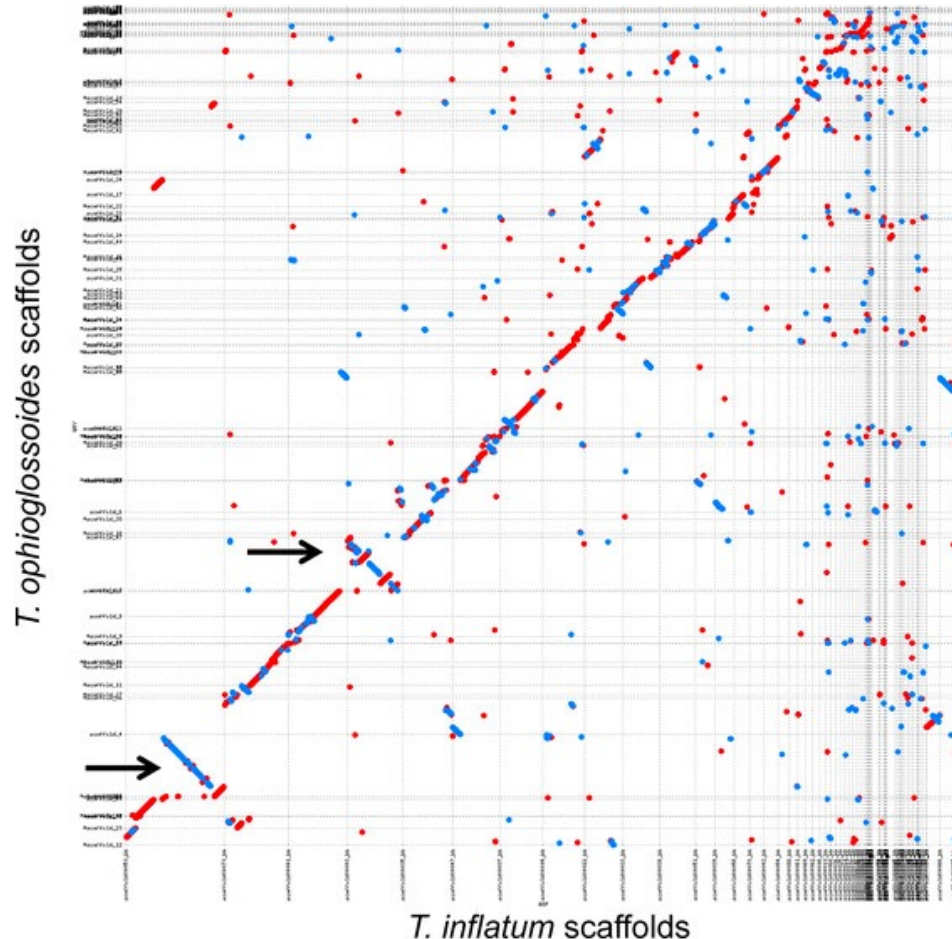
■ **Key idea:**

- ❑ 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)

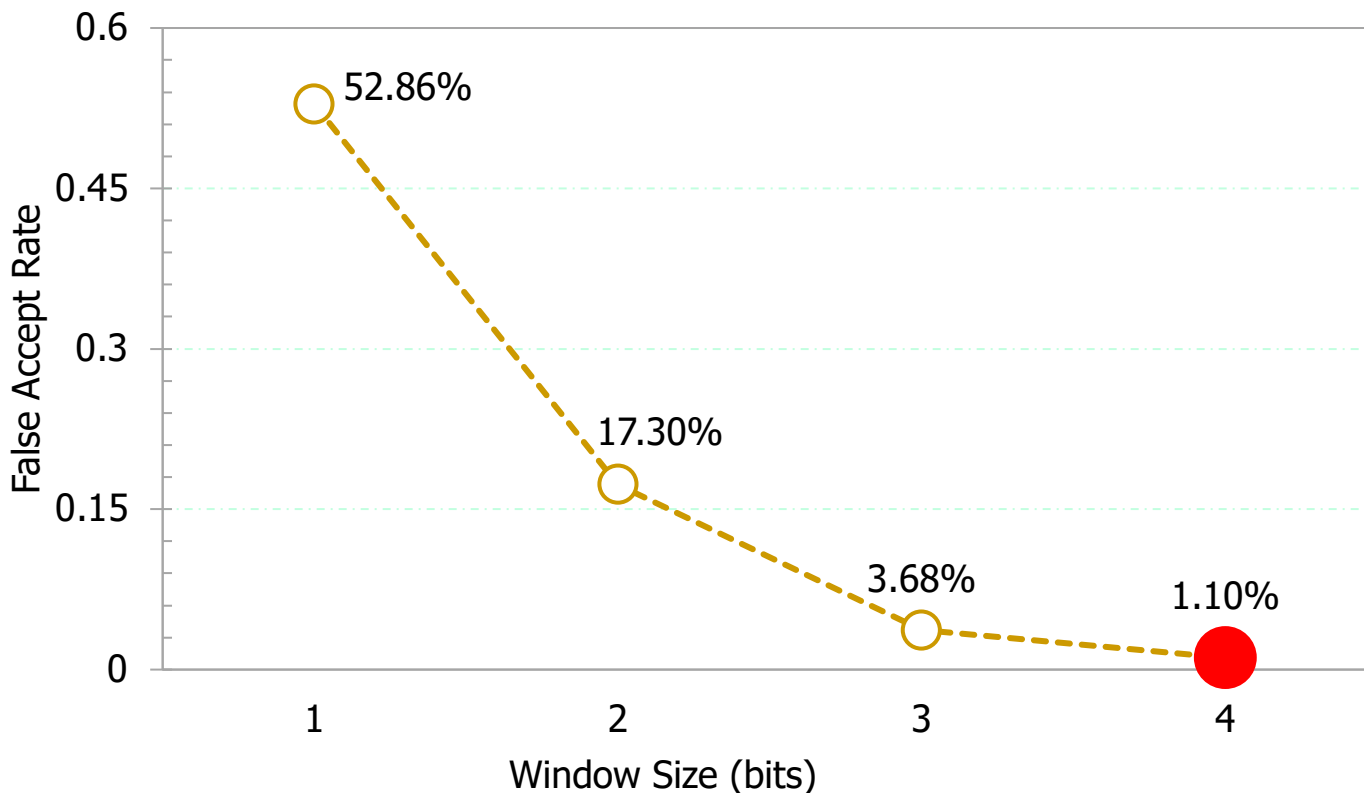
- **Key observation:**
 - Correct alignment always includes **long identical subsequences**



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

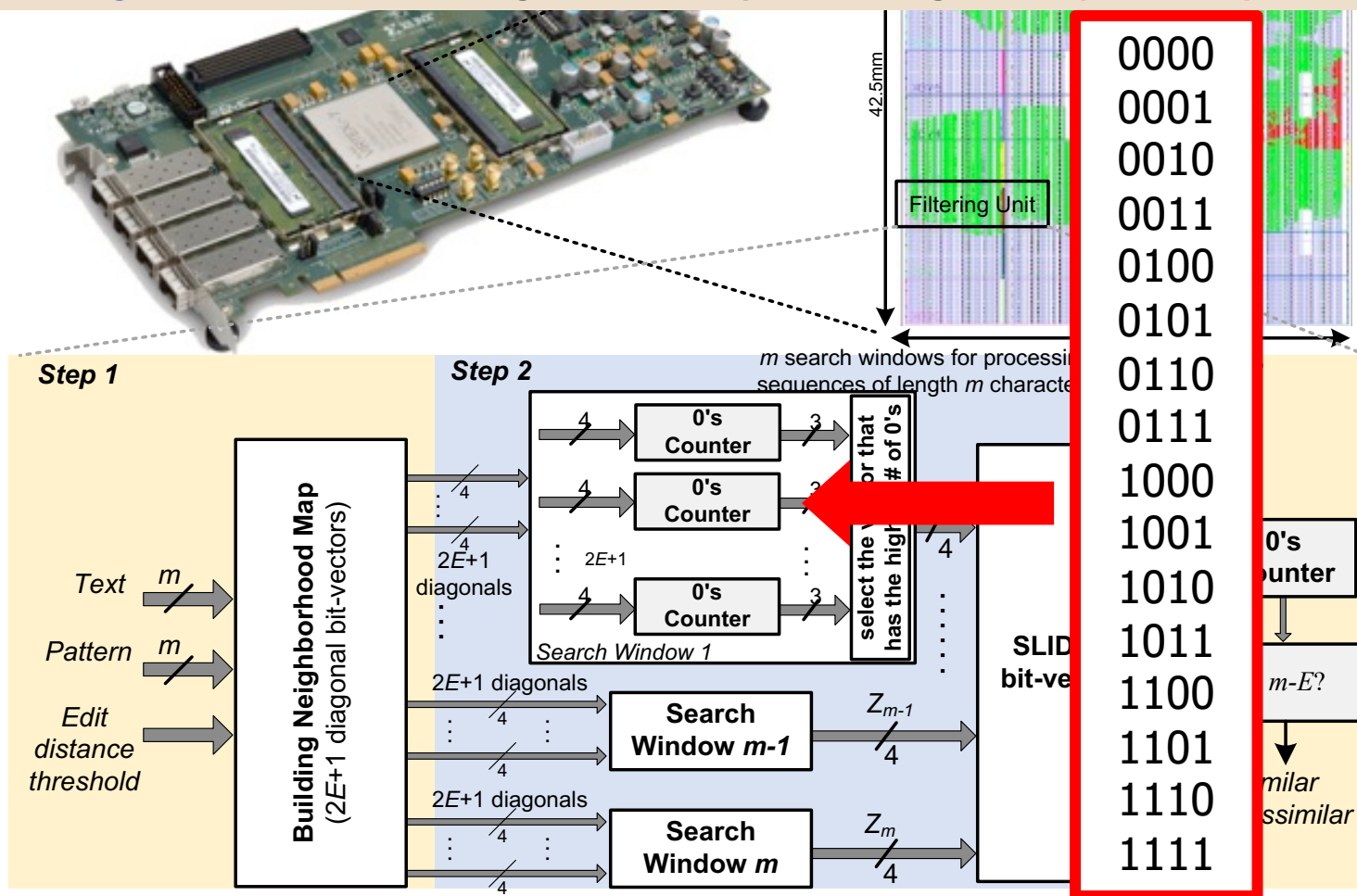
Effect of Sliding Window Size

- 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 Hassan¹, Akash Kumar², Onur Mutlu^{1,3,*} and Can Alkan^{3,*}

¹Computer Science Department, ETH Zürich, Zürich 8092, Switzerland, ²Chair for Processor Design, Center For Advancing Electronics Dresden, Institute of Computer Engineering, Technische Universität Dresden, 01062 Dresden, Germany and ³Computer 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 Shahroodi¹, Juan Gómez-Luna^{1,2},
Can Alkan^{4,*}, and Onur Mutlu^{1,2,3,4,*}**

¹Department of Computer Science, ETH Zurich, Zurich 8006, Switzerland

²Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland

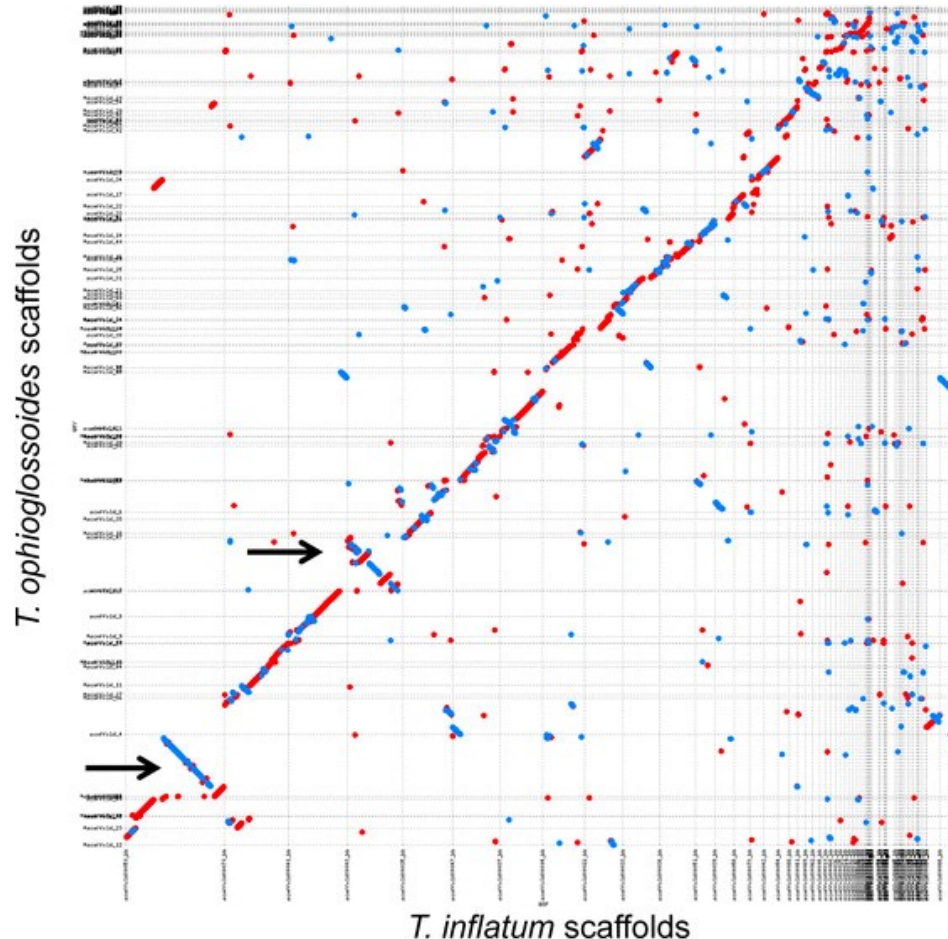
³Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh 15213, PA, USA

⁴Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey

SneakySnake

■ Key observation:

- Correct alignment is a sequence of non-overlapping long matches



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

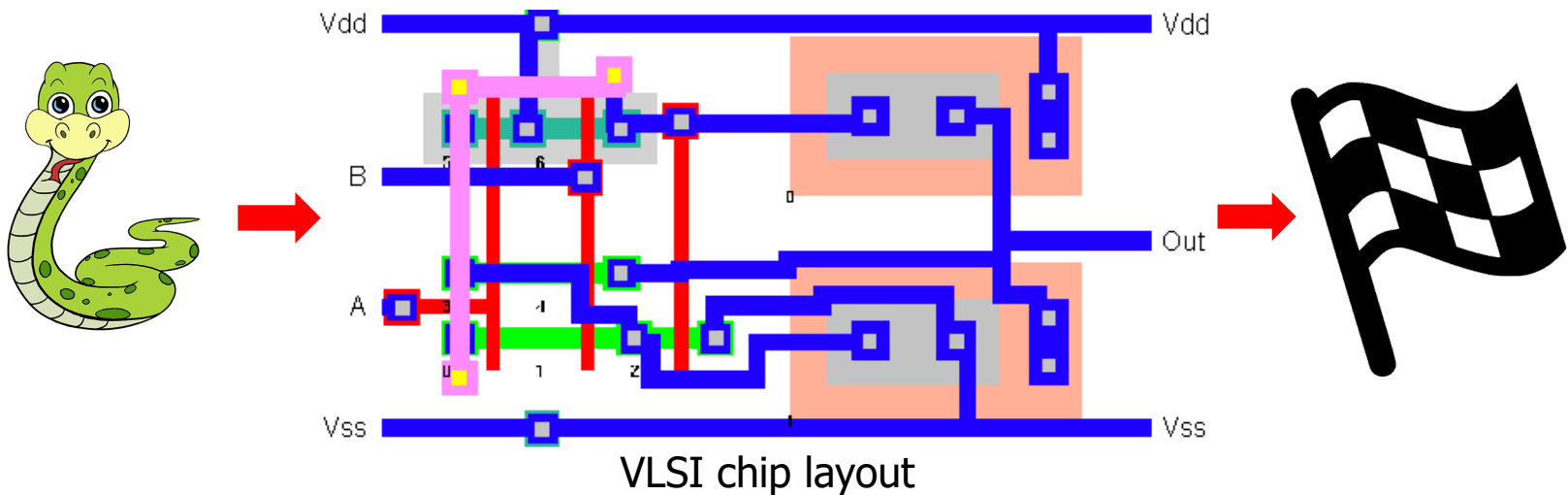
SneakySnake

■ 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



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:

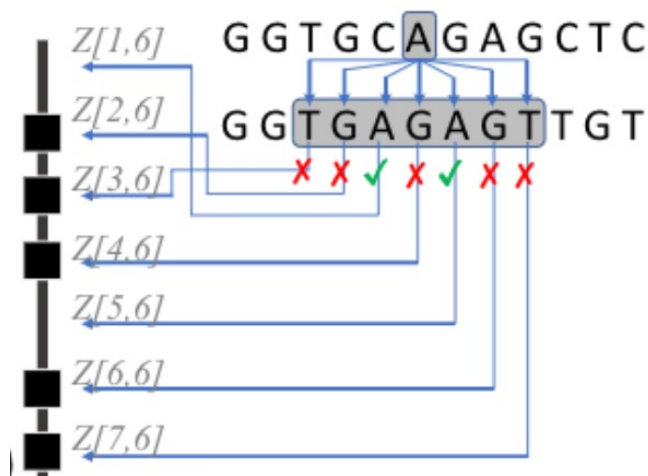
- ❑ SneakySnake is up to four orders of magnitude more accurate than Shouji (Bioinformatics'19) and GateKeeper (Bioinformatics'17)
- ❑ SneakySnake greatly accelerates state-of-the-art CPU sequence aligners, Edlib (Bioinformatics'17) and Parasail (BMC Bioinformatics'16)
 - ❑ by up to 37.7× and 43.9× (>12× 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

$E = 3$

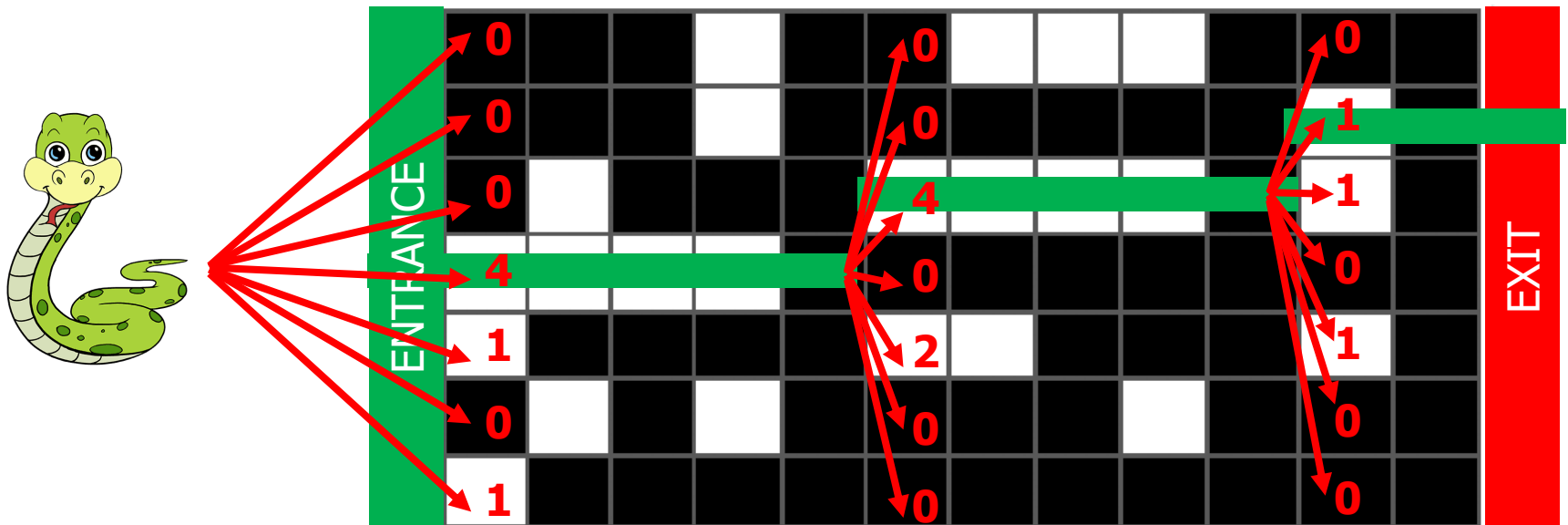
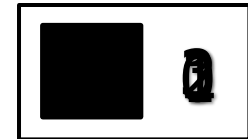
column	1	2	3	4	5	6	7	8	9	10	11	12
3 rd Upper Diagonal												
2 nd Upper Diagonal												
1 st Upper Diagonal												
Main Diagonal												
1 st Lower Diagonal												
2 nd Lower Diagonal												
3 rd Lower Diagonal												

SneakySnake Walkthrough

Building Neighborhood Map

Finding the Optimal Routing Path

Examining the Snake Survival



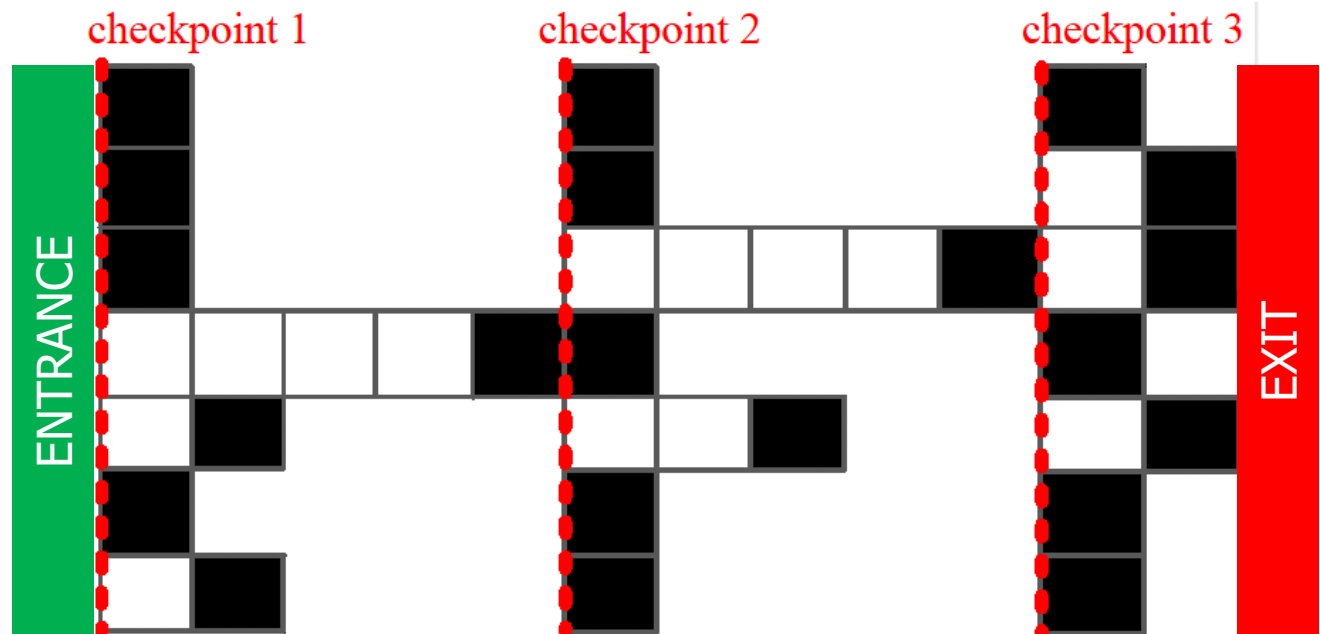
SneakySnake Walkthrough

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

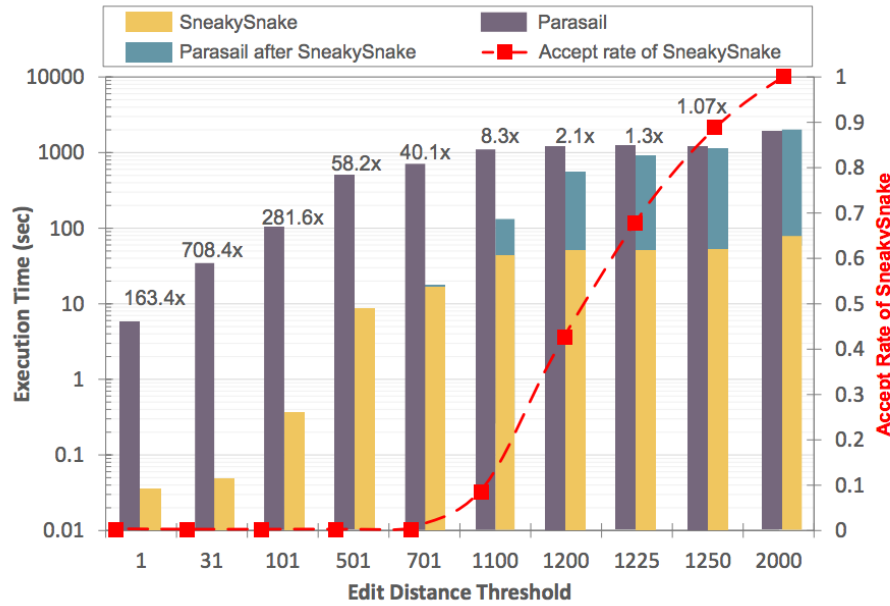
- ❑ SneakySnake is up to **four orders of magnitude more accurate** than **Shouji** (Bioinformatics'19) and **GateKeeper** (Bioinformatics'17)

- ❑ 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

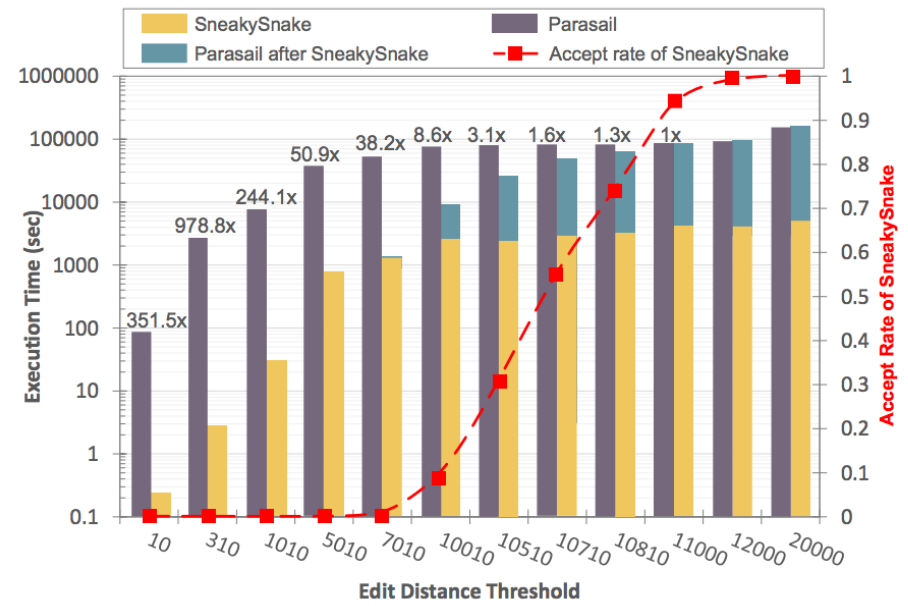
Long Read Mapping (SneakySnake vs Parasail)

10K bp reads



(a)

100K bp reads

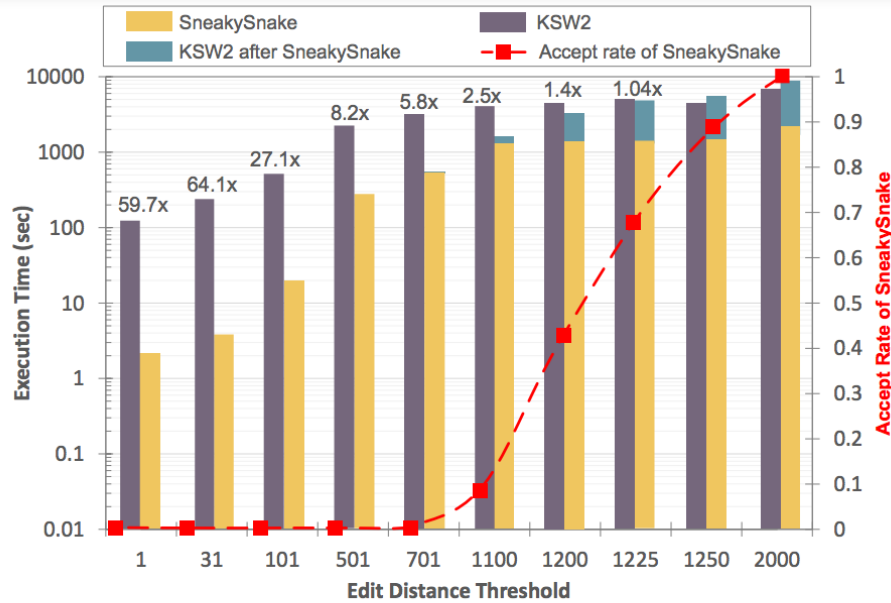


(b)

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.

