

Computer Architecture

Lecture 8:

Intelligent Genome Analysis

Dr. Mohammed Alser

ALSERM@safari.ethz.ch

ETH Zurich

Fall 2020

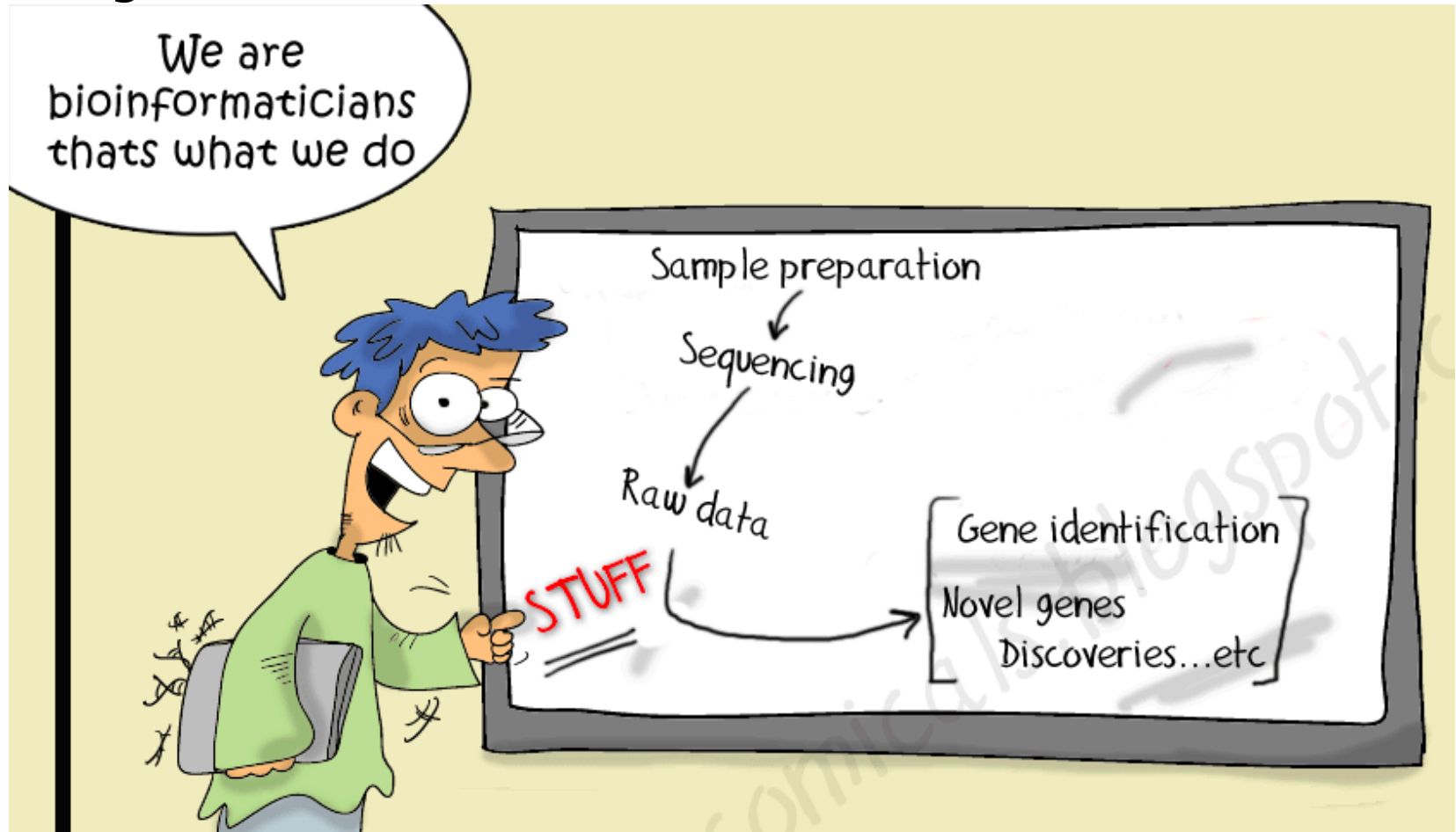
15 October 2020

Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- What Makes Read Mapper Slow?
- Algorithmic & Hardware Acceleration
 - Seed Filtering Technique
 - Pre-alignment Filtering Technique
 - Read Alignment Acceleration
- Where is Read Mapping Going Next?

Agenda for Today

- This lecture is **NOT** about how to **analyze biological** data using available tools.



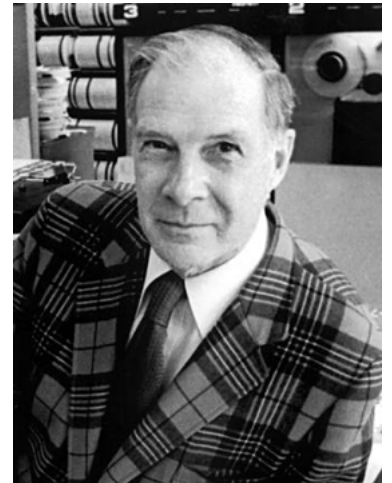
Agenda for Today

- **What is Genome Analysis?**
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- What Makes Read Mapper Slow?
- Algorithmic & Hardware Acceleration
 - Seed Filtering Technique
 - Pre-alignment Filtering Technique
 - Read Alignment Acceleration
- Where is Read Mapping Going Next?

What is Data Analysis?

“The purpose of **computing** is [to gain]
insight, not numbers”

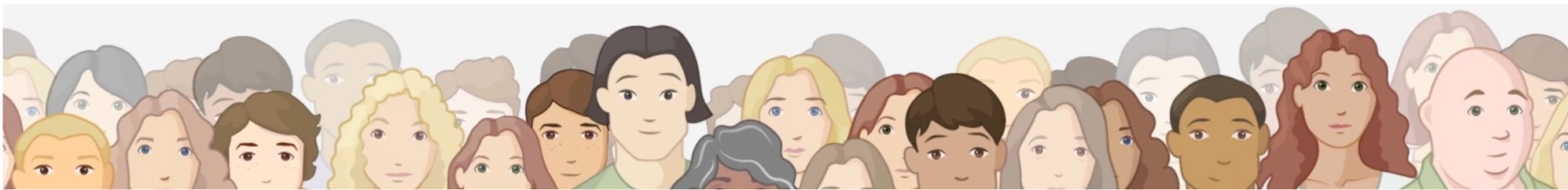
Richard Hamming



What is Genome Analysis?



What is Genome Analysis?



natureresearch

Search  Login 

nature > subjects > genomic analysis

Genomic analysis

 Atom

 RSS Feed

Genomic analysis is the identification, measurement or comparison of genomic features such as DNA sequence, structural variation, gene expression, or regulatory and functional element annotation at a genomic scale. Methods for genomic analysis typically require high-throughput sequencing or microarray hybridization and bioinformatics.

DNA Testing



Fall DNA special
Just 55 CHF ~~89 CHF~~

Order now

The promotion ends today in 12 more hours!



DNA Testing



Fall DNA special
Just 55 CHF ~~89 CHF~~



Health + Ancestry
Service

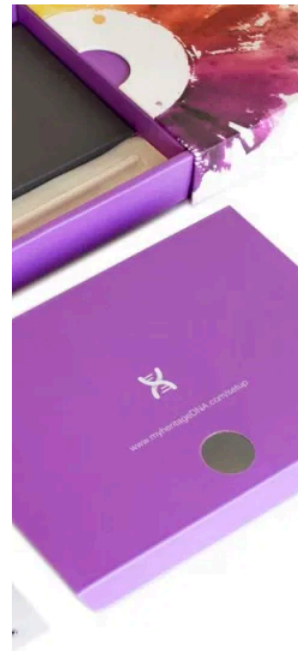
\$199

- Includes everything in Ancestry + Traits Service

PLUS

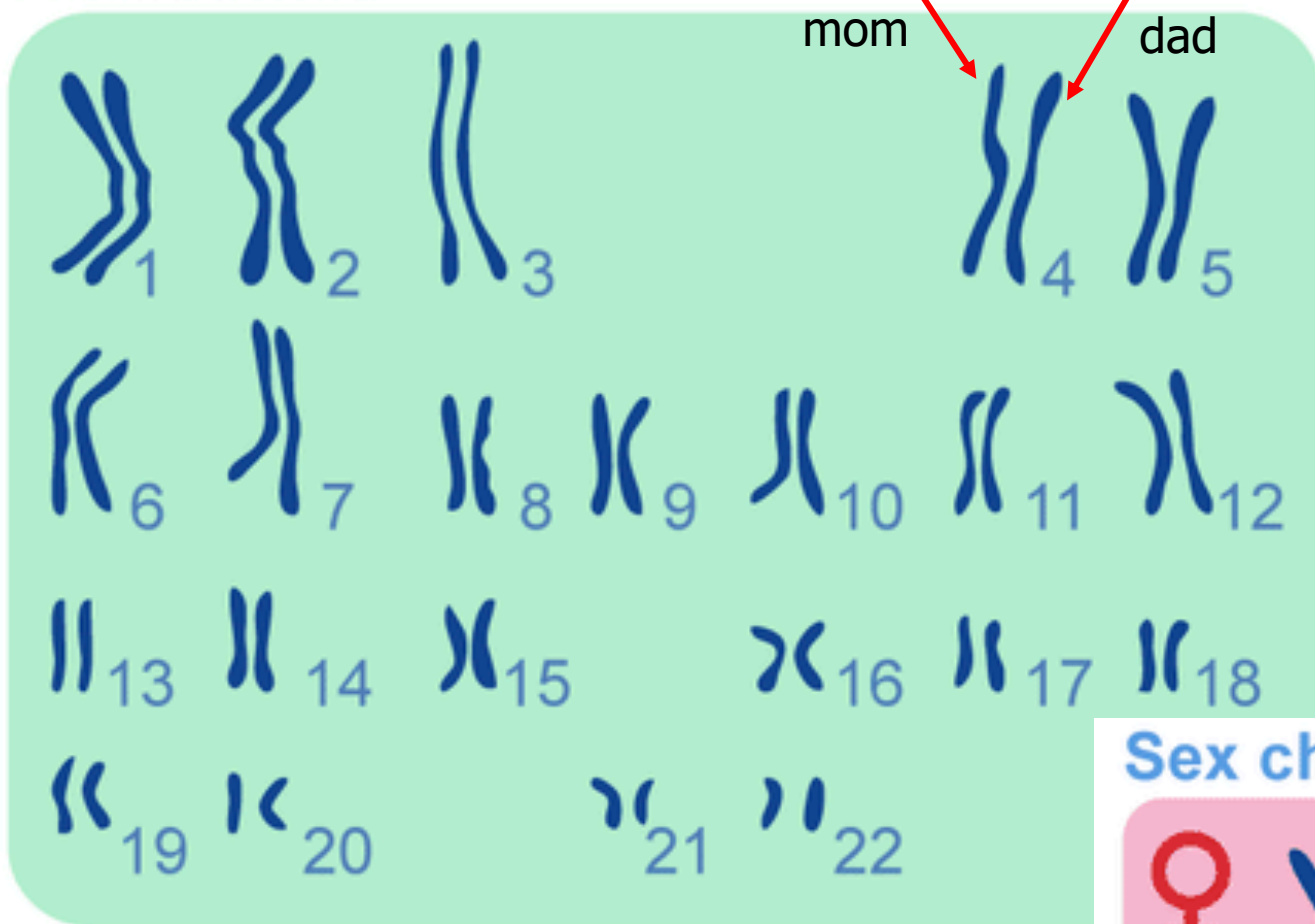
- 10+ Health Predisposition reports*
- 5+ Wellness reports
- 40+ Carrier Status reports*

now
only in 12 more hours!



Human Chromosomes (23 Pairs)

Autosomes

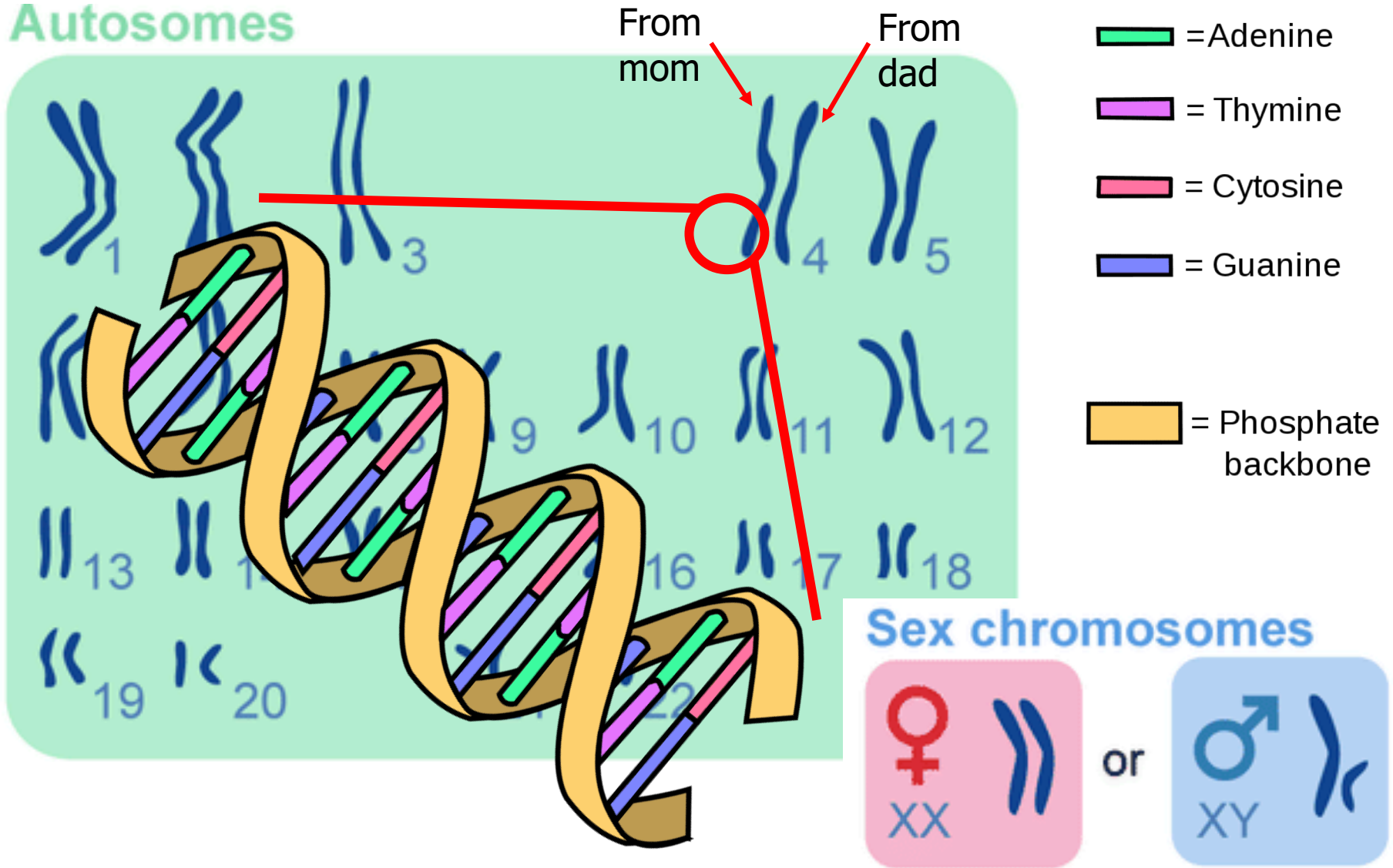


Sex chromosomes




Human Chromosomes (23 Pairs)

Autosomes



Finding SNPs Associated with Complex Trait

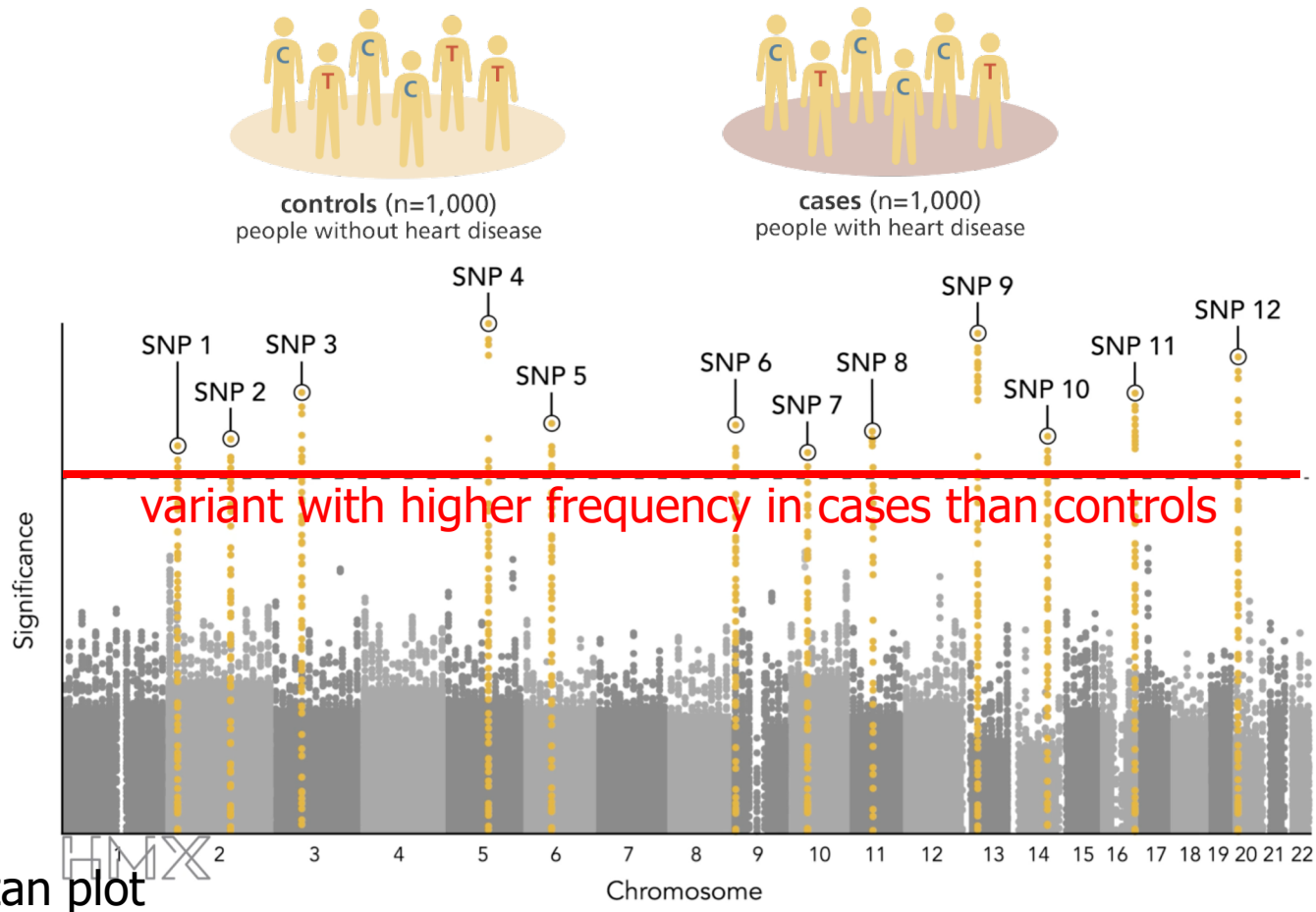
	SNP1	SNP2	Blood Pressure
Individual #1	...ACATG C CGACATTTCATA G GCC...		180
Individual #2	...ACATG C CGACATTTCATA A GCC...		175
Individual #3	...ACATG C CGACATTTCATA G GCC...		170
Individual #4	...ACATG C CGACATTTCATA A GCC...		165
Individual #5	...ACATG C CGACATTTCATA G GCC...		160
Individual #6	...ACATG 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 Study (GWAS)

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



Similar Association Studies

PERSPECTIVE

<https://doi.org/10.1038/s41588-019-0385-z>

nature
genetics

Opportunities and challenges for transcriptome-wide association studies

Michael Wainberg¹, Nasa Sinnott-Armstrong^{ID 2}, Nicholas Mancuso^{ID 3}, Alvaro N. Barbeira^{ID 4}, David A. Knowles^{ID 5,6}, David Golan², Raili Ermel⁷, Arno Ruusalepp^{7,8}, Thomas Quertermous^{ID 9}, Ke Hao^{ID 10}, Johan L. M. Björkegren^{ID 8,10,11,12*}, Hae Kyung Im^{ID 4*}, Bogdan Pasaniuc^{ID 3,13,14*}, Manuel A. Rivas^{ID 15*} and Anshul Kundaje^{ID 1,2*}

Transcriptome-wide association studies (TWAS) integrate genome-wide association studies (GWAS) and gene expression datasets to identify gene-trait associations. In this Perspective, we explore properties of TWAS as a potential approach to prioritize causal genes at GWAS loci, by using simulations and case studies of literature-curated candidate causal genes for schizophrenia, low-density-lipoprotein cholesterol and Crohn's disease. We explore risk loci where TWAS accurately prioritizes the likely causal gene as well as loci where TWAS prioritizes multiple genes, some likely to be non-causal, owing to sharing of expression quantitative trait loci (eQTL). TWAS is especially prone to spurious prioritization with expression data from non-trait-related tissues or cell types, owing to substantial cross-cell-type variation in expression levels and eQTL strengths. Nonetheless, TWAS prioritizes candidate causal genes more accurately than simple baselines. We suggest best practices for causal-gene prioritization with TWAS and discuss future opportunities for improvement. Our results showcase the strengths and limitations of using eQTL datasets to determine causal genes at GWAS loci.

Wainberg+, "Opportunities and challenges for transcriptome-wide

association studies", *Nature genetics*, 2019.

SNPs and Personalized Medicine

openSNP

Q Search

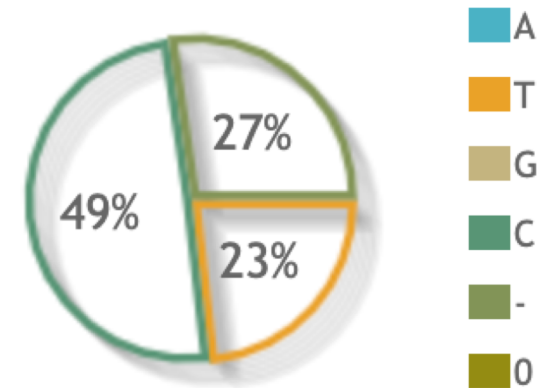
☰

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

Personalized Medicine in UK

npj | Genomic Medicine

www.nature.com/npjgenmed

PMCID: PMC5884823

PMID: [29644095](https://pubmed.ncbi.nlm.nih.gov/29644095/)

[NPJ Genom Med](#). 2018; 3: 10.