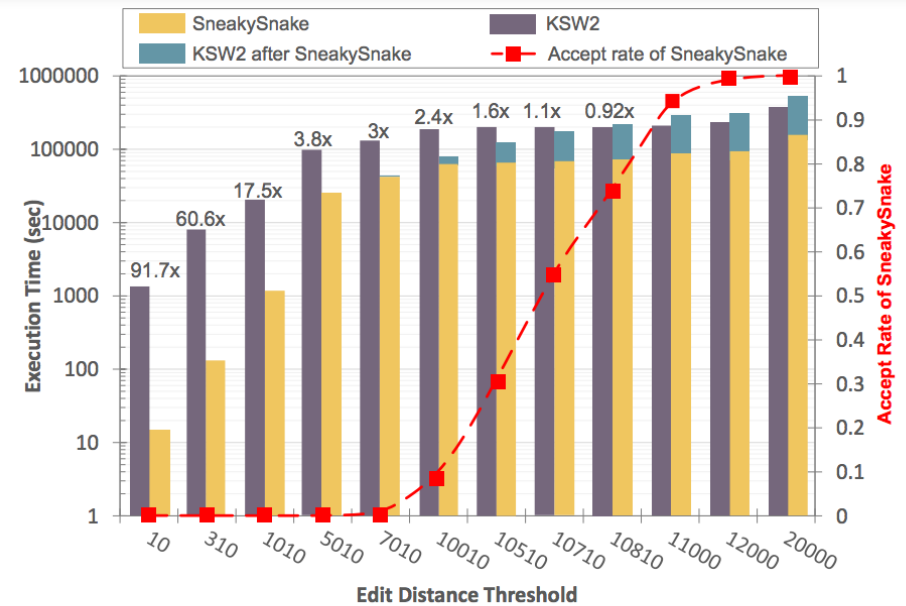
Long Read Mapping (SneakySnake vs KSW2)

10K bp reads



(a)

100K bp reads



(b)

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 Shahroodi¹, Juan Gómez-Luna^{1,2},
Can Alkan^{4,*}, and Onur Mutlu^{1,2,3,4,*}**

¹Department of Computer Science, ETH Zurich, Zurich 8006, Switzerland

²Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland

³Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh 15213, PA, USA

⁴Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey

GenASM Framework [MICRO 2020]

- 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

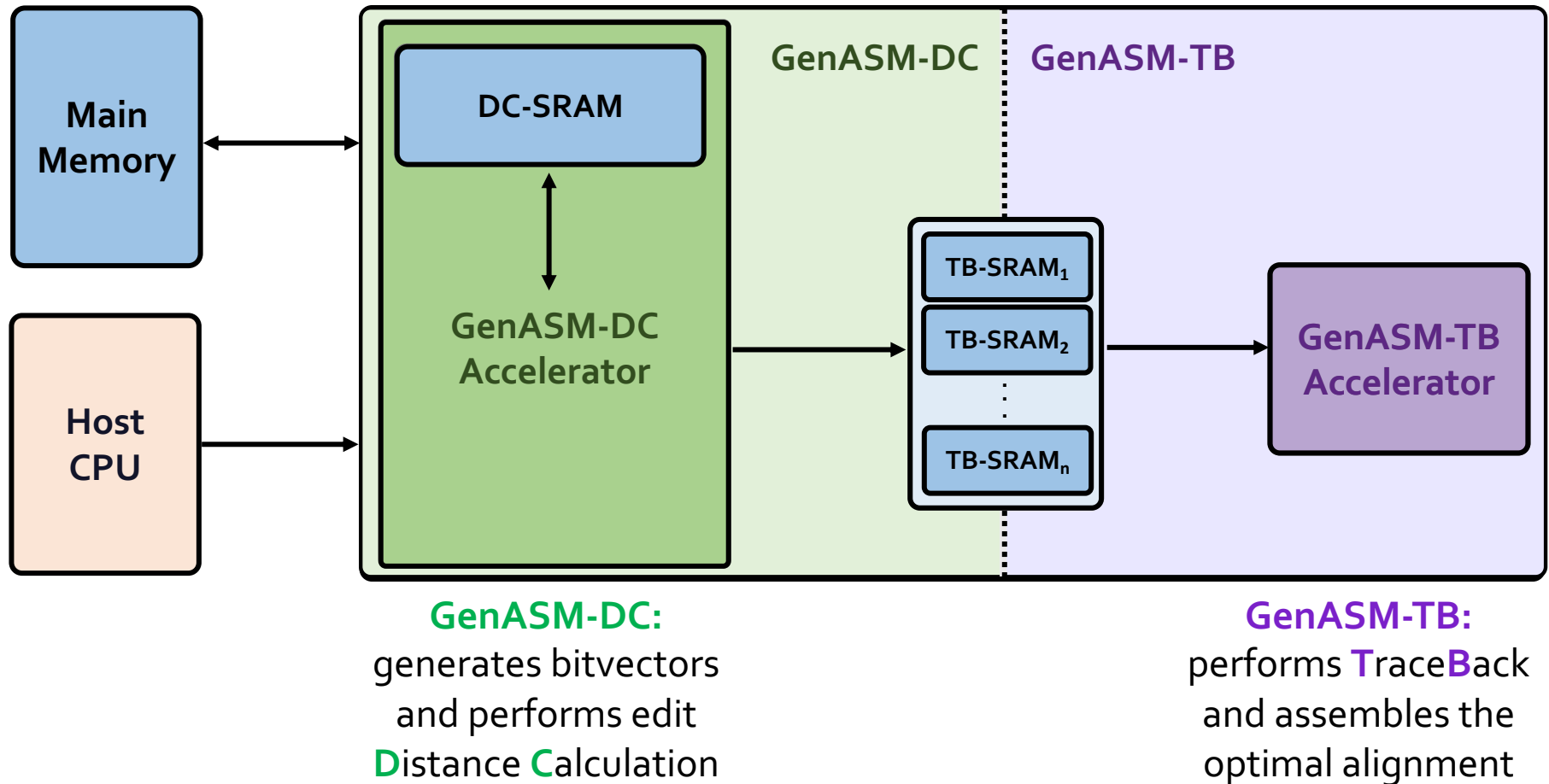
GenASM: ASM Framework for GSA

Our Goal:

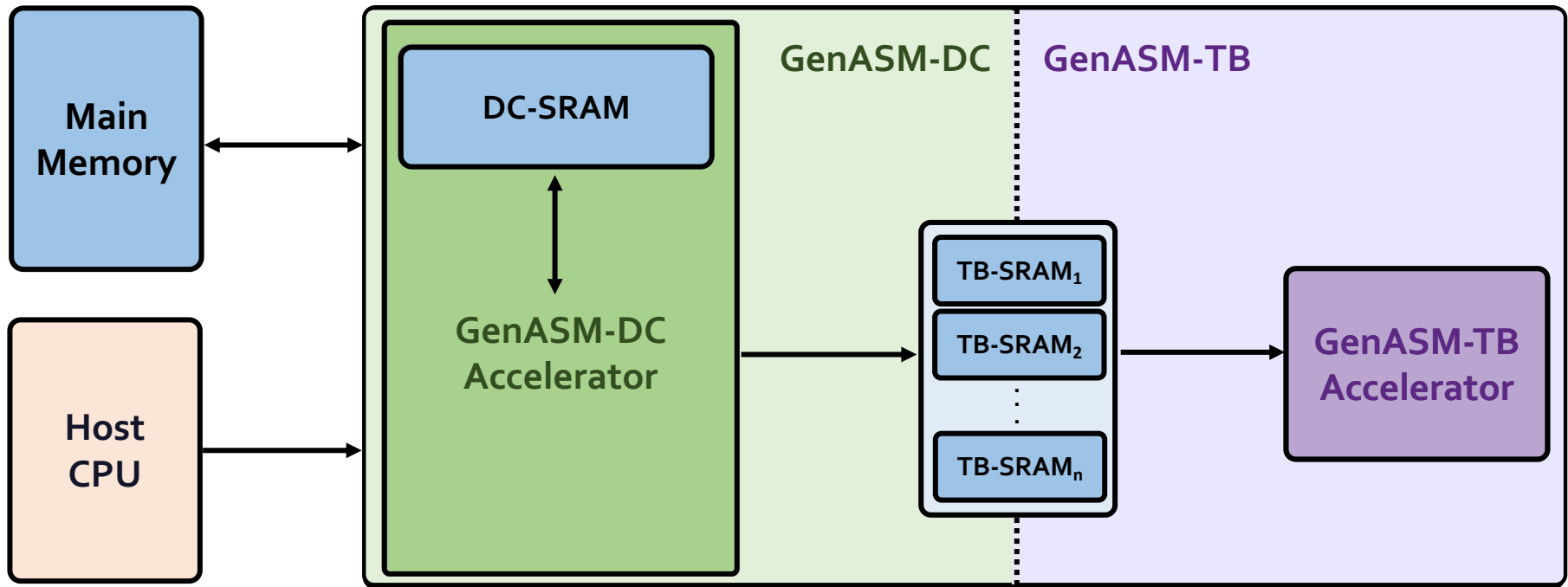
Accelerate approximate string matching
by designing a fast and flexible framework,
which can accelerate *multiple steps* of genome sequence analysis

- ❑ **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: Hardware Design

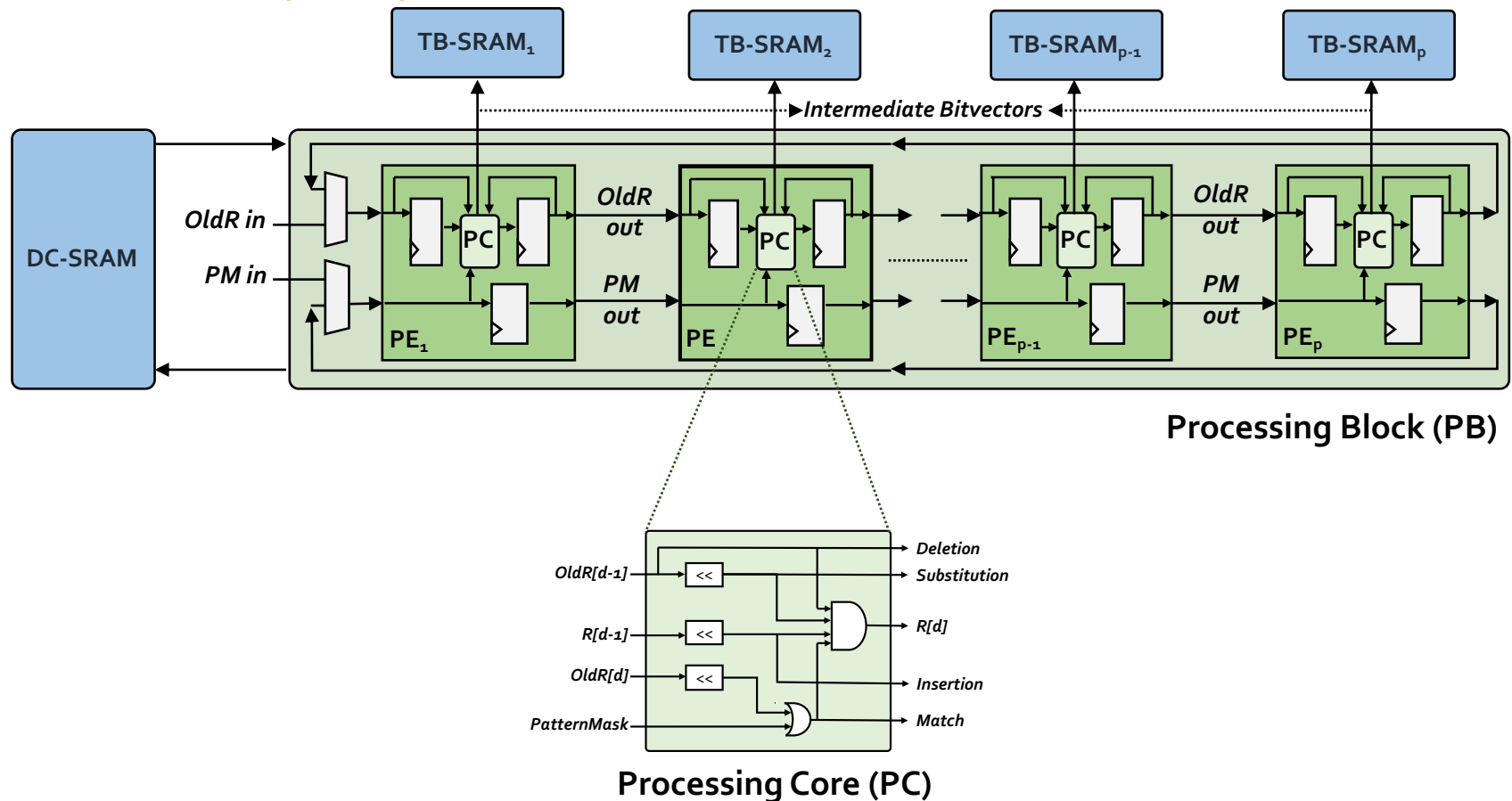


Our *specialized compute units* and *on-chip SRAMs* help us to:

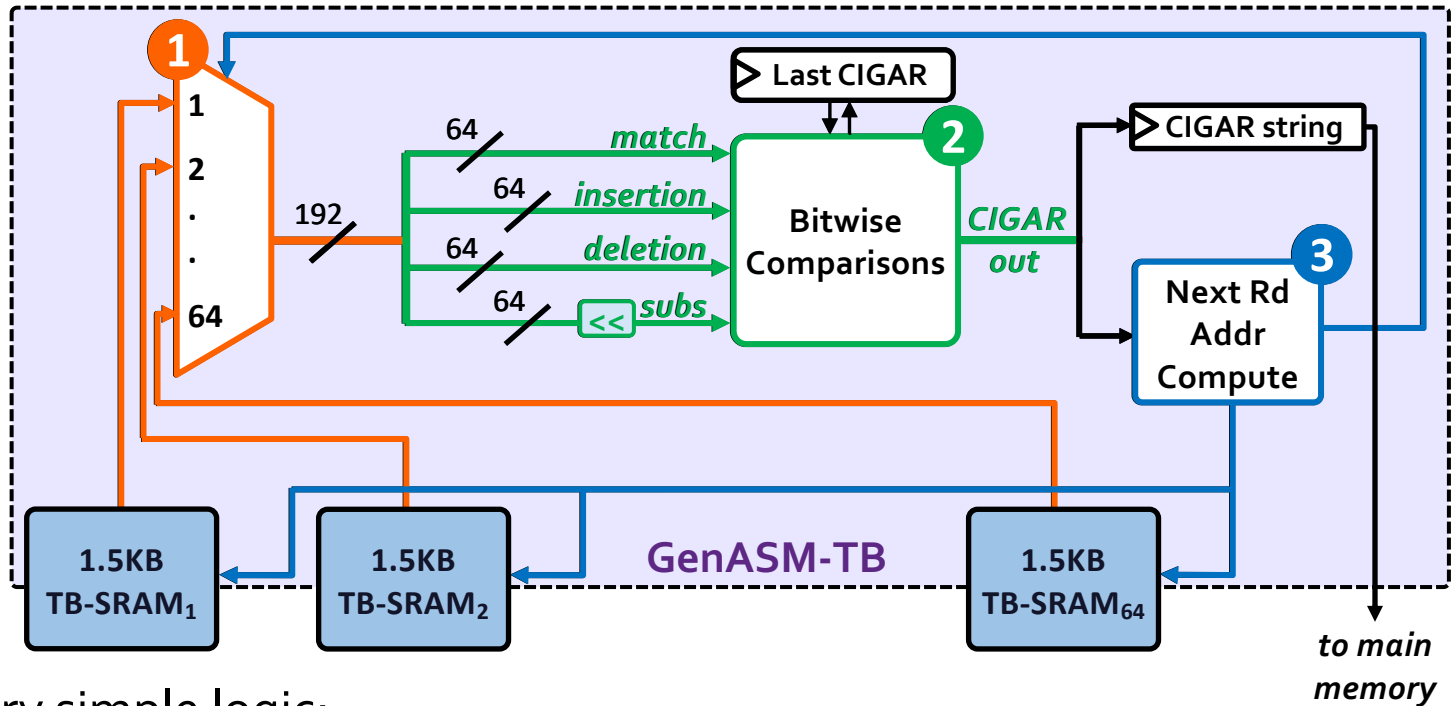
- Match **the rate of computation** with **memory capacity and bandwidth**
- **Achieve high performance and power efficiency**
- **Scale linearly in performance** with the number of parallel compute units that we add to the system

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

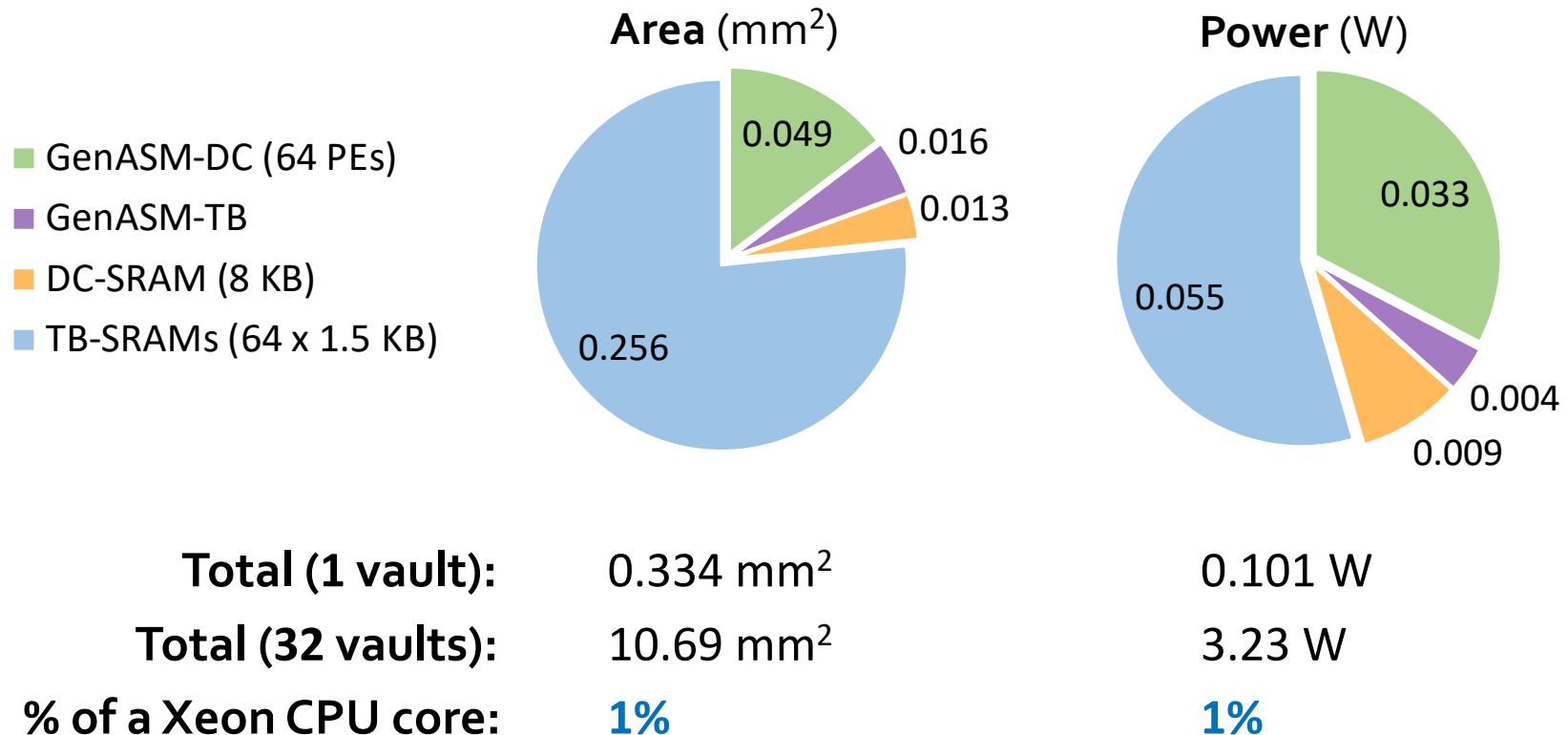


□ Very simple logic:

- 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

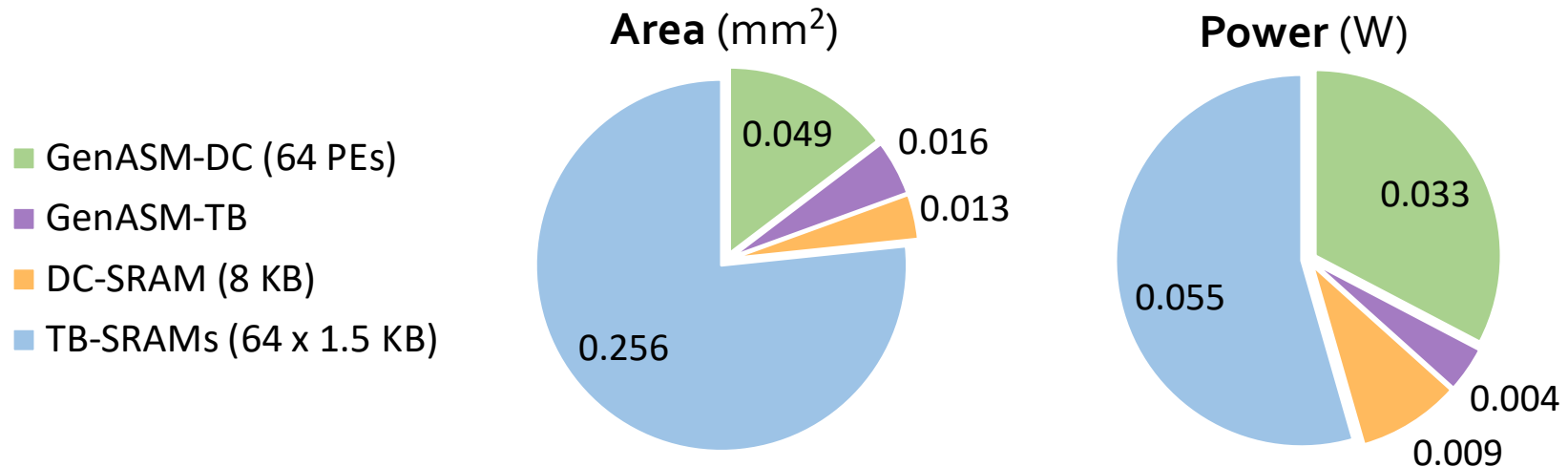
Key Results – Area and Power

- 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**



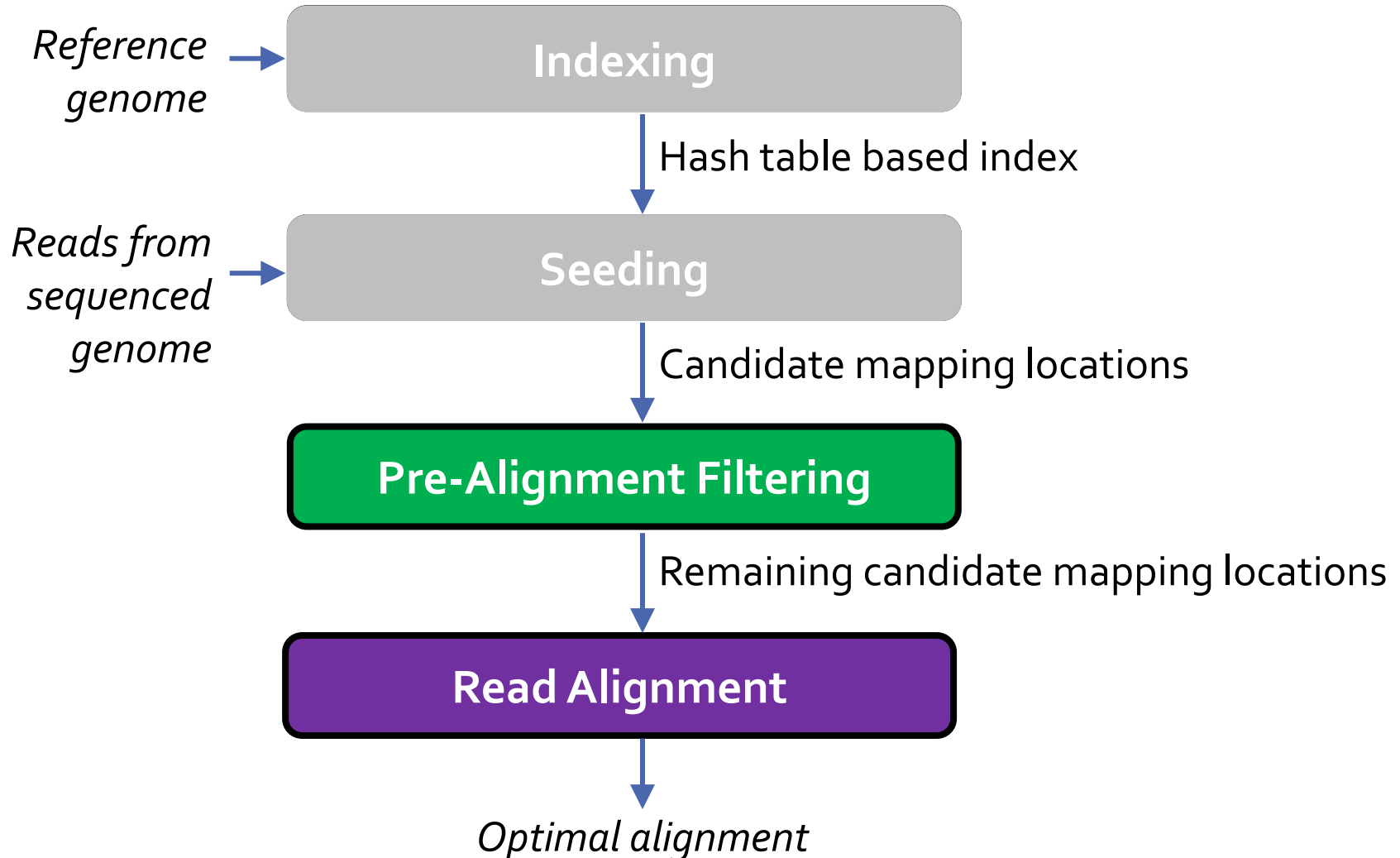
Key Results – Area and Power

- 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, 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, 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, Zülal Bingöl, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika Mansouri Ghiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,
"SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
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 Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Zülal Bingöl³
Gurpreet S. Kalsi⁴ Ziyi Zuo⁵ Can Firtina² Meryem Banu Cavlak² Jeremie Kim²
Nika Mansouri Ghiasi² Gagandeep Singh² Juan Gómez-Luna² Nour Almadhoun Alserr²
Mohammed Alser² Sreenivas Subramoney⁴ Can Alkan³ Saugata Ghose⁶ Onur Mutlu²

¹Bionano Genomics ²ETH Zürich ³Bilkent University ⁴Intel Labs

⁵Carnegie Mellon University ⁶University of Illinois Urbana-Champaign

Genome Sequence Analysis

- ❑ **Sequence-to-sequence mapping (traditional read mapping):**
 - Critical step in genome sequence analysis (GSA)
 - Maps *reads* collected from an individual to a known *linear reference genome sequence*
 - Well studied with many available *tools and accelerators*
- ❑ Recent works replace the linear reference sequence with a *graph-based representation of the reference genome (genome graph)*
 - Captures the *genetic variations* and *diversity* across many individuals in a *population*
- ❑ **Sequence-to-graph mapping** results in notable quality improvements in GSA
 - **More difficult** computational problem
 - **Much smaller number** of practical software tools currently available
 - **No prior** hardware design for graph-based GSA

SeGraM: First Graph Mapping Accelerator

Our Goal:

- **Specialized, high-performance, scalable, and low-cost** algorithm/hardware co-design that alleviates bottlenecks in *both* **the seeding and alignment steps** of sequence-to-graph mapping
 - Design an accelerator that is efficient for *both* **linear and graph-based read mapping**

SeGraM: *First universal genomic mapping accelerator*

- **MinSeed:** The *first* minimizer-based seeding accelerator
- **BitAlign:** The *first* (bitvector-based) sequence-to-graph alignment accelerator

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**)

Accelerating Sequence-to-Graph Mapping

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zülal Bingöl, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika Mansouri Ghiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,
"SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
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 Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Zülal Bingöl³
Gurpreet S. Kalsi⁴ Ziyi Zuo⁵ Can Firtina² Meryem Banu Cavlak² Jeremie Kim²
Nika Mansouri Ghiasi² Gagandeep Singh² Juan Gómez-Luna² Nour Almadhoun Alserr²
Mohammed Alser² Sreenivas Subramoney⁴ Can Alkan³ Saugata Ghose⁶ Onur Mutlu²

¹Bionano Genomics ²ETH Zürich ³Bilkent University ⁴Intel Labs

⁵Carnegie Mellon University ⁶University of Illinois Urbana-Champaign

Agenda

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

Read Mapping & Filtering

- 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](#)]



[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

◀	▶
Previous	Next
☰	Table of Contents
📖	Past Issues

Near-Memory SneakySnake

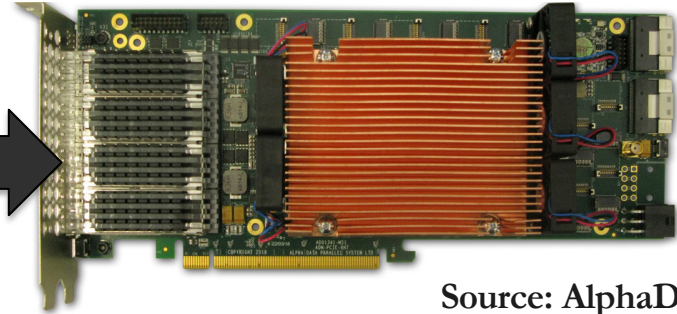
- Problem: Read mapping is heavily bottlenecked by data movement from main memory
- Solution: Perform read mapping near where data resides using specialized logic
- We carefully redesign the accelerator logic of SneakySnake to exploit near-memory computation capability on real FPGA boards that use HBM (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



Source: AlphaData

HBM-based FPGA board

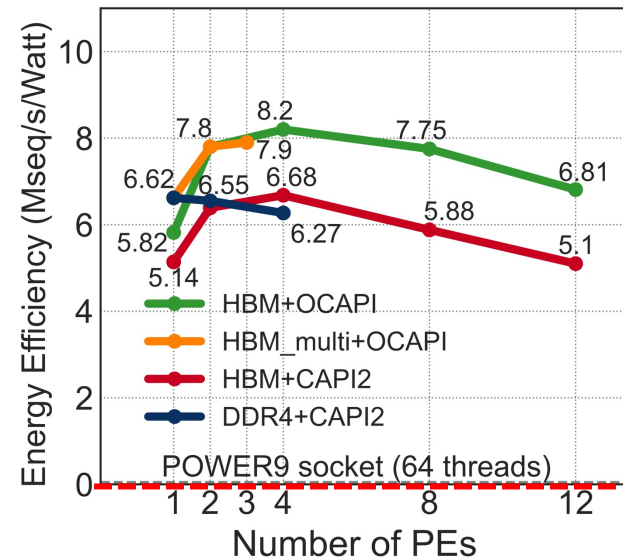
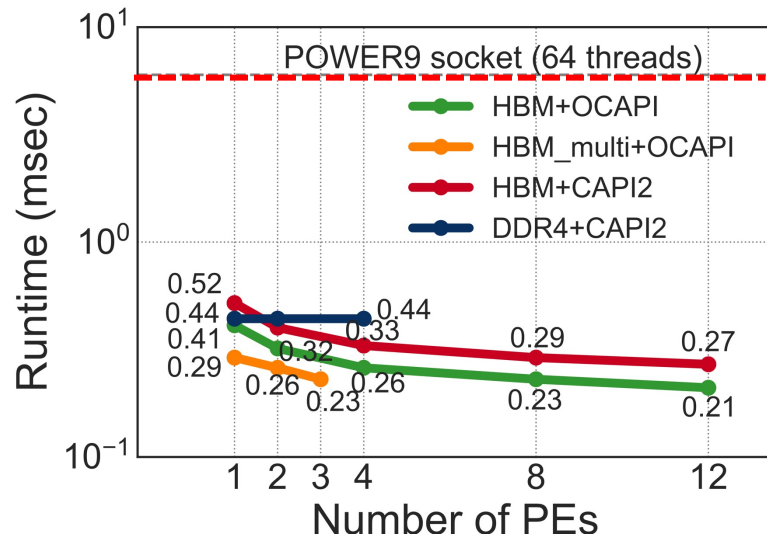
Near-HBM FPGA-based accelerator

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](#)]



[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

◀	▶
Previous	Next
☰	Table of Contents
📖	Past Issues

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

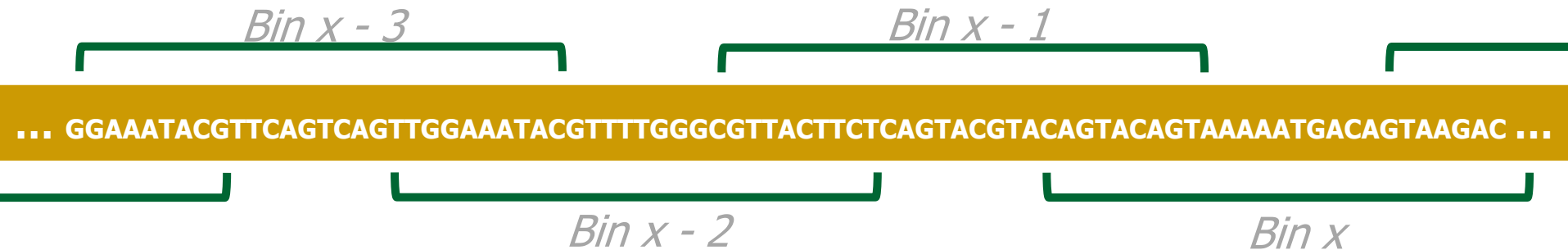
- **Key observation:** FPGA and GPU accelerators are heavily bottlenecked by **data movement**
- **Key idea:** exploit the high memory bandwidth and the logic layer of **3D-stacked memory** to perform **highly-parallel filtering** in the DRAM chip itself
- **GRIM-Filter, an algorithm-hardware co-designed 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**).