Published online 2018 Apr 4. doi: [10.1038/s41525-018-0049-4](https://doi.org/10.1038/s41525-018-0049-4)

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

[Lauge Farnaes](#),^{#1,2} [Amber Hildreth](#),^{#1,2} [Nathaly M. Sweeney](#),^{#1,2} [Michelle M. Clark](#),¹ [Shimul Chowdhury](#),¹ [Shareef Nahas](#),¹ [Julie A. Cakici](#),¹ [Wendy Benson](#),¹ [Robert H. Kaplan](#),³ [Richard Kronick](#),⁴ [Matthew N. Bainbridge](#),¹ [Jennifer Friedman](#),^{1,2,5} [Jeffrey J. Gold](#),^{1,5} [Yan Ding](#),¹ [Narayanan Veeraraghavan](#),¹ [David Dimmock](#),¹ and [Stephen F. Kingsmore](#)^{✉1}



reduced inpatient cost by

\$9.9K-\$327K

“From 2019, **all seriously ill children** in UK will be offered **whole genome sequencing** as part of their care”

NHS

**National Institute for
Health Research**

SAFARI

Farnaes+, “[Rapid whole-genome sequencing decreases infant morbidity and cost of hospitalization](#)”, *NPJ Genom Med*. 2018

Mirror Phenotypes of 593 Kb CNVs



AUTISM

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



SCHIZOPHRENIA

McCarthy, *Nat Genet* 2009
Duplication of 593 kb



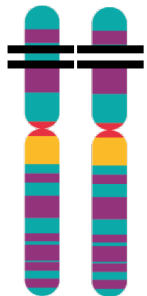
OBESITY

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)

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

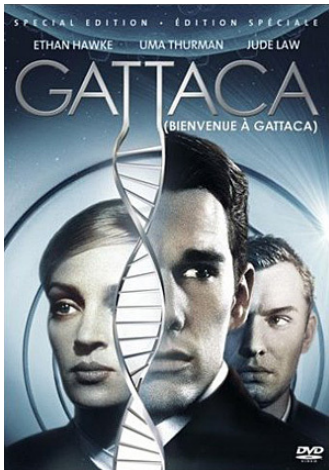
Agenda for Today

- What is Genome Analysis?
- **What is Intelligent Genome Analysis?**
- How we Analyze Genome?
- What Makes Read Mapper Slow?
- Algorithmic & Hardware Acceleration
 - Seed Filtering Technique
 - Pre-alignment Filtering Technique
 - Read Alignment Acceleration
- Where is Read Mapping Going Next?

Fast Genome Analysis?

- **Fast** genome analysis in mere seconds using **limited computational resources** (i.e., personal computer or small hardware).

1997



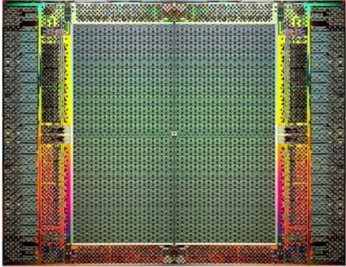
2015



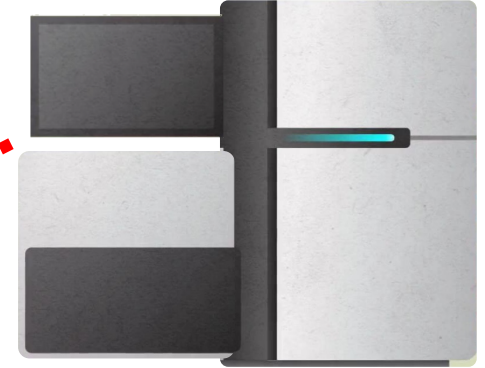
Intelligent Architecture?

Modern systems

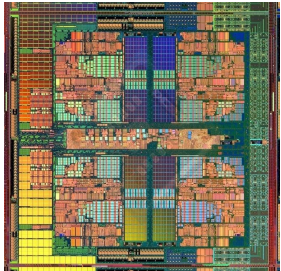
FPGAs



?



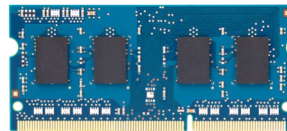
Sequencing Machine



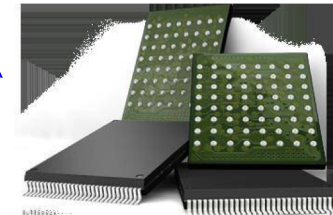
Heterogeneous Processors and Accelerators



Hybrid Main Memory



(General Purpose) GPUs

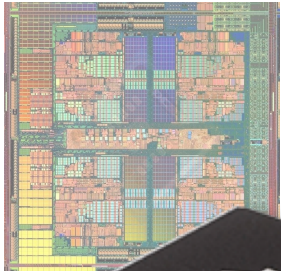
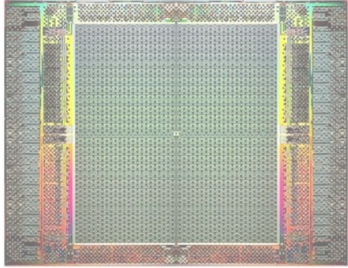


Persistent Memory/Storage

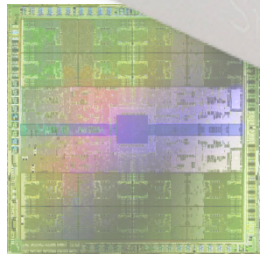
Intelligent Architecture?

Modern systems

FPGAs

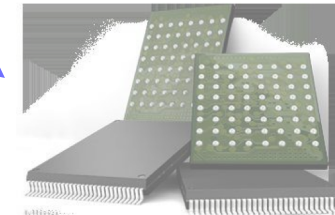


Hetero
Pro
Ac



(General Purpose) GPUs

Sequencing
Machine



Persistent Memory/Storage

Privacy-Preserving Genome Analysis?

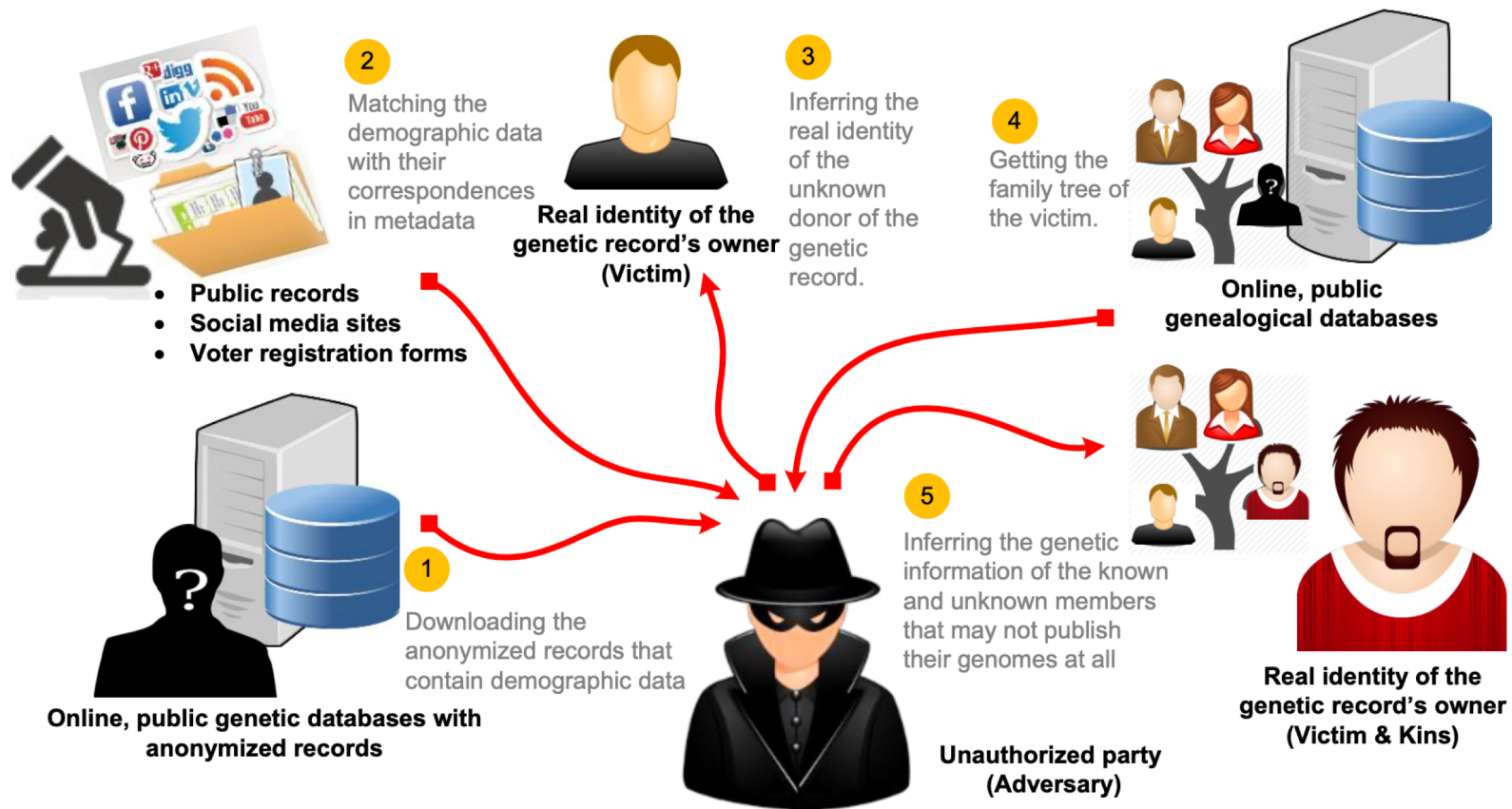


Fig. 5. A completion attack.

Alser+, "**Can you really anonymize the donors of genomic data in today's digital world?**" *10th International Workshop on Data Privacy Management (DPM)*, 2015.

Can you Really Anonymize the Donors?

(Position Paper) Can You Really Anonymize the Donors of Genomic Data in Today's Digital World?

Mohammed Alser, Nour Almadhoun, Azita Nouri, Can Alkan, and Erman Ayday

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

Abstract. The rapid progress in genome sequencing technologies leads to availability of high amounts of genomic data. Accelerating the pace of biomedical breakthroughs and discoveries necessitates not only collecting millions of genetic samples but also granting open access to genetic databases. However, one growing concern is the ability to protect the privacy of sensitive information and its owner. In this work, we survey a wide spectrum of cross-layer privacy breaching strategies to human genomic data (using both public genomic databases and other public non-genomic data). We outline the principles and outcomes of each technique, and assess its technological complexity and maturation. We then review potential privacy-preserving countermeasure mechanisms for each threat.

Keywords: Genomics, Privacy, Bioinformatics

DPM 2015

Vienna, Austria
September 21-22, 2015

Alser+, "**Can you really anonymize the donors of genomic data in today's digital world?**" *10th International Workshop on Data Privacy Management (DPM), 2015.*

Rapid Surveillance of Disease Outbreaks?

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



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

Scalable SARS-CoV-2 Testing



THE PREPRINT SERVER FOR HEALTH SCIENCES



Cold
Spring
Harbor
Laboratory



Yale

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

Population-Scale Microbiome Profiling



City-Scale Microbiome Profiling

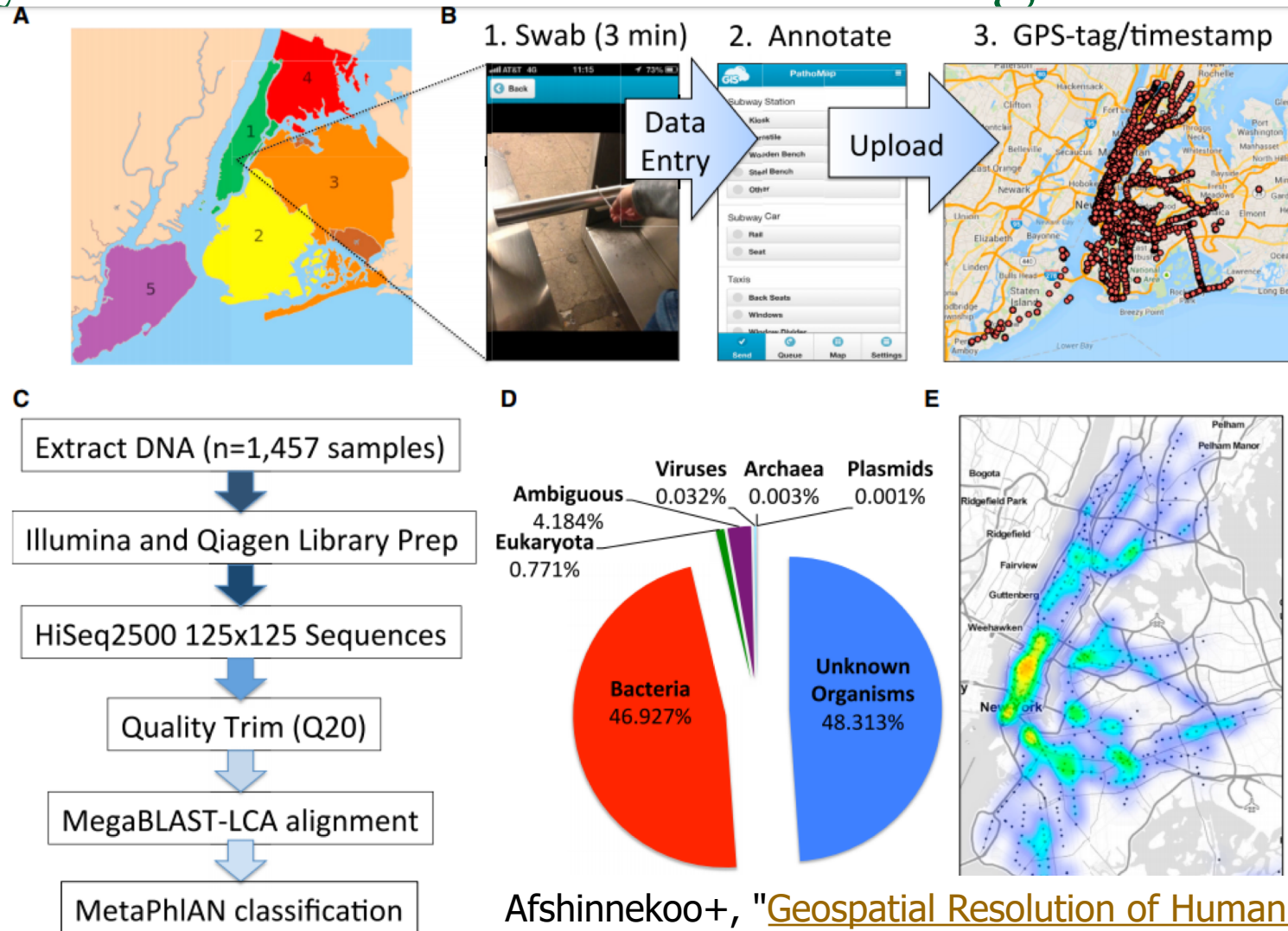


Figure 1. The Metagenome of New York City

(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

Plague in New York Subway System?

Plague (Yersinia Pestis)



Harvard Health Publishing
HARVARD MEDICAL SCHOOL

Trusted advice for a healthier life

What Is It?

Published: December, 2018

Plague is caused by Yersinia pestis bacteria. It can be a life-threatening infection if not treated promptly. Plague has caused several major epidemics in Europe and Asia over the last 2,000 years. Plague has most famously been called "the Black Death" because it can cause skin sores that form black scabs. A plague epidemic in the 14th century killed more than one-third of the population of Europe within a few years. In some cities, up to 75% of the population died within days, with fever and swollen skin sores.

Plague in New York Subway System?

Plague (Yersinia)

What Is It?

Published: December, 2018

Plague is caused by Yersinia treated promptly. Plague has last 2,000 years. Plague has cause skin sores that form b than one-third of the popul the population died within

The New York Times
Bubonic Plague in the Subway System? Don't Worry About It



In October, riders were not deterred after reports that an Ebola-infected man had ridden the subway just before he fell ill. Robert Stolarik for The New York Times

<https://www.nytimes.com/2015/02/07/nyregion/bubonic-plague-in-the-subway-system-dont-worry-about-it.html>

The findings of Yersinia Pestis in the subway received wide coverage in the lay press, causing some alarm among New York residents

Failure of Bioinformatics



data. Rob Knight, a professor in the department of pediatrics at the University of California, San Diego, calls this type of error “a **failure of bioinformatics**,” in that Mason had assumed the gene fragments were unique to the pathogens, when in fact they can also be detected in other

Living in a microbial world

Charles Schmidt

Nature Biotechnology, **volume 35**, pages 401–403 (2017)

<https://www.nature.com/articles/nbt.3868>

There is a critical need for **fast** and
accurate genome analysis.

What is Intelligent Data Analysis?

How and where to enable

fast, accurate, cheap,

privacy-preserving, and exabyte scale

analysis of genomic data?

Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- **How we Analyze Genome?**
- What Makes Read Mapper Slow?
- Algorithmic & Hardware Acceleration
 - Seed Filtering Technique
 - Pre-alignment Filtering Technique
 - Read Alignment Acceleration
- Where is Read Mapping Going Next?

Genome Analysis

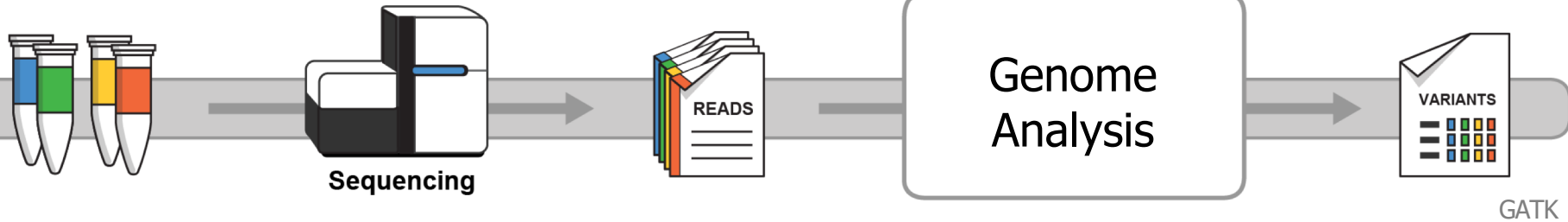


NO machine can read the *entire* content of a genome



```
>CCTCCTCAGTGCCACCCAGCCCACTGGCAGCTCCCAAACAGGCTCTTATTAACACCCCTGTTCCCTGCCCTTGGAGTGAGGTGTCAAG  
GACCTAACTAAAAAAAAAAAAAAAAAGAAAAAGAAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTCTT  
CATGTCAAGGACCTAATGTGCTAAACAGCACTTTTTTGACCATTATTTTGGATCTGAAAGAAATCAAGAATAAATGAAGGACTTGATACATTG  
GAAGAGGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAAAAAGAAAAAGAAAAAGAAAAAGAATTAAATTTAAGTAATTCTTTGAAAAAA  
ACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCTGTGTTGCAGGTCTTCTTGCATTTCCCTGTCAAAAGAAAAAGAATTTAAATTT  
AAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCAGGCCAAGAGTTGCAAAAAAAAAAAAAAGAAAAA  
GAAAAGAAAAAGAATTTAAATTTAAAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTAGCCAGAATGG  
TTGTGGGATGGGAGCCTCTGTGGACCGACCAGGTAGCTCTCTTTCCACACTGTAGTCTCAAAGCTTCTTCATGTGGTCTTCTGAGTGAAA  
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

High-Throughput Sequencers



Illumina MiSeq



Pacific
Biosciences
Sequel II

Oxford
Nanopore
PromethION



Illumina NovaSeq 6000



Pacific Biosciences RS II



Oxford Nanopore MinION

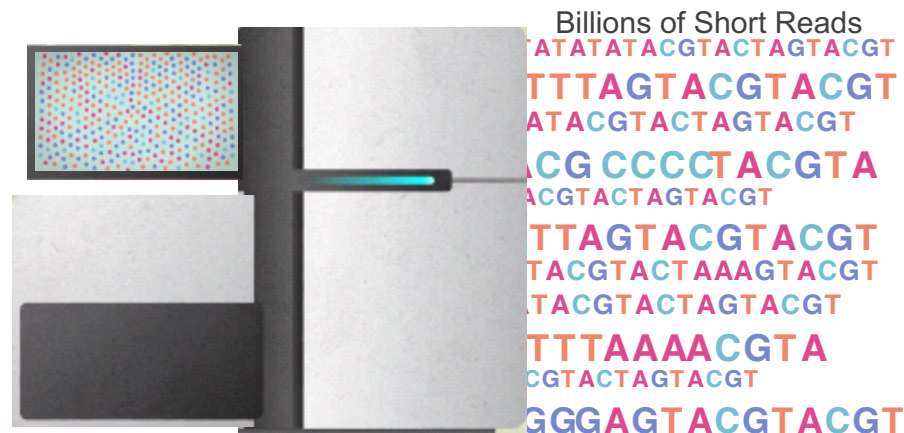


Oxford
Nanopore
SmidgION

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

How Does HTS Machine Work?

Reads lack information about their order and location (which part of genome they are originated from)



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 ($\sim 0.1\%$)

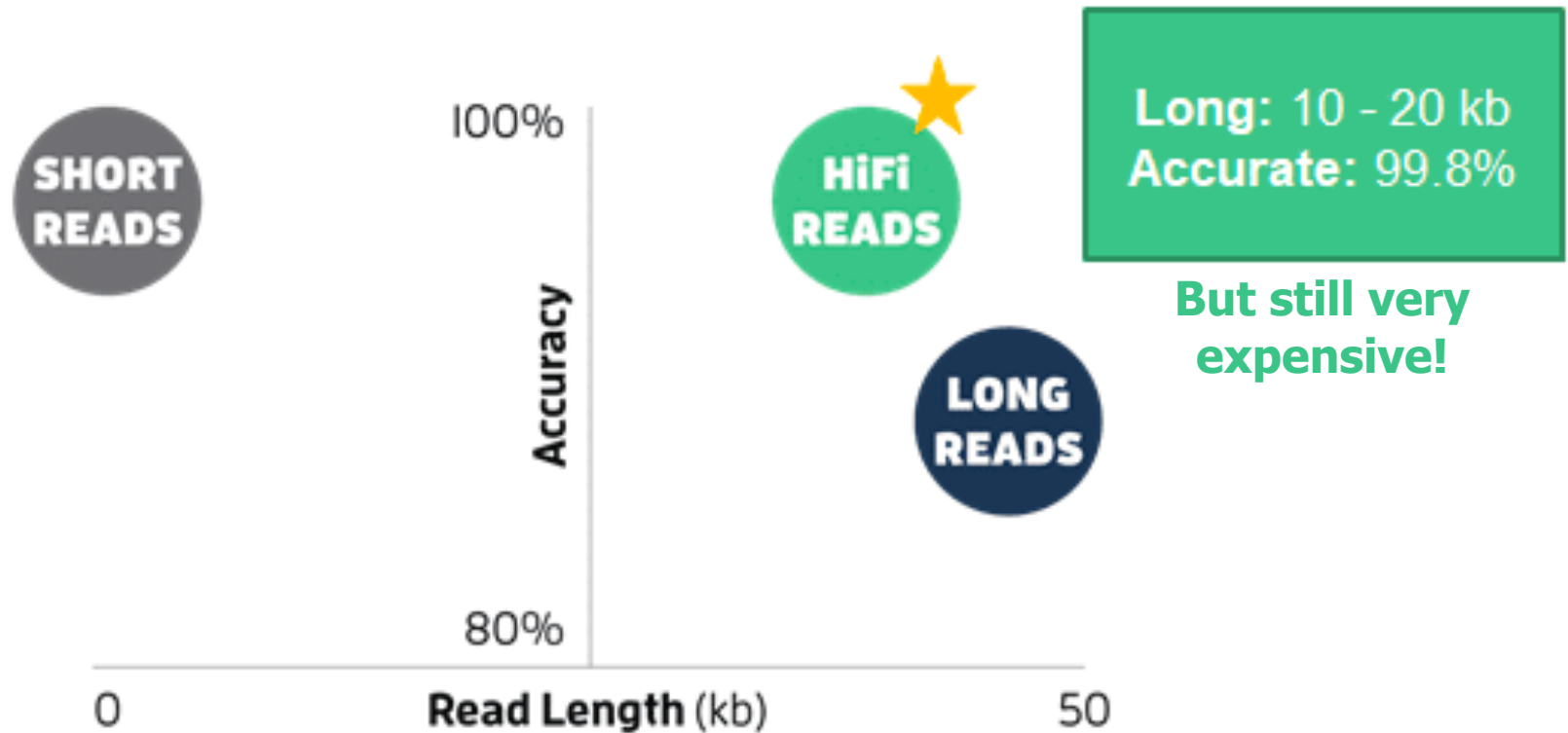
☐ 500-2M bp

☐ high error rate ($\sim 15\%$)

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

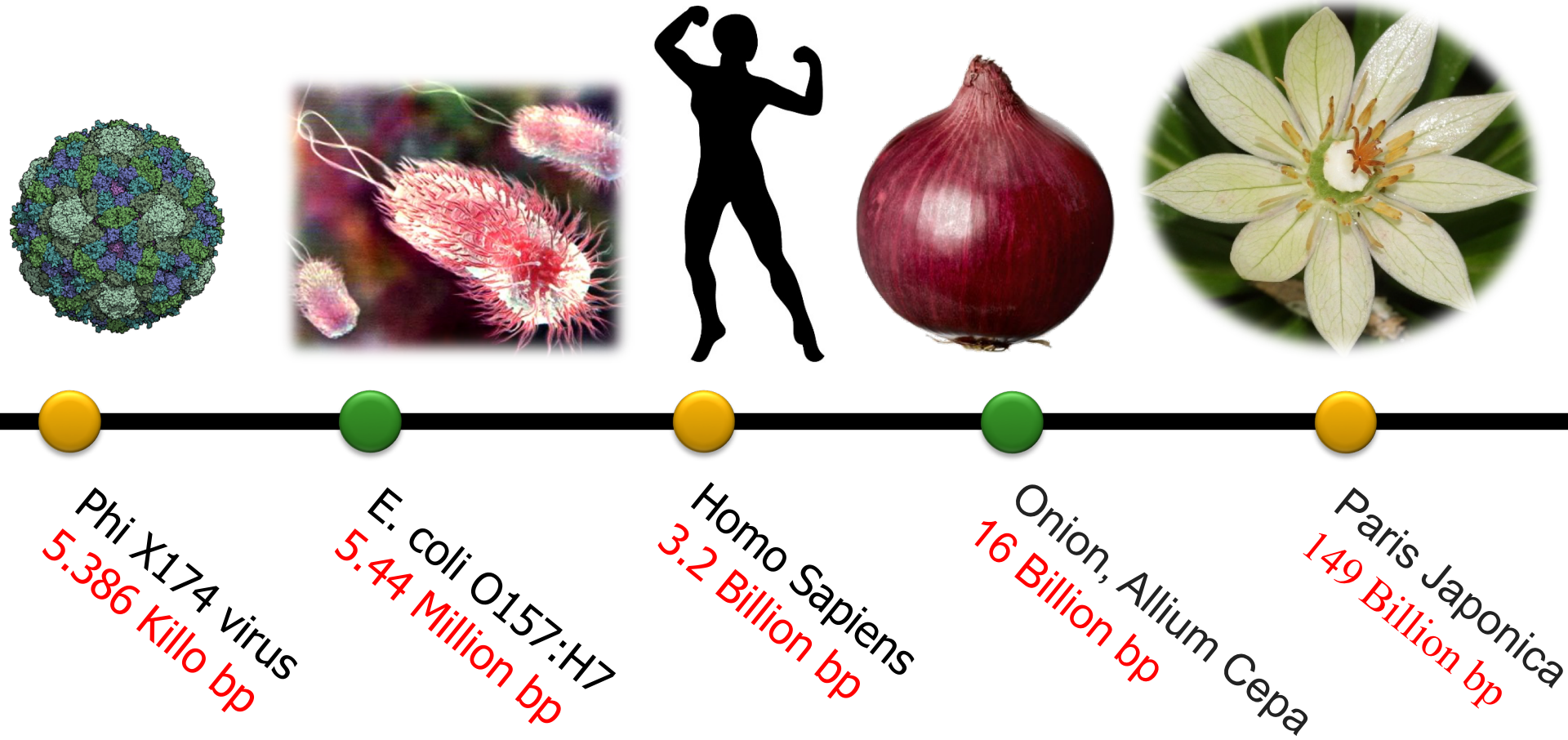
HiFi Reads (PacBio)

HIFI READS ARE LONG AND ACCURATE



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 3.2×10^9
G T bases

 13 years

 $> 3 \times 10^9$ \$

Obtaining the Human Reference Genome

■ **GRCh38.p13**

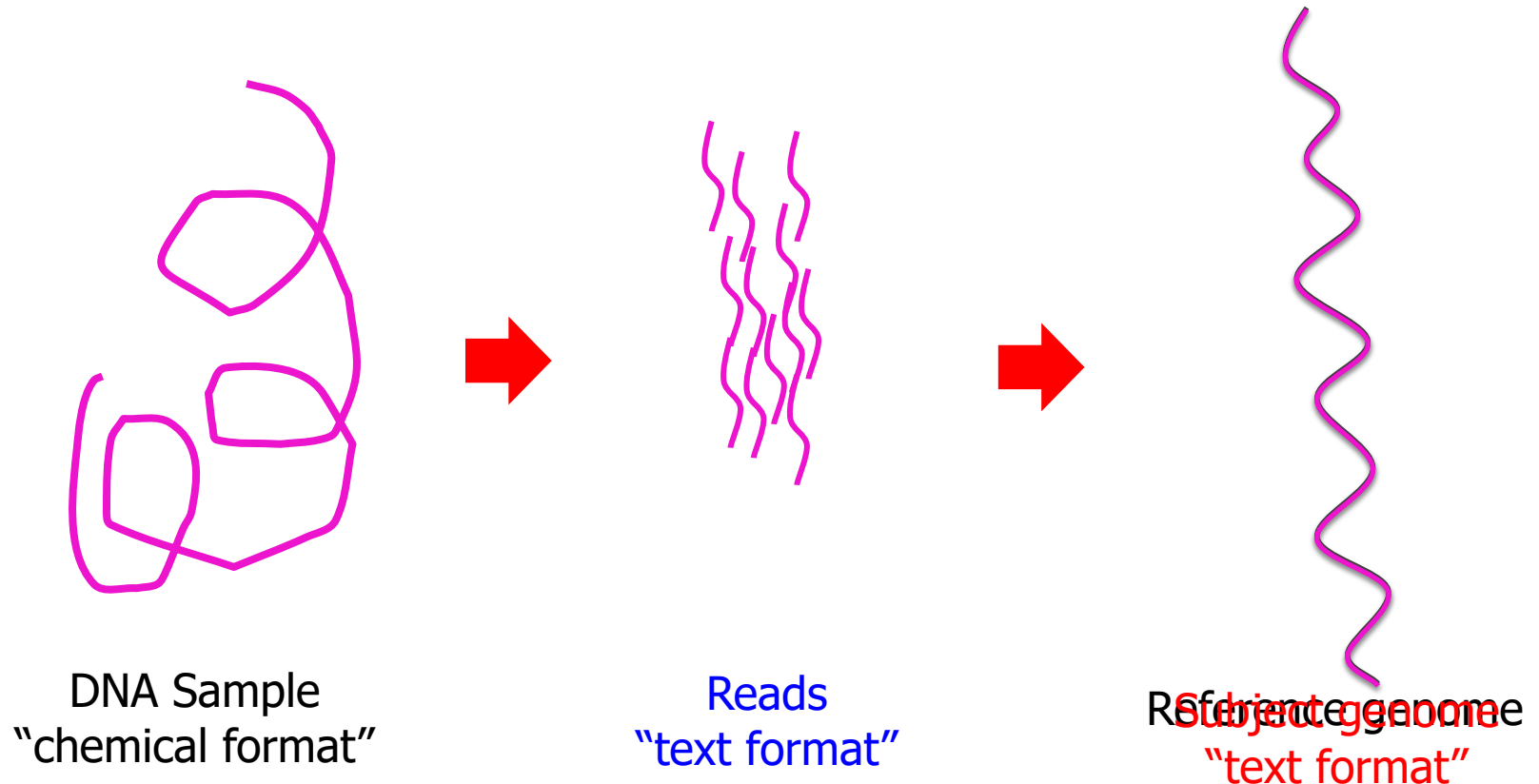
- Description: Genome Reference Consortium Human Build 38 patch release 13 (GRCh38.p13)
- Organism name: Homo sapiens (human)
- Date: 2019/02/28
- 3,099,706,404 bases
- Compressed .fna file (964.9 MB)
- https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39

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

[illegible]

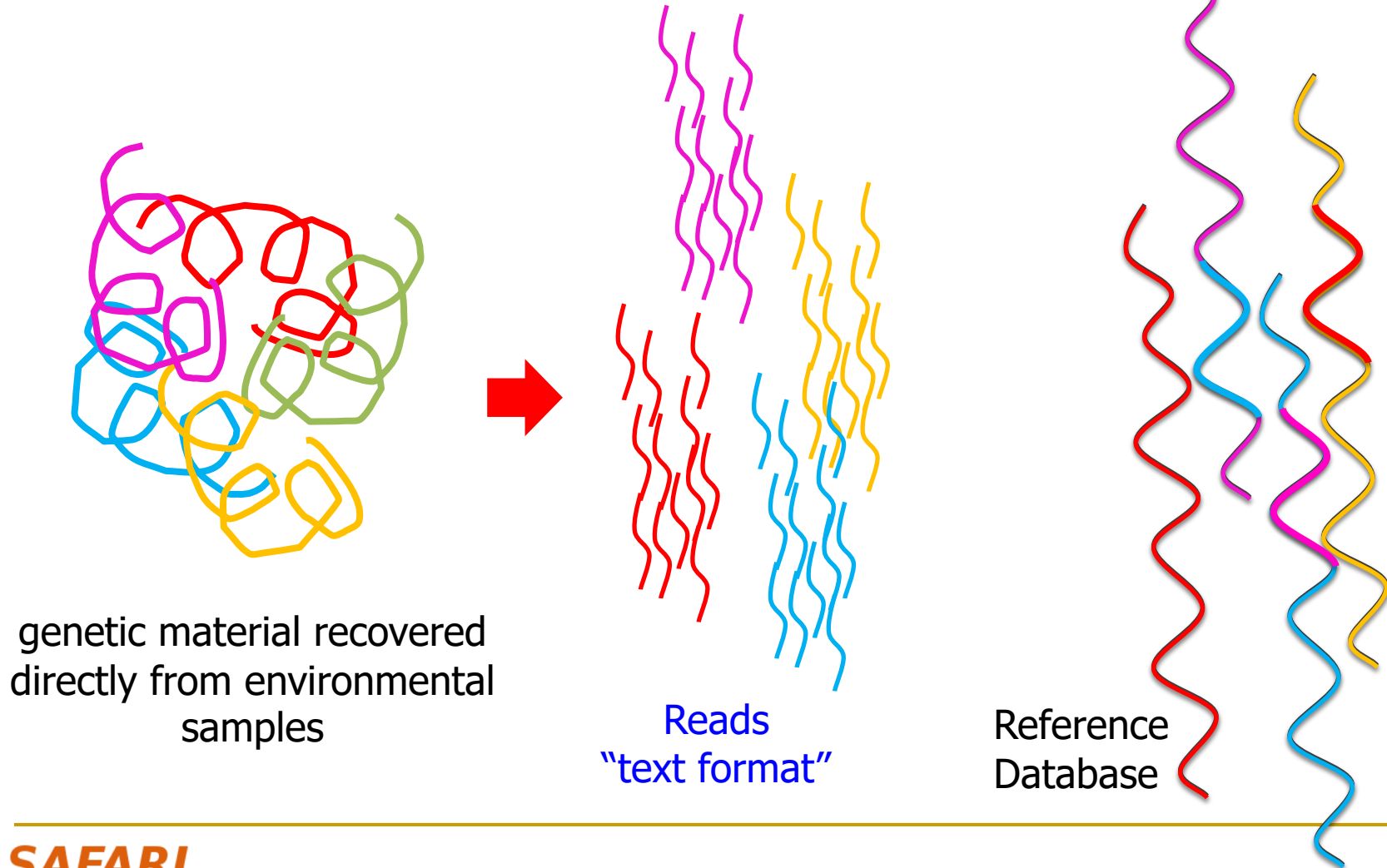
Genome Analysis

Map **reads** to a known reference genome with some minor differences allowed



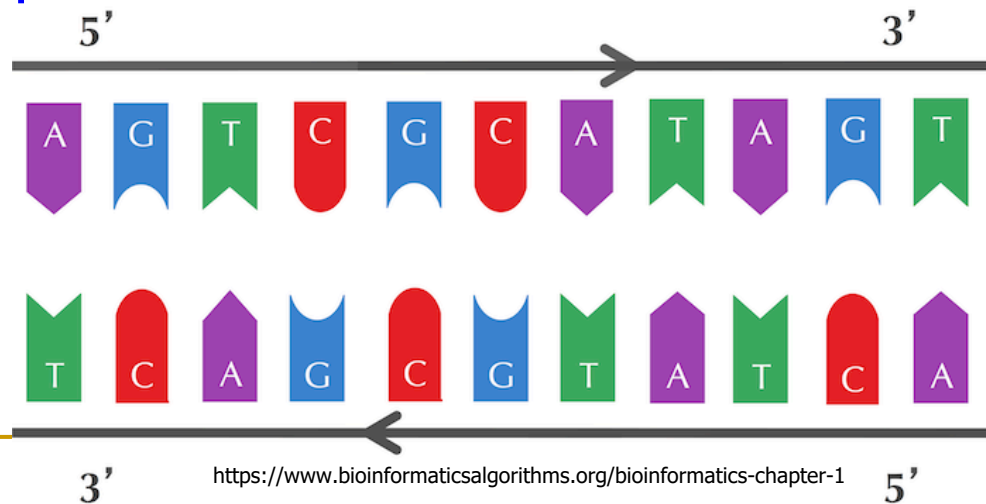
Metagenomics Analysis

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

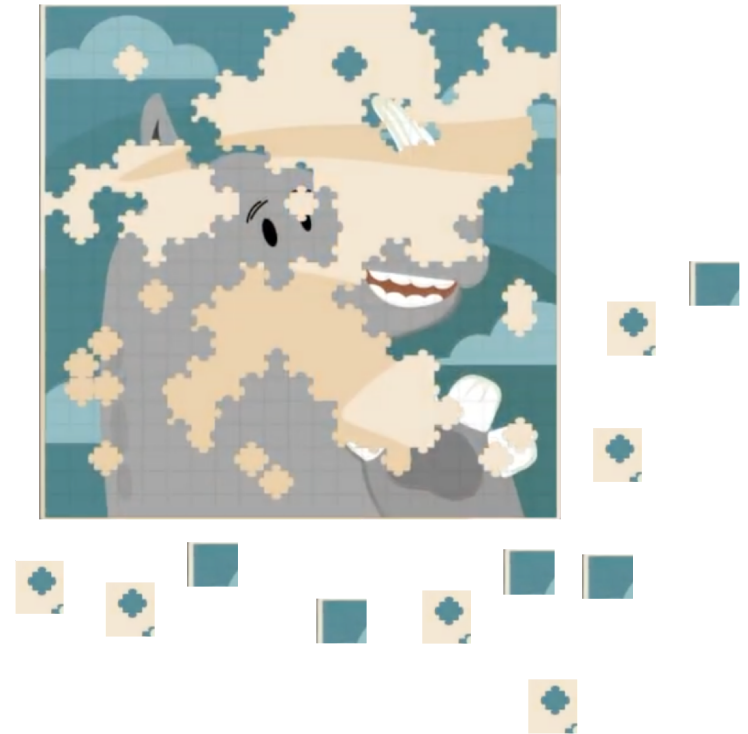


Challenges in Read Mapping

- 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”

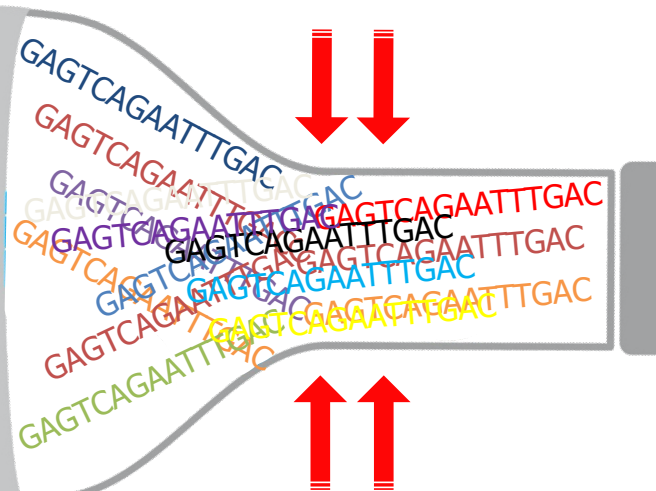
Bottlenecked in Read Mapping!!

48 Human whole
genomes

at 30× coverage

in about 2 days

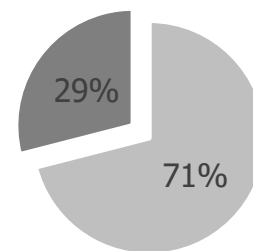
Illumina NovaSeq 6000



1 Human
genome

32 CPU hours

on a 48-core processor



■ Read Mapping ■ Others

Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- **What Makes Read Mapper Slow?**
- Algorithmic & Hardware Acceleration
 - Seed Filtering Technique
 - Pre-alignment Filtering Technique
 - Read Alignment Acceleration
- Where is Read Mapping Going Next?

Let's First Learn How to Map a Read

Read Mapping in 111 pages!

Analyzing 107 read mappers (1988-2020) in depth

arXiv.org > q-bio > arXiv:2003.00110

Search...

Help | Advanced

Quantitative Biology > Genomics

[Submitted on 28 Feb 2020 (v1), last revised 9 Jul 2020 (this version, v3)]

Technology dictates algorithms: Recent developments in read alignment

Mohammed Alser, Jeremy Rotman, 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

Alser+, "[Technology dictates algorithms: Recent developments in read alignment](#)", arXiv, 2020

GitHub: https://github.com/Mangul-Lab-USC/review_technology_dictates_algorithms

Read Mapping: A Brute Force Algorithm

Reference



Read