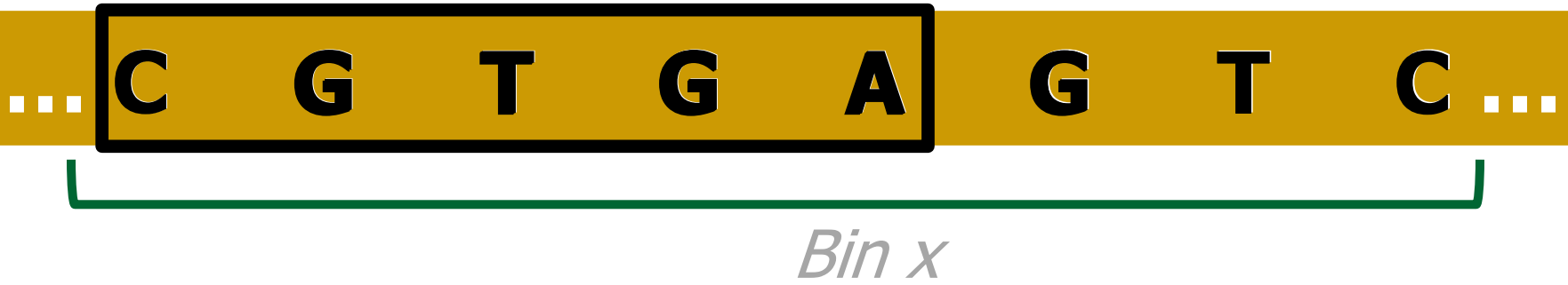


- Represent each bin with a **bitvector** that holds the occurrence of all permutations of a small string (**token**) in the bin
- To account for matches that straddle bins, we employ overlapping bins
 - A read will now always completely fall within a single bin

Bitvector

AAAAA	1	AAAAA exists in bin x
AAAAC	0	
AAAAT	1	
...	...	
CCCCC	1	
CCCCT	0	CCCCT doesn't exist in bin x
CCCCG	0	
...	...	
GGGGG	1	

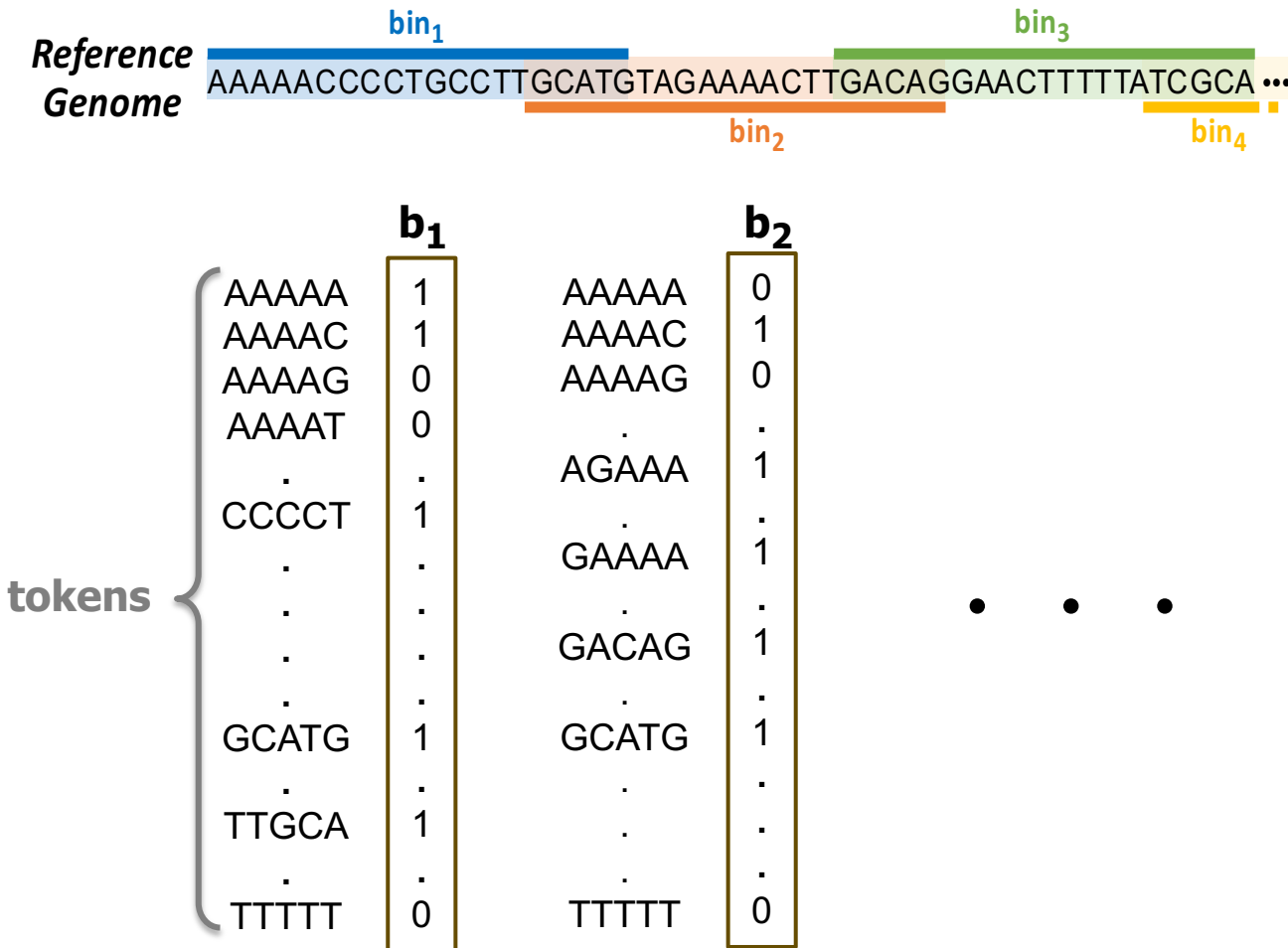
GRIM-Filter: Bitvectors



Bin x Bitvector

AAAAA	0
...	...
CGTGA	0
...	...
TGAGT	0
...	...
GAGTC	0
...	...
GTGAG	0
...	...

GRIM-Filter: Bitvectors



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

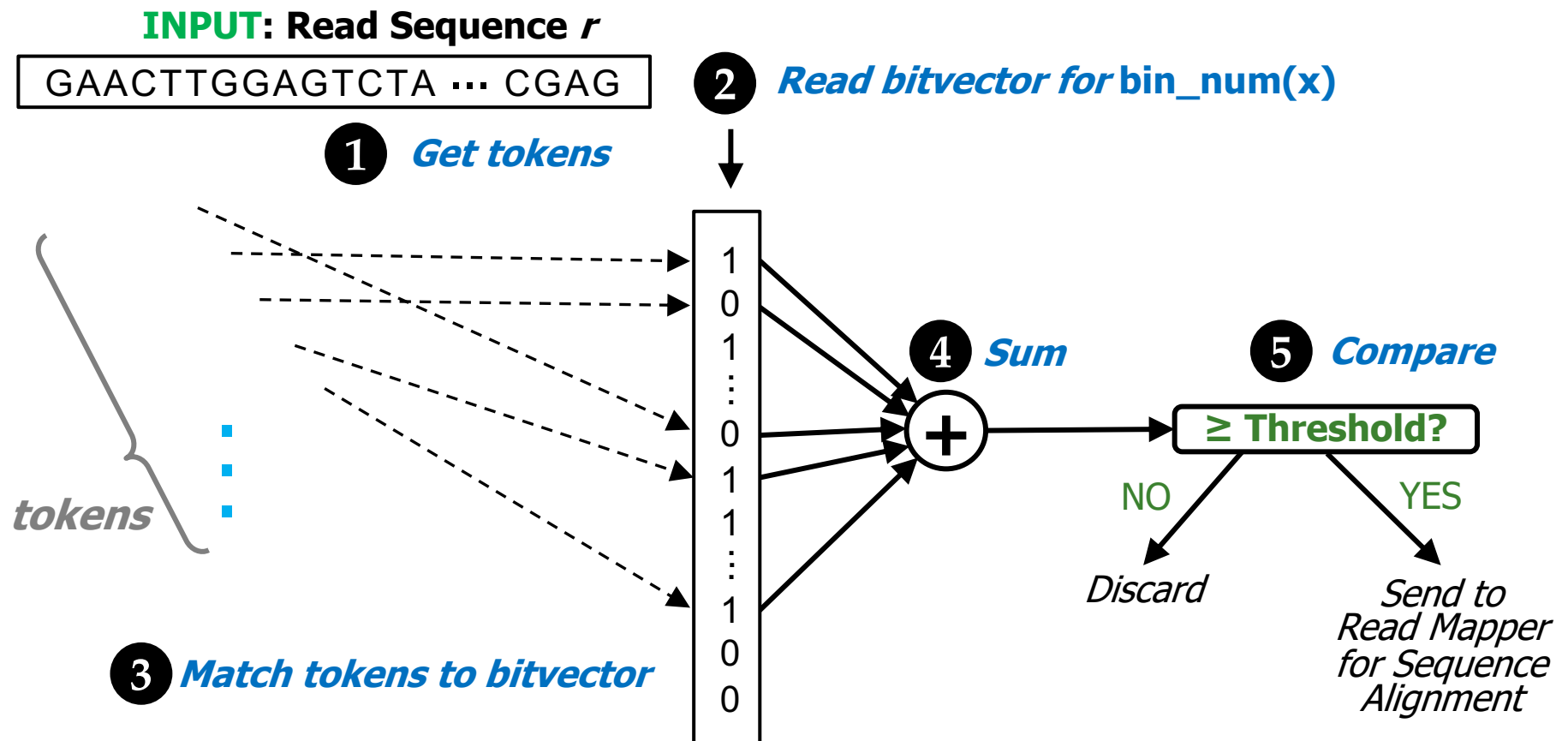
For **bin size** ~ 200 , and **n** = 5, **memory footprint** ~ 3.8 GB

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

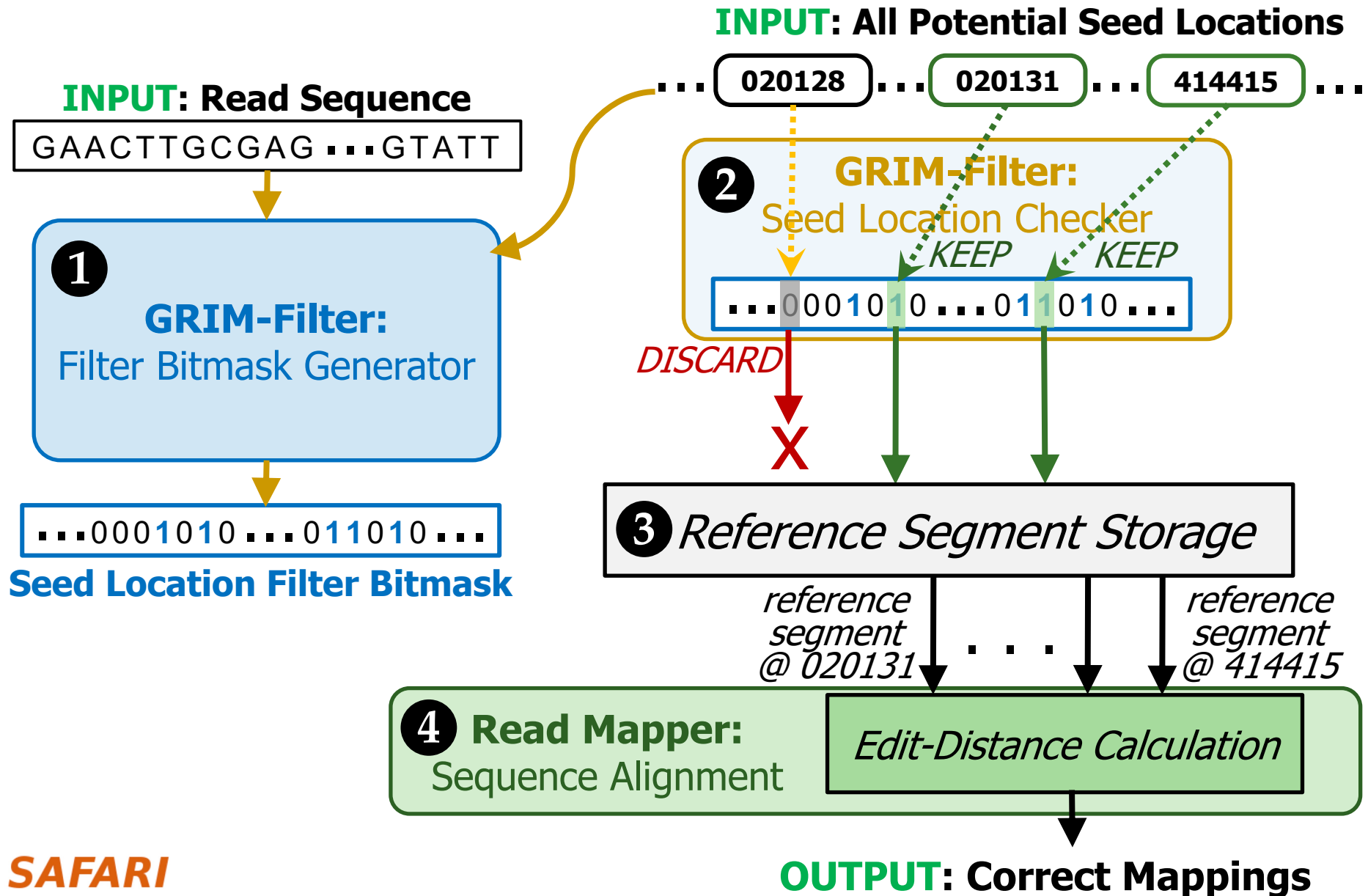
How GRIM-Filter determines whether to **discard** potential match locations in a given bin **prior** to alignment



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:

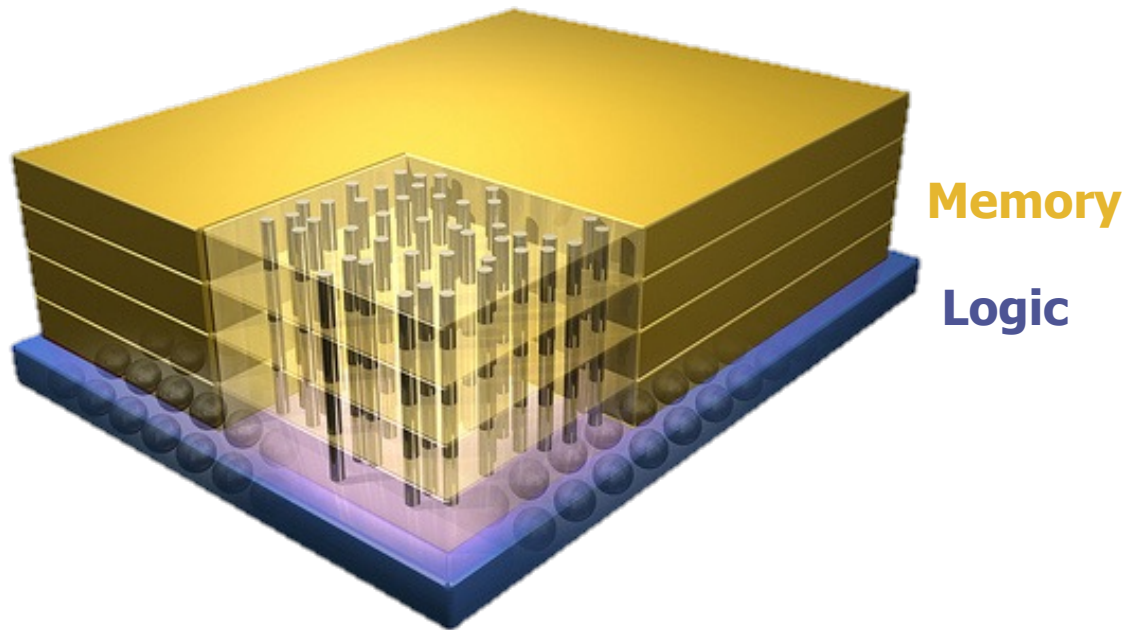
- ❑ To check a given bin, find the **sum** of all bits corresponding to each token in the read
- ❑ **Compare** against threshold to determine whether to align

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

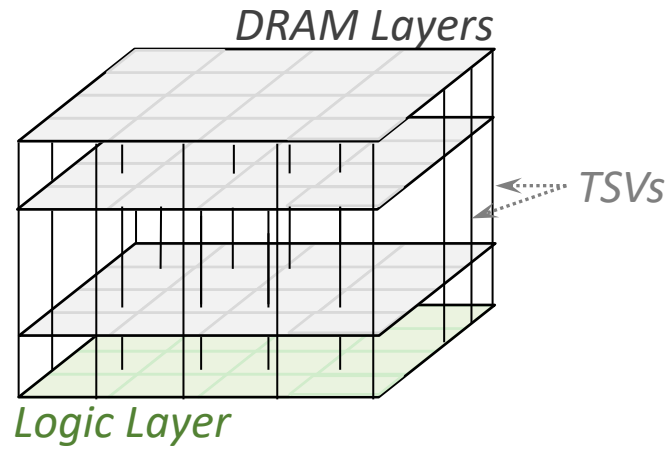
DRAM Landscape (circa 2015)

<i>Segment</i>	<i>DRAM Standards & Architectures</i>
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.

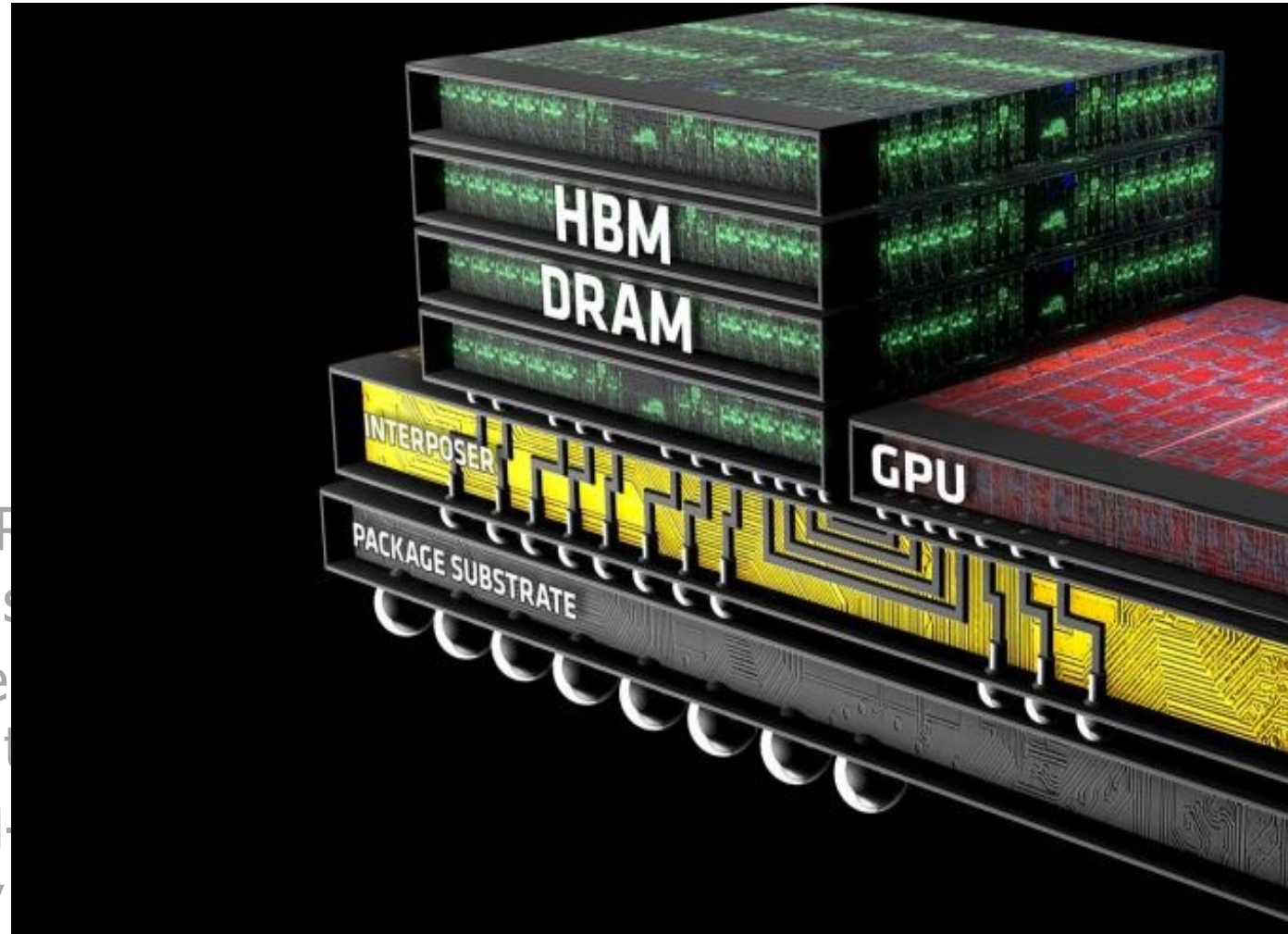
3D-Stacked Memory



- 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

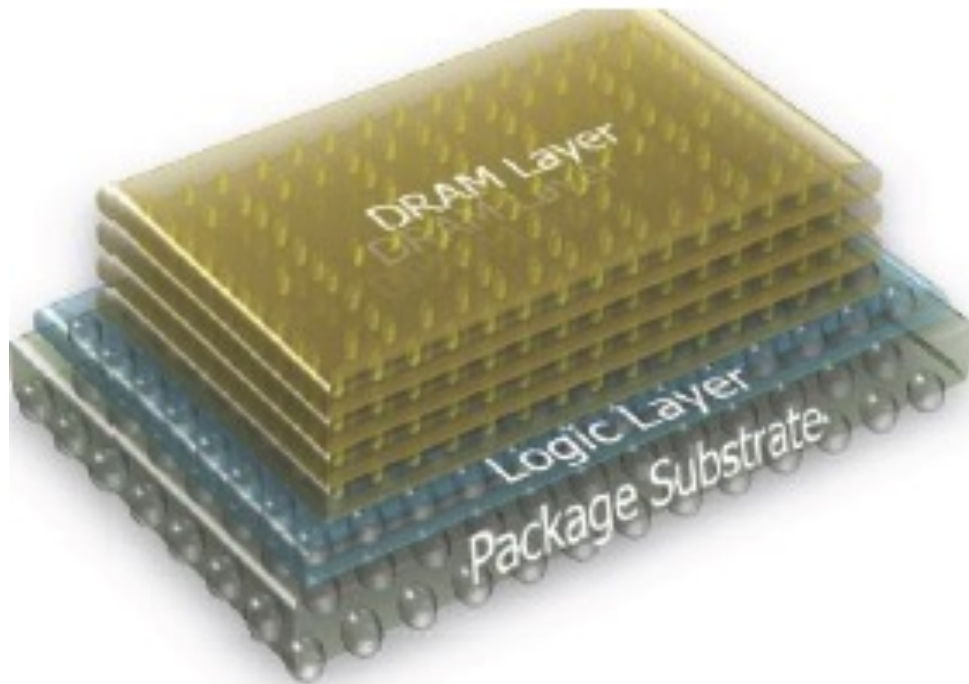
3D-Stacked Memory

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



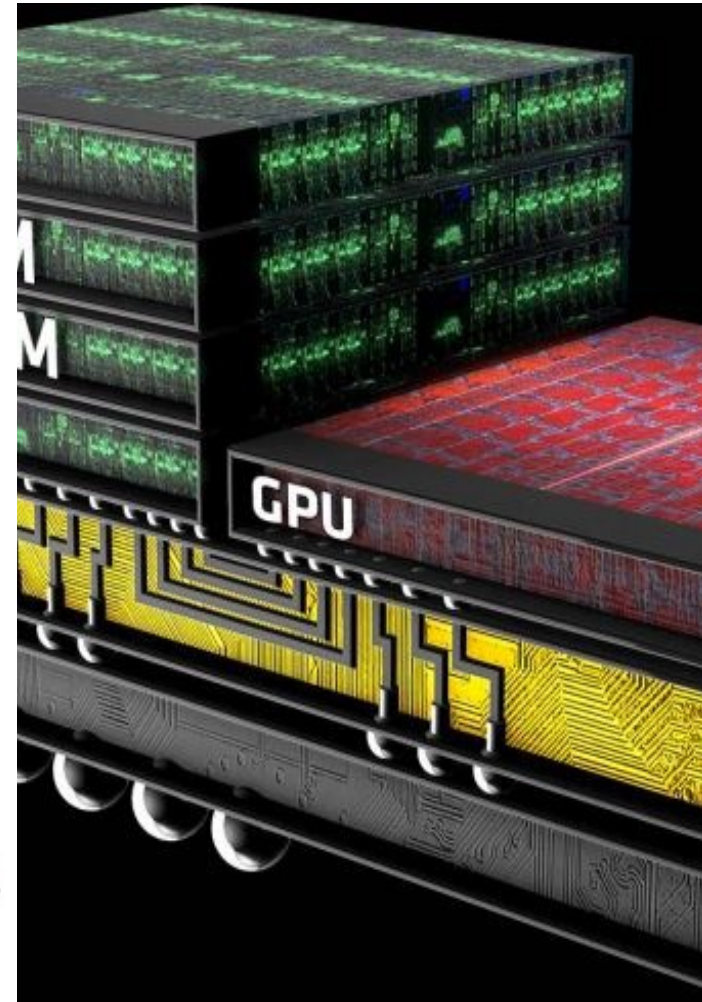
3D-Stacked Memory

Micron's HMC



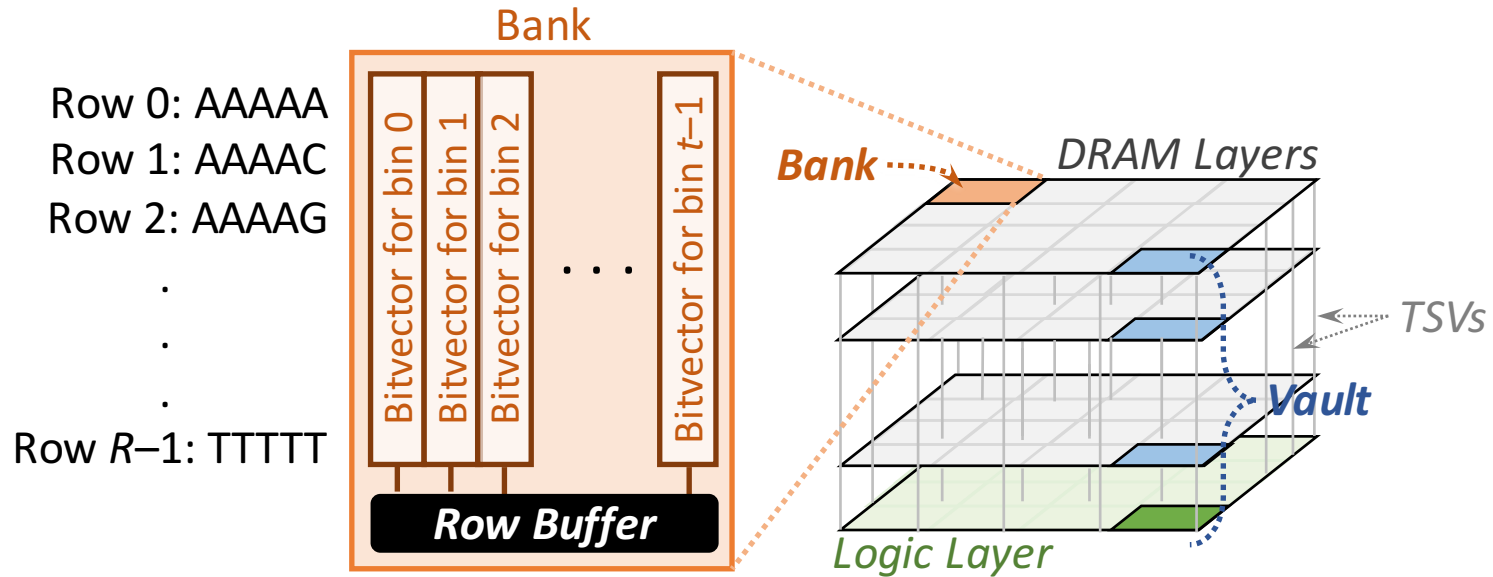
Micron has working demonstration components

http://images.anandtech.com/doci/9266/HBMCa_678x452.jpg



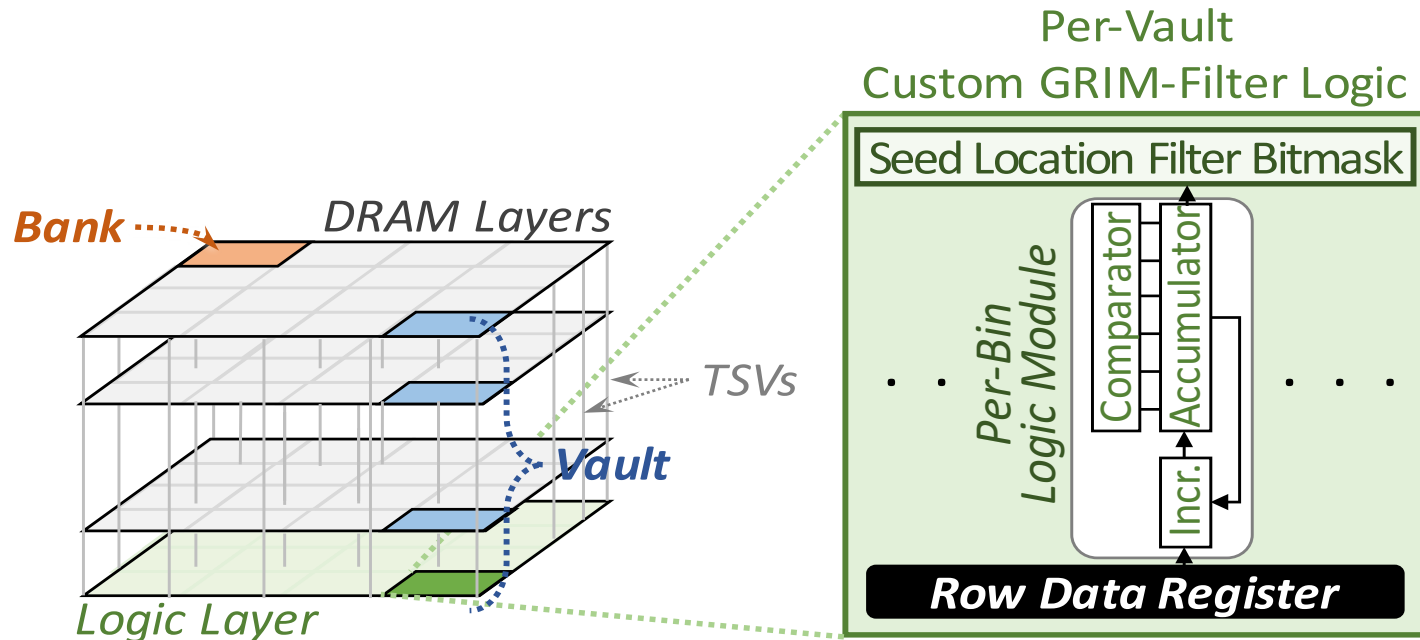
<http://i1-news.softpedia-static.com/images/news2/Micron-and-Samsung-Join-Force-to-Create-Next-Gen-Hybrid-Memory-2.png>

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

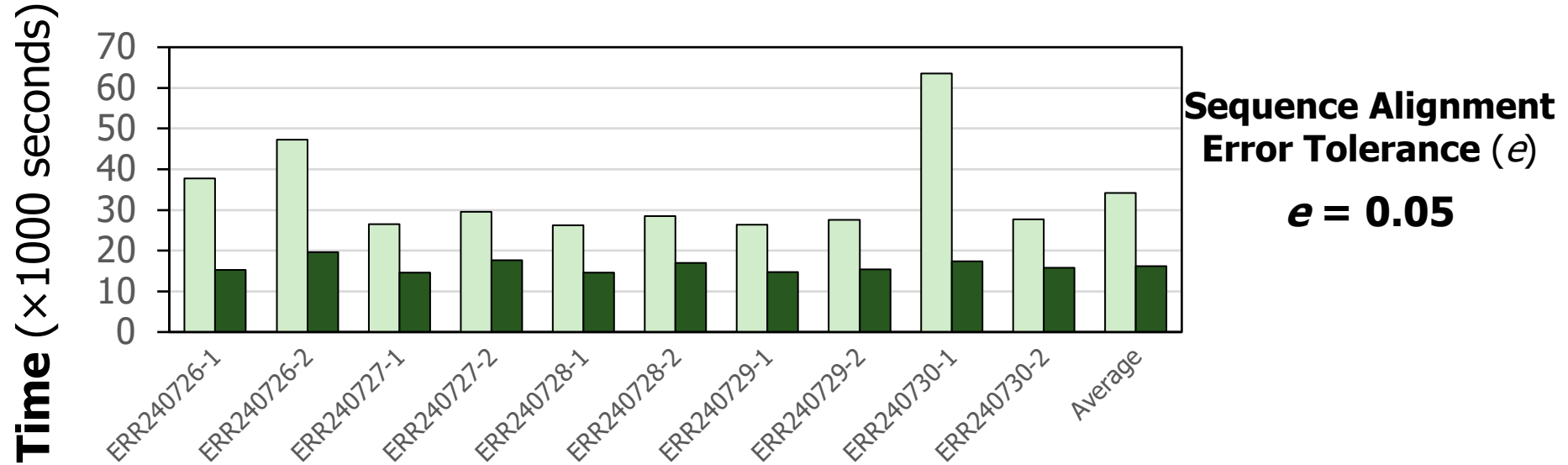
Methodology

- 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

Benchmarks and their Execution Times

FastHASH filter GRIM-Filter



1.8x-3.7x performance benefit across real data sets

2.1x average performance benefit

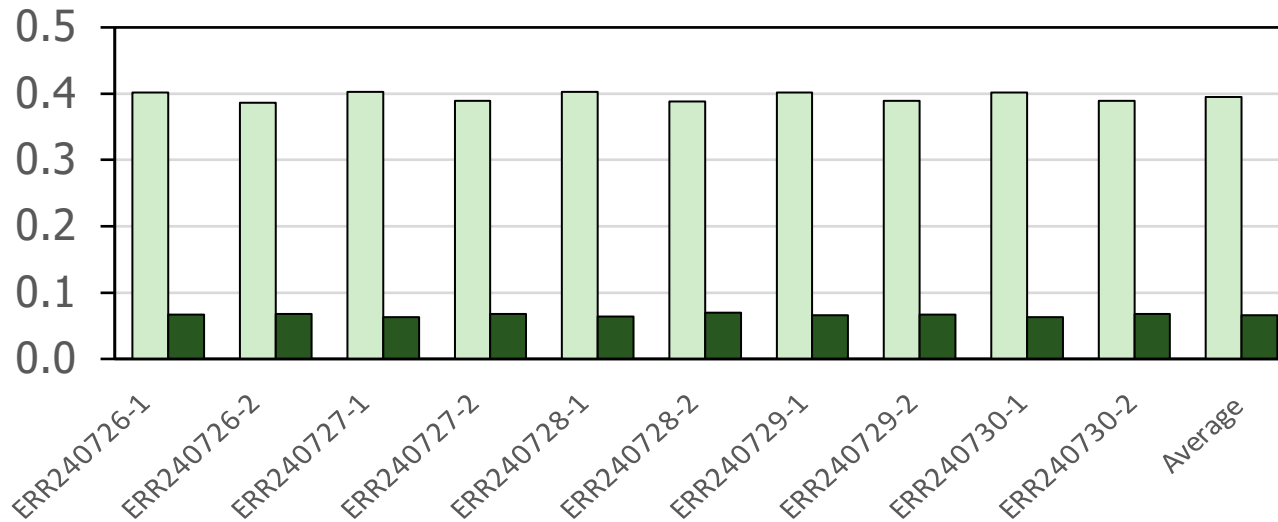
GRIM-Filter gets performance due to its hardware-software co-design

GRIM-Filter False Negative Rate

Benchmarks and their False Negative Rates

FastHASH filter GRIM-Filter

False Negative Rate



Sequence Alignment
Error Tolerance (e)

$e = 0.05$

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)]

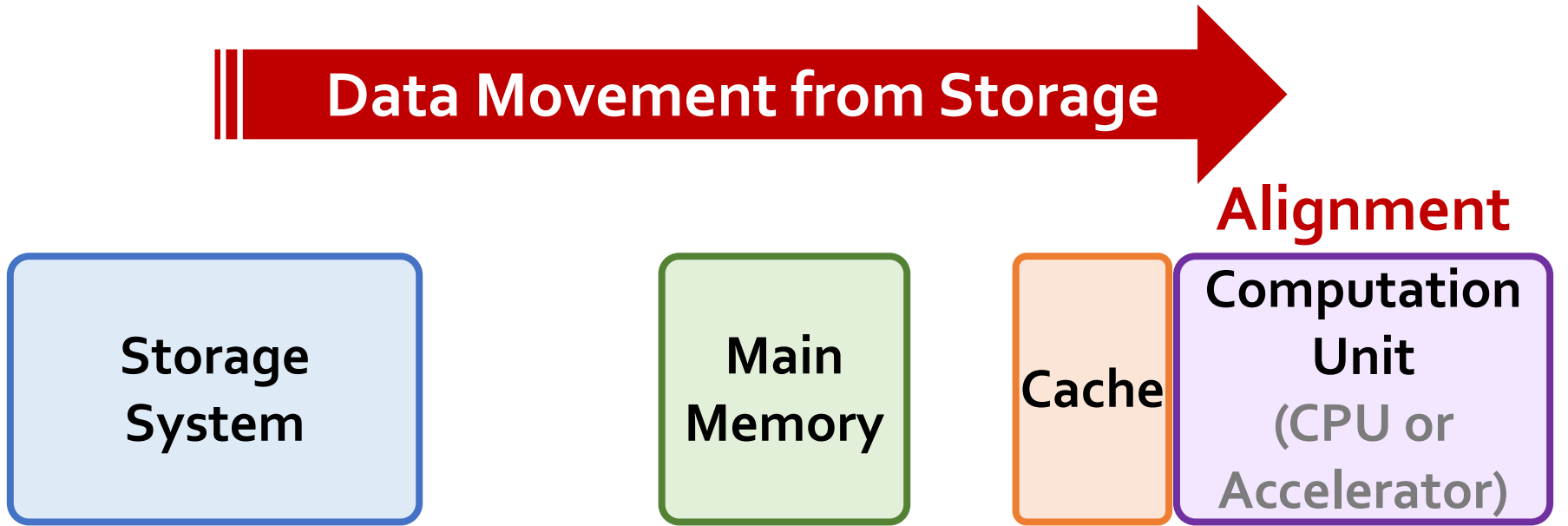
GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

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

¹ETH Zürich ²Bionano Genomics ³KMUTNB ⁴University of Toronto

Genome Sequence Analysis

Data Movement from Storage



The diagram illustrates the data flow in genome sequence analysis. A large red arrow points from left to right, labeled 'Data Movement from Storage'. Below the arrow, four components are arranged horizontally: a light blue box for 'Storage System', a light green box for 'Main Memory', a light orange box for 'Cache', and a light purple box for 'Computation Unit (CPU or Accelerator)'. The 'Computation Unit' box is also labeled 'Alignment' in red text above it.