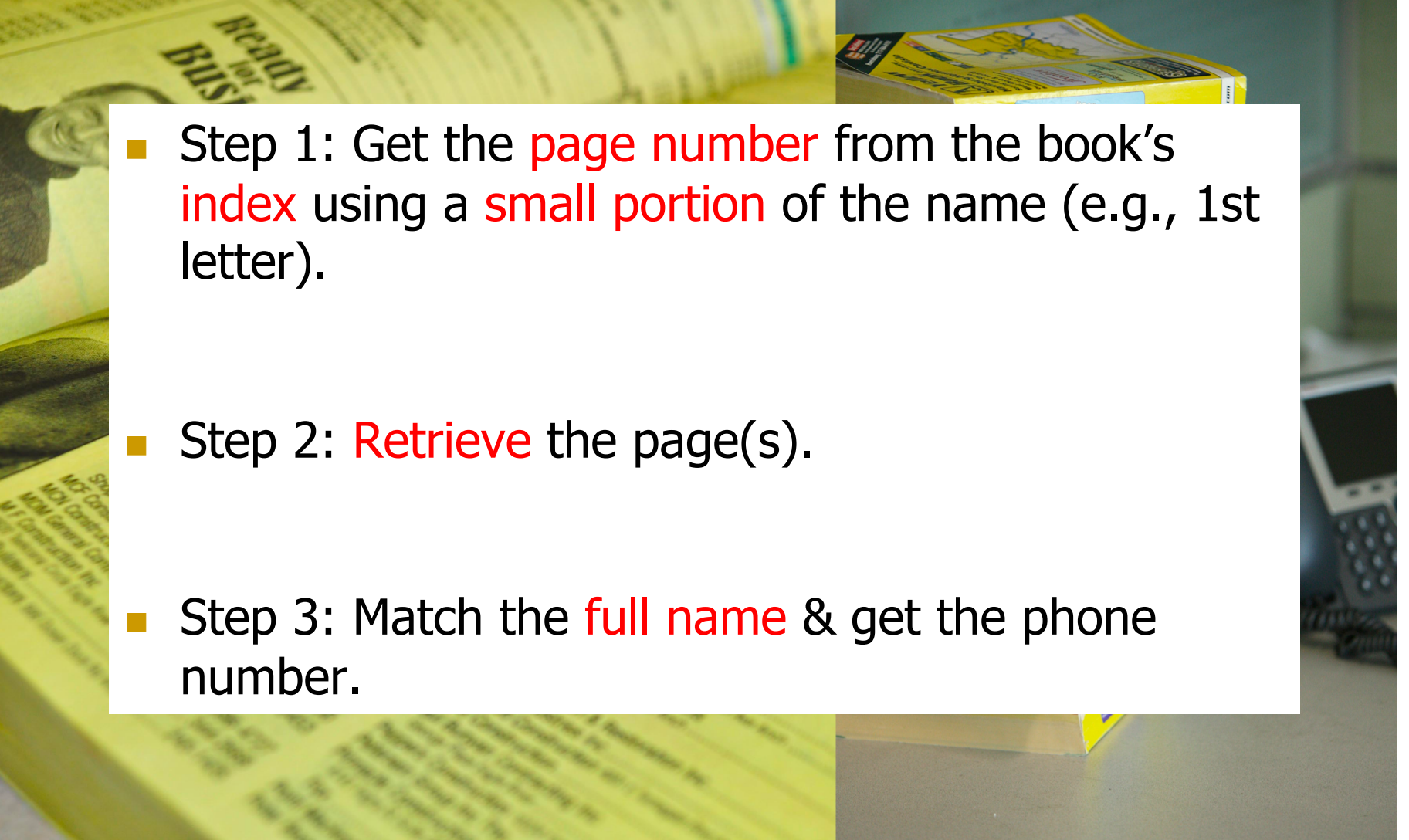
Very Expensive!
 $O(m^2kn)$

m : read length

k : no. of reads

n : reference genome length

Similar to Searching Yellow Pages!

- 
- Step 1: Get the **page number** from the book's **index** using a **small portion** of the name (e.g., 1st letter).
 - Step 2: **Retrieve** the page(s).
 - Step 3: Match the **full name** & get the phone number.

Mapping a Read is
Similar to Querying
the Yellow Pages!

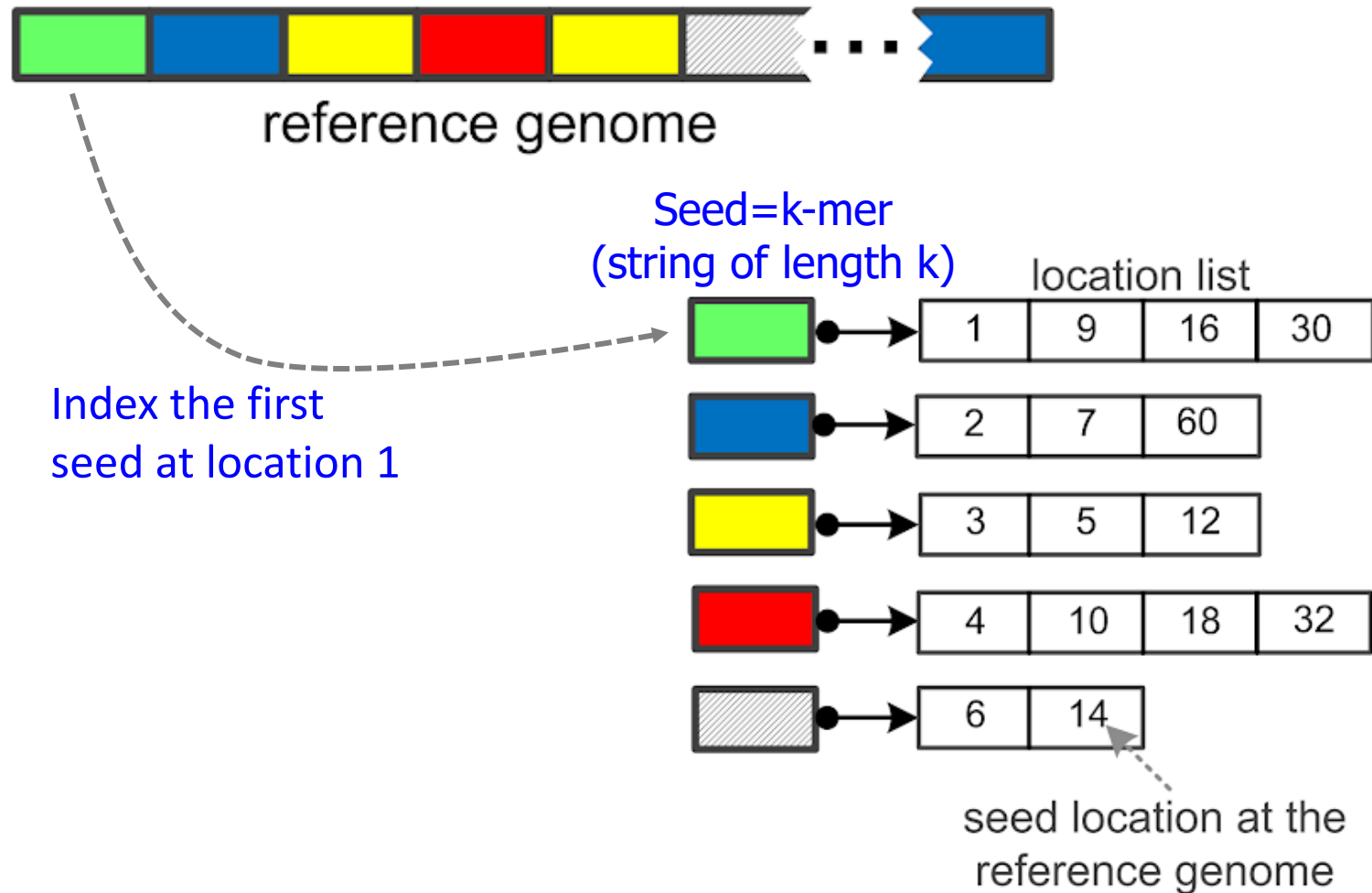
Step 1: Indexing the Reference Genome



Hashing is the most popular indexing technique for read mapping since 1988

Alser+, "[Technology dictates algorithms: Recent developments in read alignment](#)", arXiv, 2020

Step 1: Indexing the Reference Genome



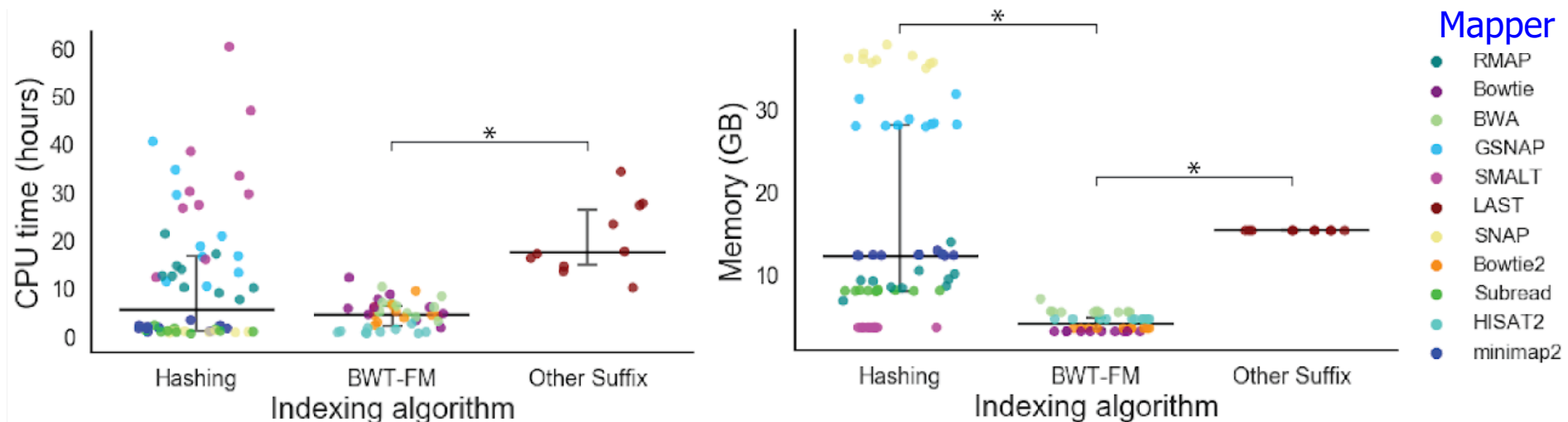
Genome Index Properties

- The index is built **only once** for each reference.
- **Seeds** can be overlapping, non-overlapping, spaced, adjacent, non-adjacent, minimizers, compressed, ...

Tool	Version	Index Size [*]	Indexing Time
mrFAST	2.2.5	16.5 GB	20.00 min
minimap2	0.12.7	7.2 GB	3.33 min
BWA-MEM	0.7.17	4.7 GB	49.96 min

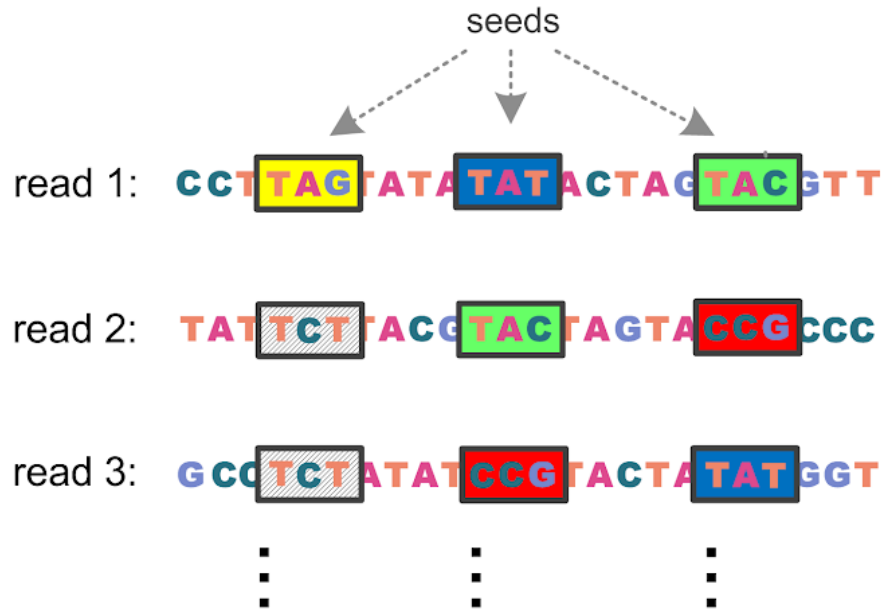
^{*}Human genome = 3.2 GB

Performance of Human Genome Indexing

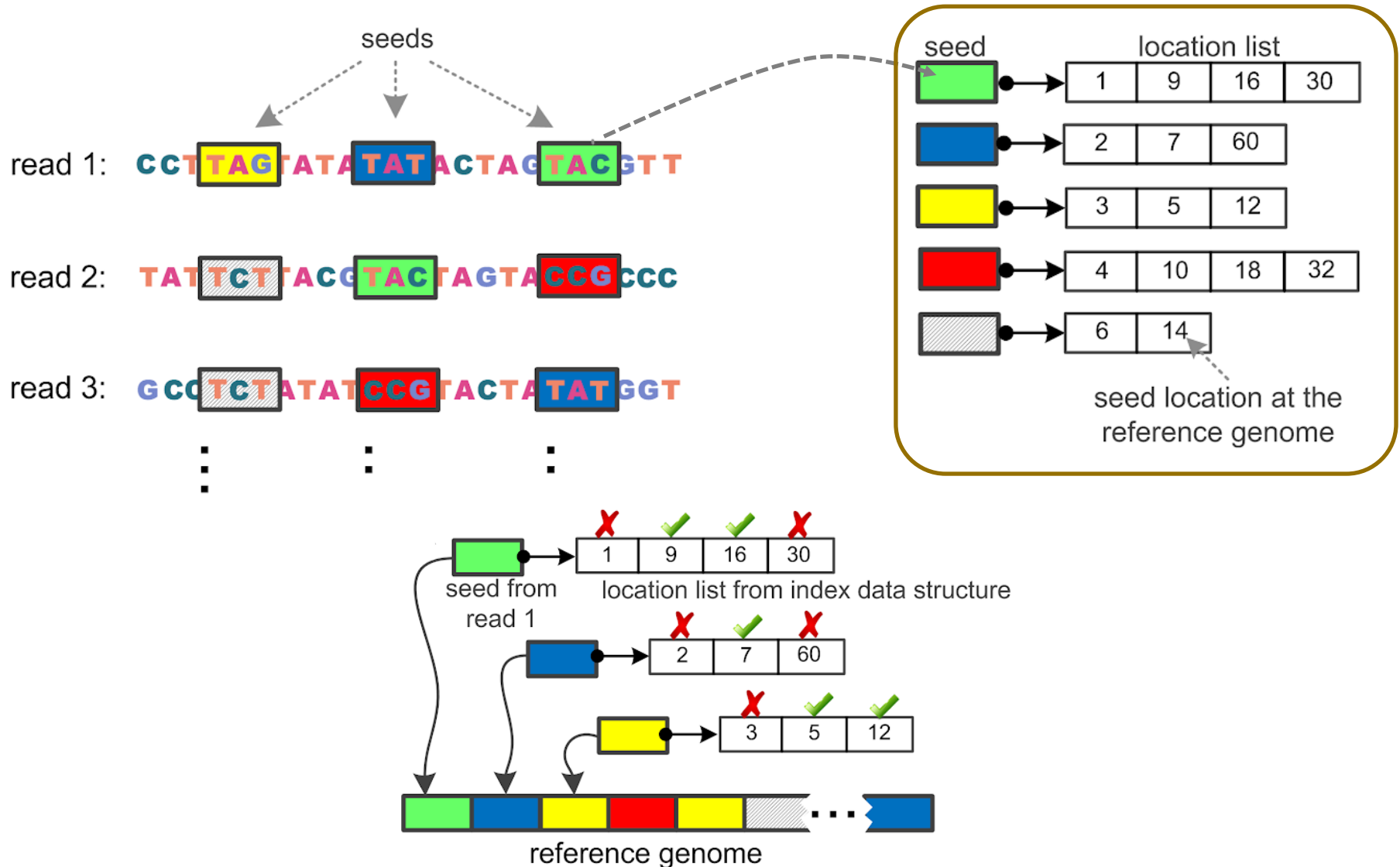


Alser+, "[Technology dictates algorithms: Recent developments in read alignment](#)", arXiv, 2020

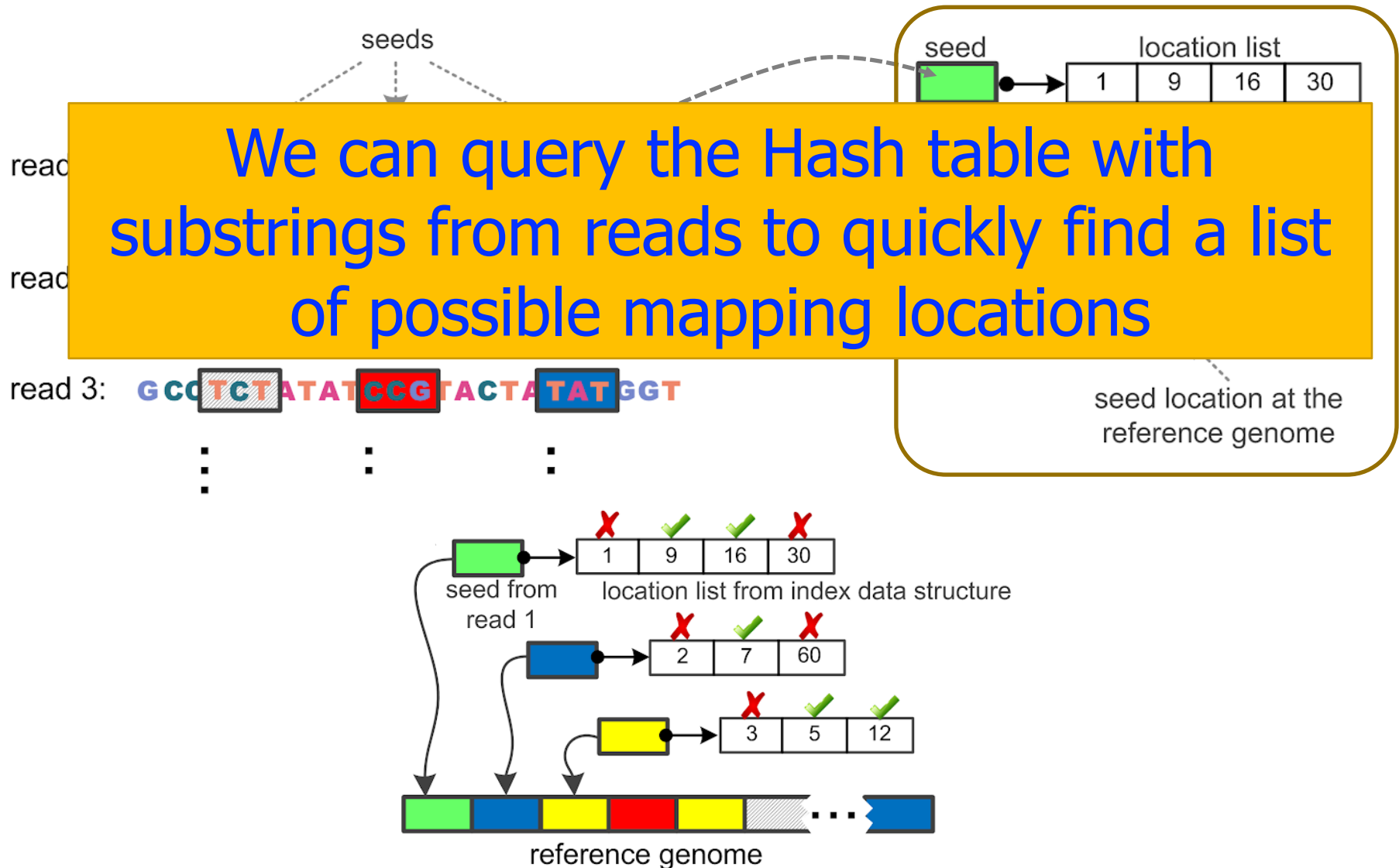
Step 2: Query the Index Using Read Seeds



Step 2: Query the Index Using Read Seeds

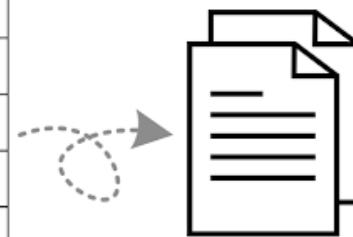


Step 2: Query the Index Using Read Seeds



Step 3: Read Alignment (Verification)

	C	G	T	T	A	G	T	C	T	A	...
C	0	0	0	0	0	0	0	0	0	0	
C	0	2	2	2	2	2	2	2	2	2	
T	0	2	3	3	3	3	3	3	4	4	
T	0	2	3	5	5	5	5	5	5	6	
A	0	3	3	5	7	9	9	9	9	9	
G	0	2	4	5	7	9	11	11	11	11	
T	0	2	4	6	7	9	11	13	13	13	
A	0	2	4	6	7	9	11	13	14	14	
T	0	2	4	6	8	9	11	13	14	16	
⋮											



.bam/.sam file contains
necessary alignment
information (e.g., type,
location, and number of
each edit)

Step 3: 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.

organization x operation

Ref	o	-	-	r	g	a	n	i	z	a	t	i	o	n
Read	o	p	e	r	-	-	-	-	-	a	t	i	o	n

Ref	o	-	-	r	g	a	n	i	z	a	t	i	o	n
Read	o	p	e	r	-	a	-	-	-	-	t	i	o	n

Edit distance = 7

match
deletion
insertion
mismatch

organization x translation

Ref	o	r	g	a	n	i	z	-	a	t	i	o	n
Read	t	r	-	a	n	-	s	-	a	t	i	o	n

Ref	o	r	g	a	n	-	i	z	a	t	i	o	n
Read	t	r	-	a	n	s	-	-	a	t	i	o	n

Ref	o	r	g	a	n	i	z	a	t	i	o	n
Read	t	r	-	a	n	s	-	a	t	i	o	n

Edit distance = 4

Smith-Waterman remains
the most popular algorithm
since 1988

Hamming distance is
the second most popular technique
since 2008

Alser+, "[Technology dictates algorithms: Recent developments in read alignment](#)", arXiv, 2020

An Example of Hash Table Based Mappers

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

nature
genetics

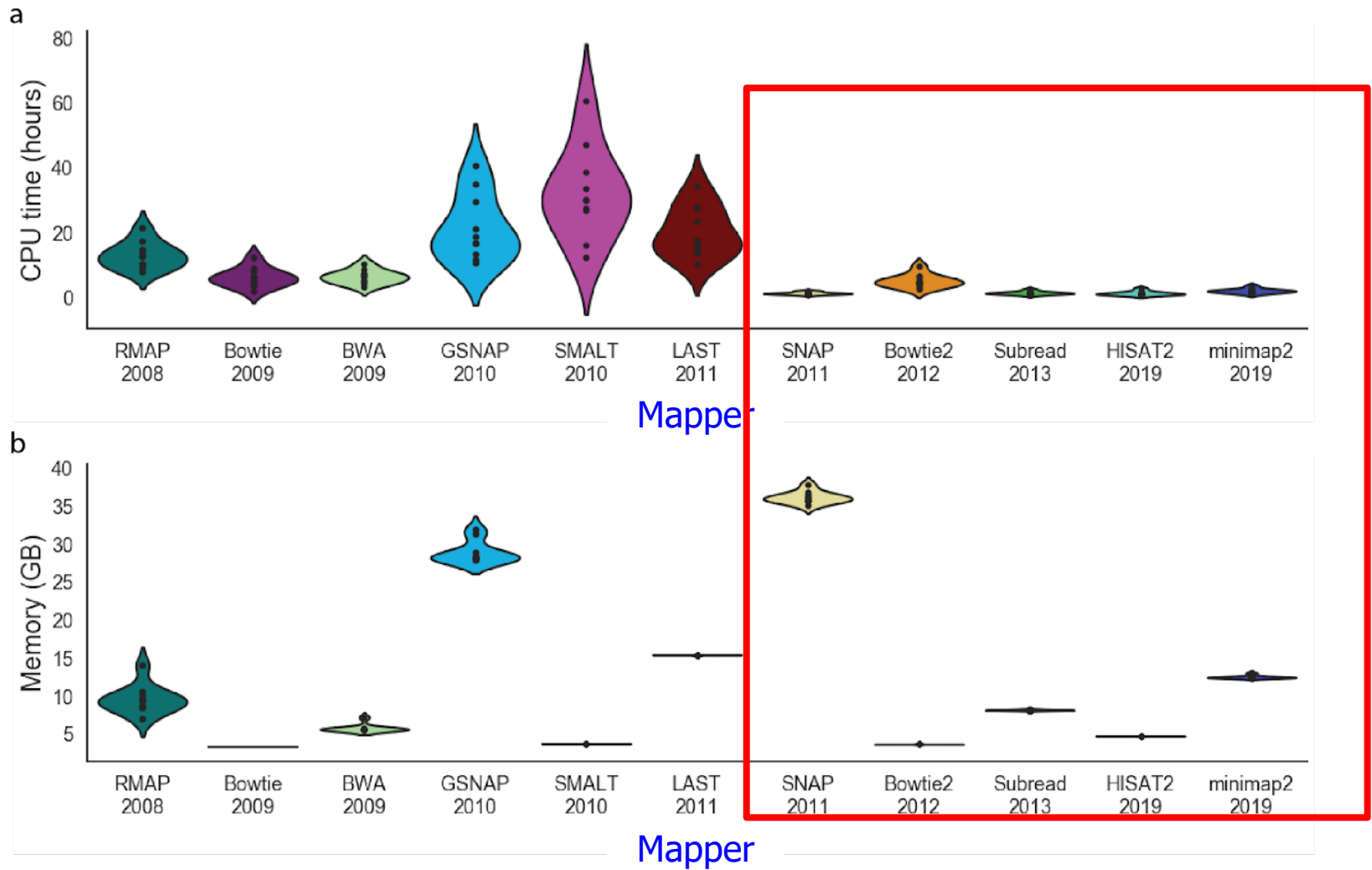
<https://github.com/BilkentCompGen/mrfast>

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, Fereydoon Hormozdiari⁴, Jacob O Kitzman¹, Carl Baker¹, Maika Malig¹, Onur Mutlu⁵, S Cenk Sahinalp⁴, Richard A Gibbs⁶ & Evan E Eichler^{1,2}

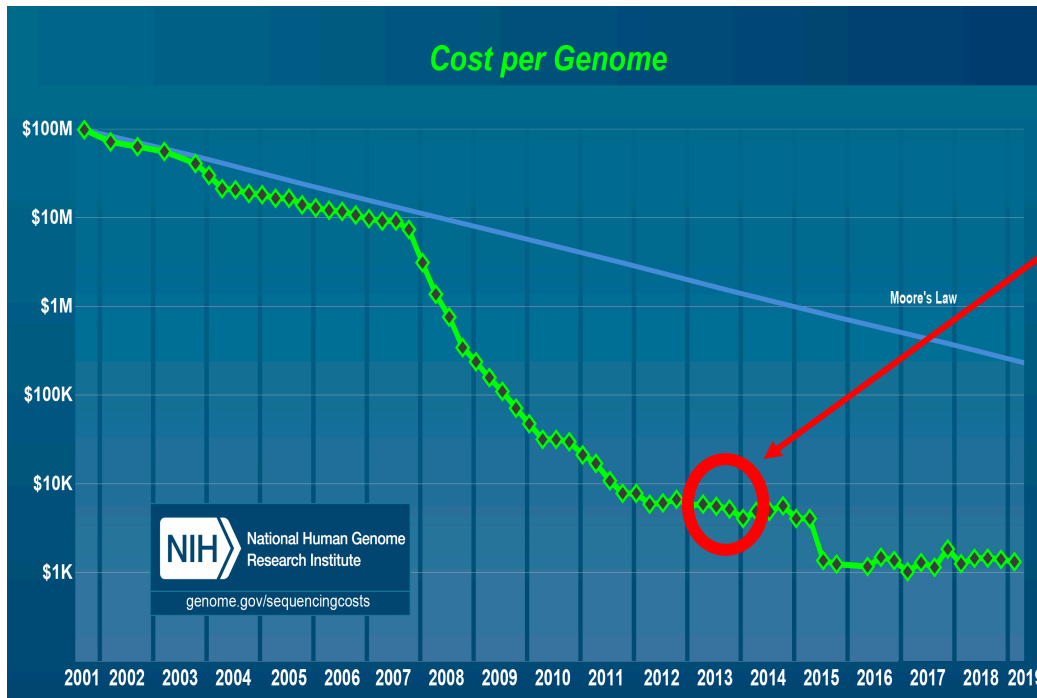
Alkan+, "[Personalized copy number and segmental duplication maps using next-generation sequencing](#)", Nature Genetics 2009.

Performance of Read Mapping

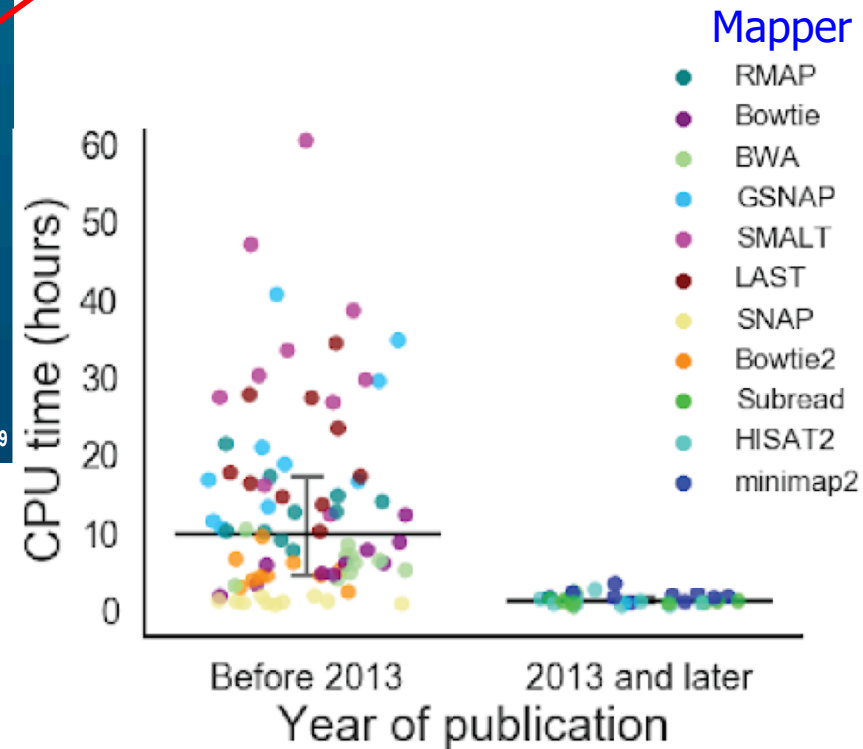


Alser+, "[Technology dictates algorithms: Recent developments in read alignment](#)", arXiv, 2020

The Need for Speed



Did we realize the **need** for **faster** genome analysis?



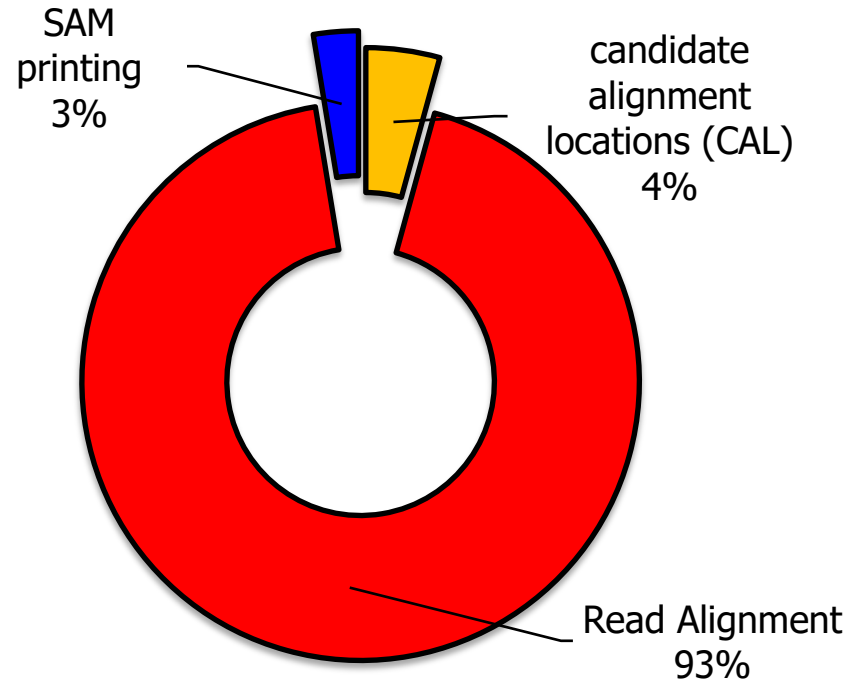
Alser+, "Technology dictates algorithms: Recent developments in read alignment", arXiv, 2020

What Makes Read Mapper **Slow**?

What Makes Read Mapper Slow?

Key Observation # 1

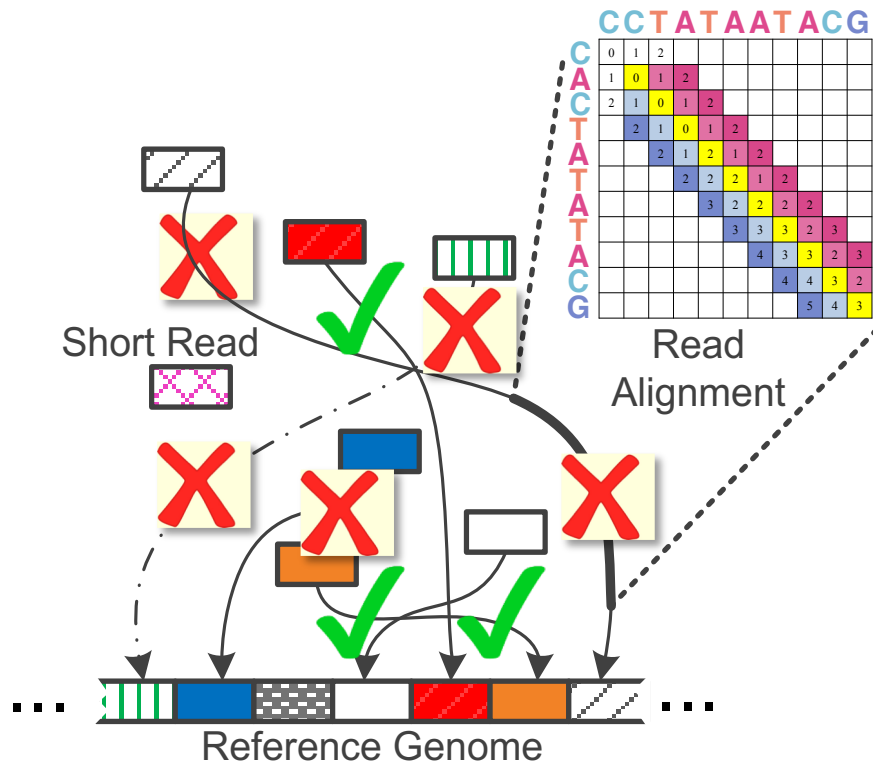
93%
of the read mapper's
execution time is spent
in read alignment.



Alser et al, Bioinformatics (2017)

What Makes Read Mapper Slow? (cont'd)

Key Observation # 2



98%
of candidate locations
have high dissimilarity
with a given read.

Cheng *et al*, *BMC bioinformatics* (2015)
Xin *et al*, *BMC genomics* (2013)

What Makes Read Mapper Slow? (cont'd)

Key Observation # 3

- **Quadratic-time** dynamic-programming algorithm **WHY?!**

Enumerating all possible prefixes

- NETHERLANDS x SWITZERLAND
NETHERLANDS x S
NETHERLANDS x SW
NETHERLANDS x SWI
NETHERLANDS x SWIT
NETHERLANDS x SWITZ
NETHERLANDS x SWITZE
NETHERLANDS x SWITZER
NETHERLANDS x SWITZERL
NETHERLANDS x SWITZERLA
NETHERLANDS x SWITZERLAN
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	1	2	3	4	5	6	7	8	9	10	11	
I	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	

What Makes Read Mapper Slow? (cont'd)

Key Observation # 3

- **Quadratic-time** dynamic-programming algorithm

Enumerating all possible prefixes

- **Data dependencies** limit the computation parallelism

Processing row (or column) after another

- **Entire matrix** is computed even though strings can be dissimilar.

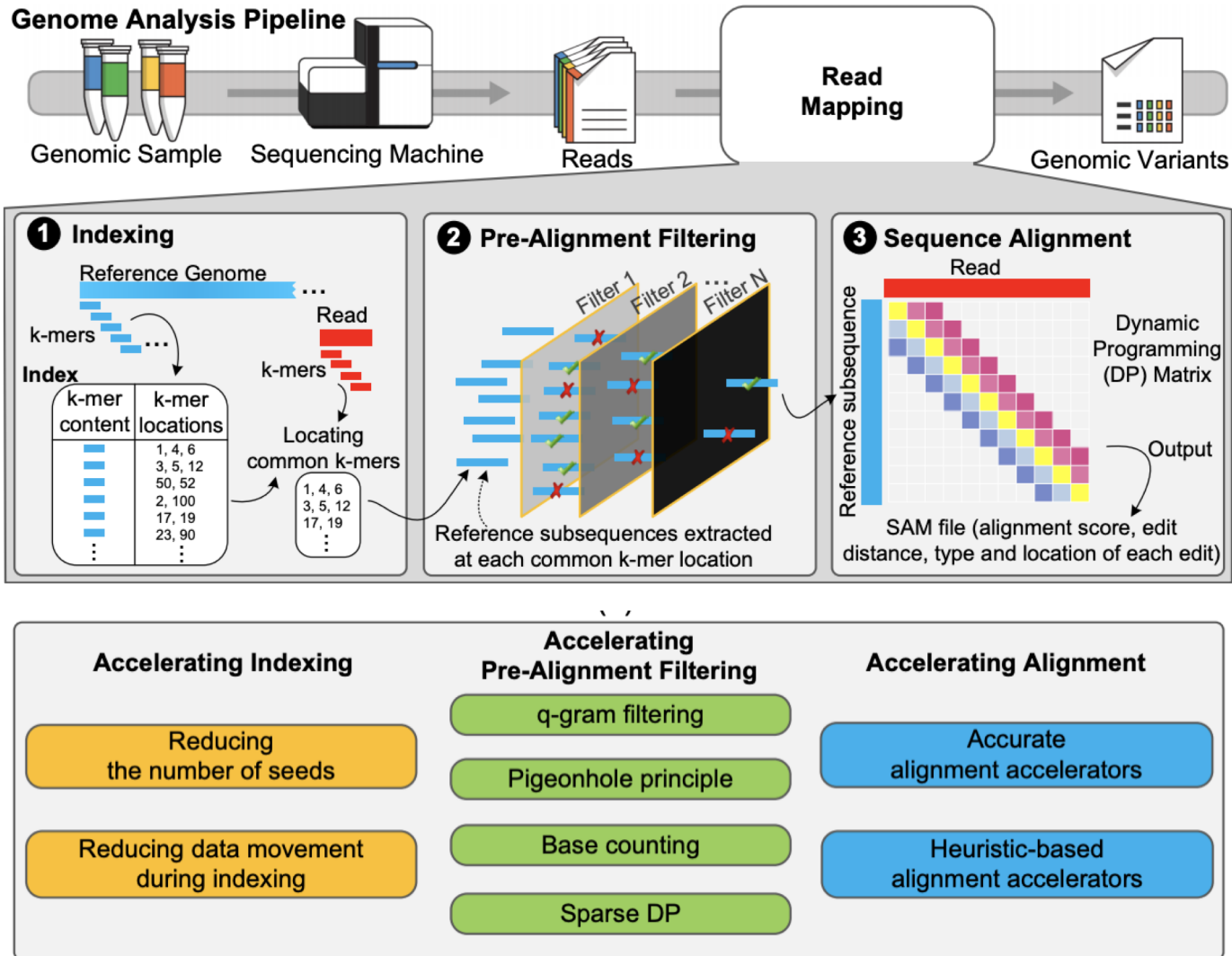
Number of differences is computed only at the backtraking step.

		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

Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- What Makes Read Mapper Slow?
- **Algorithmic & Hardware Acceleration**
 - Seed Filtering Technique
 - Pre-alignment Filtering Technique
 - Read Alignment Acceleration
- Where is Read Mapping Going Next?

Accelerating Read Mapping



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

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

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

Ongoing Directions

■ **Seed Filtering Technique:**

- **Goal:** Reducing the number of seed (k-mer) locations.
 - **Heuristic** (limits the number of mapping locations for each seed).
 - Supports **exact** matches only.

■ **Pre-alignment Filtering Technique:**

- **Goal:** Reducing the number of *invalid mappings* ($>E$).
 - Supports both **exact and inexact** matches.
 - Provides some **falsely-accepted** mappings.

■ **Read Alignment Acceleration:**

- **Goal:** Performing read alignment at scale.
 - Limits the **numeric range** of each cell in the DP table and hence supports **limited scoring** function.
 - May not support **backtracking** step due to random memory accesses.

Ongoing Directions

■ **Seed Filtering Technique:**

- **Goal:** Reducing the number of seed (k-mer) locations.
 - **Heuristic** (limits the number of mapping locations for each seed).
 - Supports **exact** matches only.

■ **Pre-alignment Filtering Technique:**

- **Goal:** Reducing the number of *invalid mappings* ($>E$).
 - Supports both **exact and inexact** matches.
 - Provides some **falsely-accepted** mappings.

■ **Read Alignment Acceleration:**

- **Goal:** Performing read alignment at scale.
 - Limits the **numeric range** of each cell in the DP table and hence supports **limited scoring** function.
 - May not support **backtracking** step due to random memory accesses.

FastHASH

- **Goal:** Reducing the number of seed (k-mer) locations.
 - **Heuristic** (limits the number of mapping locations for each seed).
 - Supports **exact** matches only.

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

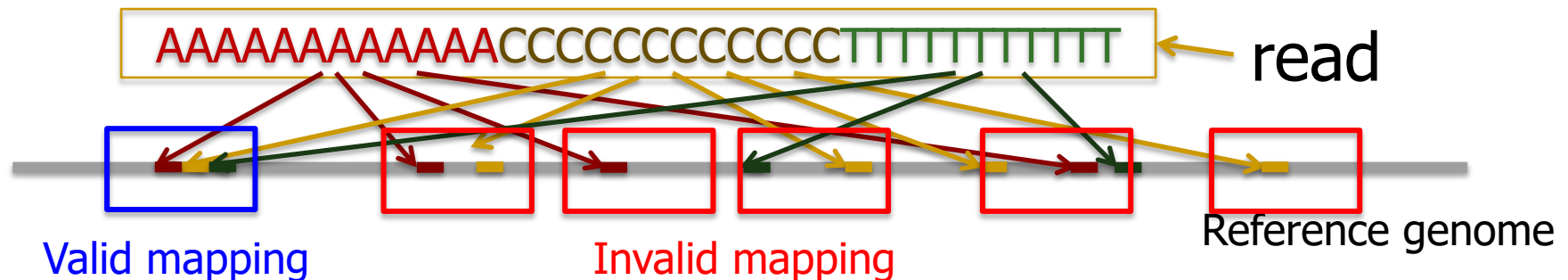
Hongyi Xin¹, Donghyuk Lee¹, Farhad Hormozdiari², Samihan Yedkar¹, Onur Mutlu^{1*}, Can Alkan^{3*}

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

Key Observations

■ Observation 1 (Adjacent k-mers)

- ❑ **Key insight:** Adjacent k-mers in the read should also be adjacent in the reference genome
- ❑ **Key idea:** 1) sort the location list based on their number of locations and 2) search for adjacent locations in the k-mers' location lists



Key Observations

■ Observation 1 (Adjacent k-mers)