Alignment

Storage
System

Main
Memory

Cache

Computation
Unit
(CPU or
Accelerator)

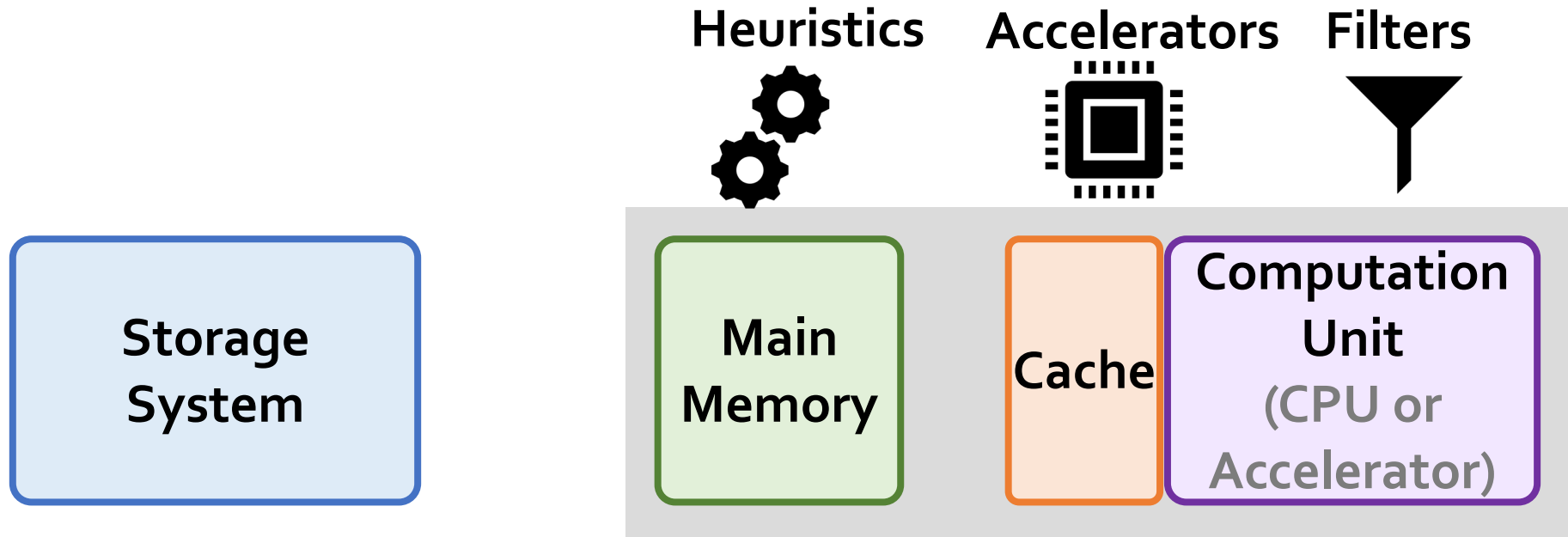


Computation overhead



Data movement overhead

Accelerating Genome Sequence Analysis



Computation overhead

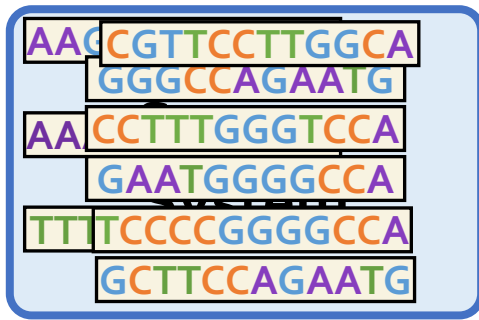


Data movement overhead

Key Idea



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



Filtered Reads

**Main
Memory**

Cache

**Computation
Unit
(CPU or
Accelerator)**

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

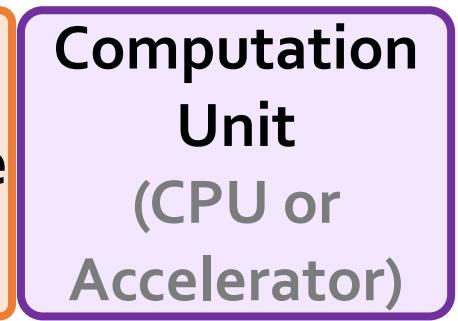
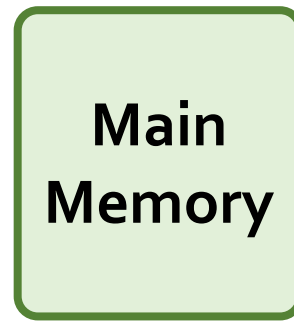
Do not have potential matching locations, so they skip alignment

- 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*



Filtered Reads

Read mapping workloads can exhibit different behavior

There are **limited hardware resources**
in the storage system

GenStore



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

GenStore-Enabled
Storage
System

Main
Memory

Cache

Computation
Unit
(CPU or
Accelerator)



Computation overhead



Data movement overhead

GenStore provides significant speedup (1.4x - 33.6x) and
energy reduction (3.9x - 29.2x) at low cost

In-Storage Genome Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,
"GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"
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 Ghiasi¹ Jisung Park¹ Harun Mustafa¹ Jeremie Kim¹ Ataberk Olgun¹
Arvid Gollwitzer¹ Damla Senol Cali² Can Firtina¹ Haiyu Mao¹ Nour Almadhoun Alserr¹
Rachata Ausavarungnirun³ Nandita Vijaykumar⁴ Mohammed Alser¹ Onur Mutlu¹

¹ETH Zürich ²Bionano Genomics ³KMUTNB ⁴University of Toronto

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.*

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]

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>

Memory-Centric Computing Systems



Onur Mutlu

omutlu@gmail.com

<https://people.inf.ethz.ch/omutlu>

12 December 2020

IEDM Tutorial

SAFARI

ETH zürich

Carnegie Mellon



0:06 / 1:51:05



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

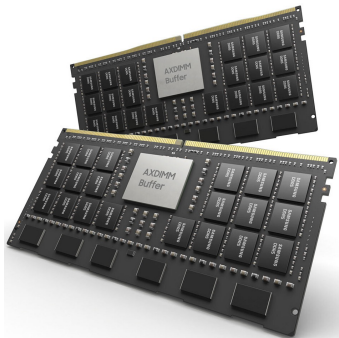
<https://www.youtube.com/watch?v=H3sEaINPBOE>

ANALYTICS

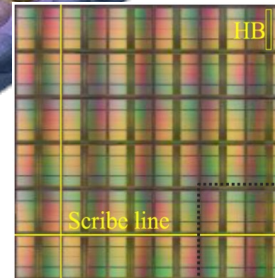
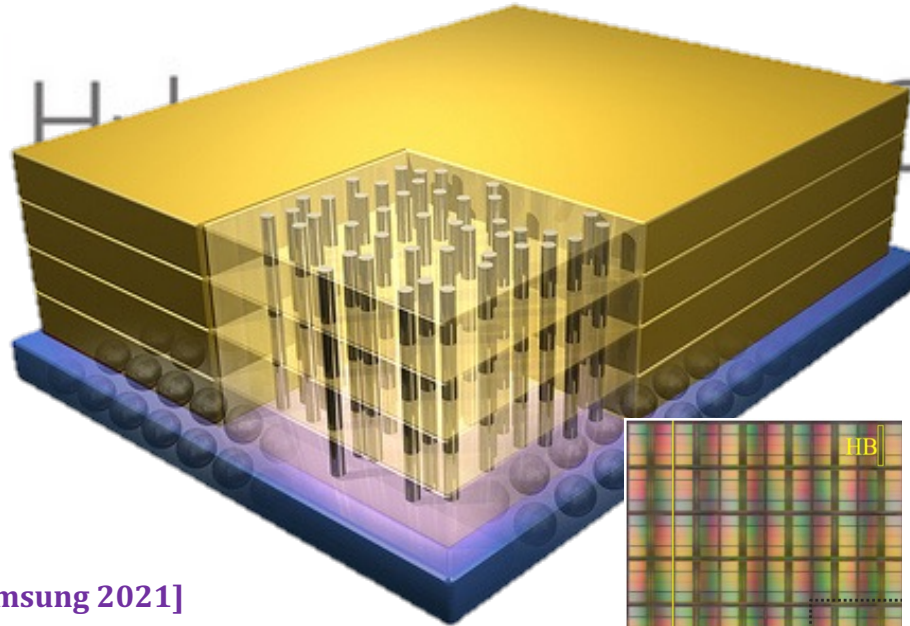
EDIT VIDEO

<https://www.youtube.com/onurmutlulectures>

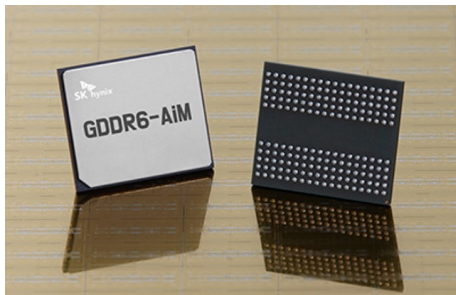
Processing-in-Memory Landscape Today



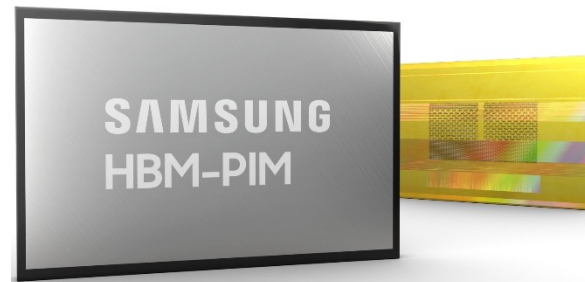
[Samsung 2021]



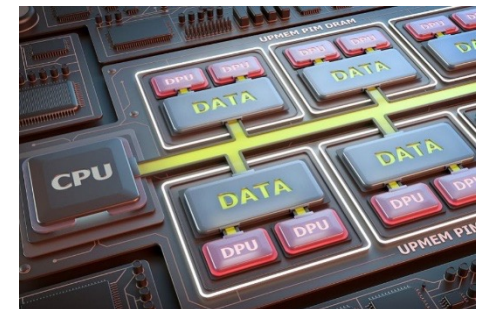
[Alibaba 2022]



[SK Hynix 2022]



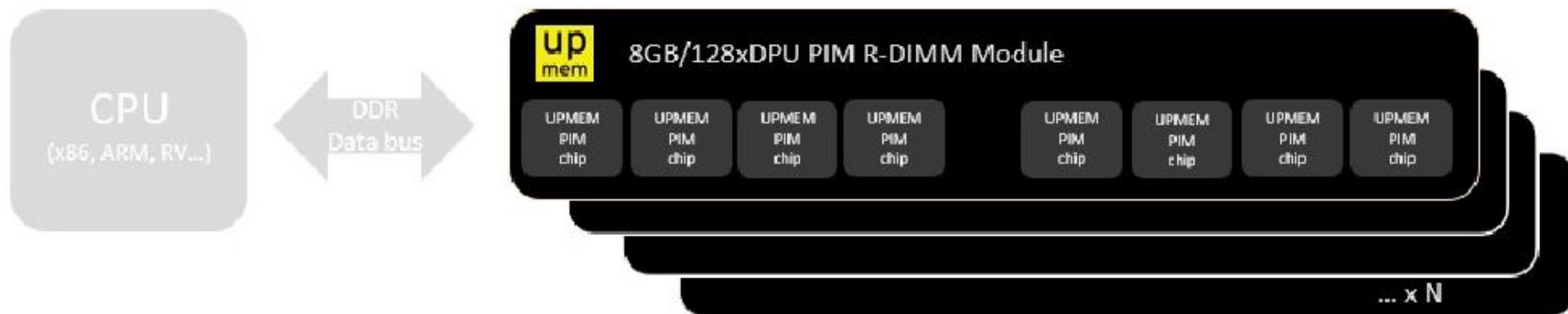
[Samsung 2021]



[UPMEM 2019]

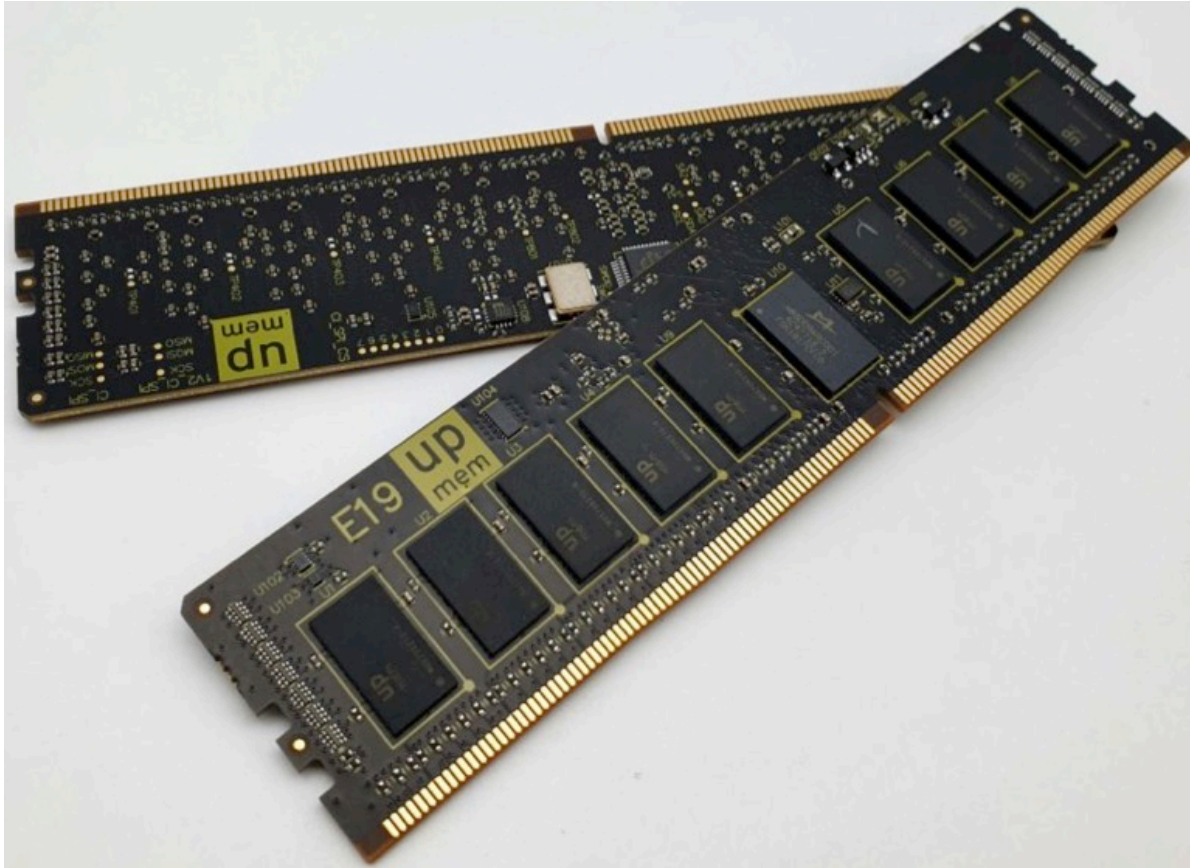
UPMEM Processing-in-DRAM Engine (2019)

- **Processing in DRAM Engine**
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.
- Replaces **standard DIMMs**
 - DDR4 R-DIMM modules
 - 8GB+128 DPUs (16 PIM chips)
 - Standard 2x-nm DRAM process
 - **Large amounts of** compute & memory bandwidth

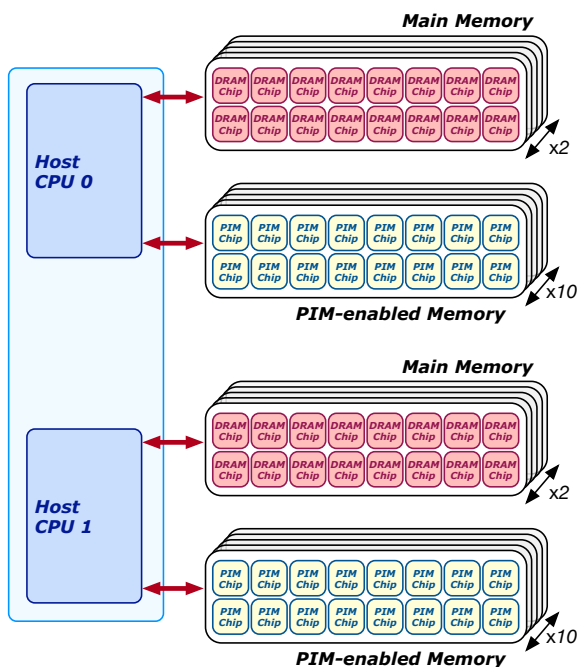


UPMEM Memory Modules

- E19: 8 chips DIMM (1 rank). DPUs @ 267 MHz
- P21: 16 chips DIMM (2 ranks). DPUs @ 350 MHz



2,560-DPU Processing-in-Memory System



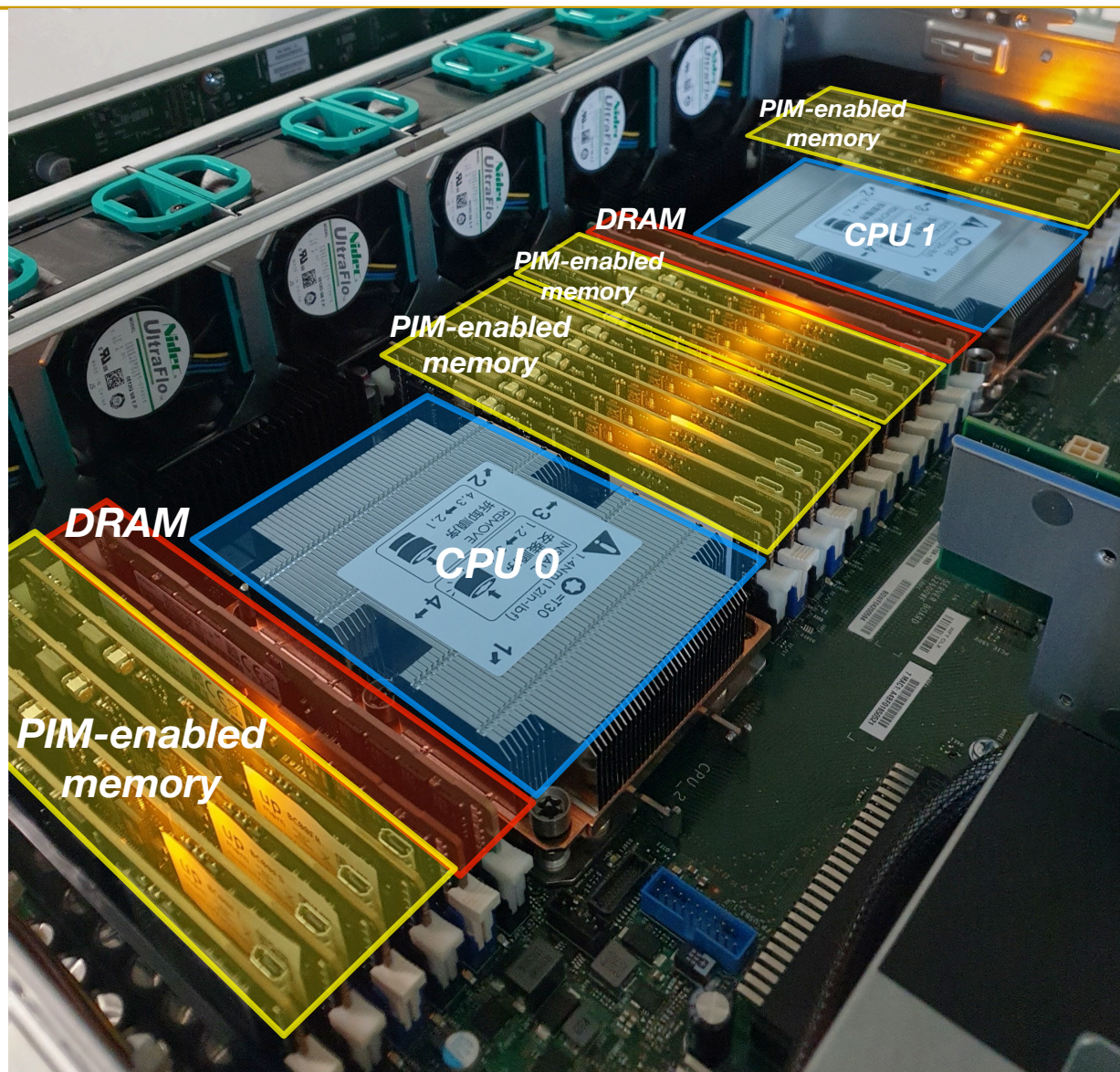
Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

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

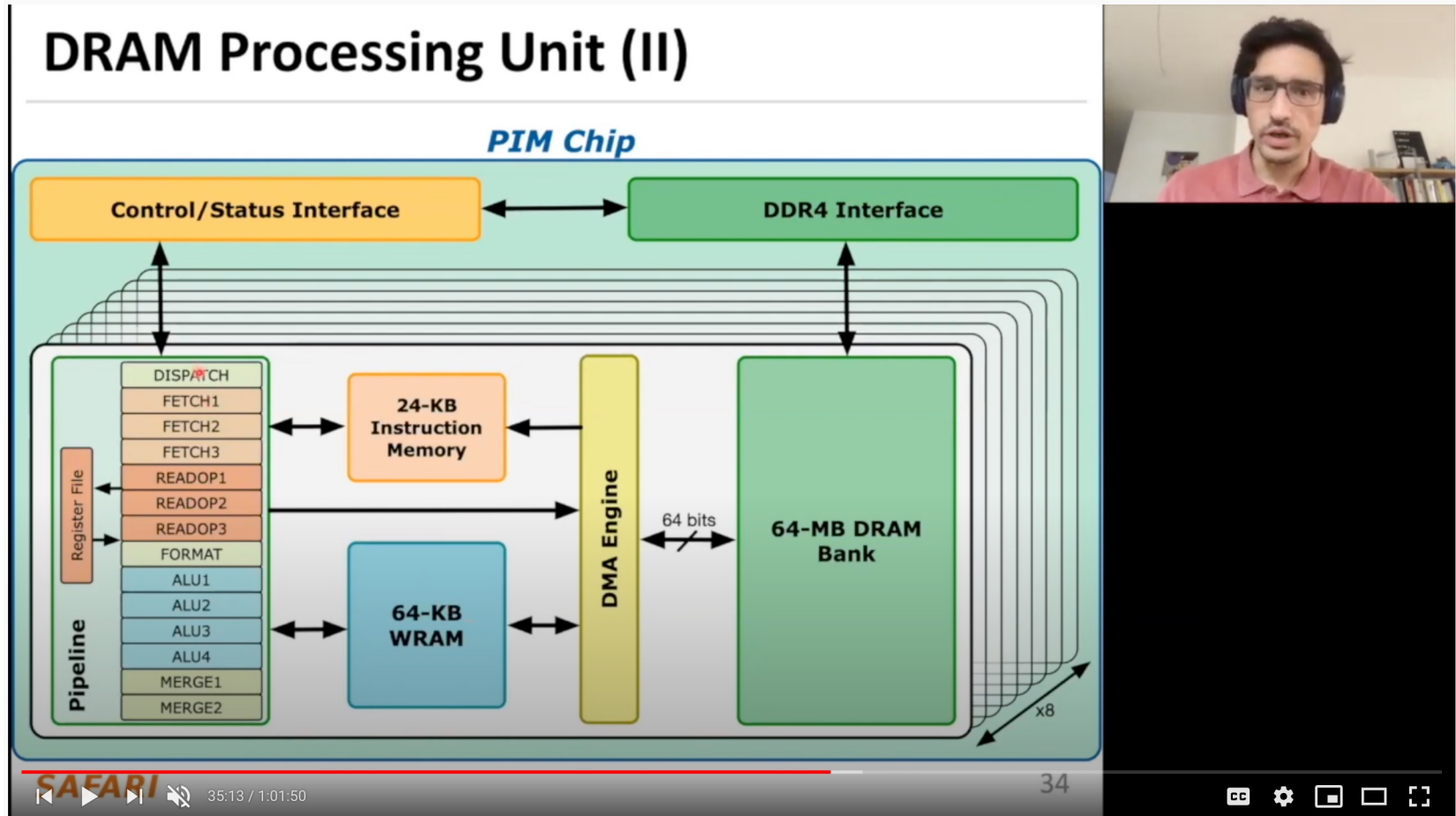
Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware 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 2,560 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.



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

More on the UPMEM PIM System



ETH ZÜRICH HAUPTGEBÄUDE

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

1,120 views • Oct 31, 2020

30 0 SHARE SAVE ...



Onur Mutlu Lectures
16.7K subscribers

ANALYTICS

EDIT VIDEO

<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.

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

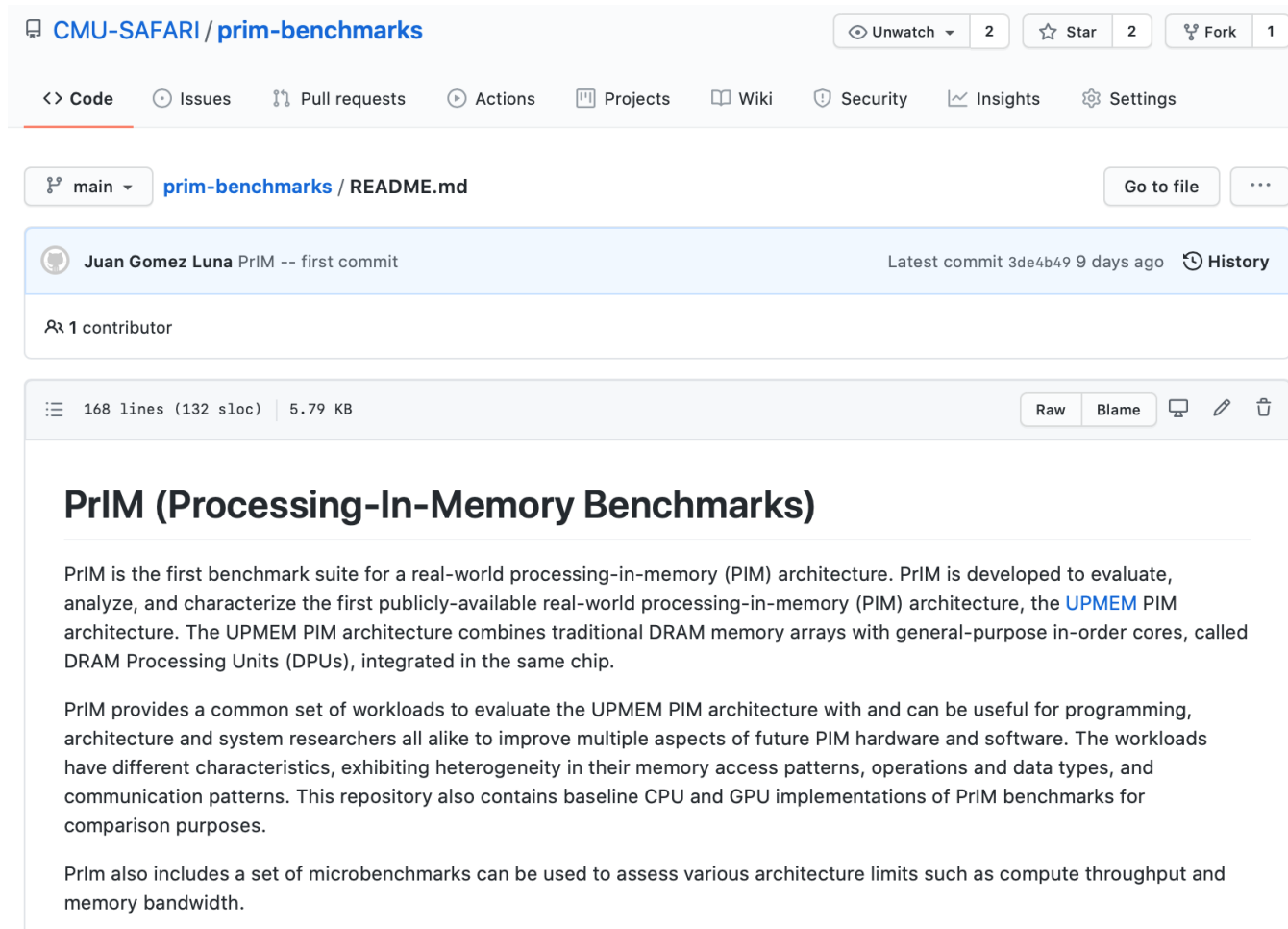
Onur Mutlu
ETH Zürich

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>



CMU-SAFARI / **prim-benchmarks** Unwatch 2 Star 2 Fork 1

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main prim-benchmarks / README.md Go to file ...