- ❑ **Key insight:** Adjacent k-mers in the read should also be adjacent in the reference genome
- ❑ **Key idea:** 1) sort the location list based on their number of locations and 2) search for adjacent locations in the k-mers' location lists

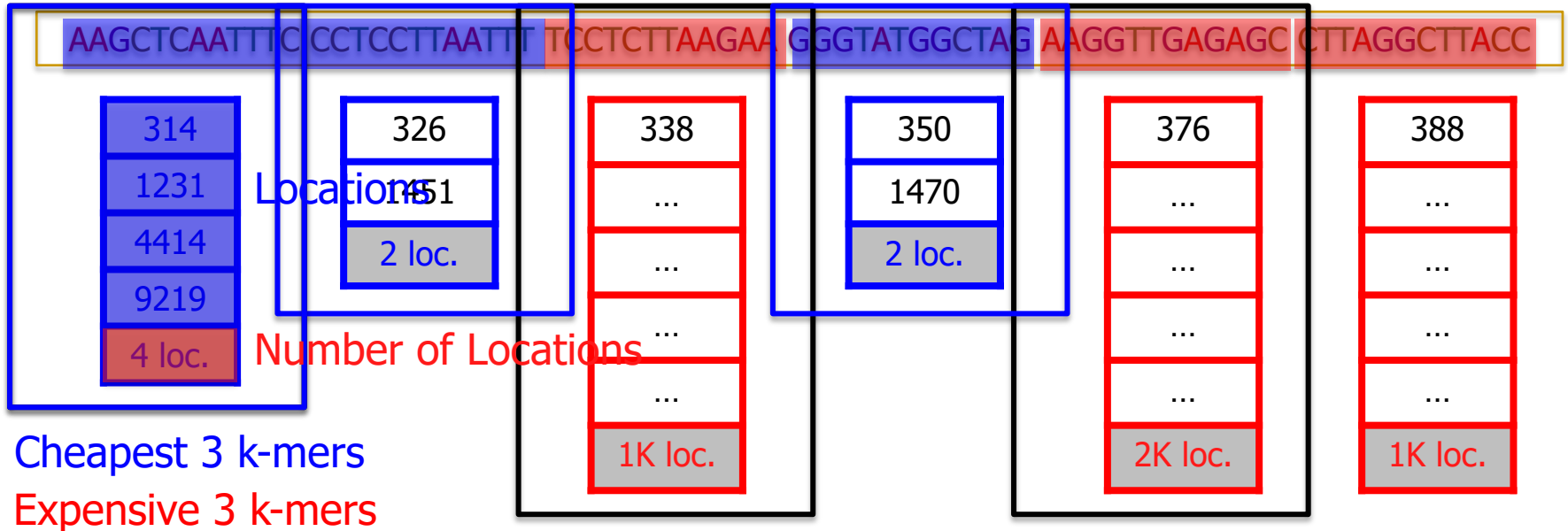
■ Observation 2 (Cheap k-mers)

- ❑ **Key insight:** Some k-mers are cheaper to verify than others because they have shorter location lists (they occur less frequently in the reference genome)
- ❑ **Key Idea:** Read mapper can choose the cheapest k-mers and verify their locations

Cheap K-mer Selection

- occurrence threshold = 500

read



Previous work needs to verify:

3004 locations



FastHASH verifies only:

8 locations

FastHASH Conclusion

- **Problem:** Existing **read mappers** perform **poorly** in mapping billions of short reads to the reference genome, in the presence of errors
- **Observation:** Most of the **verification** calculations are unnecessary → filter them out
- **Key Idea:** To reduce the cost of unnecessary verification
 - ❑ Select **Cheap** and **Adjacent** k-mers.
- **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

Ongoing Directions

■ **Seed Filtering Technique:**

- **Goal:** Reducing the number of seed (k-mer) locations.
 - **Heuristic** (limits the number of mapping locations for each seed).
 - Supports **exact** matches only.

■ **Pre-alignment Filtering Technique:**

- **Goal:** Reducing the number of *invalid mappings* ($>E$).
 - Supports both **exact and inexact** matches.
 - Provides some **falsely-accepted** mappings.

■ **Read Alignment Acceleration:**

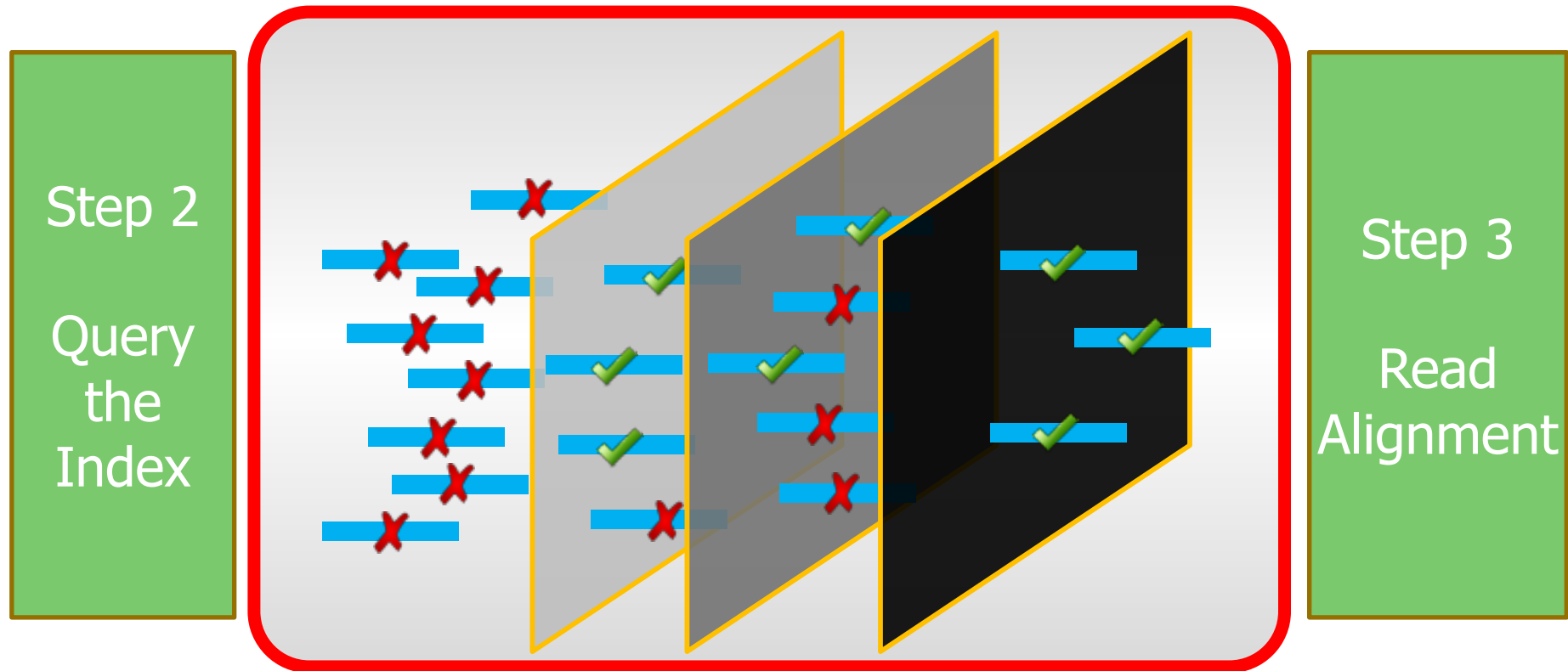
- **Goal:** Performing read alignment at scale.
 - Limits the **numeric range** of each cell in the DP table and hence supports **limited scoring** function.
 - May not support **backtracking** step due to random memory accesses.

Pre-alignment Filtering Technique

Read Alignment is expensive

Our goal is to reduce the need for dynamic programming algorithms

Ideal Filtering Algorithm



1. **Filter out** most of incorrect mappings.
2. **Preserve** all correct mappings.
3. Do it **quickly**.

Bioinformatics



Article Navigation

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

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

Bioinformatics, Volume 33, Issue 21, 01 November 2017, Pages 3355–3363,

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

Published: 31 May 2017 **Article history** ▼

Alser+, "GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping", *Bioinformatics*, 2017.

GateKeeper

■ Key observation:

- If two strings differ by E edits, then every bp match can be aligned in at most $2E$ shifts.

■ Key idea:

- Compute “Shifted Hamming Distance”: AND of $2E+1$ Hamming vectors of two strings, to identify invalid mappings
 - Uses *bit-parallel operations* that nicely map to FPGA architectures

■ Key result:

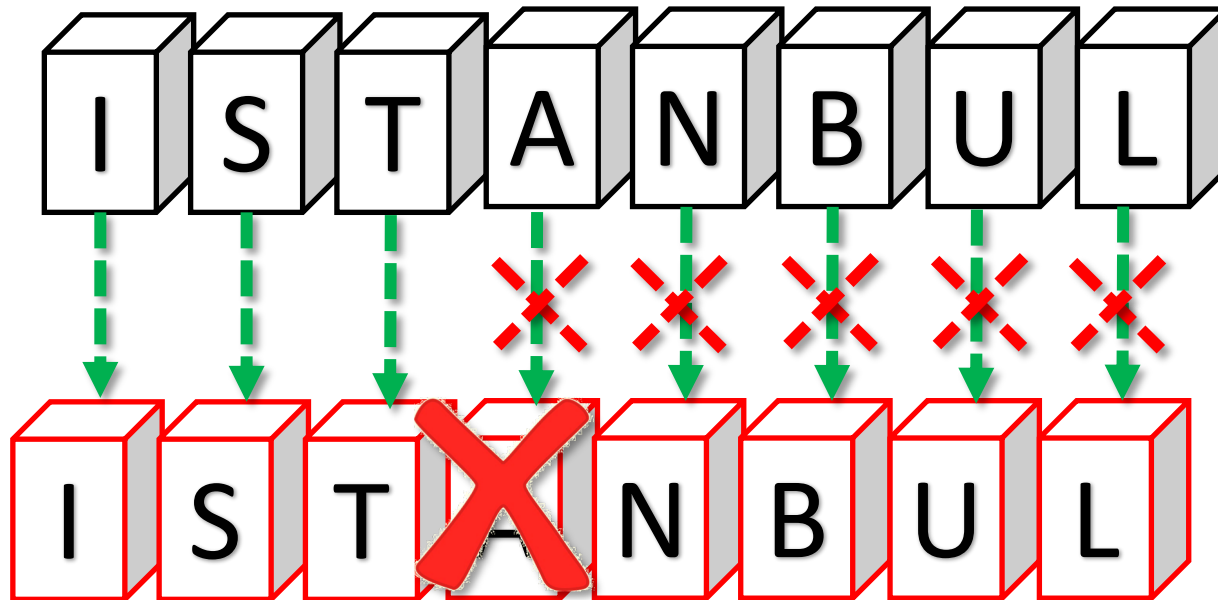
- GateKeeper is 90x-130x faster than SHD (Xin et al., 2015) and the Adjacency Filter (Xin et al., 2013), with only a 7% false positive rate
- The addition of GateKeeper to the mrFAST mapper (Alkan et al., 2009) results in 10x end-to-end speedup in read mapping

Hamming Distance ($\Sigma \oplus$)

3 matches

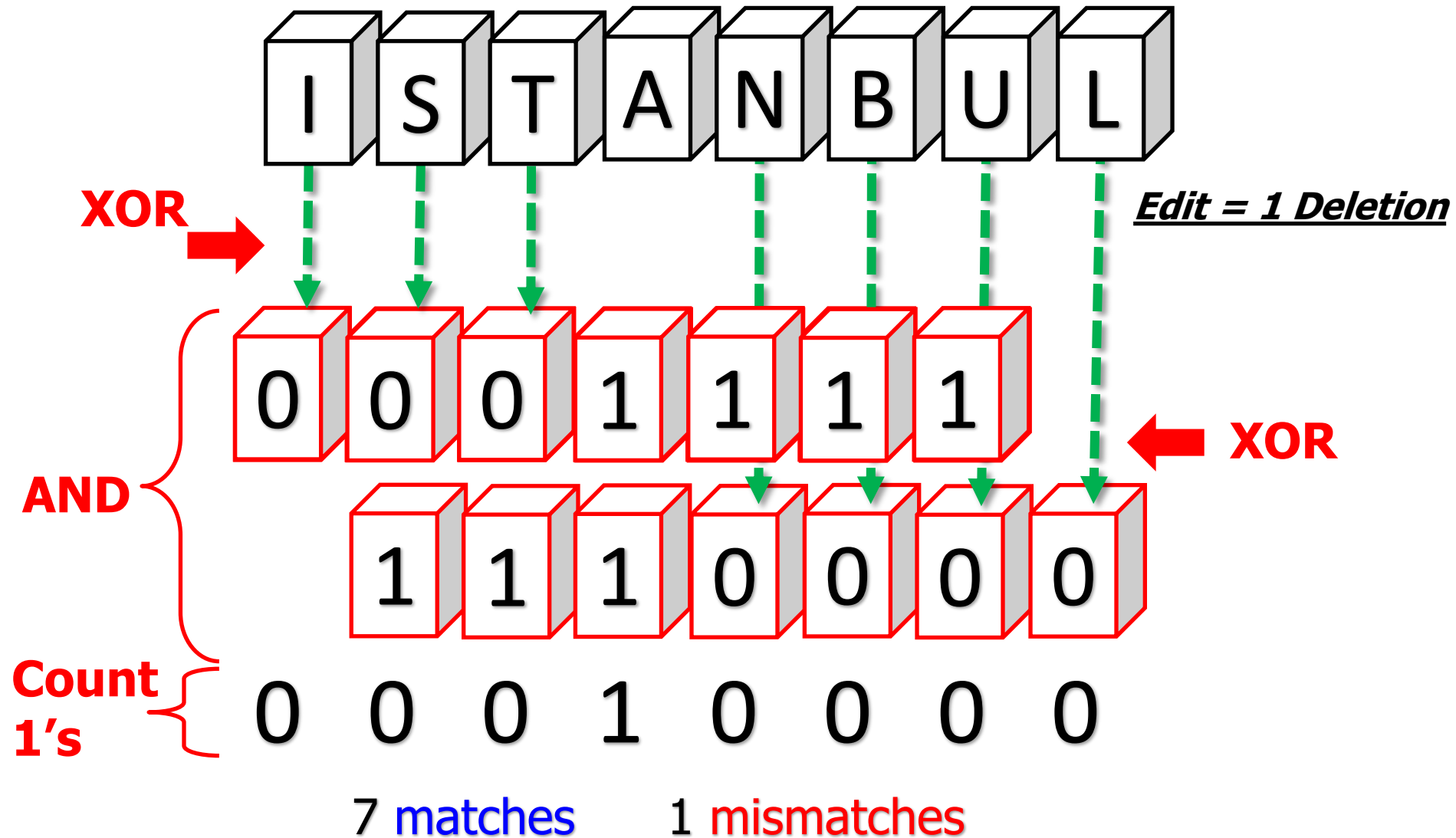
5 mismatches

Edit = 1 Deletion



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

Shifted Hamming Distance (Xin+ 2015)



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 :GAGAGAGATAGTTAGTGTTCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGGCCGG

Hamming Mask : 00000000001000000000000011111111011110001110110101101111111100010000101110110110010101

[illegible]

2-Deletion Mask :000000000101101110011111111111111011110001110110101101111111111000100100111101101001010

3-Deletion Mask :1111111111110111011001101110111011100010010011111111111111001011001101011011101101111