Juan Gomez Luna PrIM -- first commit Latest commit 3de4b49 9 days ago History

1 contributor

168 lines (132 sloc) 5.79 KB Raw Blame

PrIM (Processing-In-Memory Benchmarks)

PrIM is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publicly-available real-world processing-in-memory (PIM) architecture, the [UPMEM](#) 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.

PrIM provides a common set of workloads to evaluate the UPMEM PIM architecture with and can be useful for programming, architecture and system researchers all alike to improve multiple aspects of future PIM hardware and software. The workloads have different characteristics, exhibiting heterogeneity in their memory access patterns, operations and data types, and communication patterns. This repository also contains baseline CPU and GPU implementations of PrIM benchmarks for comparison purposes.

PrIm also includes a set of microbenchmarks can be used to assess various architecture limits such as compute throughput and memory bandwidth.

Understanding a Modern PIM Architecture

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

**JUAN GÓMEZ-LUNA¹, IZZAT EL HAJJ², IVAN FERNANDEZ^{1,3}, CHRISTINA GIANNOULA^{1,4},
GERALDO F. OLIVEIRA¹, AND ONUR MUTLU¹**

¹ETH Zürich

²American University of Beirut

³University of Malaga

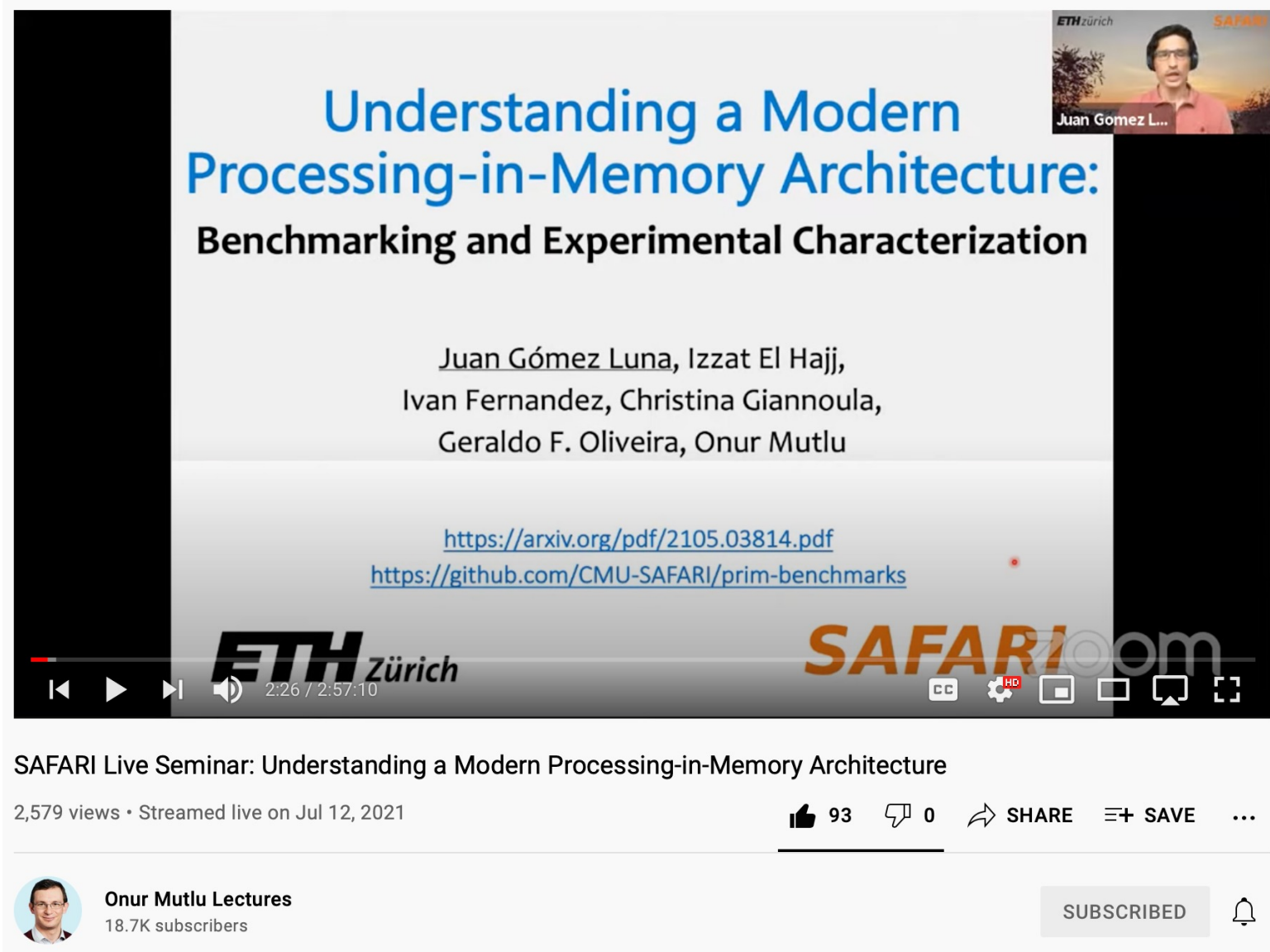
⁴National Technical University of Athens

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



The image shows a YouTube video player interface. The video title is "Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization". The speaker is Juan Gómez Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu. The video is from the channel "Onur Mutlu Lectures" and has 18.7K subscribers. The video is titled "SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture" and has 2,579 views. The video is streamed live on Jul 12, 2021. The video player shows the title, speaker names, and links to the arXiv paper and GitHub repository. The video player also shows the ETH Zürich and SAFARI logos. The video player has a progress bar at 2:26 / 2:57:10. The video player has a like button with 93 likes, a comment button with 0 comments, a share button, a save button, and a bell icon. The video player has a "SUBSCRIBED" button.

Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

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

<https://arxiv.org/pdf/2105.03814.pdf>
<https://github.com/CMU-SAFARI/prim-benchmarks>

ETH Zürich SAFARI

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

2,579 views • Streamed live on Jul 12, 2021

93 0 SHARE SAVE ...

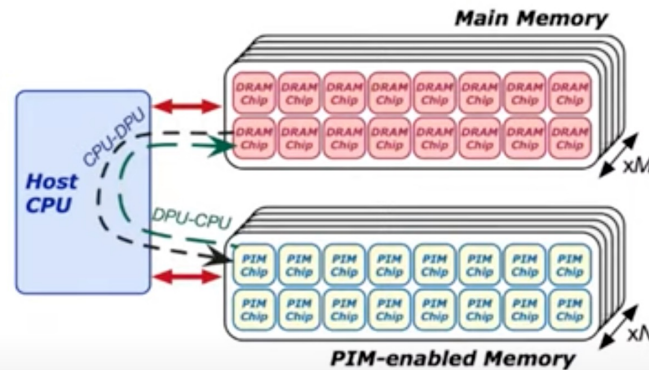
 **Onur Mutlu Lectures**
18.7K subscribers

SUBSCRIBED

More on Analysis of the UPMEM PIM Engine

Inter-DPU Communication

- There is **no direct communication channel between DPUs**



- Inter-DPU communication takes place via the host CPU using CPU-DPU and DPU-CPU transfers
- Example communication patterns:
 - Merging of partial results to obtain the final result
 - Only DPU-CPU transfers
 - Redistribution of intermediate results for further computation
 - DPU-CPU transfers and CPU-DPU transfers



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

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

ANALYTICS

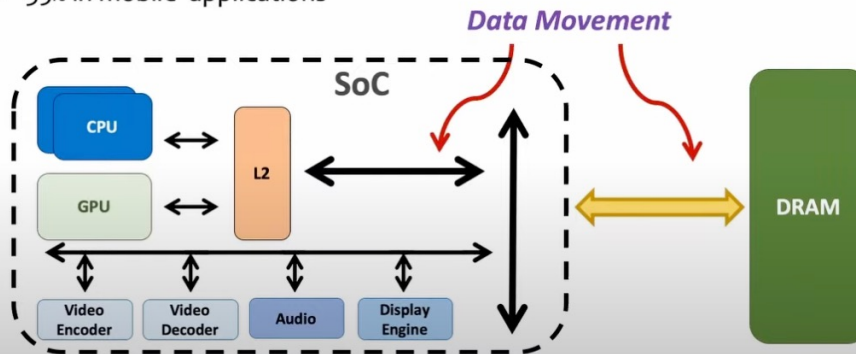
EDIT VIDEO

https://www.youtube.com/watch?v=D8Hjy2IU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9

More on Analysis of the UPMEM PIM Engine

Data Movement in Computing Systems

- **Data movement** dominates **performance** and is a major system **energy bottleneck**
- **Total system energy**: data movement accounts for
 - 62% in consumer applications*,
 - 40% in scientific applications*,
 - 35% in mobile applications*



* Boroumand et al., "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS 2018

* Kestor et al., "Quantifying the Energy Cost of Data Movement in Scientific Applications," IISWC 2013

* Pandiyan and Wu, "Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms," IISWC 2014

SAFARI

3

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	Izzat El Hajj	Ivan Fernandez	Christina Giannoula	Geraldo F. Oliveira	Onur Mutlu
<i>ETH Zürich</i>	<i>American University of Beirut</i>	<i>University of Malaga</i>	<i>National Technical University of Athens</i>	<i>ETH Zürich</i>	<i>ETH Zürich</i>

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 Mansouri Ghiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,
"SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
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 Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Zülal Bingöl³
Gurpreet S. Kalsi⁴ Ziyi Zuo⁵ Can Firtina² Meryem Banu Cavlak² Jeremie Kim²
Nika Mansouri Ghiasi² Gagandeep Singh² Juan Gómez-Luna² Nour Almadhoun Alserr²
Mohammed Alser² Sreenivas Subramoney⁴ Can Alkan³ Saugata Ghose⁶ Onur Mutlu²

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

New Applications: Ref Genome Updates

RESEARCH

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

Jeremie S. Kim¹, Can Firtina¹, Meryem Banu Cavlak², Damla Senol Cali³, Nastaran Hajinazar^{1,4}, Mohammed Alser¹, Can Alkan² and Onur Mutlu^{1,2,3*}

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

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
sequencing 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

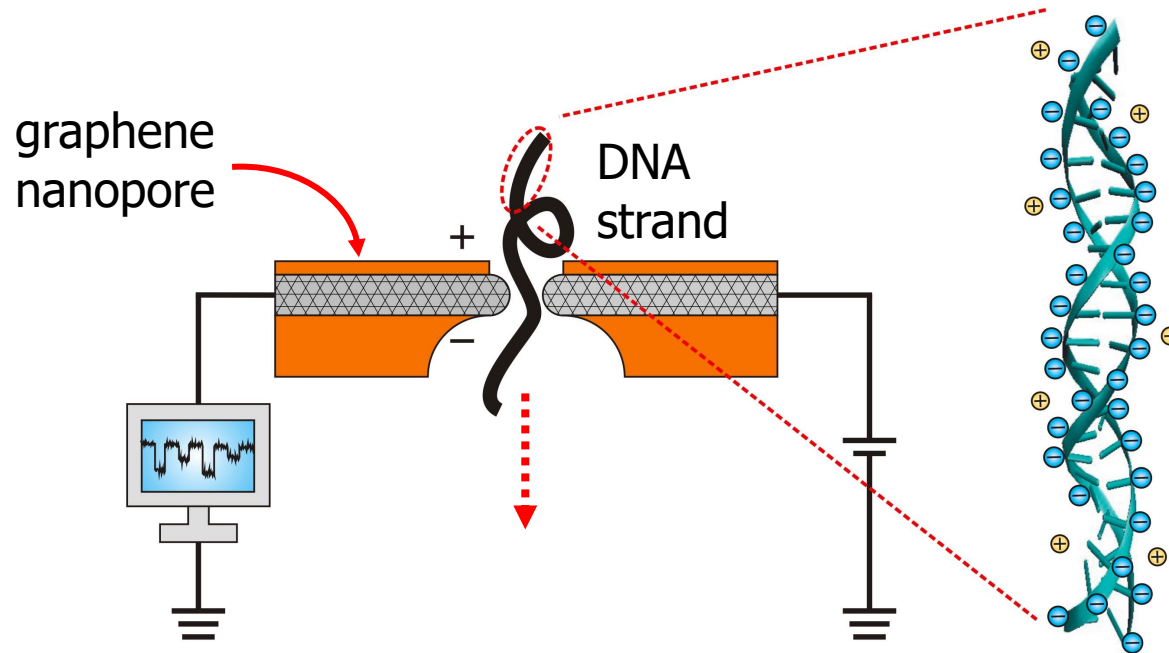
	MinION Mk1B	MinION Mk1C	GridION Mk1	PromethION 24	PromethION 48
Read length	> 2Mb	> 2Mb	> 2Mb	> 2Mb	> 2Mb
Yield per flow cell	50 Gb	50 Gb	50 Gb	220 Gb	220 Gb
Number of flow cells per device	1	1	5	24	48
Yield per device	<50 Gb	<50 Gb	<250 Gb	<5.2 Tb	<10.5 Tb
Starting price	\$1,000	\$4,990	\$49,995	\$195,455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

Illumina Sequencers



	iSeq 100	MiniSeq	MiSeq	NextSeq 550	NextSeq 2000	NovaSeq 6000
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 ($<20\text{nm}$).
- In nanopore sequencers, an **ionic current** passes through the nanopores
- When the DNA strand passes through the nanopore, the sequencer measures 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

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

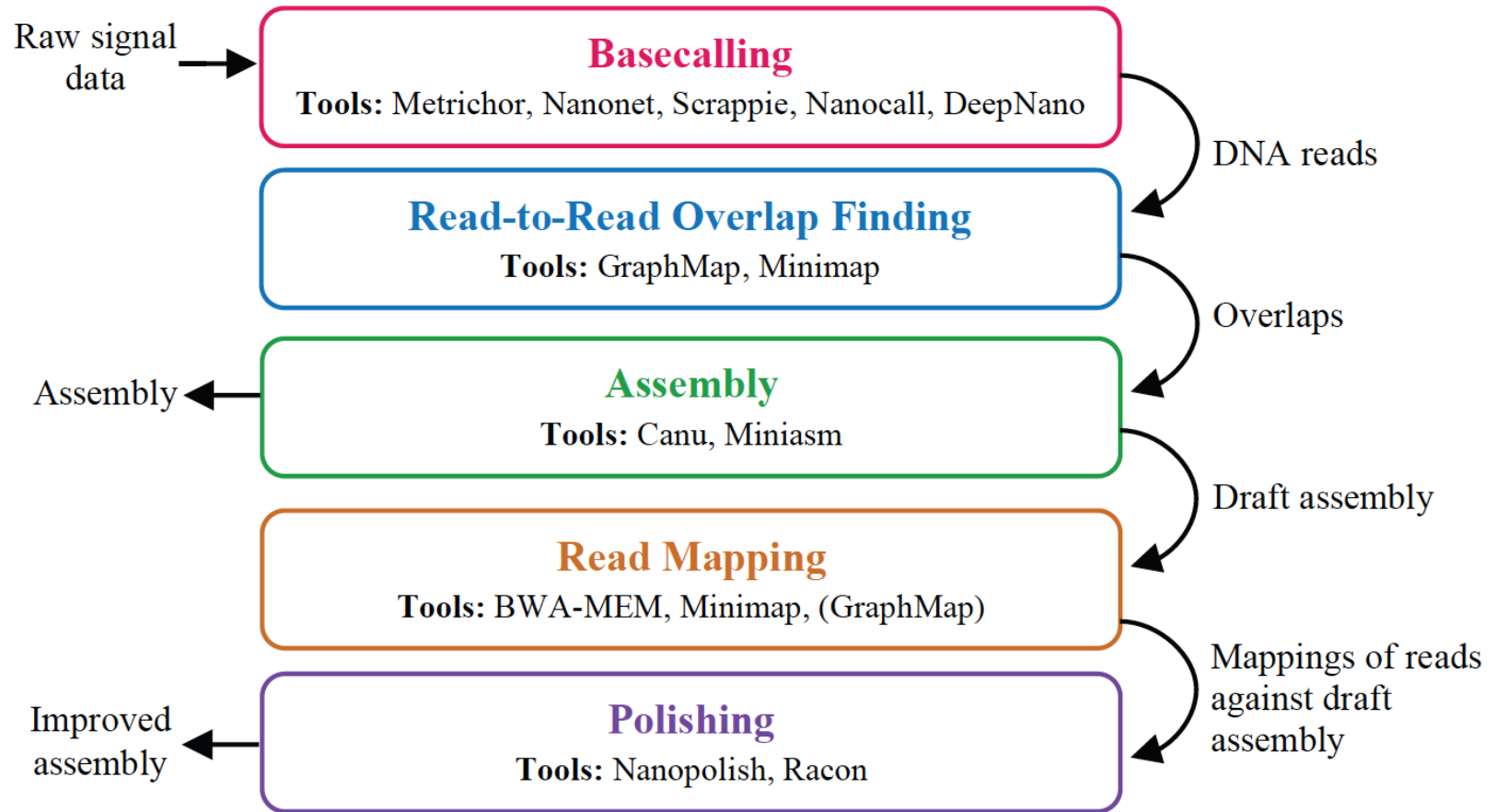


Figure 1. The analyzed genome assembly pipeline using nanopore sequence data, with its five steps and the associated tools for each step.

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

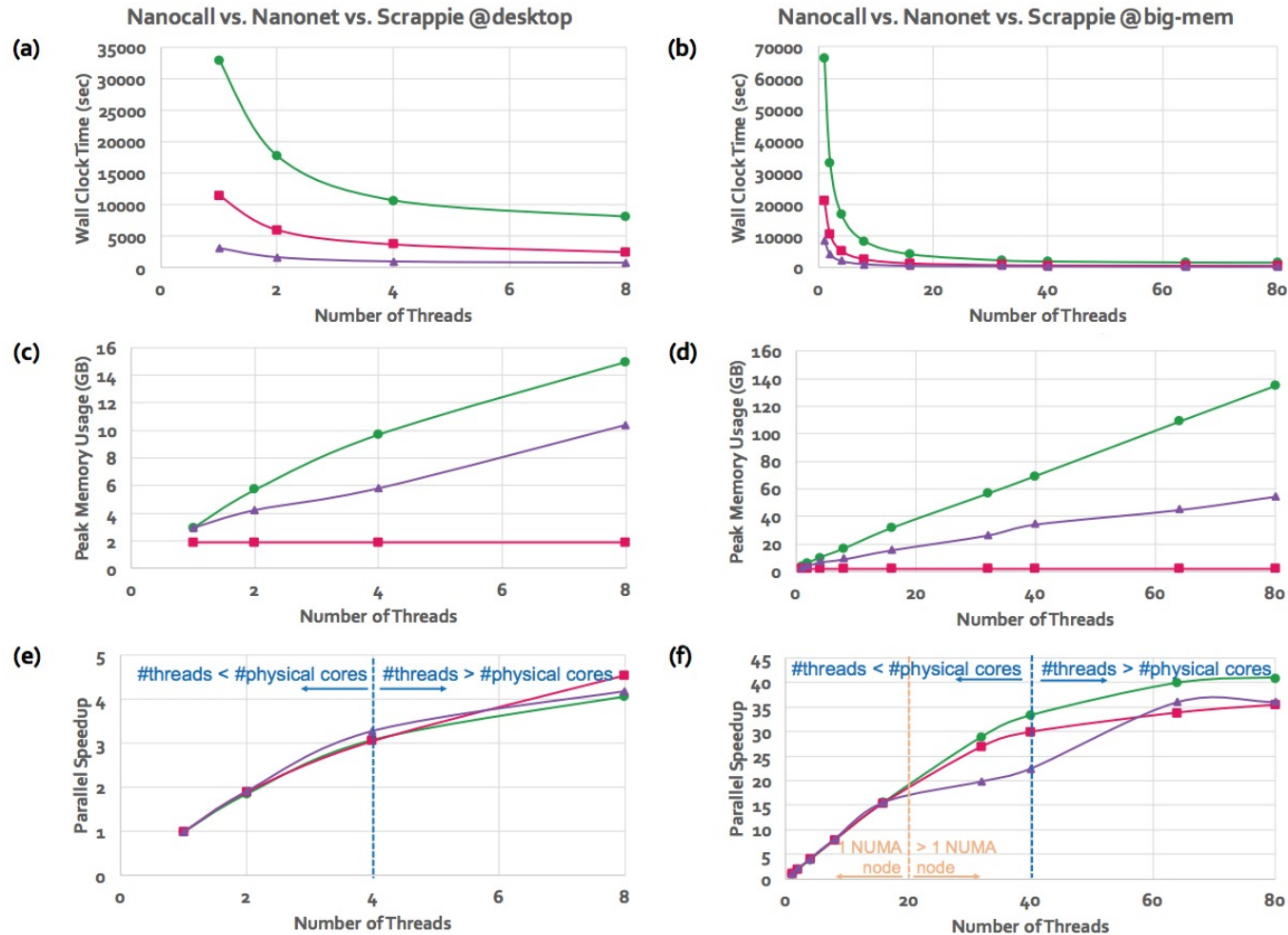
Nanopore Genome Assembly Tools (II)

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

Nanopore Genome Assembly Tools (III)

● Nanocall ■ Nanonet ▲ Scrappie



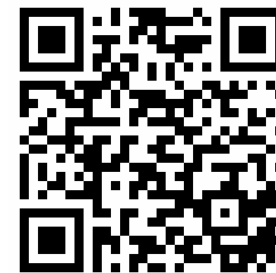
More on Nanopore Sequencing & Tools

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.

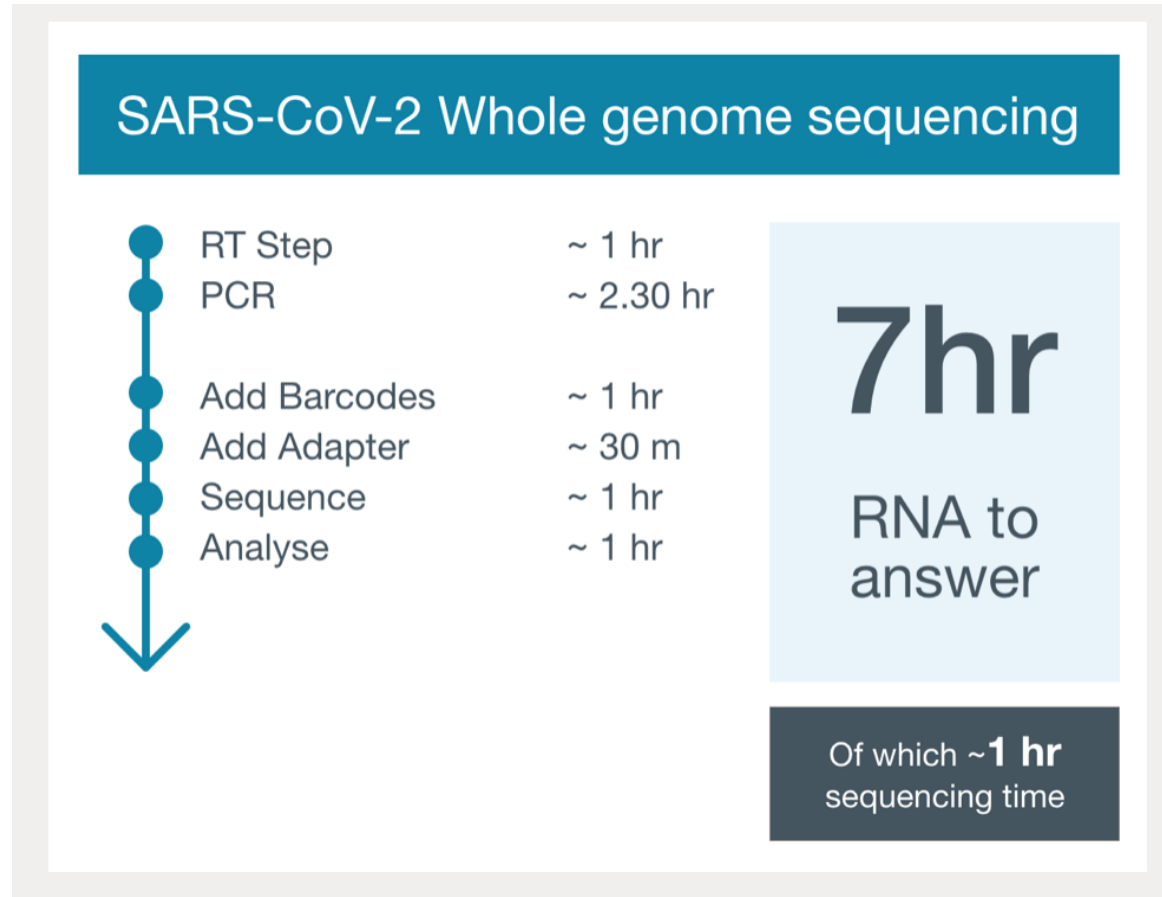


700Kg of Oxford Nanopore sequencers and consumables are on their way for use by Chinese scientists in understanding the current coronavirus outbreak.

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)

How are scientists using nanopore sequencing to research COVID-19?



Samples are collected

Validated SARS-CoV-2 RT-PCR test performed



SARS-CoV-2 positive samples



SARS-CoV-2 negative samples: used as negative controls

How can this be used?
Genomic epidemiology: analyse variants & mutation rate, track spread of virus, identify clusters of transmission

What are the results?
From RNA to full SARS-CoV-2 consensus sequence in ~7 hours

How?
Targeted amplification of SARS-CoV-2 genome + multiplexed, rapid nanopore sequencing

Targeted SARS-CoV-2 nanopore sequencing



Metagenomic nanopore sequencing

How?
1 x RNA metagenomic sequencing run
1 x DNA metagenomic sequencing run

What are the results?
RNA: data for RNA viruses (including SARS-CoV-2) + microbial transcripts
DNA: data for bacteria + DNA viruses

How can this be used?
Characterise co-infecting bacteria & viruses, identify any correlation of risk factors, research potential future treatment implications

SARS-CoV-2 Direct RNA whole genome sequencing: assess viral genome in its native RNA form and the effect of base modifications

Immune repertoire: assess response of the immune system to SARS-CoV-2 infection by sequencing of full-length immune cell receptor genes and transcripts

Whole human genome sequencing: investigate what might cause different responses to the virus in different people based on their genome

What's next?



Find out more at nanoporetech.com/covid19

MinION™



GridION™



PromethION™



Oxford Nanopore Technologies, the Wheel icon, GridION, PromethION and MinION are registered trademarks of Oxford Nanopore Technologies in various countries. © 2020 Oxford Nanopore Technologies. All rights reserved. Oxford Nanopore Technologies' products are currently for research use only. IG_1061[EN]_V1_03Apr2020

• 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](https://doi.org/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](https://doi.org/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

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](https://doi.org/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](https://doi.org/10.1109/MM.2021.3088396)



SmidgION from ONT

Resources & Acknowledgments

Accelerating Genome Analysis: Overview

- Mohammed Alser, Zülal Bingöl, 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.*

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]

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>

Memory-Centric Computing Systems



Onur Mutlu

omutlu@gmail.com

<https://people.inf.ethz.ch/omutlu>

12 December 2020

IEDM Tutorial

SAFARI

ETH zürich

Carnegie Mellon



0:06 / 1:51:05



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>

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>

A Longer Version of This Lecture...

- Onur Mutlu,
"Accelerating Genome Analysis: A Primer on an Ongoing Journey"
Invited Lecture at [Technion](#), Virtual, 26 January 2021.
[[Slides \(pptx\)](#) ([pdf](#))]
[[Talk Video](#) (1 hour 37 minutes, including Q&A)]
[[Related Invited Paper \(at IEEE Micro, 2020\)](#)]

Insight: Shifting a String Helps Similarity Search

7 matches 1 mismatch

ISTANBUL

ISTNBUL

ISTNBUL

81

46:08 / 1:37:37

Onur Mutlu - Invited Lecture @Technion: Accelerating Genome Analysis: A Primer on an Ongoing Journey

566 views · Premiered Feb 6, 2021

31 0 SHARE SAVE ...

Onur Mutlu Lectures
13.9K subscribers

ANALYTICS EDIT VIDEO

Comp Arch (Fall 2021)

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

- Computer Architecture FS20: Course Webpage
- Computer Architecture FS20: Lecture Videos
- Digitaltechnik SS21: Course Webpage
- Digitaltechnik SS21: Lecture Videos
- Moodle
- HotCRP
- Verilog Practice Website (HDLBits)

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>

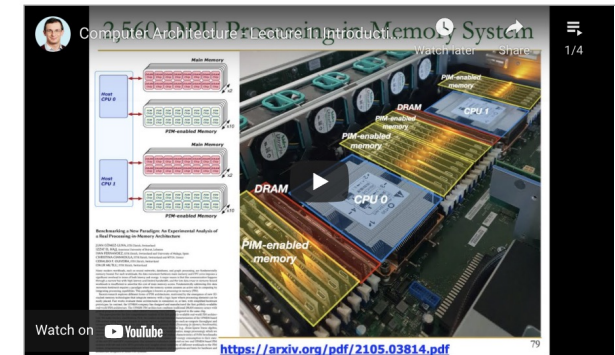
Master's level course

- 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

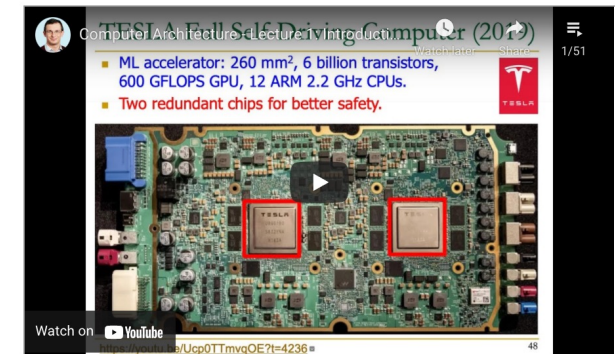
<https://www.youtube.com/onurmutlulectures>

Lecture Video Playlist on YouTube

Livestream Lecture Playlist



Recorded Lecture Playlist



Fall 2021 Lectures & Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	30.09 Thu.	YouTube Live	L1: Introduction and Basics (PDF) (PPT)	Required Mentioned	Lab 1 Out	HW 0 Out
	01.10 Fri.	YouTube Live	L2: Trends, Tradeoffs and Design Fundamentals (PDF) (PPT)	Required Mentioned		
W2	07.10 Thu.	YouTube Live	L3a: Memory Systems: Challenges and Opportunities (PDF) (PPT)	Described Suggested		HW 1 Out
			L3b: Course Info & Logistics (PDF) (PPT)			
			L3c: Memory Performance Attacks (PDF) (PPT)	Described Suggested		
	08.10 Fri.	YouTube Live	L4a: Memory Performance Attacks (PDF) (PPT)	Described Suggested	Lab 2 Out	
			L4b: Data Retention and Memory Refresh (PDF) (PPT)	Described Suggested		
			L4c: RowHammer (PDF) (PPT)	Described Suggested		

DDCA (Spring 2022)

Spring 2022 Edition:

- <https://safari.ethz.ch/digitaltechnik/spring2022/duku.php?id=schedule>

Spring 2021 Edition:

- <https://safari.ethz.ch/digitaltechnik/spring2021/duku.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_uej3aY39YB5pfW4SJ7LIN

Bachelor's course

- 2nd semester at ETH Zurich
- Rigorous introduction into "How Computers Work"
- Digital Design/Logic
- Computer Architecture
- 10 FPGA Lab Assignments

SAFARI

<https://www.youtube.com/onurmutlulectures>



Digital Design and Computer Architecture -
Spring 2021

Search
Recent Changes Media Manager Site

Trace: - schedule

Home

Announcements

Materials

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Optional HWs
- Labs
- Extra Assignments
- Exams
- Technical Docs

Resources

- Computer Architecture (CMU) SS15: Lecture Videos
- Computer Architecture (CMU) SS15: Course Website
- Digitaltechnik SS18: Lecture Videos
- Digitaltechnik SS18: Course Website
- Digitaltechnik SS19: Lecture Videos
- Digitaltechnik SS19: Course Website
- Digitaltechnik SS20: Lecture Videos
- Digitaltechnik SS20: Course Website
- Moodle

Lecture Video Playlist on YouTube

Livestream Lecture Playlist

Recorded Lecture Playlist

Spring 2021 Lectures/Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	25.02 Thu.	YouTube Live	L1: Introduction and Basics G20a (PDF) G20b (PPT)	Required Suggested Mentioned		
	26.02 Fri.	YouTube Live	L2a: Tradeoffs, Metrics, Mindset G20a (PDF) G20b (PPT)	Required		
			L2b: Mysteries in Computer Architecture G20a (PDF) G20b (PPT)	Required Mentioned		
W2	04.03 Thu.	YouTube Live	L3a: Mysteries in Computer Architecture II G20a (PDF) G20b (PPT)	Required Suggested Mentioned		

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):

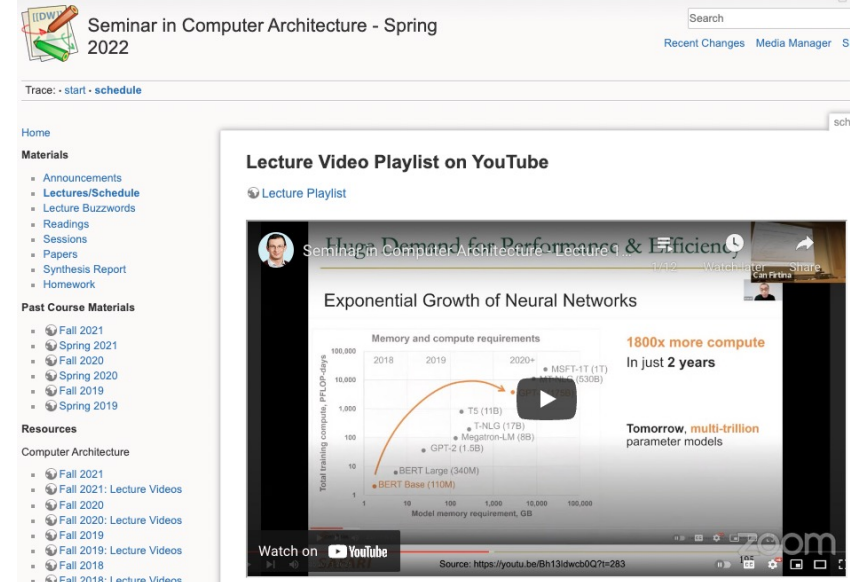
- https://www.youtube.com/watch?v=rS9UPk509AQ&list=PL5Q2soXY2Zi_hxizriwKmFHgcoe2Q8-m0

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



Seminar in Computer Architecture - Spring 2022

Trace: start - schedule

Home

Materials

- Announcements
- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Sessions
- Papers
- Synthesis Report
- Homework

Past Course Materials

- Fall 2021
- Spring 2021
- Fall 2020
- Spring 2020
- Fall 2019
- Spring 2019

Resources

Computer Architecture

- Fall 2021
- Fall 2021: Lecture Videos
- Fall 2020
- Fall 2020: Lecture Videos
- Fall 2019
- Fall 2019: Lecture Videos
- Fall 2018
- Fall 2018: Lecture Videos

Digital Design and Computer Architecture

- Spring 2021
- Spring 2021: Lecture Videos
- Spring 2020
- Spring 2020: Lecture Videos
- Spring 2019
- Spring 2019: Lecture Videos

Lecture Video Playlist on YouTube

Lecture Playlist

Exponential Growth of Neural Networks

Memory and compute requirements

1800x more compute in just 2 years

Tomorrow, multi-trillion parameter models

Watch on YouTube

Source: <https://youtu.be/8h13idwcb0Q?t=283>

Week	Date	Livestream	Lecture	Readings	Assignments
W1	24.02 Thu.	Yes	L1a: Course Logistics PDF (PPT)	Suggested	
			L1b: Introduction and Basics PDF (PPT)	Suggested	
			L1c: Architectural Design Fundamentals PDF (PPT)	Suggested	
W2	03.03 Thu.	Yes	L2: Memory-Centric Computing PDF (PPT)	Suggested	
W3	10.03 Thu.	Yes	L3: Memory-Centric Computing II PDF (PPT)	Suggested	
W4	17.03 Thu.	Yes	L4: Memory-Centric Computing III PDF (PPT)	Suggested	
W5	24.03 Thu.	Yes	L5: Accelerating Genome Analysis PDF (PPT)	Suggested	
W6	31.03 Thu.	Yes	L6a: Rethinking Virtual Memory I PDF (PPT)	Suggested	
			L6b: Rethinking Virtual Memory II PDF (PPT)	Suggested	
W7	07.04 Thu.	Yes	S1.1: A Logic-in-Memory Computer IEEE Trans. Comput., 1970 PDF (PPT)		
			S1.2: SCA: Scalable Computing Architecture PDF (PPT)		

PIM Course (Fall 2021)

Fall 2021 Edition:

- https://safari.ethz.ch/projects_and_seminars/fall2021/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

PIM Review and Open Problems
Processing in Memory Course: Meeting 1: Ex...