```

-Insertion Mask :1111111111101111101111110111101100010010011111111111111110010110011000 01011110111011111110

```

2-Insertion Mask :00000010011111001111111111001000110101010011010111111111111110111001 11 111000111101100

3-Insertion Mask :1111111101110110011000111111111010110111111001100101110111111110110111010111001000

--- Masks after amendment ---

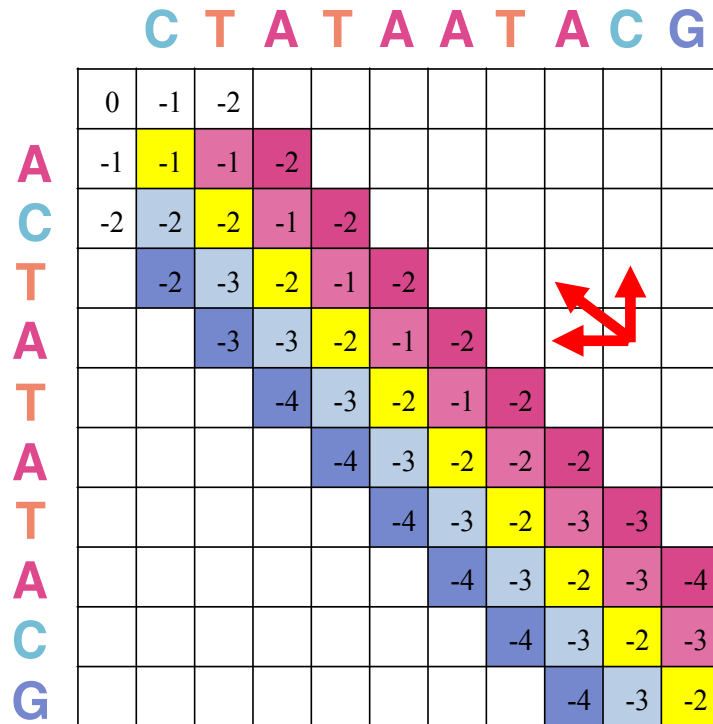
Our goal to track the diagonally consecutive matches in the neighborhood map.

.GAGAGAGATATTTAGTGTTGCAG-CACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGGAACATTGTTGGGCCGG

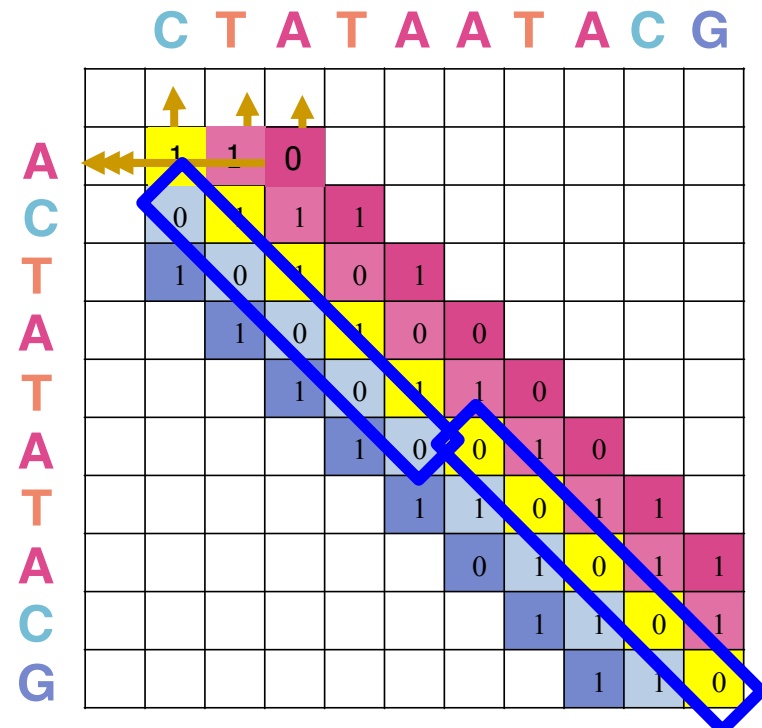
.GAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGGCCCG

Alignment Matrix vs. Neighborhood Map

Needleman-Wunsch

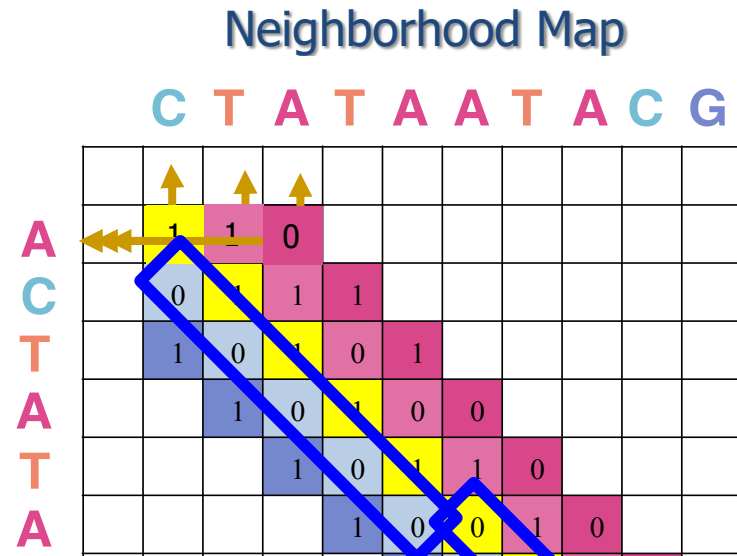
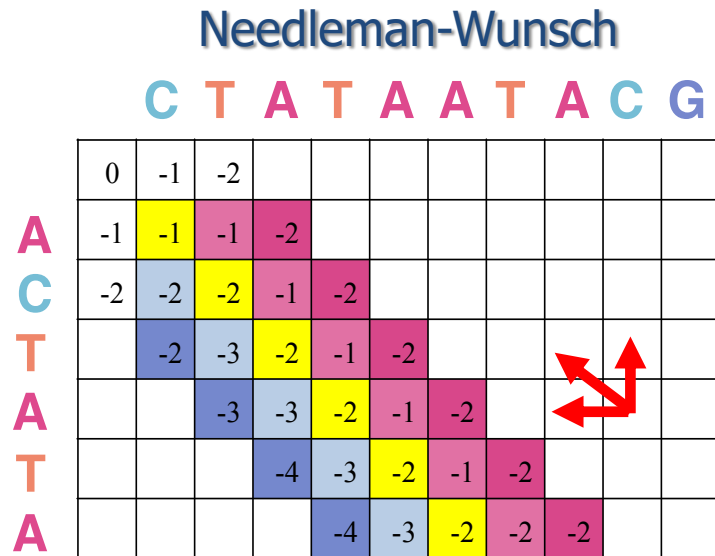


Neighborhood Map

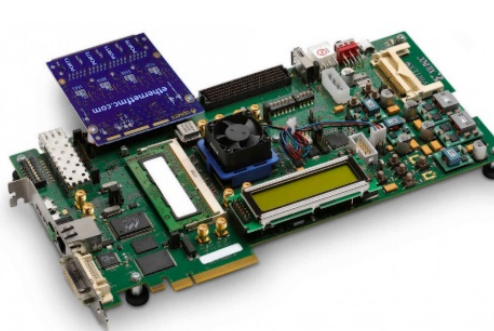


Our goal to track the diagonally consecutive matches in the neighborhood map.

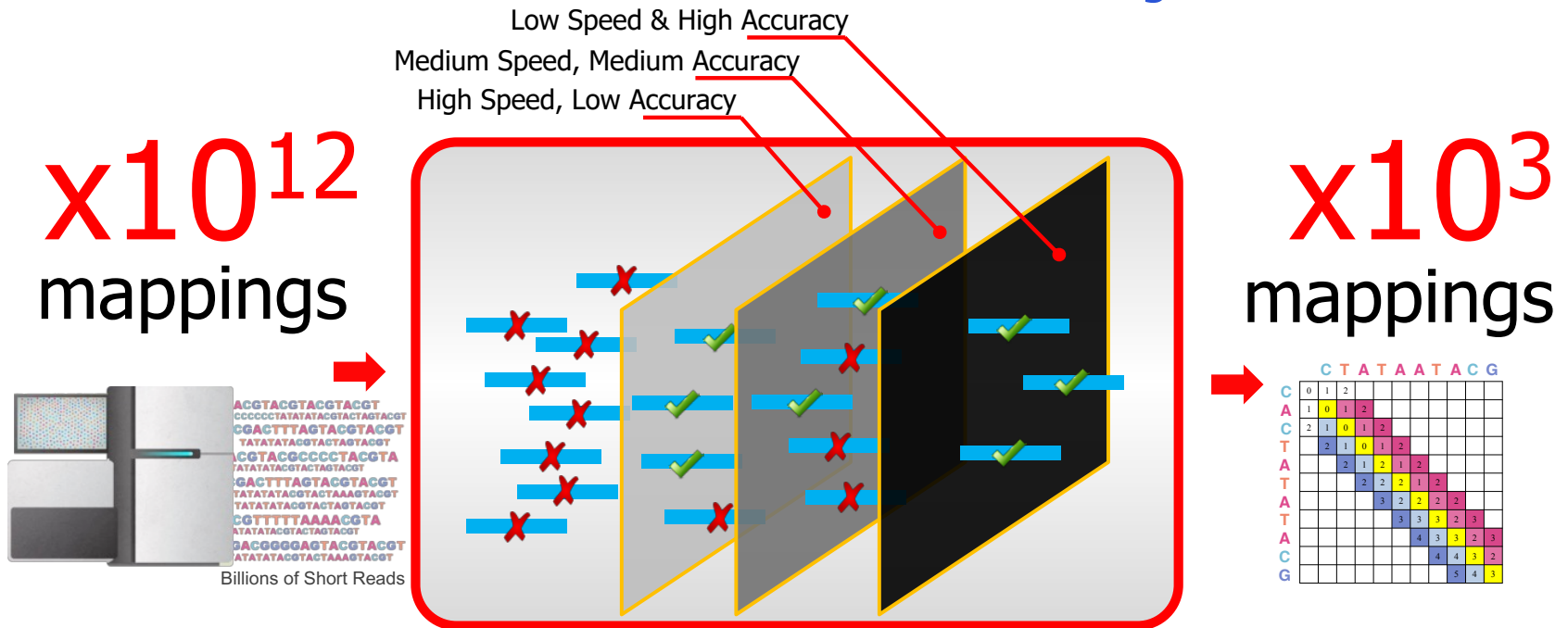
Alignment Matrix vs. Neighborhood Map



Independent vectors can be processed in parallel using hardware technologies



Our Solution: GateKeeper



- 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 Walkthrough (cont'd)

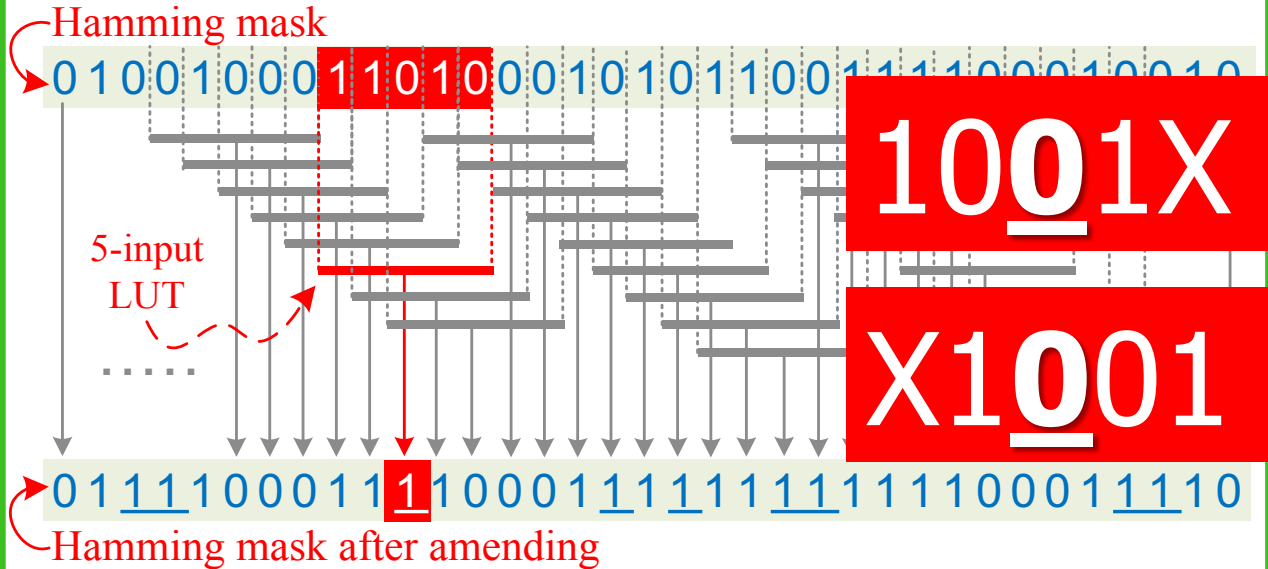
Generate $2E+1$ masks

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.

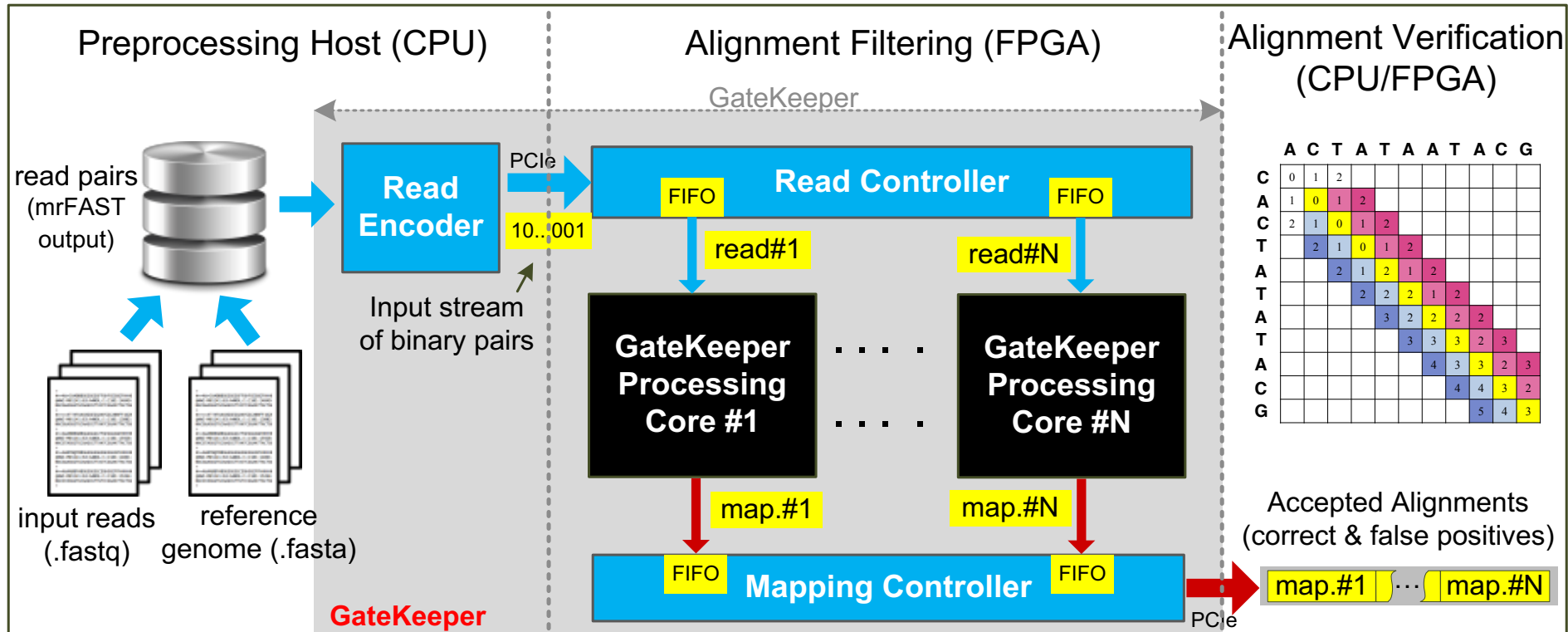
- $(2E) * (\text{ReadLength})$ 2-AND operations.
- $(\text{ReadLength}/4)$ 5-input LUT.
- $\log_2 \text{ReadLength}$ -bit counter.



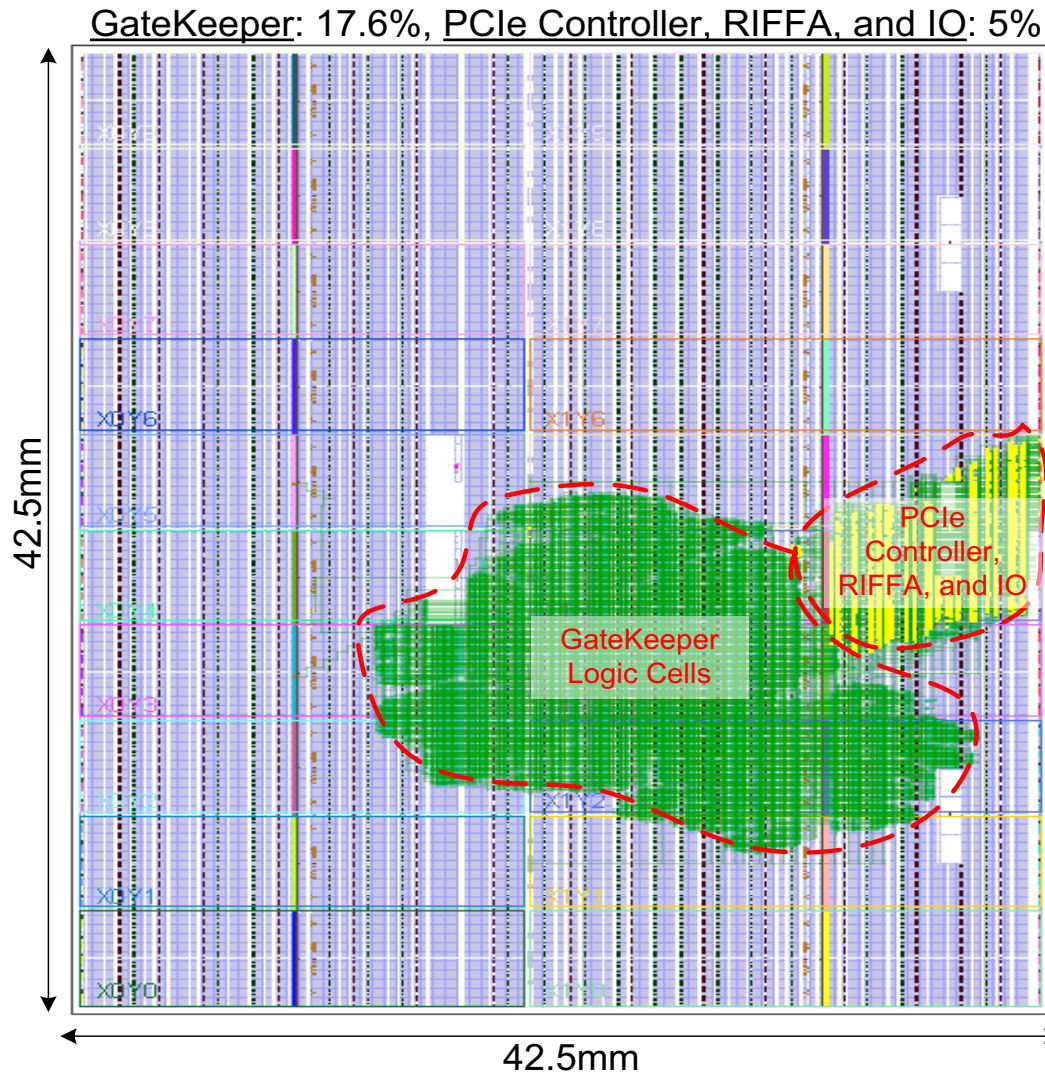
- $(2E+1) * (\text{ReadLength})$ 5-input LUT.

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



300 bp

E=15

GateKeeper: Speed & Accuracy Results

90x-130x faster filter

than SHD (Xin et al., 2015) and the Adjacency Filter (Xin et al., 2013)

4x lower false accept rate

than the Adjacency Filter (Xin et al., 2013)

10x speedup in read mapping

with the addition of GateKeeper to the mrFAST mapper (Alkan et al., 2009)

Freely available online

github.com/BilkentCompGen/GateKeeper

GateKeeper Conclusions

- FPGA-based pre-alignment greatly speeds up read mapping
 - 10x speedup of a state-of-the-art mapper (mrFAST)
- FPGA-based pre-alignment can be integrated with the sequencer
 - It can help to hide the complexity and details of the FPGA
 - Enables real-time filtering while sequencing

More on SHD (SIMD Implementation)

- Download and test for yourself
- <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

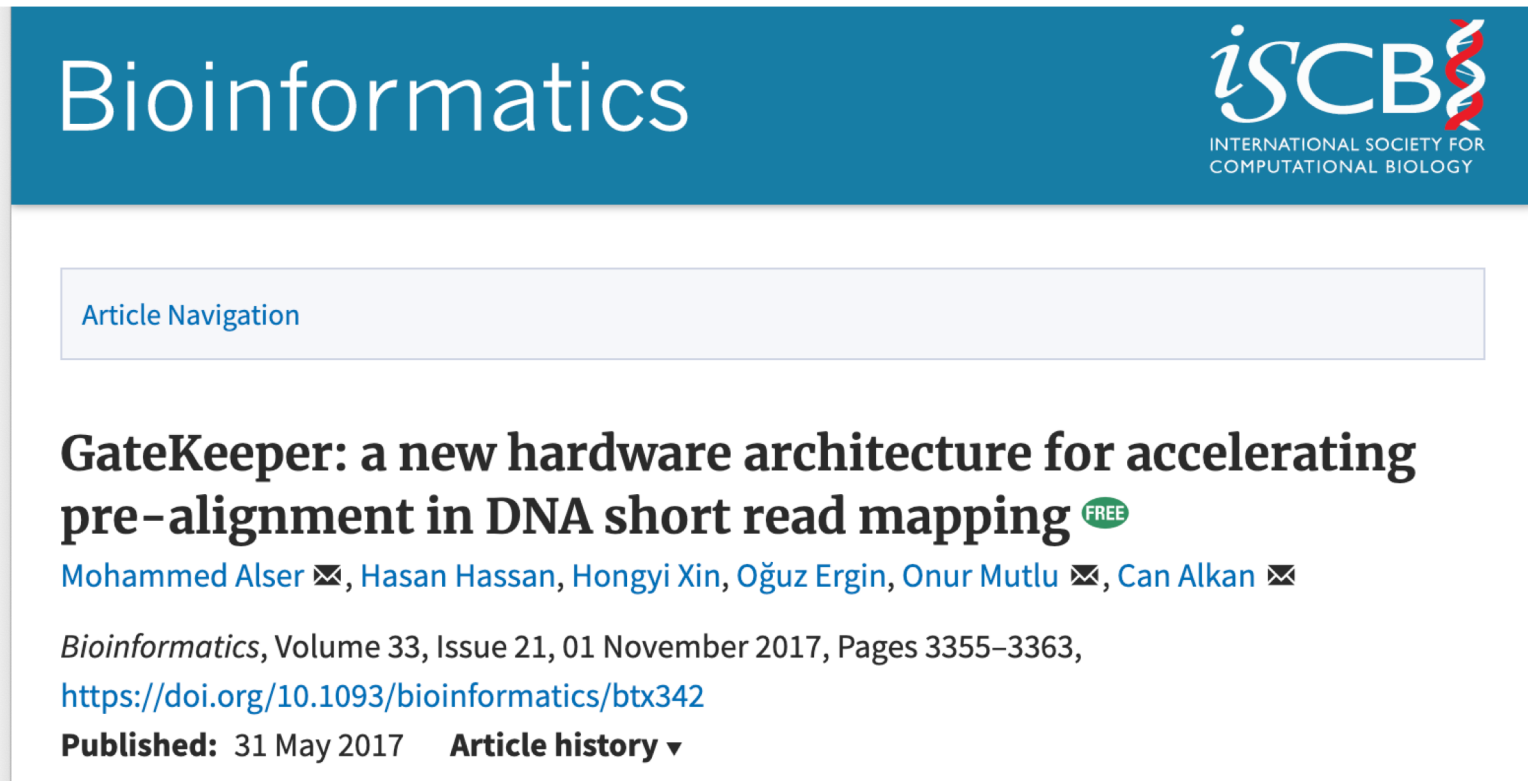
Shifted Hamming distance: a fast and accurate SIMD-friendly filter to accelerate alignment verification in read mapping

**Hongyi Xin^{1,*}, John Greth², John Emmons², Gennady Pekhimenko¹,
Carl Kingsford³, Can Alkan^{4,*} and Onur Mutlu^{2,*}**

More on GateKeeper

- Download and test for yourself

<https://github.com/BilkentCompGen/GateKeeper>



The screenshot shows the top section of a Bioinformatics article page. At the top, there is a blue header bar with the word "Bioinformatics" in white on the left and the "iSCB" logo (International Society for Computational Biology) on the right. Below the header, there is a light blue box labeled "Article Navigation". The main title of the article is "GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping", with a green "FREE" badge next to it. Below the title, the authors are listed: "Mohammed Alser", "Hasan Hassan", "Hongyi Xin", "Oğuz Ergin", "Onur Mutlu", and "Can Alkan", each followed by an email icon. Below the authors, the journal information is given: "Bioinformatics, Volume 33, Issue 21, 01 November 2017, Pages 3355–3363," followed by the DOI link "https://doi.org/10.1093/bioinformatics/btx342". At the bottom of the article preview, it says "Published: 31 May 2017" and "Article history" with a dropdown arrow.

Bioinformatics

iSCB
INTERNATIONAL SOCIETY FOR
COMPUTATIONAL BIOLOGY

Article Navigation

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

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

Bioinformatics, Volume 33, Issue 21, 01 November 2017, Pages 3355–3363,
<https://doi.org/10.1093/bioinformatics/btx342>

Published: 31 May 2017 **Article history** ▼

Alser+, "[GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping](#)", *Bioinformatics*, 2017.

Can we do better? Scalability?

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

Alser+, "[Shouji: a fast and efficient pre-alignment filter for sequence alignment](https://doi.org/10.1093/bioinformatics/btz234)", *Bioinformatics* 2019, <https://doi.org/10.1093/bioinformatics/btz234>

Shouji

■ **Key observation:**

- ❑ Correct alignment always includes **long identical subsequences**.