Watch later Share 1/10

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

^aETH Zürich
^bCarnegie Mellon University
^cUniversity of Illinois at Urbana-Champaign
^dKing Mongkut's University of Technology North Bangkok

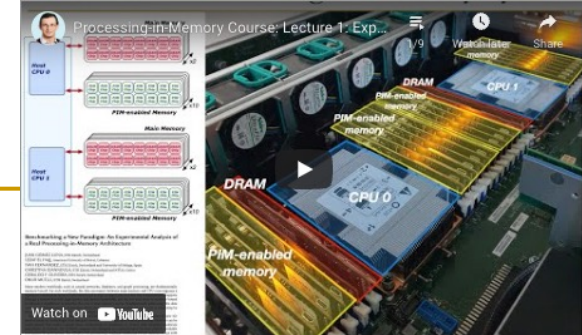
Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun, "A Modern Primer on Processing in Memory" Invited Book Chapter in *Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann*, Springer, to be published in 2021.

Watch on YouTube <https://arxiv.org/pdf/1903.03988.pdf> 108

Fall 2021 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	05.10 Tue.	YouTube Live	M1: P&S PIM Course Presentation PDF (PDF) PPT (PPT)	Required Materials Recommended Materials	HW 0 Out
W2	12.10 Tue.	YouTube Live	M2: Real-World PIM Architectures PDF (PDF) PPT (PPT)		
W3	19.10 Tue.	YouTube Live	M3: Real-World PIM Architectures II PDF (PDF) PPT (PPT)		
W4	26.10 Tue.	YouTube Live	M4: Real-World PIM Architectures III PDF (PDF) PPT (PPT)		
W5	02.11 Tue.	YouTube Live	M5: Real-World PIM Architectures IV PDF (PDF) PPT (PPT)		
W6	09.11 Tue.	YouTube Live	M6: End-to-End Framework for Processing-using-Memory PDF (PDF) PPT (PPT)		
W7	16.11 Tue.	YouTube Live	M7: How to Evaluate Data Movement Bottlenecks PDF (PDF) PPT (PPT)		
W8	23.11 Tue.	YouTube Live	M8: Programming PIM Architectures PDF (PDF) PPT (PPT)		
W9	30.11 Tue.	YouTube Live	M9: Benchmarking and Workload Suitability on PIM PDF (PDF) PPT (PPT)		
W10	07.12 Tue.	YouTube Live	M10: Bit-Serial SIMD Processing using DRAM PDF (PDF) PPT (PPT)		

PIM Course (Current)



Recorded Lecture Playlist



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.		M10: Real-world PIM: Alibaba HB-PIM (PDF) (PPT)		

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-841fUYUYUK9EsXKhQKRPyX

Project course

- Taken by Bachelor's/Master's students
- Processing-in-Memory lectures
- Hands-on research exploration
- Many research readings

Genomics (Spring 2022)

Fall 2021 Edition:

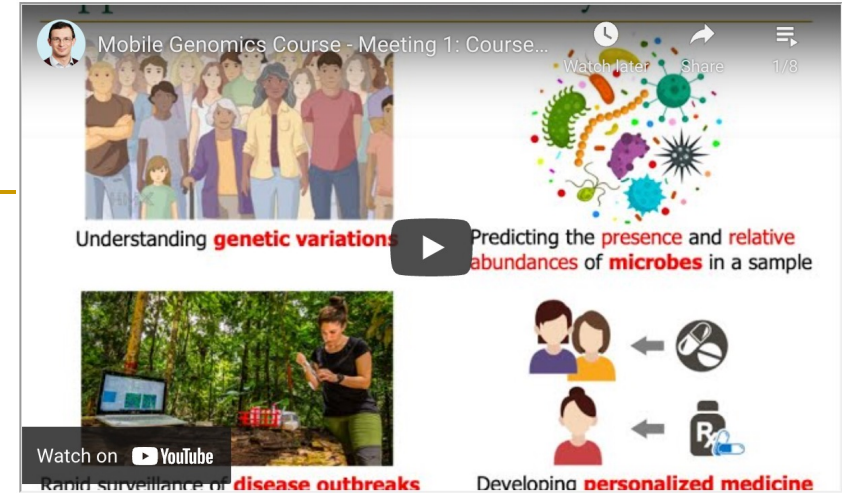
- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics

Youtube Livestream:

- https://www.youtube.com/playlist?list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18

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 (PDF) PPT (PPT) YouTube Video	Required Materials Recommended Materials	
W2	20.10 Wed.	YouTube Live	M2: Introduction to Sequencing PDF (PDF) PPT (PPT)		
W3	27.10 Wed.	YouTube Live	M3: Read Mapping PDF (PDF) PPT (PPT)		
W4	3.11 Wed.	YouTube Live	M4: GateKeeper PDF (PDF) PPT (PPT)		
W5	10.11 Wed.	YouTube Live	M5: MAGNET & Shouji PDF (PDF) PPT (PPT)		
W6	17.11 Wed.		M6.1: SneakySnake PDF (PDF) PPT (PPT) YouTube Video		
			M6.2: GRIM-Filter PDF (PDF) PPT (PPT) YouTube Video		
W7	24.11 Wed.		M7: GenASM PDF (PDF) PPT (PPT) YouTube Video		
W8	01.12 Wed.	YouTube Live	M8: Genome Assembly PDF (PDF) PPT (PPT)		
W9	13.12 Mon.	YouTube Live	M9: GRIM-Filter PDF (PDF) PPT (PPT)		
W10	15.12 Wed.	YouTube Live	M10: Genomic Data Sharing Under Differential Privacy PDF (PDF) PPT (PPT)		

Genomics (Fall 2021)

Fall 2021 Edition:

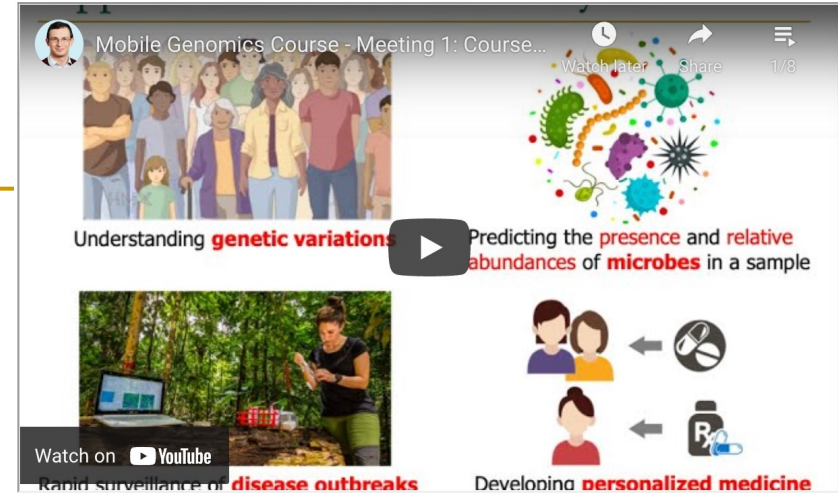
- https://safari.ethz.ch/projects_and_seminars/fall2021/doku.php?id=bioinformatics

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 (PDF) PPT (PPT) YouTube Video	Required Materials Recommended Materials	
W2	20.10 Wed.	YouTube Live	M2: Introduction to Sequencing PDF (PDF) PPT (PPT)		
W3	27.10 Wed.	YouTube Live	M3: Read Mapping PDF (PDF) PPT (PPT)		
W4	3.11 Wed.	YouTube Live	M4: GateKeeper PDF (PDF) PPT (PPT)		
W5	10.11 Wed.	YouTube Live	M5: MAGNET & Shouji PDF (PDF) PPT (PPT)		
W6	17.11 Wed.		M6.1: SneakySnake PDF (PDF) PPT (PPT) YouTube Video		
			M6.2: GRIM-Filter PDF (PDF) PPT (PPT) YouTube Video		
W7	24.11 Wed.		M7: GenASM PDF (PDF) PPT (PPT) YouTube Video		
W8	01.12 Wed.	YouTube Live	M8: Genome Assembly PDF (PDF) PPT (PPT)		
W9	13.12 Mon.	YouTube Live	M9: GRIM-Filter PDF (PDF) PPT (PPT)		
W10	15.12 Wed.	YouTube Live	M10: Genomic Data Sharing Under Differential Privacy PDF (PDF) PPT (PPT)		

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

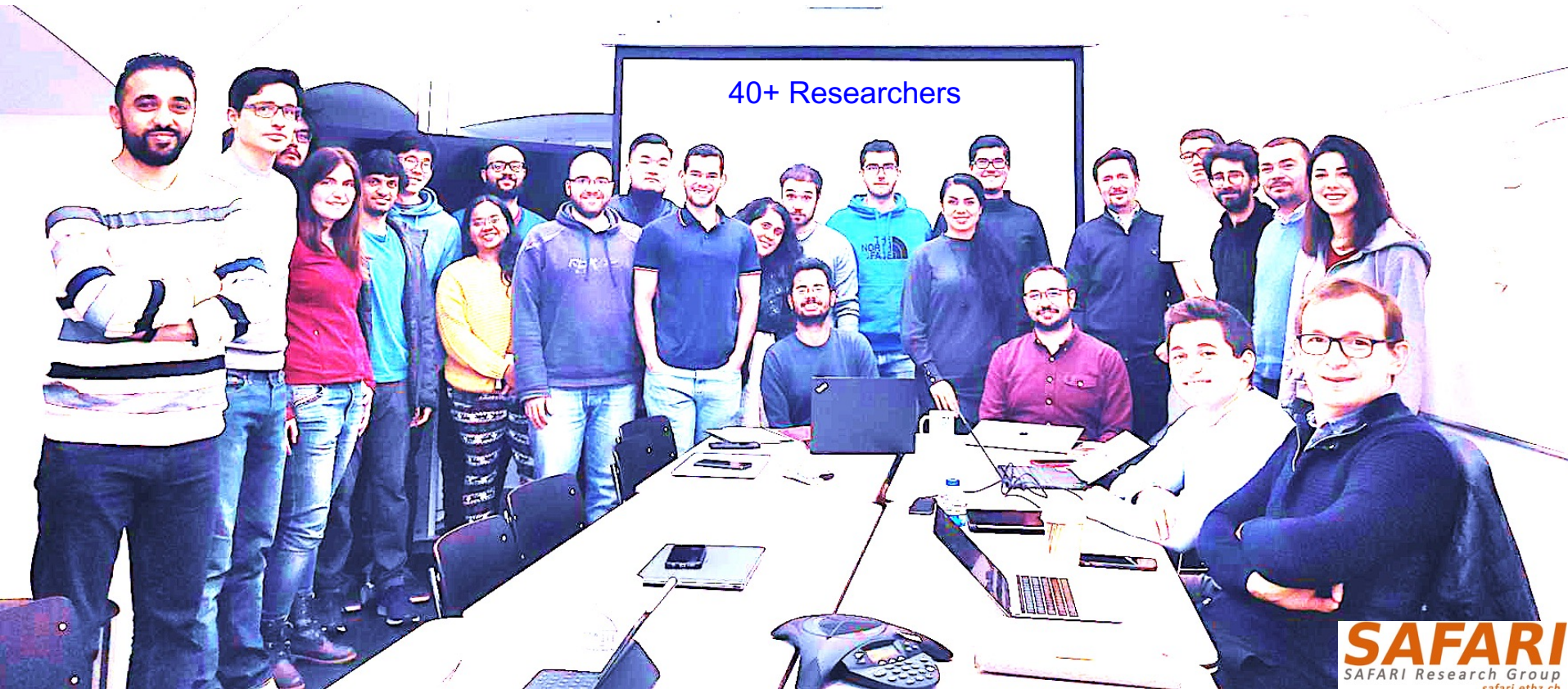
Think BIG, Aim HIGH!

<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/>



SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

SAFARI

<https://safari.ethz.ch>

SAFARI Newsletter April 2020 Edition

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



[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



SAFARI Research Group at ETH Zurich and Carnegie Mellon University

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

📍 ETH Zurich and Carnegie Mellon U... 🔗 <https://safari.ethz.ch/> ✉ omutlu@gmail.com

🏠 Overview 📁 Repositories 71 📁 Projects 📁 Packages 👤 Teams 1 👤 People 44 ⚙ Settings

Pinned

Customize pins

📁 ramulator Public

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

🔴 C++ ☆ 311 🍷 161

📁 prim-benchmarks Public

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

⬛ C ☆ 53 🍷 21

📁 DAMOV Public

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

🔴 C++ ☆ 26 🍷 4

📁 SneakySnake Public

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

🔵 VHDL ☆ 41 🍷 8

📁 MQSim Public

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

🔴 C++ ☆ 146 🍷 93

📁 rowhammer Public

Source code for testing the Row Hammer error mechanism in DRAM devices. Described in the ISCA 2014 paper by Kim et al. at http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer_isca14.pdf.

⬛ C ☆ 189 🍷 41

<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>

18 June 2022

AACBB Workshop Keynote @ ISCA 2022

SAFARI

ETH zürich

Carnegie Mellon

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=AIE1rD9G_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

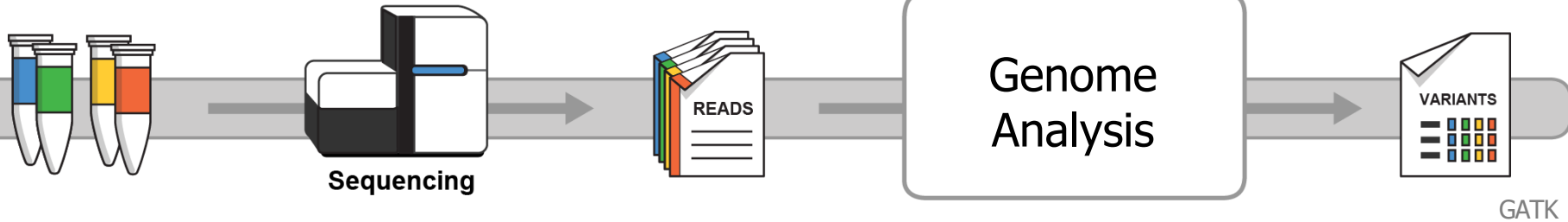


NO machine can read the *entire* content of a genome



```
>CCTCCTCAGTGCCACCCAGCCCACTGGCAGCTCCCAAACAGGCTCTTATTAACACCCCTGTTCCCTGCCCCTTGGAGTGAGGTGTCAAG
GACCTAACTAAAAAAAAAAAAAAAAAGAAAAAGAAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTCTT
CATGTCAAGGACCTAATGTGCTAAACAGCACTTTTTTGACCATTATTTTGGATCTGAAAGAAATCAAGAATAAATGAAGGACTTGATACATTG
GAAGAGGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAAAAAGAAAAAGAAAAAGAAAAAGAATTAAATTTAAGTAATTCTTTGAAAAAA
ACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCTGTGTTGCAGGTCTTCTTGCATTTCCCTGTCAAAAGAAAAAGAATTTAAATTT
AAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTGAGGCCAAGAGTTGCAAAAAAAAAAAAAAGAAAAA
GAAAAGAAAAAGAATTTAAATTTAAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTAGCCAGAATGG
TTGTGGGATGGGAGCCTCTGTGGACCGACCAGGTAGCTCTCTTTCCACACTGTAGTCTCAAAGCTTCTTCATGTGGTTTTCTCTGAGTGAAA
AAAAAAAAAAGAAAAAGAAAAAGAAAAAGAATTTAAATTTAAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTTTCATGTCAAGGACC
TAATGTAGCTATACTGAACGTTATCTAGGGGAAAGATTGAAGGGGAGCTCTAAGGTCAACACACCACCACTTCCCAGAAAGCTTCTTCA.....
```


Genome Sequencer is a Chopper



CCCCCTATATATACGTACTAGTACGT
ACGACTTTAGTACGTACGT
TATATATACGTACTAGTACGT
ACGTACGCCCCTACGTA
TATATATACGTACTAGTACGT
ACGACTTTAGTACGTACGT
TATATATACGTACTAAAGTACGT
TATATATACGTACTAGTACGT
ACGTTTTTAAACGTA
TATATATACGTACTAGTACGT
ACGACGGGGAGTACGTACGT



1×10^{12} bases*



44 hours*



<1000 \$

* NovaSeq 6000

Oxford Nanopore Sequencers



MinION Mk1B



MinION Mk1C



GridION Mk1



PromethION 24/48

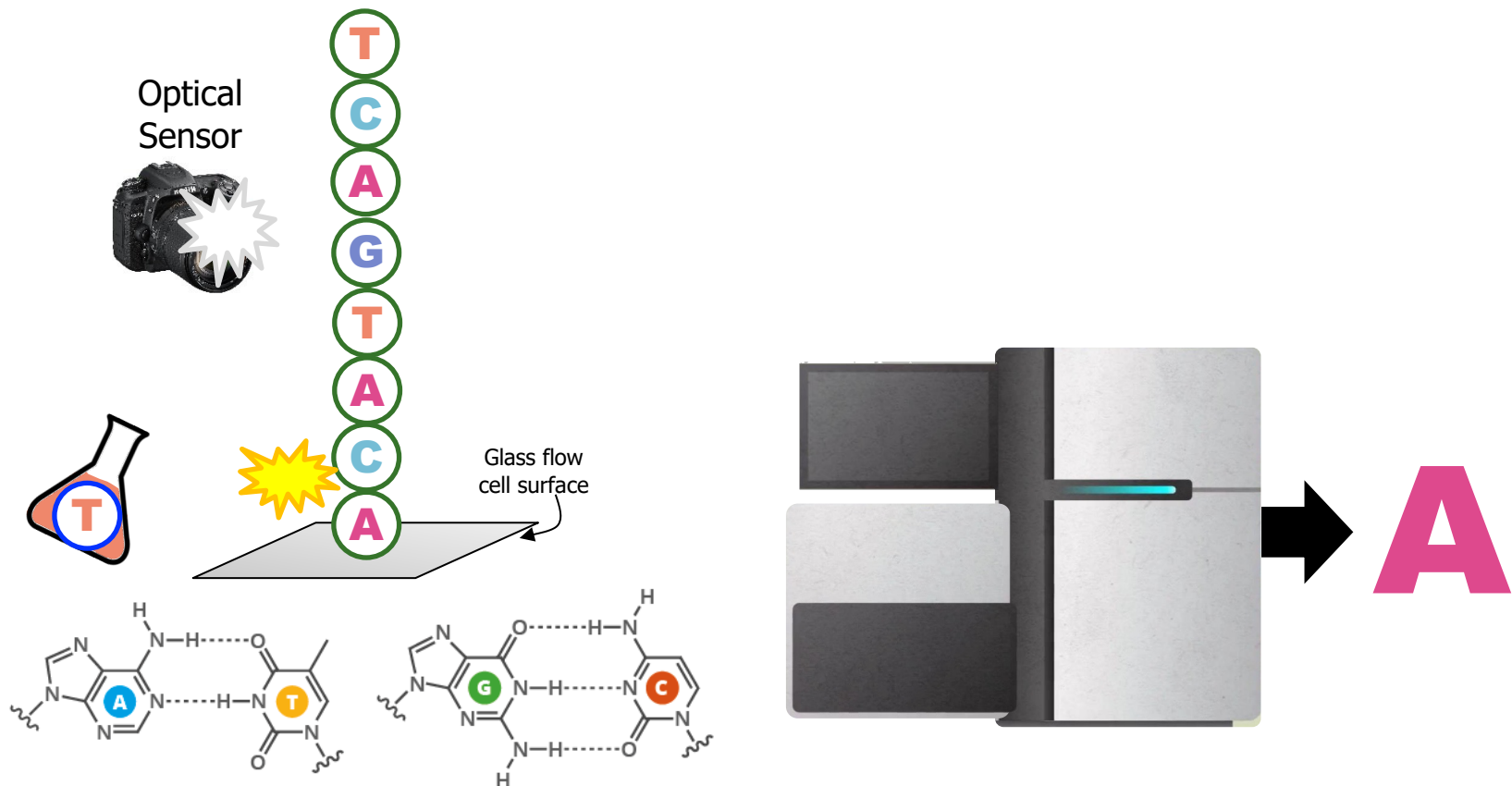
	MinION Mk1B	MinION Mk1C	GridION Mk1	PromethION 24	PromethION 48
Read length	> 2Mb	> 2Mb	> 2Mb	> 2Mb	> 2Mb
Yield per flow cell	50 Gb	50 Gb	50 Gb	220 Gb	220 Gb
Number of flow cells per device	1	1	5	24	48
Yield per device	<50 Gb	<50 Gb	<250 Gb	<5.2 Tb	<10.5 Tb
Starting price	\$1,000	\$4,990	\$49,995	\$195,455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

Illumina Sequencers

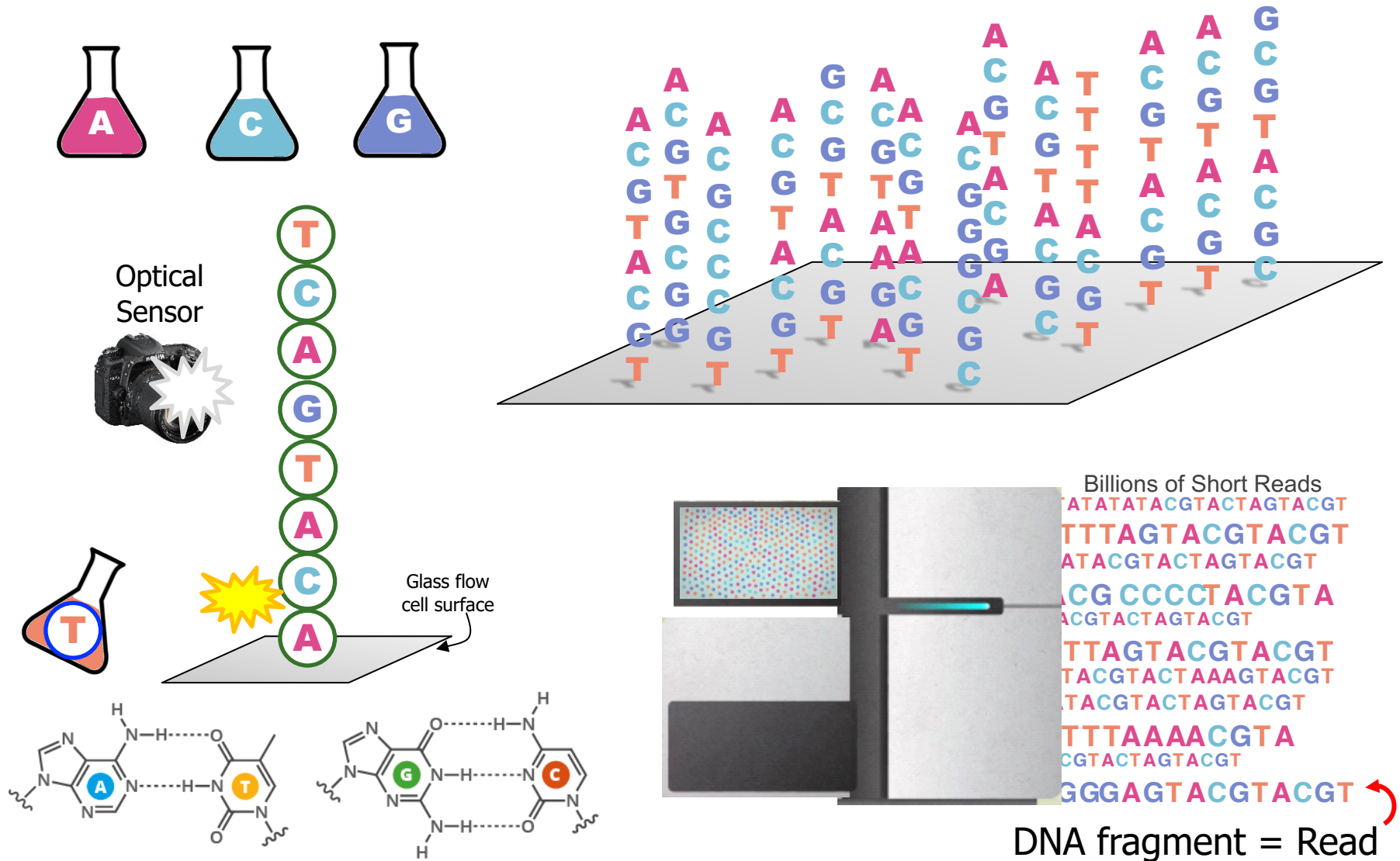


	iSeq 100	MiniSeq	MiSeq	NextSeq 550	NextSeq 2000	NovaSeq 6000
Run time	9.5–19 hrs	4–24 hrs	4–55 hrs	12–30 hrs	24–48 hrs	13–44 hrs
Max. reads per run	4 million	25 million	25 million	400 million	1 billion	20 billion
Max. read length	2 × 150 bp	2 × 150 bp	2 × 300 bp	2 × 150 bp	2 × 150 bp	2 x 250
Max. output	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?

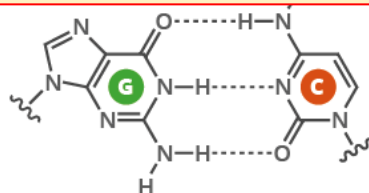
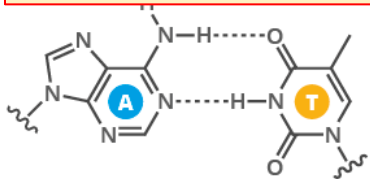


How Does Illumina Machine Work?

[illegible]

Check Illumina virtual tour:

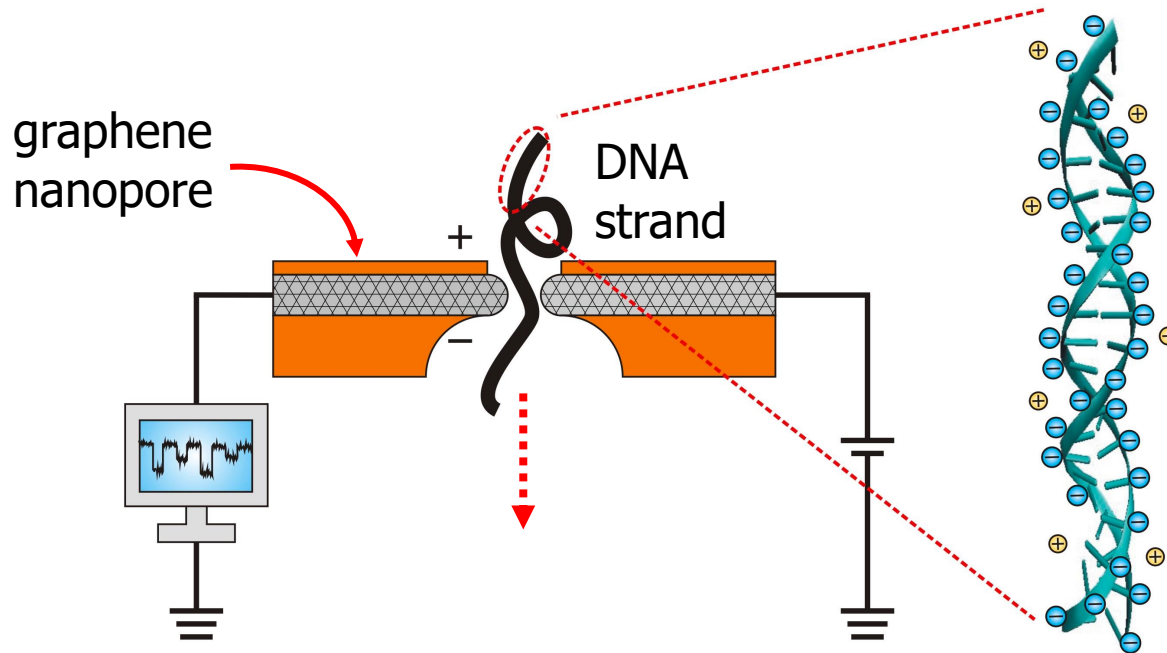
<https://emea.illumina.com/systems/sequencing-platforms/iseq/tour.html>



TTTAAACGTA
CGTACTAGTACGT
GGGAGTACGTACGT

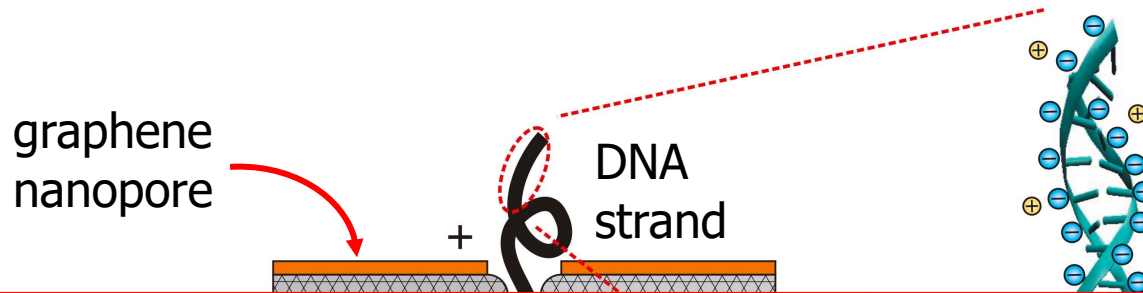
DNA fragment = Read

How Does Nanopore Machine Work?



- **Nanopore** is a nano-scale hole ($<20\text{nm}$).
- In nanopore sequencers, an **ionic current** passes through the nanopores
- When the DNA strand passes through the nanopore, the sequencer measures 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

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

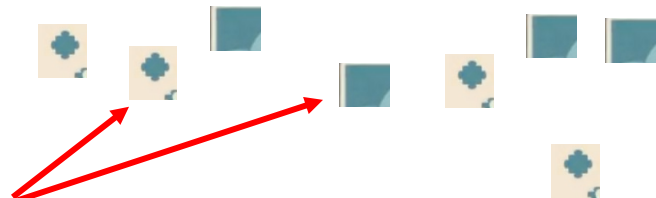
Solving the Puzzle



Reference
genome



Reads



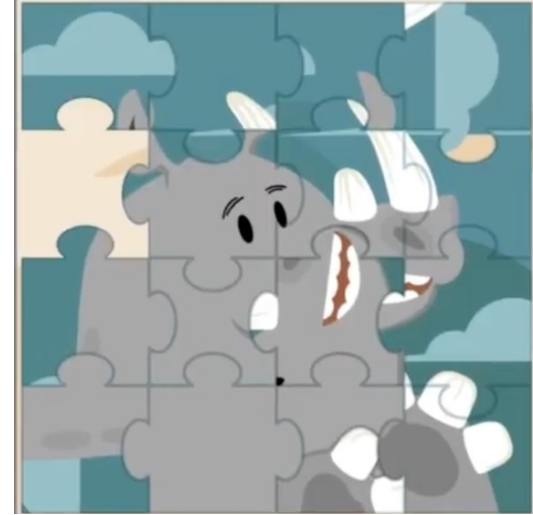
<https://www.pacb.com/smrt-science/smrt-sequencing/hifi-reads-for-highly-accurate-long-read-sequencing/>

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

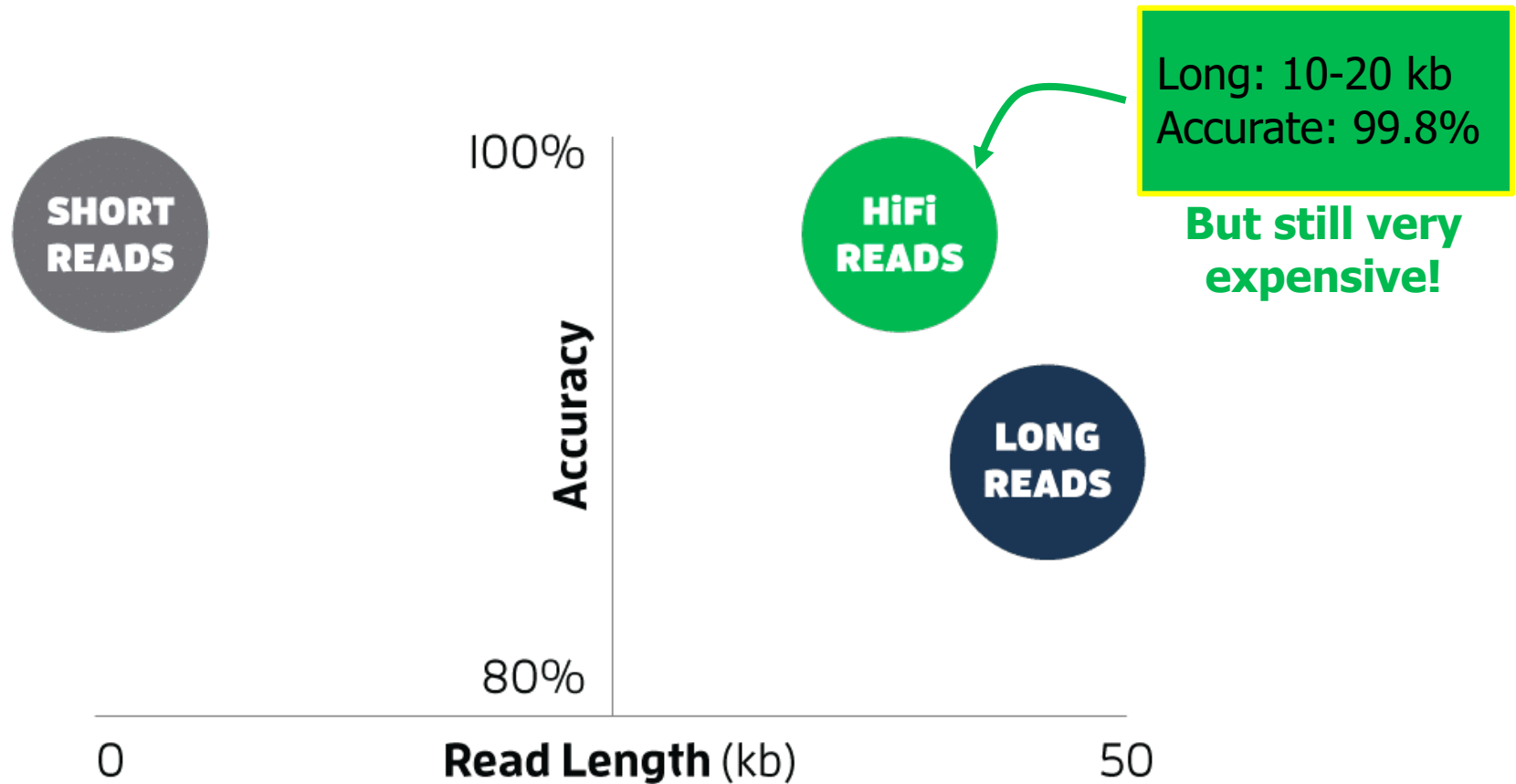
☐ low error rate ($\sim 0.1\%$)

☐ 500-2M bp

☐ high error rate ($\sim 15\%$)

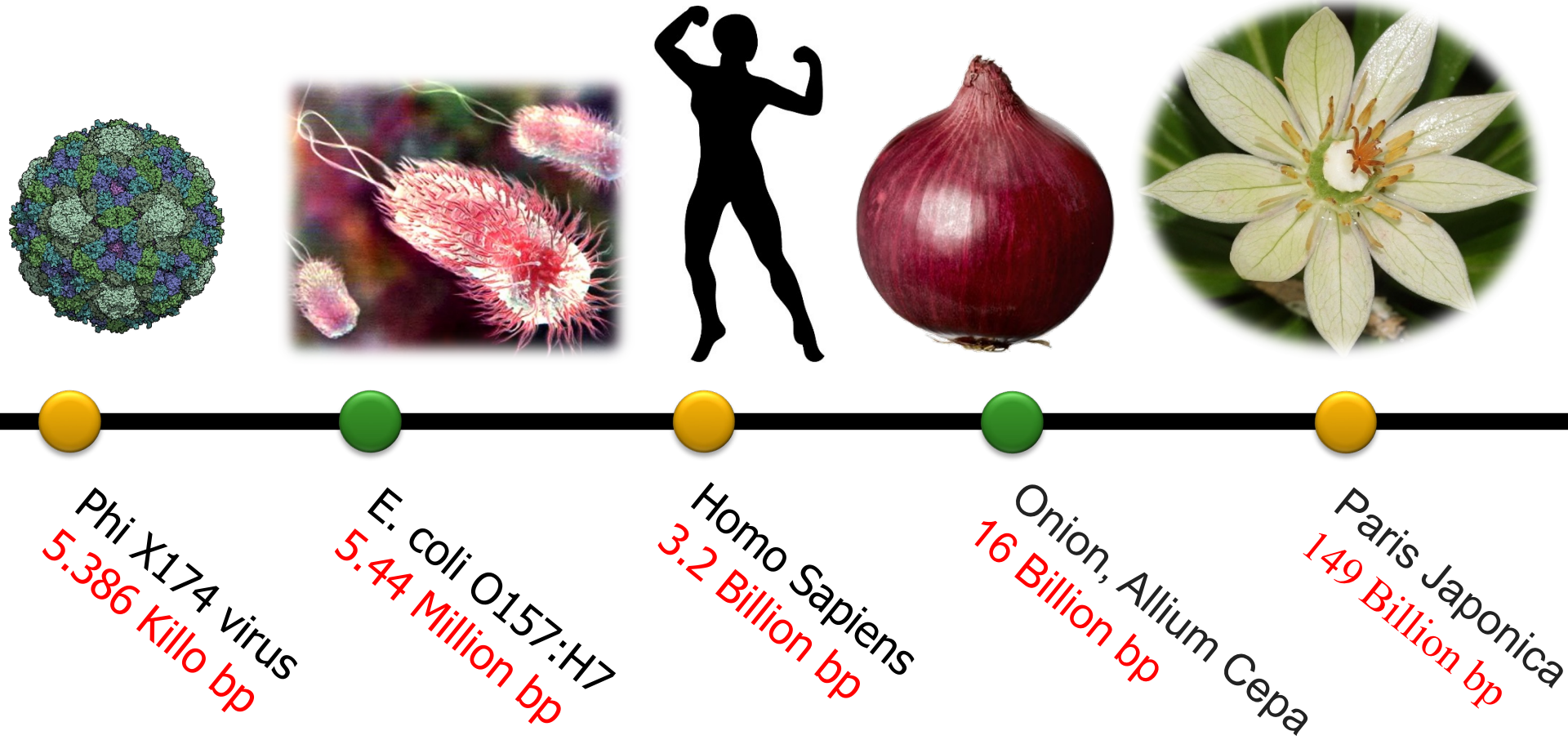
<https://www.pacb.com/smrt-science/smrt-sequencing/hifi-reads-for-highly-accurate-long-read-sequencing/>

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?



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.



A C **G T** 3.2×10^9 bases

 13 years

 $> 3 \times 10^9$ \$

Obtaining the Human Reference Genome

■ **GRCh38.p13**

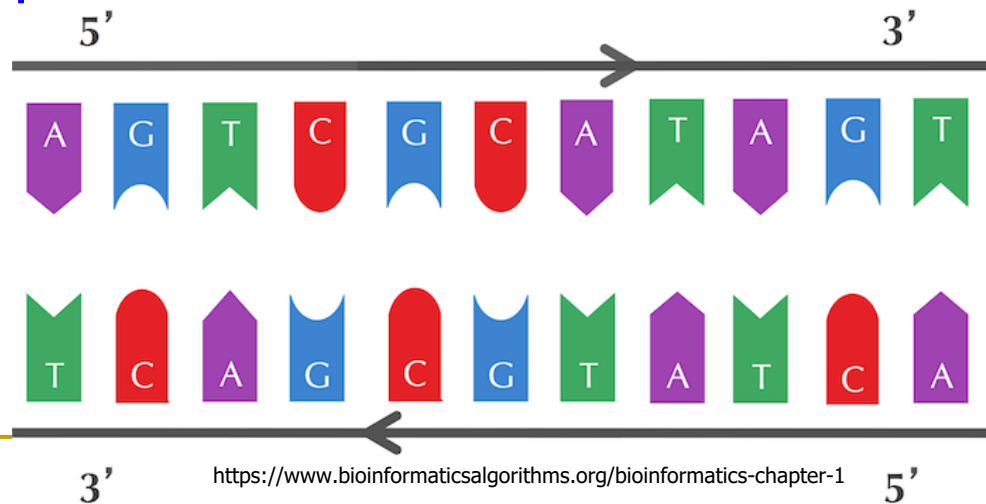
- Description: Genome Reference Consortium Human Build 38 patch release 13 (GRCh38.p13)
- Organism name: Homo sapiens (human)
- Date: 2019/02/28
- 3,099,706,404 bases
- Compressed .fna file (964.9 MB)
- https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39

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

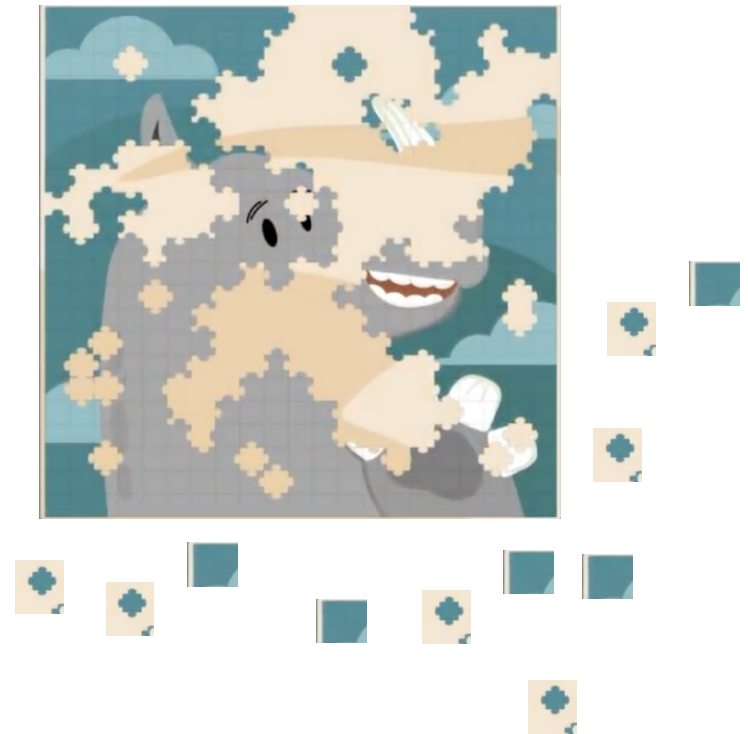
[illegible]

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**



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”

Time to Change the Reference Genome

Genome Biology

[Home](#) [About](#) [Articles](#) [Submission Guidelines](#)

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.

[illegible]

MAGNET Walkthrough

Build Neighborhood Map

Track the Diagonally Consecutive Matches

ACCEPT iff number of '1' $\leq$ Threshold

[illegible]

Find the longest segment of consecutive zeros

Exclude the errors from the search space

Divide the problem into two subproblems and repeat

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

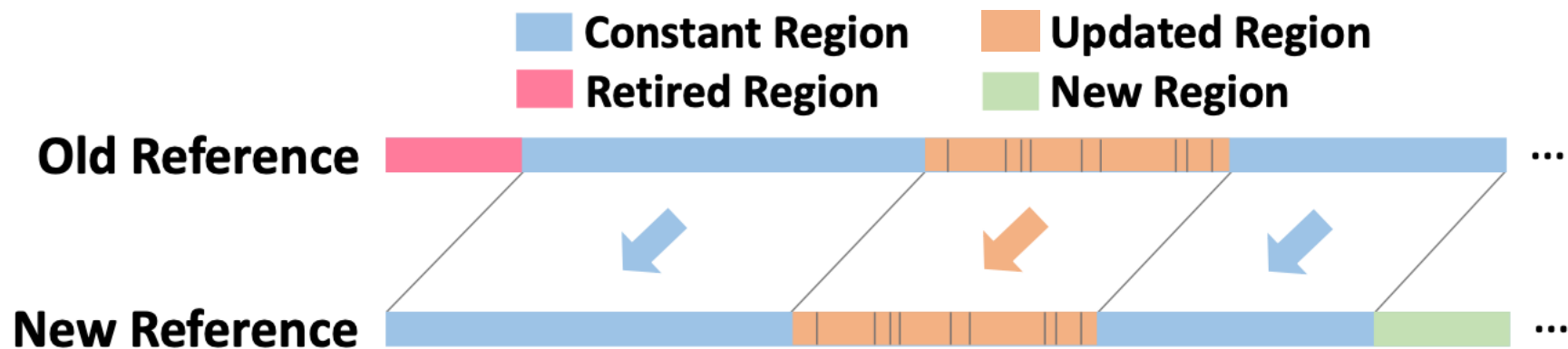


Fig. 2. Reference Genome Regions.

More Details on AirLift

arXiv.org > q-bio > arXiv:1912.08735

Search...

Help | Advanced Search

Quantitative Biology > Genomics

[Submitted on 18 Dec 2019]

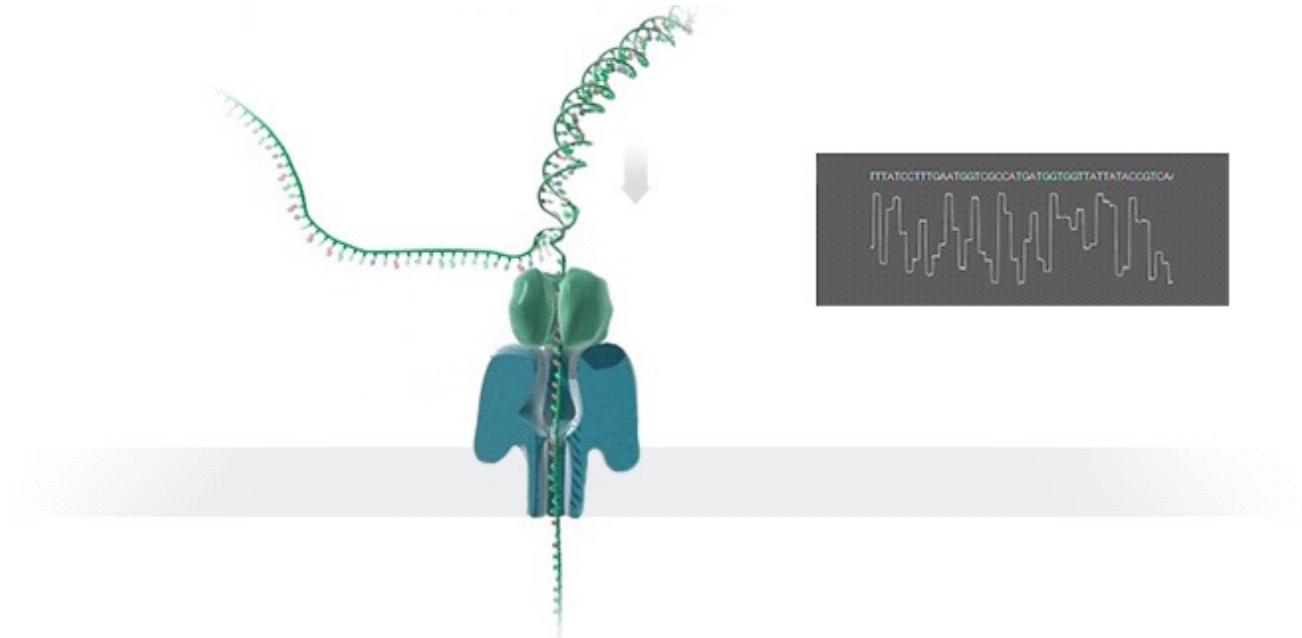
AirLift: A Fast and Comprehensive Technique for Translating Alignments between Reference Genomes

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

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

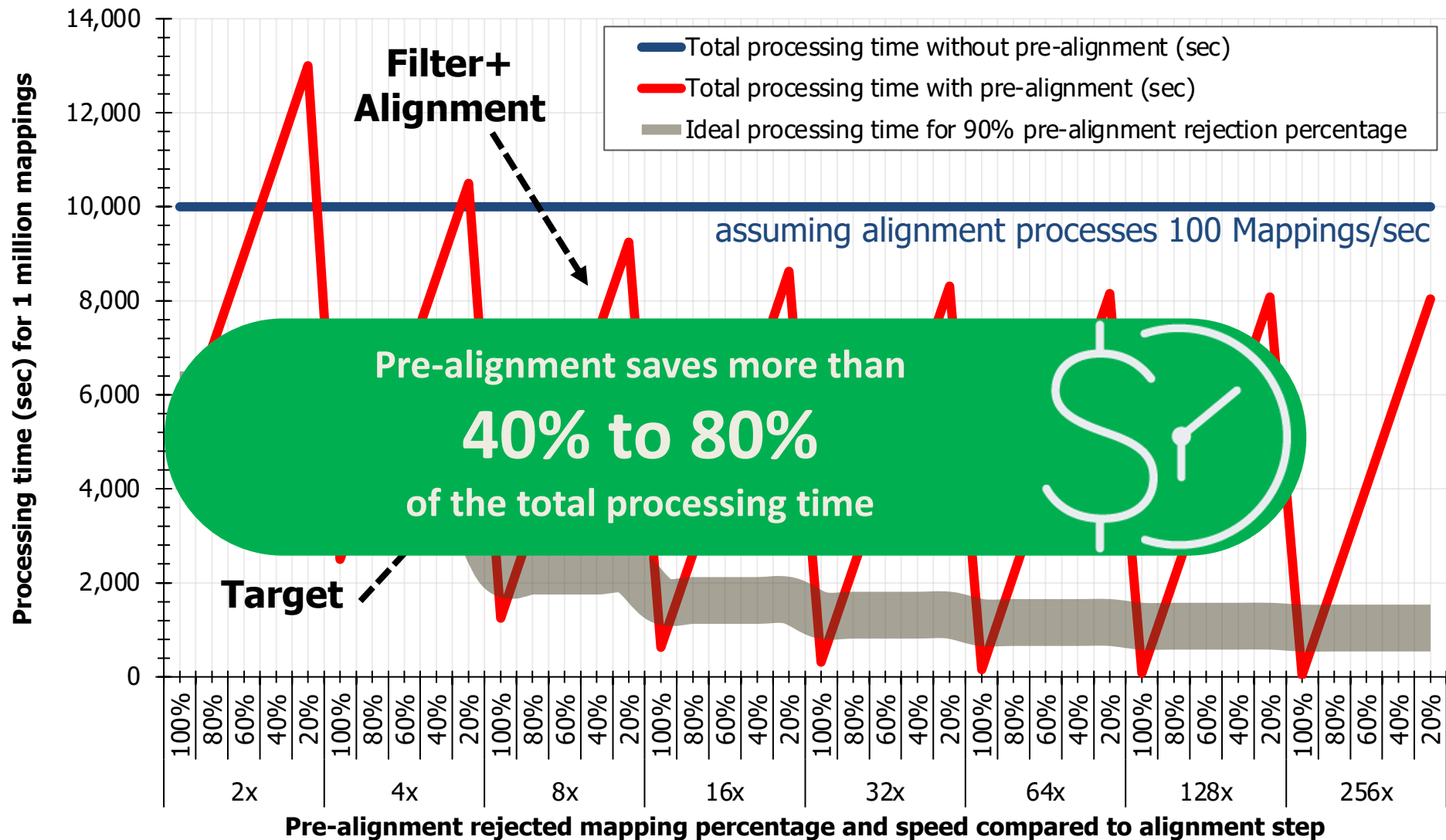
Kim+, "[AirLift: A Fast and Comprehensive Technique for Translating Alignments between Reference Genomes](#)", arXiv, 2020

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 **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

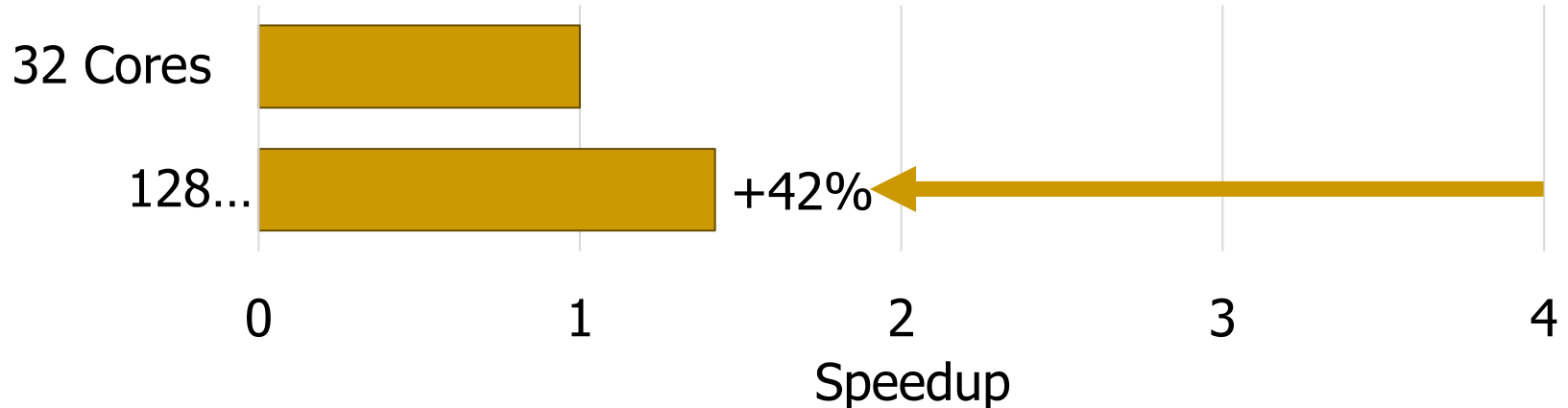


300 Million
Twitter Users



30 Billion
Instagram Photos

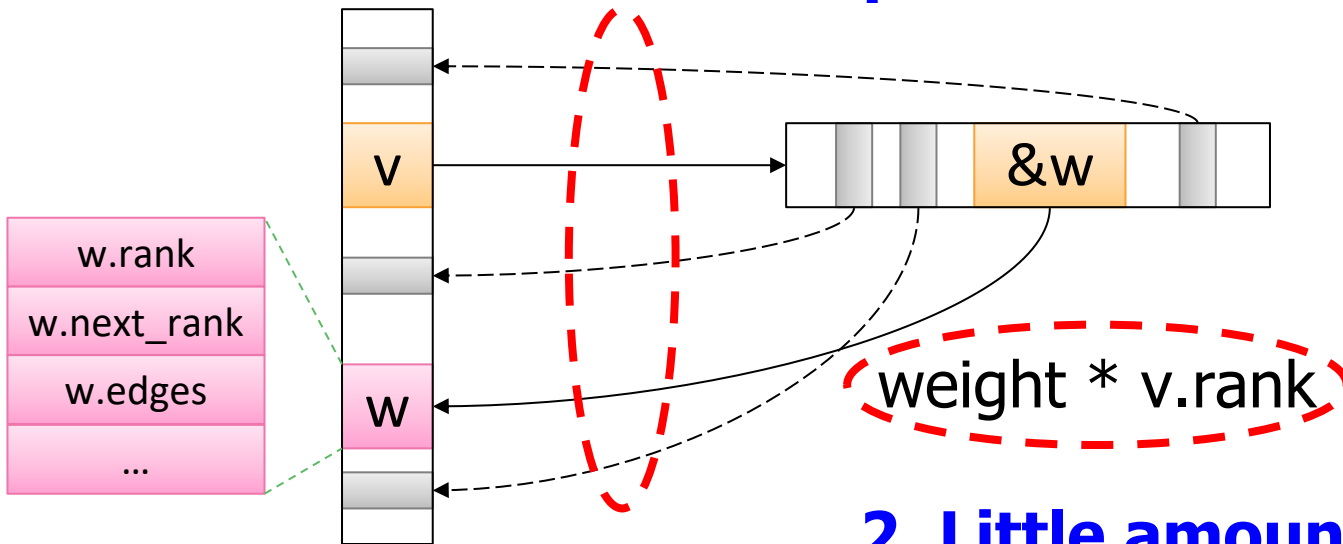
- Scalable large-scale graph processing is challenging



Key Bottlenecks in Graph Processing

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

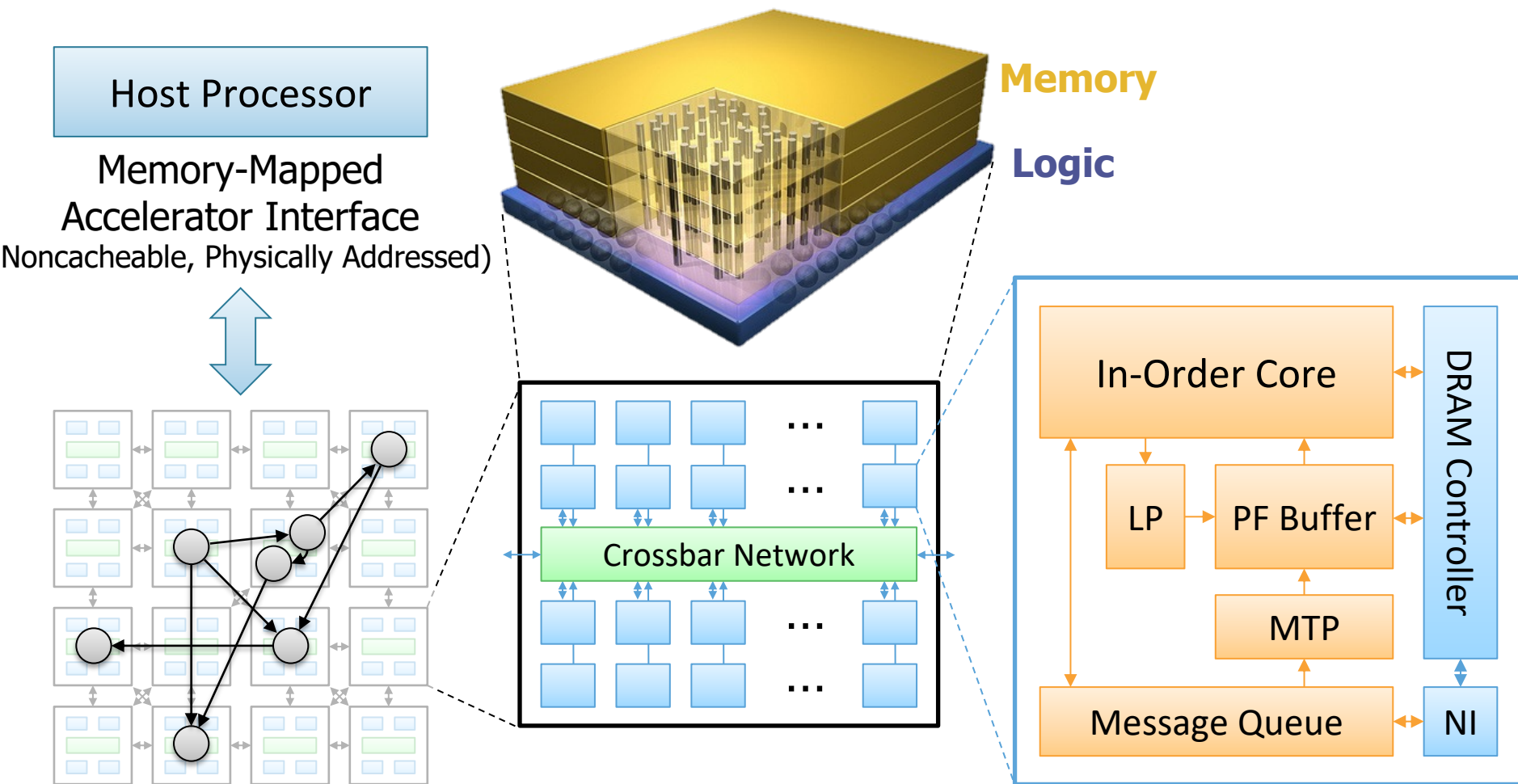
1. Frequent random memory accesses



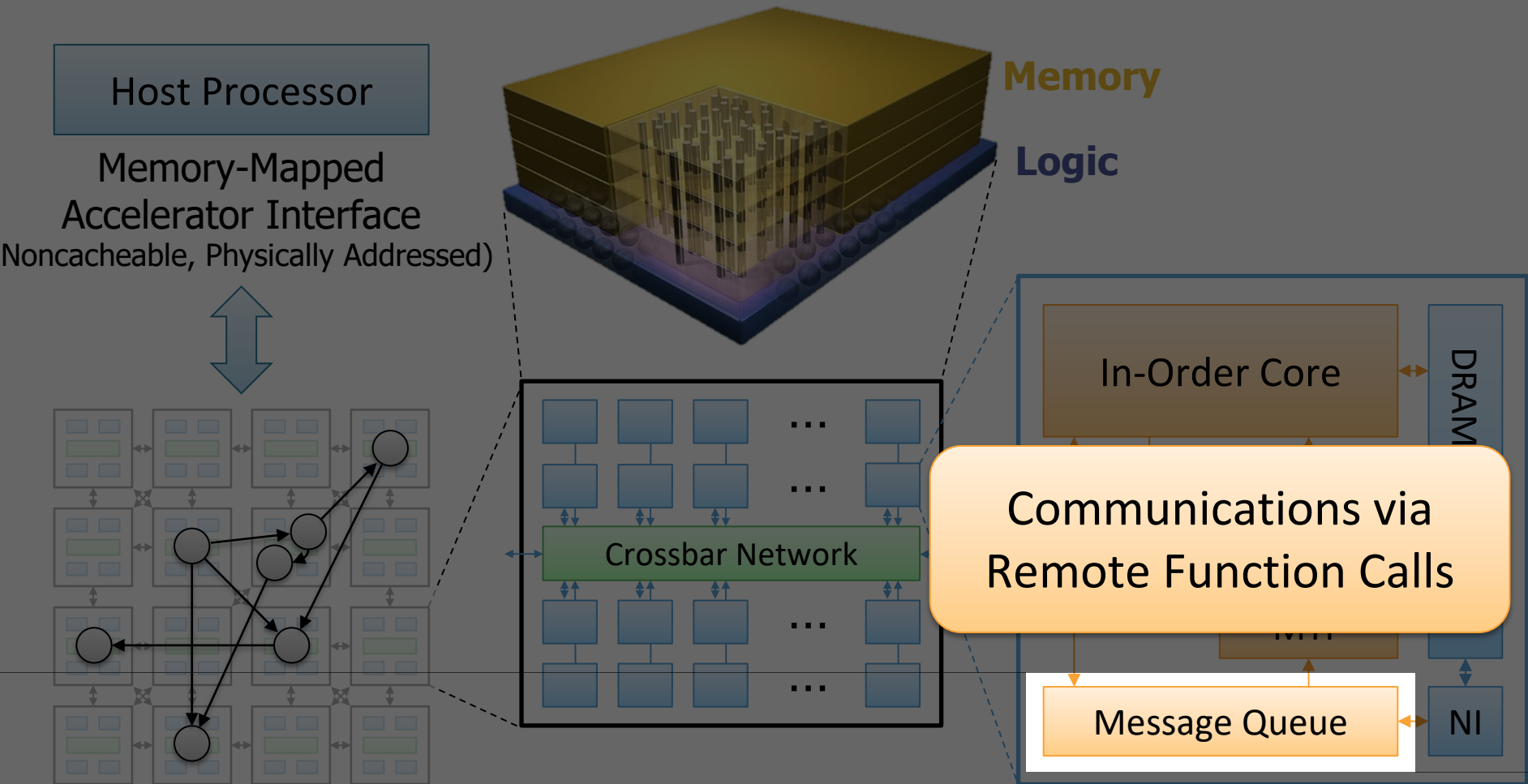
2. Little amount of computation

Tesseract System for Graph Processing

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

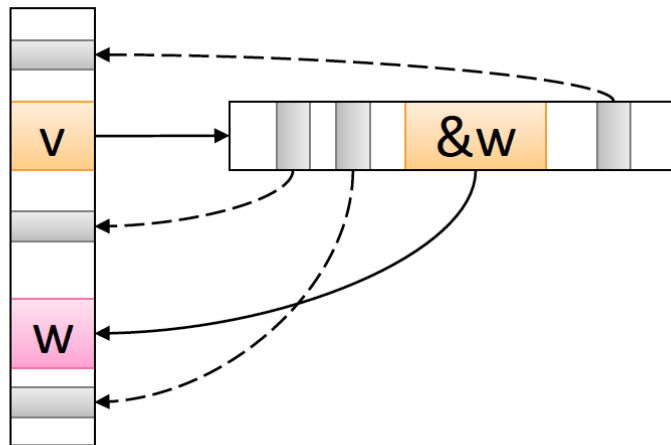


Tesseract System for Graph Processing



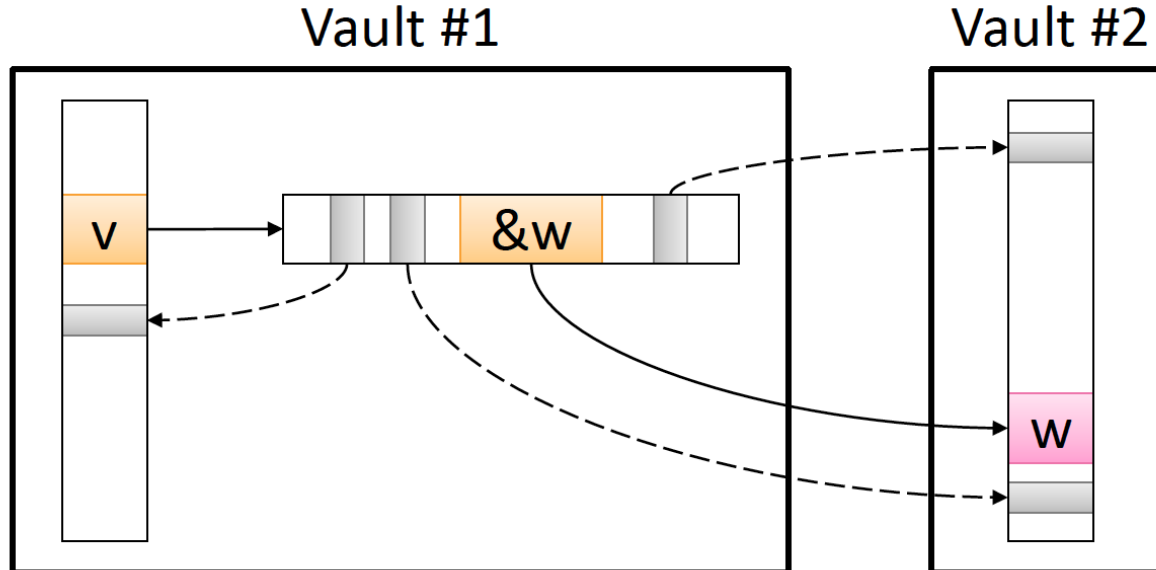
Communications In Tesseract (I)

```
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;  
  }  
}
```

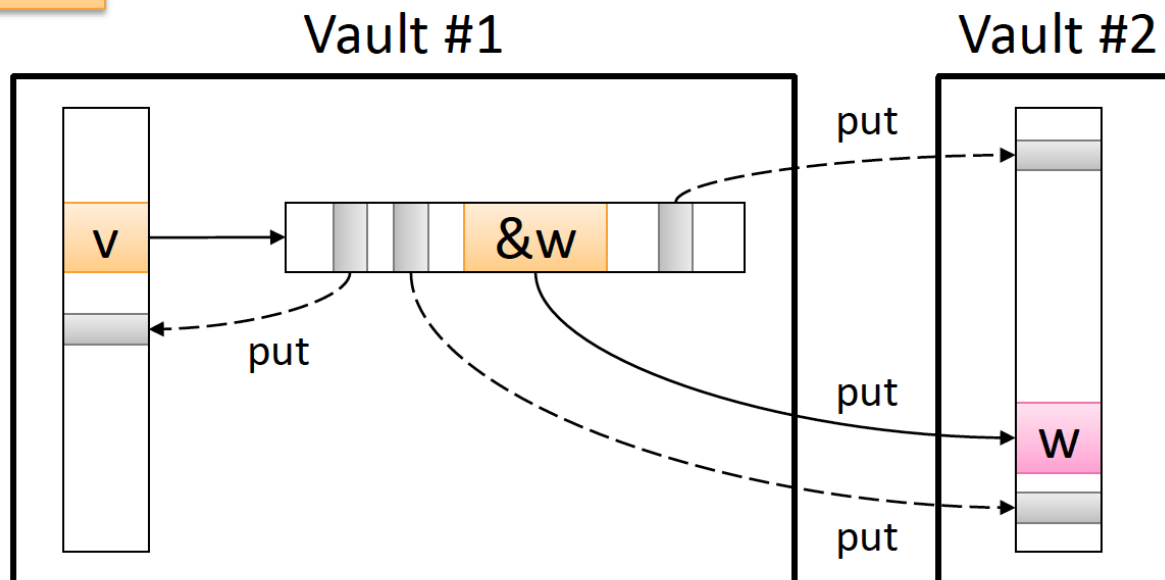


Communications In Tesseract (III)

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

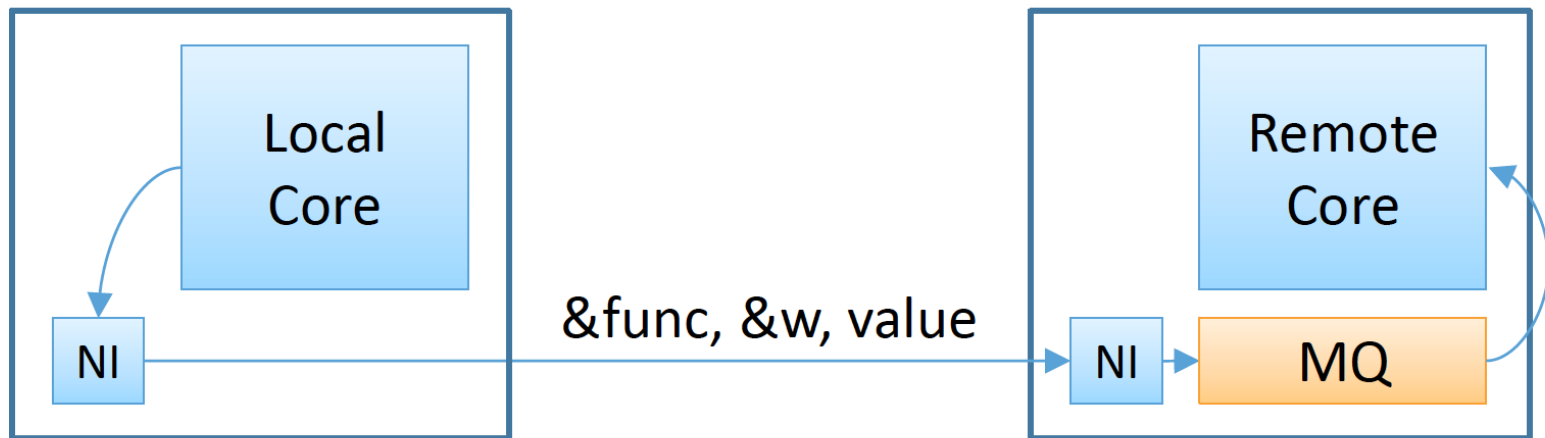
Non-blocking Remote Function Call

Can be **delayed** until the nearest barrier



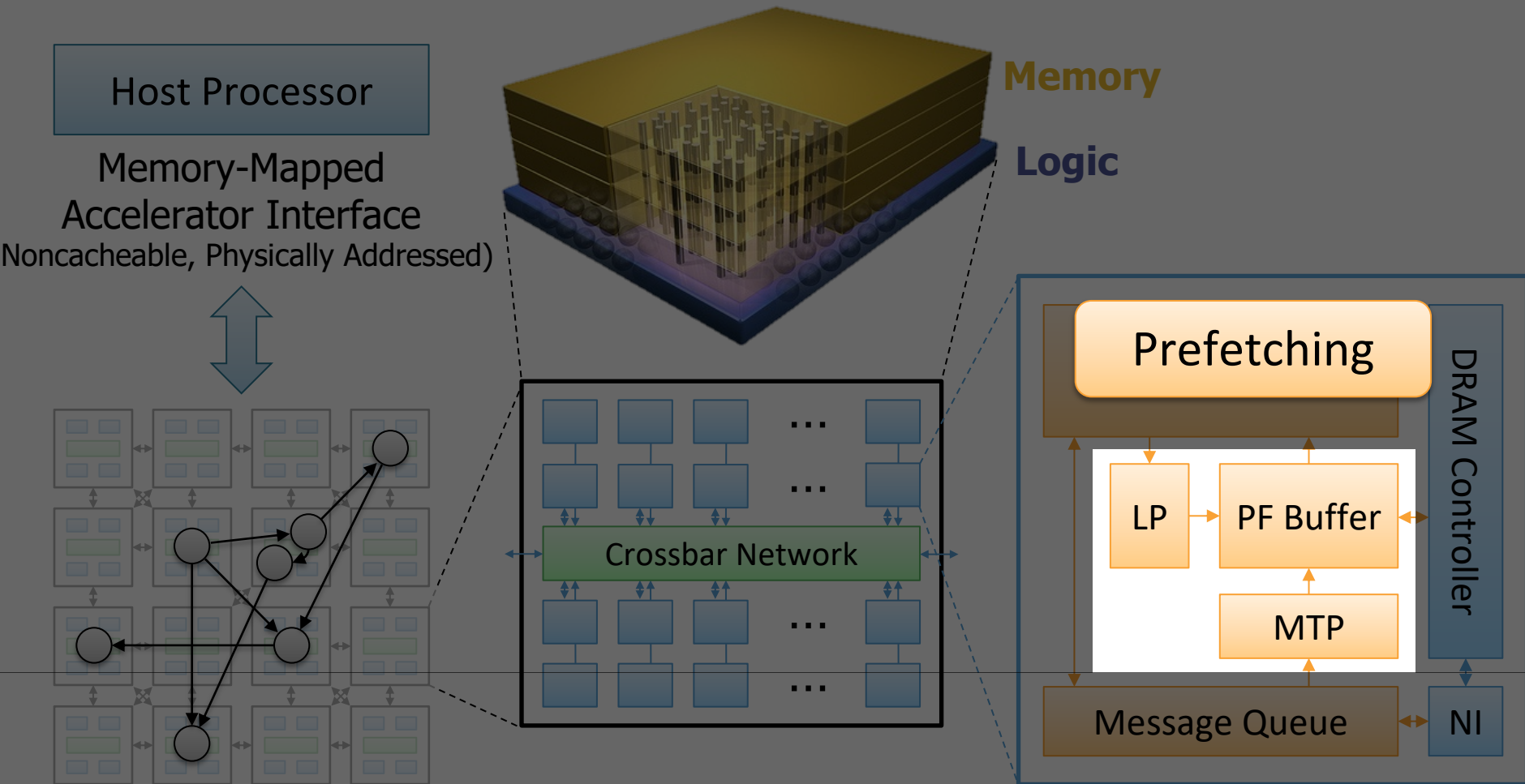
Remote Function Call (Non-Blocking)

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



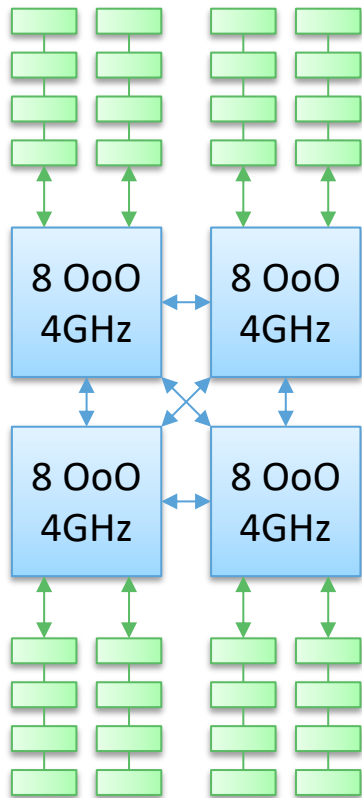
```
put(w.id, function() { w.next_rank += value; })
```

Tesseract System for Graph Processing



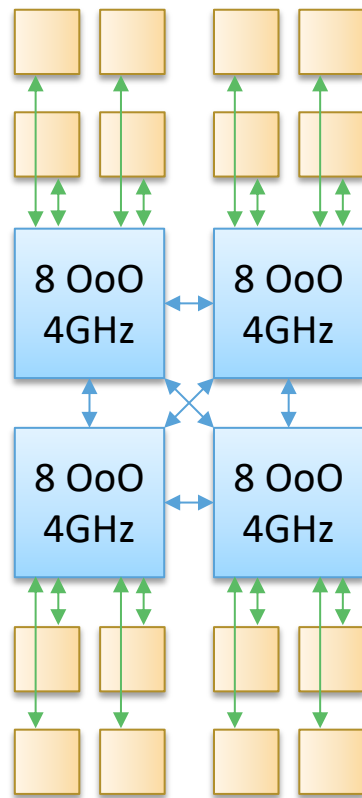
Evaluated Systems

DDR3-OoO



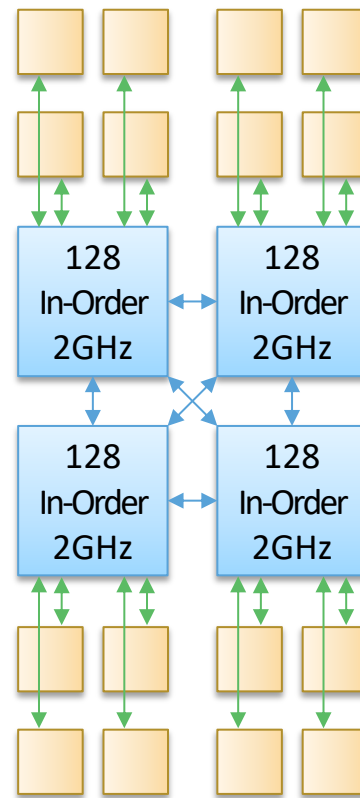
102.4GB/s

HMC-OoO



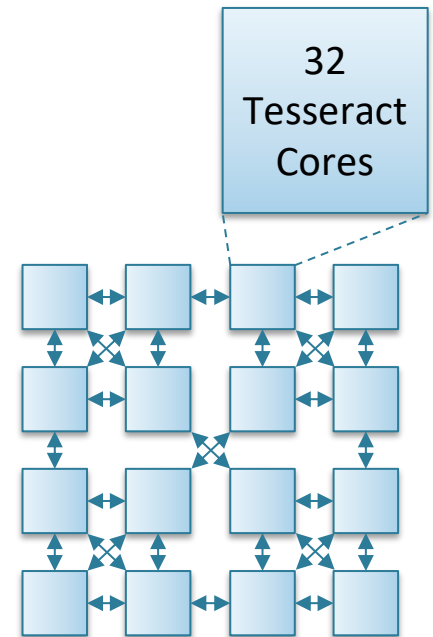
640GB/s

HMC-MC



640GB/s

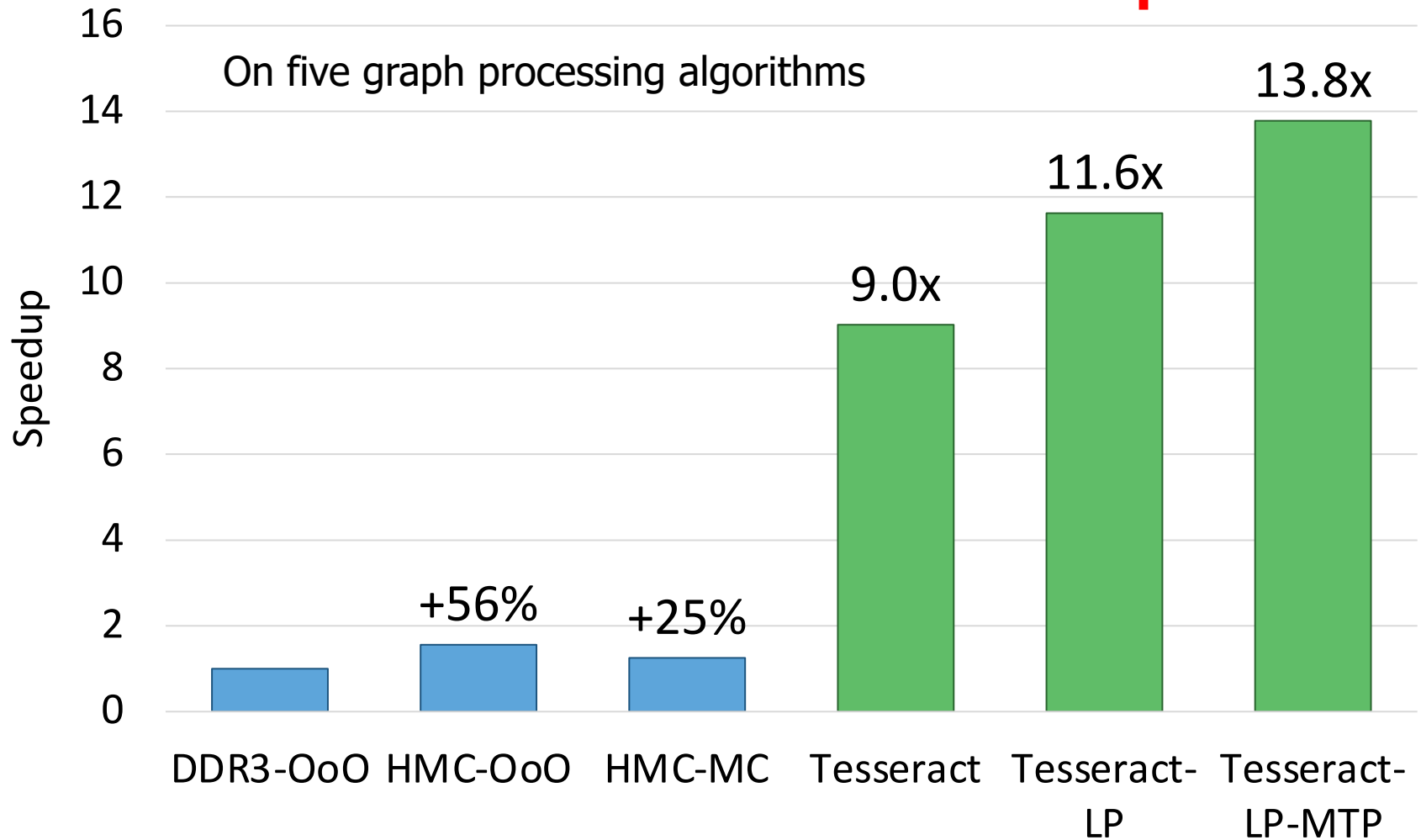
Tesseract



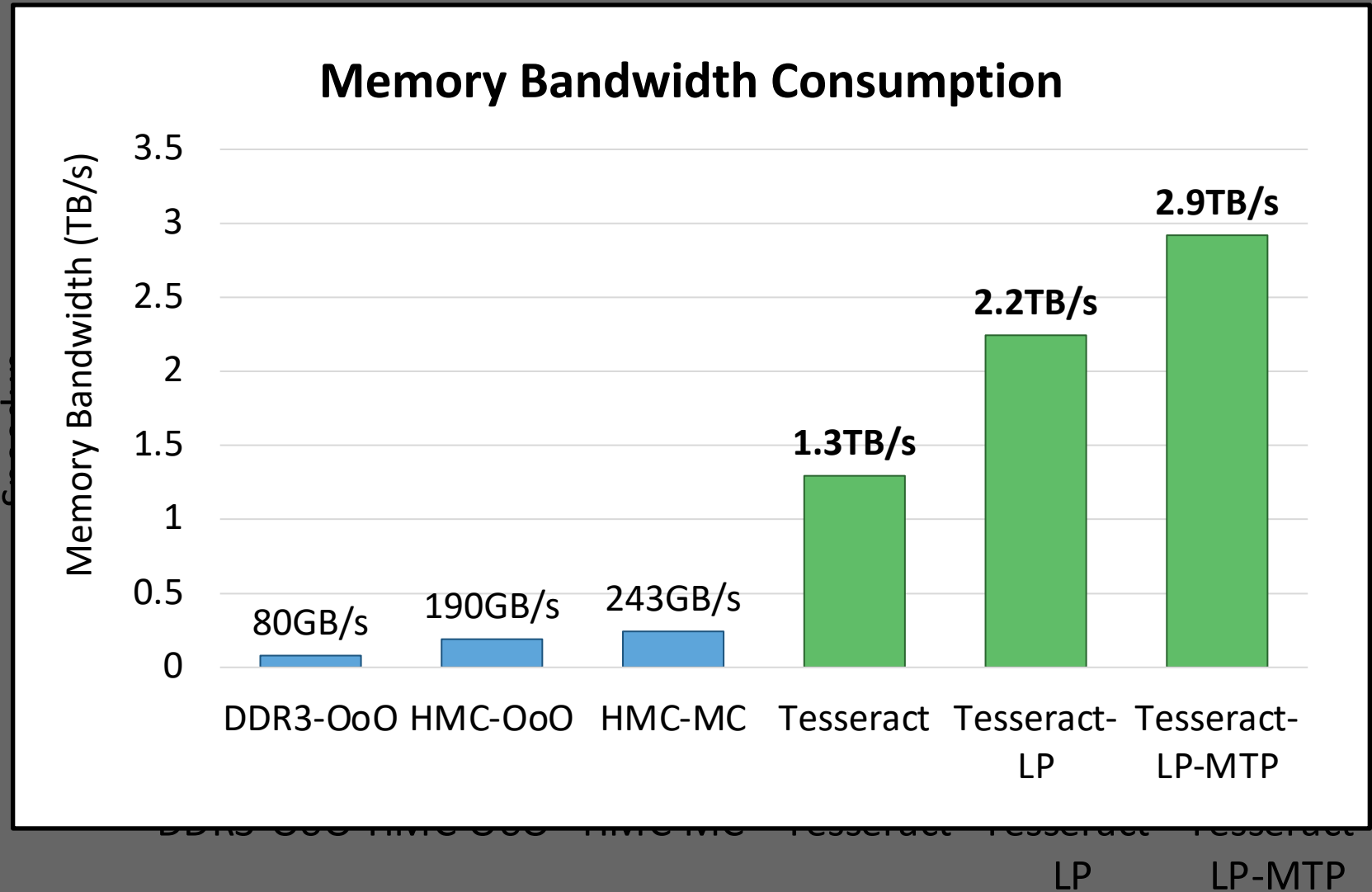
8TB/s

Tesseract Graph Processing Performance

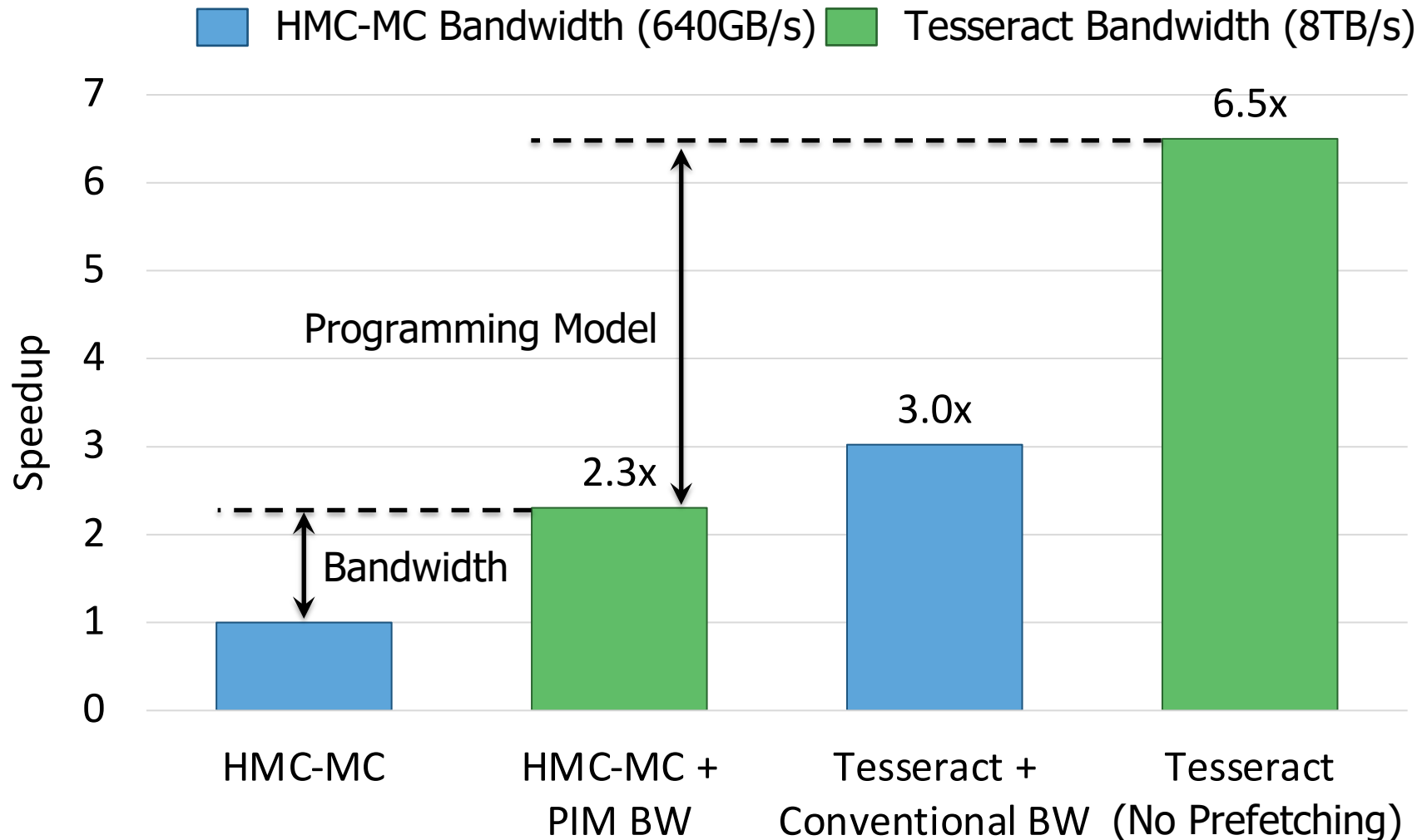
>13X Performance Improvement



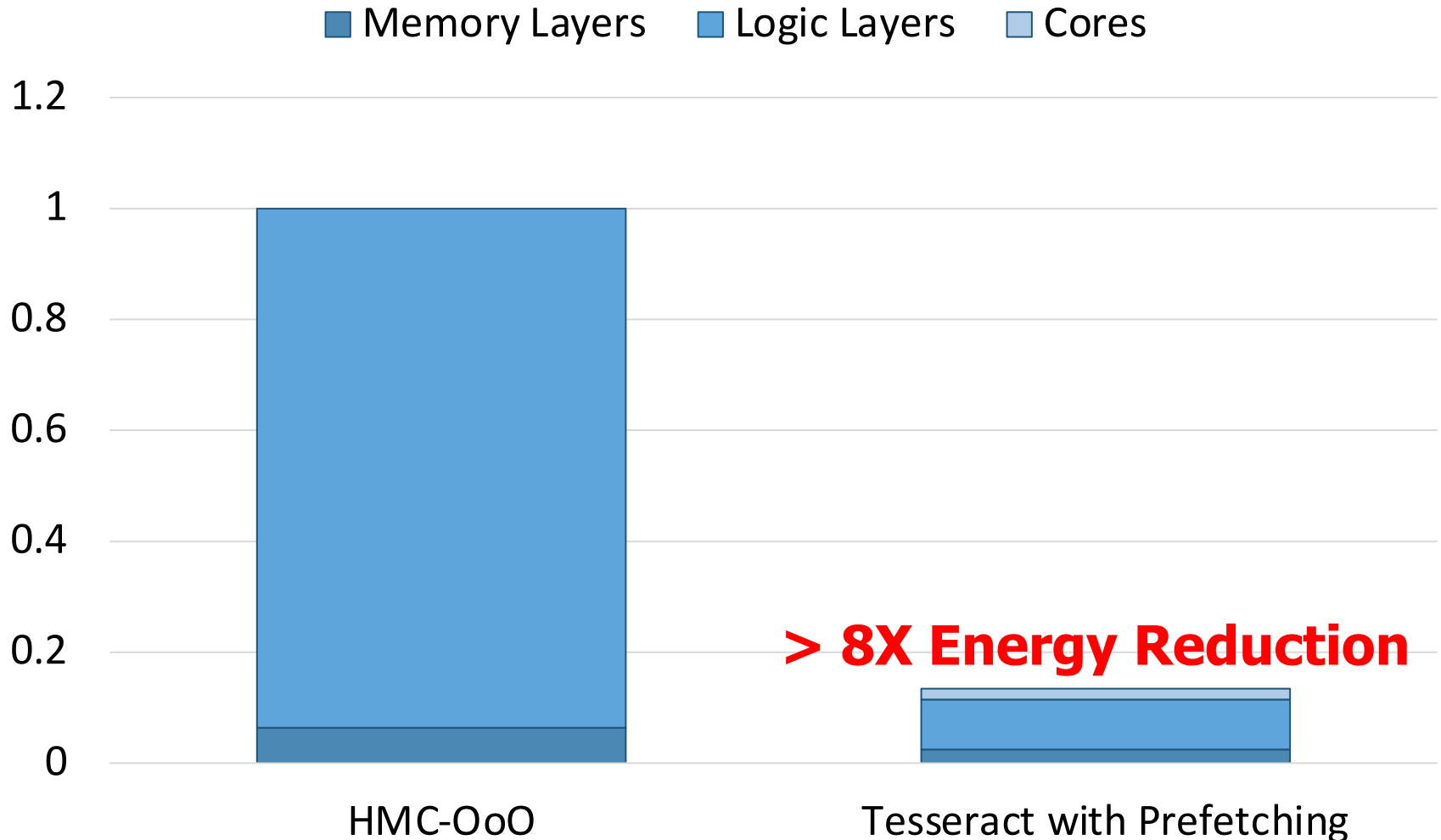
Tesseract Graph Processing Performance



Effect of Bandwidth & Programming Model



Tesseract Graph Processing System Energy



More on Tesseract

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoungh 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[†] Kiyoungh 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