- ❑ Processing the entire mapping at once is ineffective for hardware design.

■ **Key idea:**

- ❑ Use **overlapping sliding window** approach to quickly and accurately find all long segments of **consecutive zeros**.

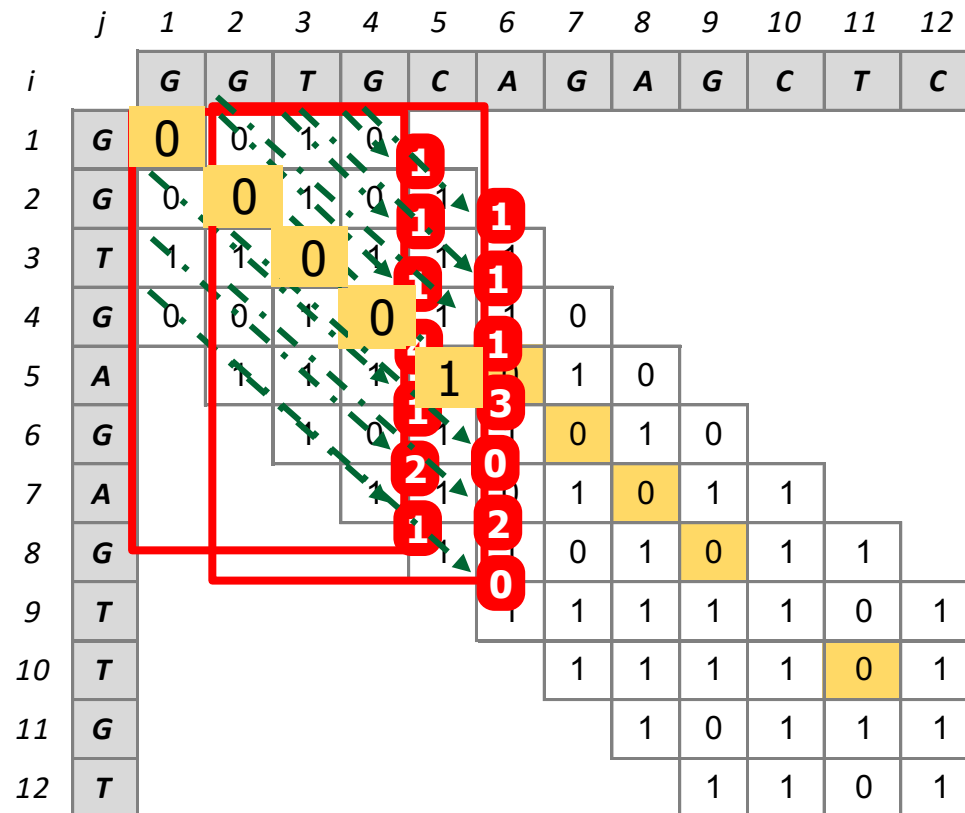
■ **Key result:**

- ❑ Shouji on FPGA is **up to three orders of magnitude faster** than its CPU implementation.
- ❑ Shouji accelerates **best-performing CPU read aligner Edlib** (Bioinformatics 2017) by **up to 18.8x** using 16 filtering units that work in parallel.
- ❑ Shouji is **2.4x to 467x more accurate** than GateKeeper (Bioinformatics 2017) and SHD (Bioinformatics 2015).

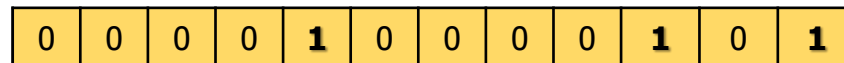
Shouji Walkthrough

Building the
Neighborhood Map

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



Storing it @ Shouji Bit-vector

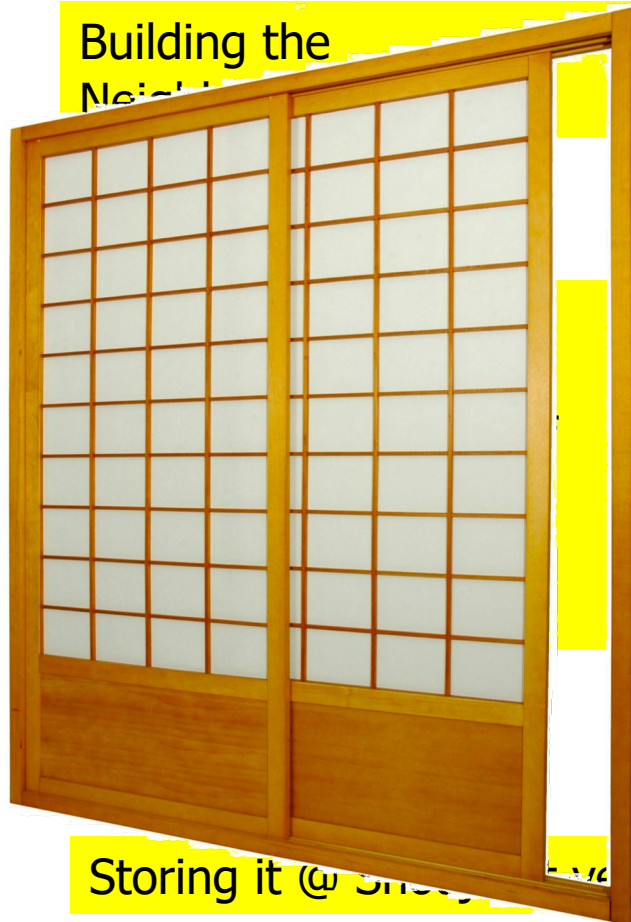


ACCEPT iff number of '1' \leq Threshold

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

Shouji Walkthrough

Building the
Neighbor



Storing it @ Shouji Vector

	<i>j</i>	1	2	3	4	5	6	7	8	9	10	11	12
<i>i</i>		G	G	T	G	C	A	G	A	G	C	T	C
1	G	0	0	1	0								
2	G	0	0	1	0	1							
3	T	1	1	0	1	1	1						
4	G	0	0	1	0	1	1	0					
5	A		1	1	1	1	0	1	0				
6	G			1	0	1	1	0	1	0			
7	A				1	1	0	1	0	1	1		
8	G					1	1	0	1	0	1	1	
9	T						1	1	1	1	1	0	1
10	T							1	1	1	1	0	1
11	G								1	0	1	1	1
12	T									1	1	0	1

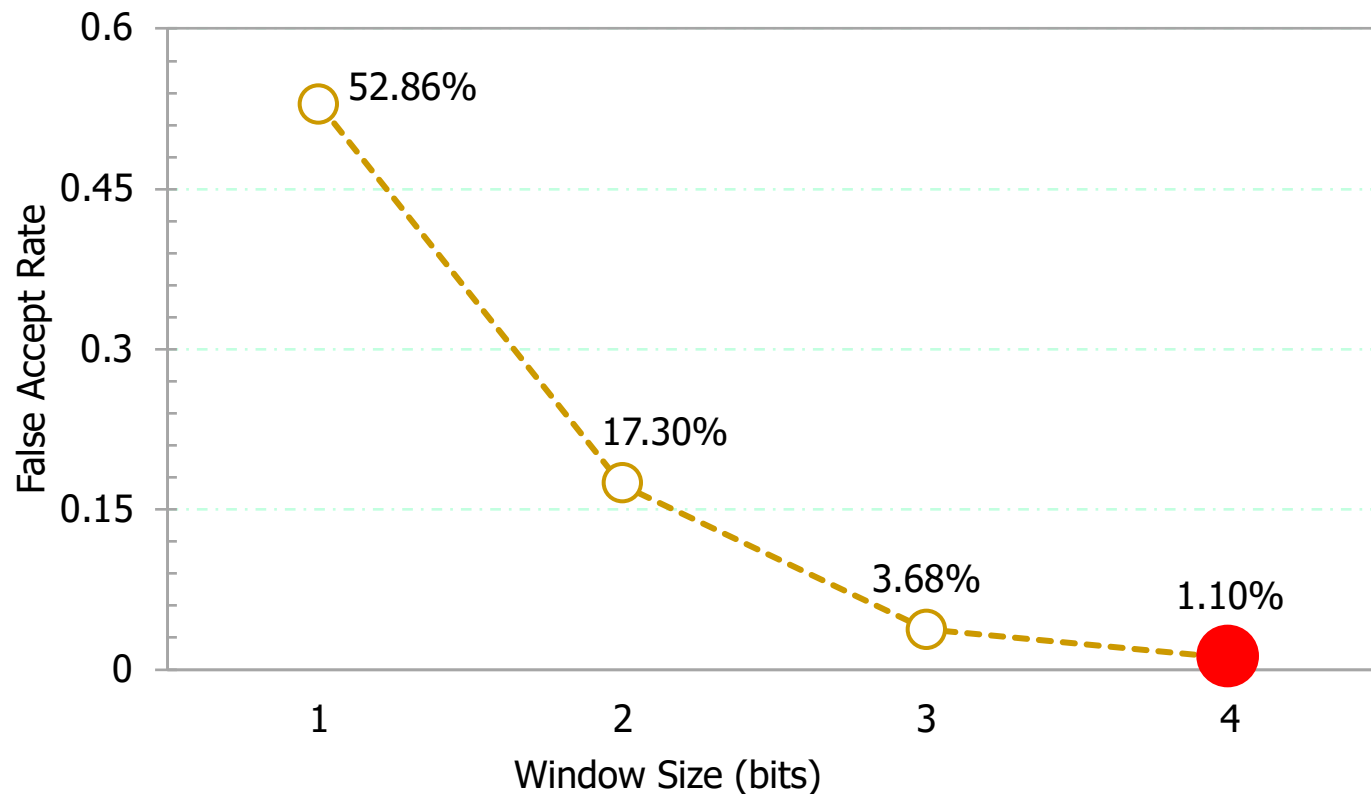
0 0 0 0 1 0 0 0 0 0 1 0 1

ACCEPT iff number of '1' ≤ Threshold

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

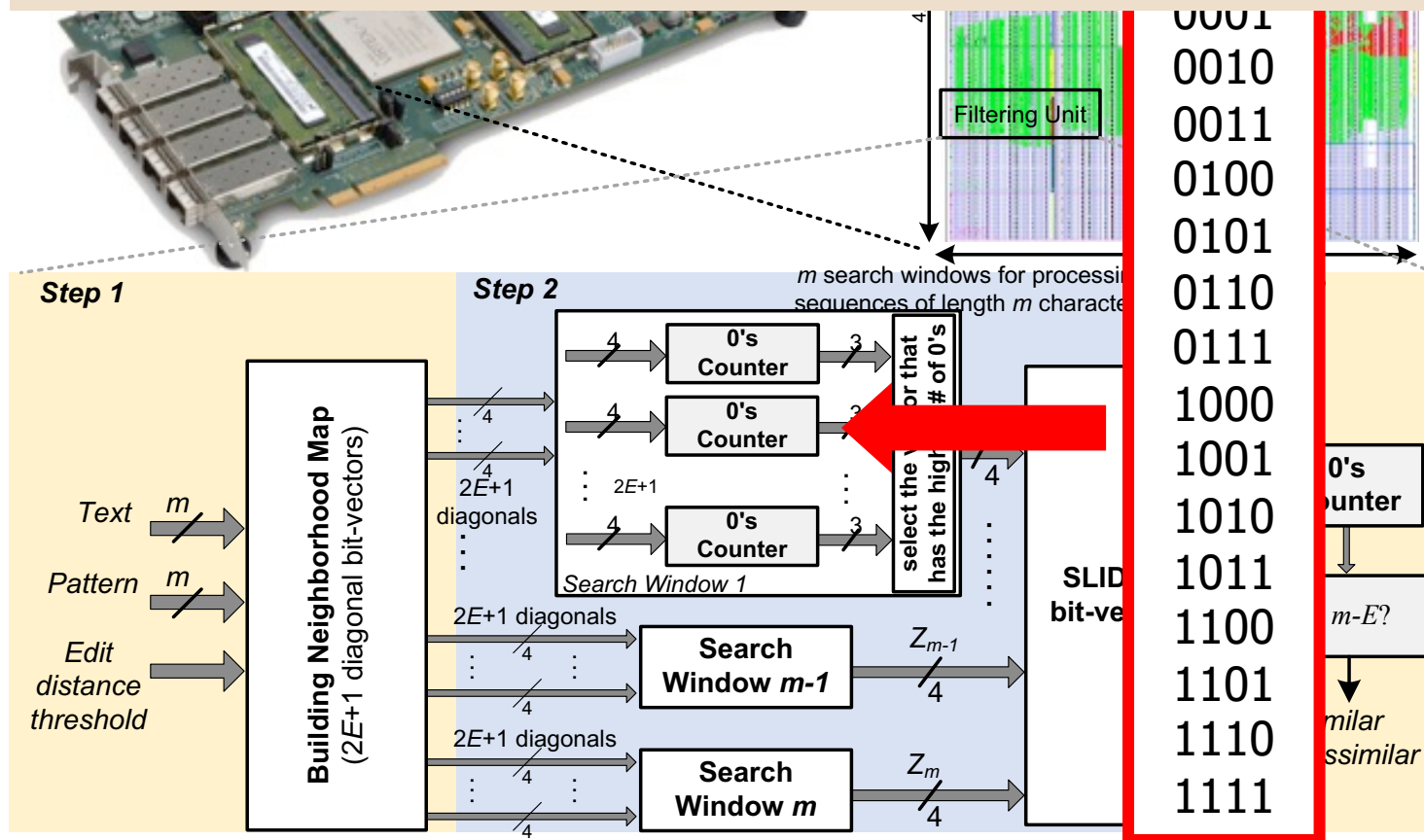
Sliding Window Size

- The reason behind the selection of the window size is due to the minimal possible length of the identical subsequence that is a single match (e.g., such as `101').



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

Download and test for yourself

<https://github.com/CMU-SAFARI/Shouji>

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

Alser+, "[Shouji: a fast and efficient pre-alignment filter for sequence alignment](https://doi.org/10.1093/bioinformatics/btz234)", *Bioinformatics* 2019, <https://doi.org/10.1093/bioinformatics/btz234>

SneakySnake

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

Mohammed Alser^{1,3}, Taha Shahroodi¹, Juan Gómez-Luna¹, Can Alkan³, and Onur Mutlu^{1,2,3}

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

²Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh 15213, PA, USA

³Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey

Alser + "[SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs.](#)" *arXiv preprint* (2019).

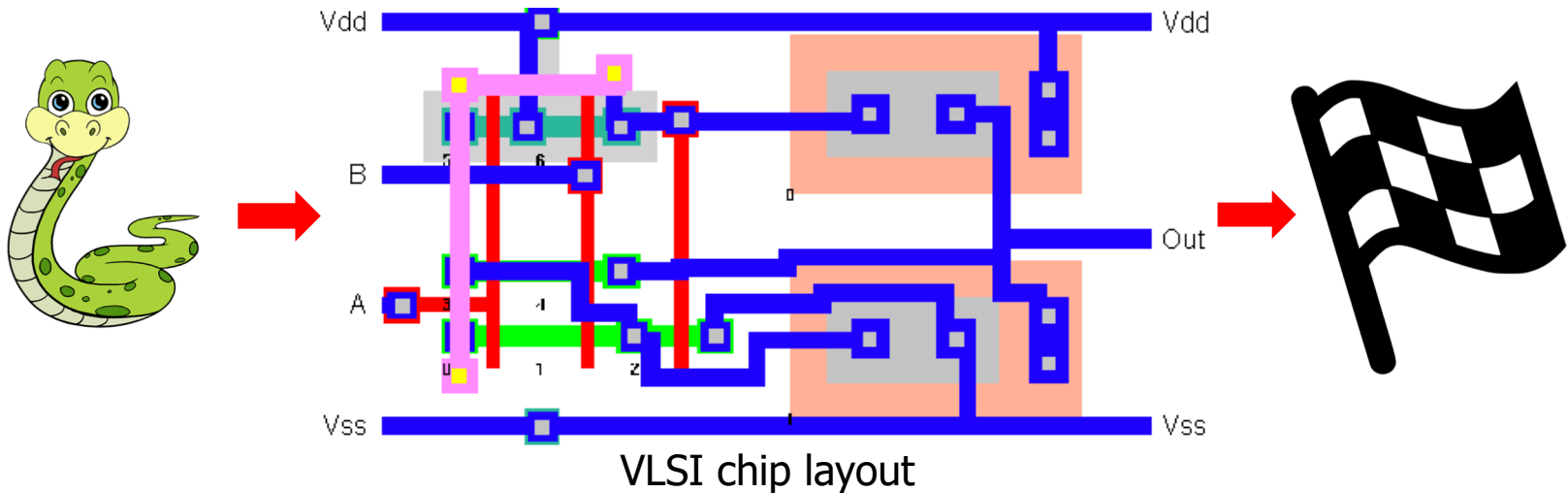
SneakySnake

■ Key observation:

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

■ Key idea:

- Approximate edit distance calculation is similar to Single Net Routing problem in VLSI chip.



SneakySnake

■ **Key observation:**

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

■ **Key idea:**

- Approximate edit distance calculation is similar to **Single Net Routing problem** in VLSI chip.

■ **Key result:**

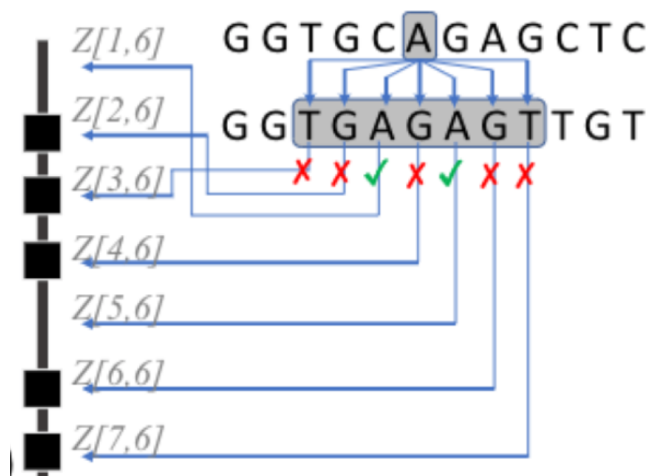
- SneakySnake is up to **four orders of magnitude more accurate** than Shouji (Bioinformatics'19) and GateKeeper (Bioinformatics'17).
- SneakySnake **accelerates** the state-of-the-art CPU-based sequence aligners, Edlib (Bioinformatics'17) and Parasail (BMC Bioinformatics'16), **by up to 37.6× and 43.9× (>12× on average)**, respectively, *without requiring hardware acceleration*, and by up to **413× and 689× (>400× on average)**, respectively, *using hardware acceleration*.

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
<i>3rd Upper Diagonal</i>												
<i>2nd Upper Diagonal</i>												
<i>1st Upper Diagonal</i>												
<i>Main Diagonal</i>												
<i>1st Lower Diagonal</i>												
<i>2nd Lower Diagonal</i>												
<i>3rd Lower Diagonal</i>												

ENTRANCE

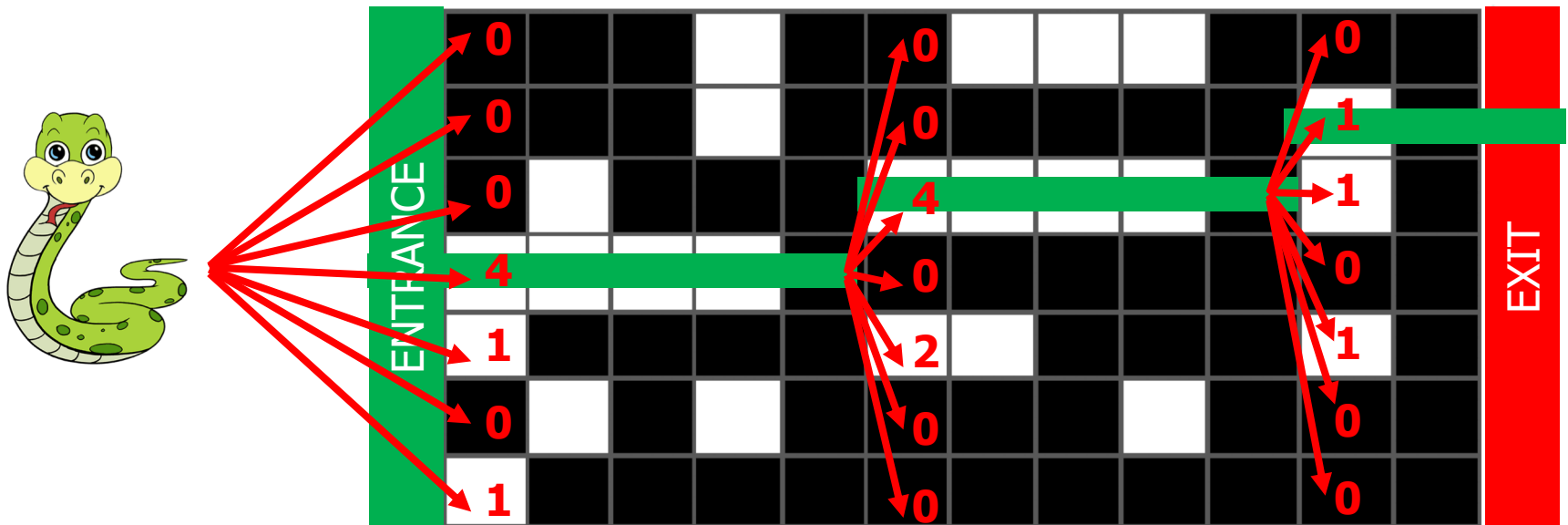
EXIT

SneakySnake Walkthrough

Building Neighborhood Map

Finding the Optimal Routing Path

Examining the Snake Survival



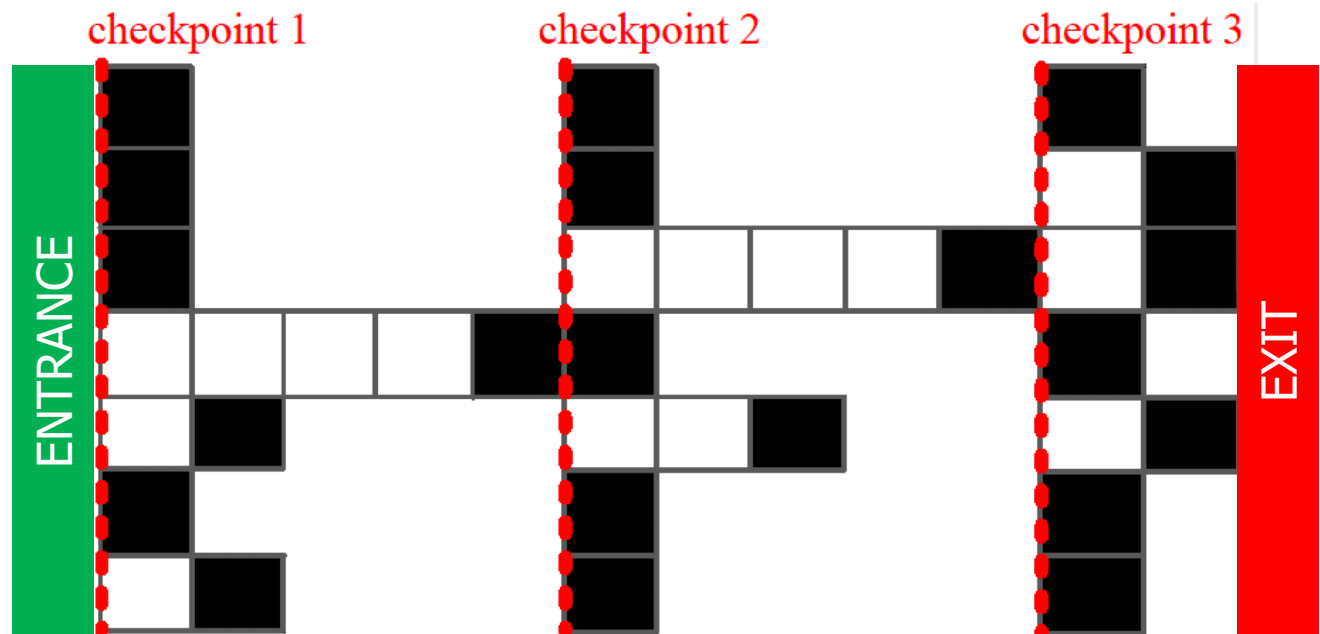
SneakySnake Walkthrough

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!**



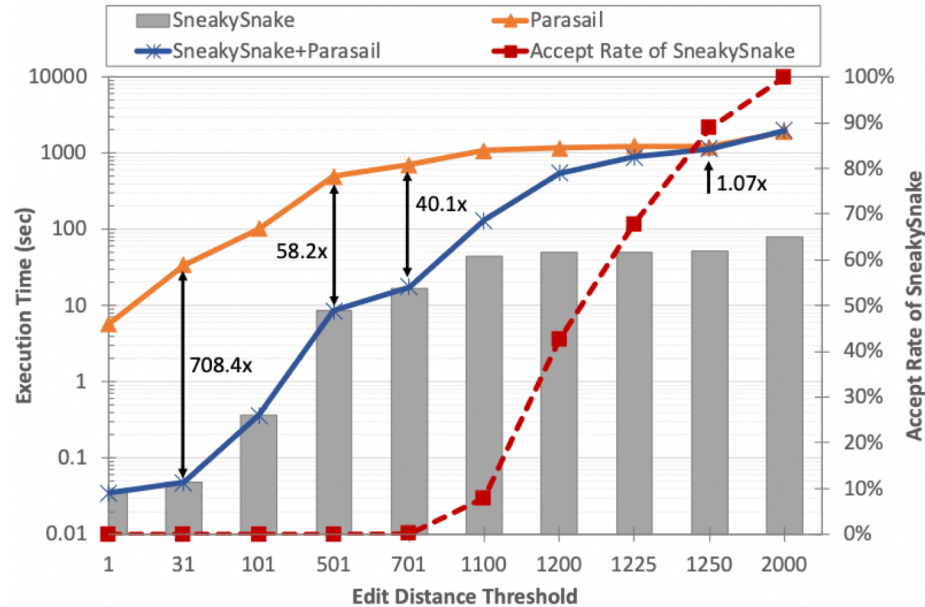
FPGA Resource Analysis

- 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).

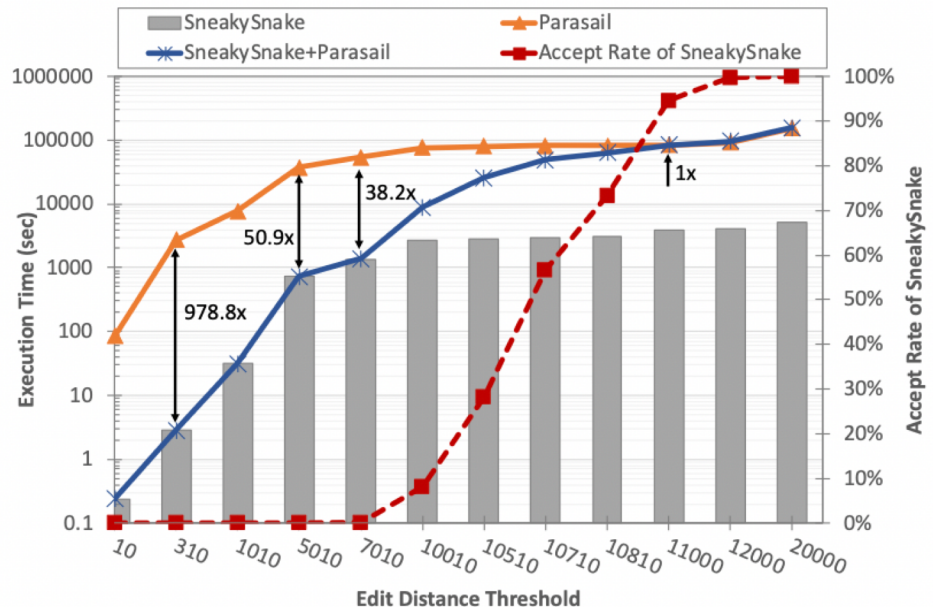
	E (bp)	Slice LUT	Slice Register	No. of Filtering Units
GateKeeper	2	0.39%	0.01%	16
	5	0.71%	0.01%	16
Shouji	2	0.69%	0.08%	16
	5	1.72%	0.16%	16
Snake-on-Chip	2	0.68%	0.16%	16
	5	1.42%	0.34%	16

Long Sequence Filtering (SneakySnake vs Parasail)

10K bp dataset



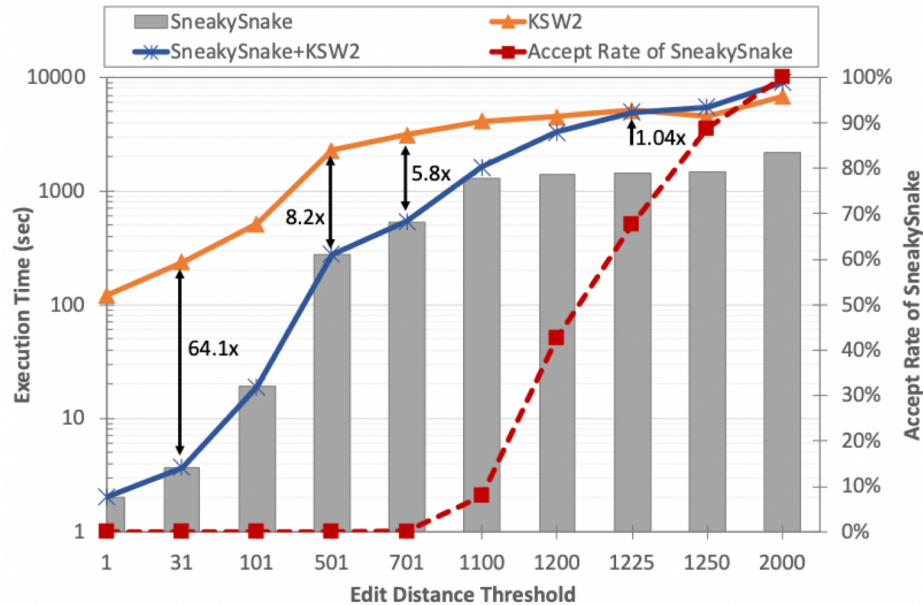
100K bp dataset



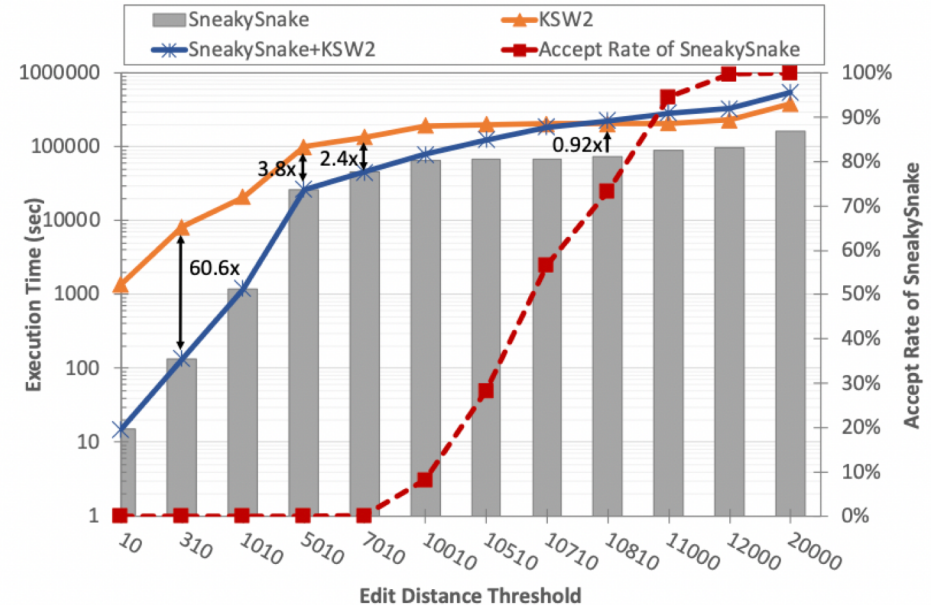
The execution time of SneakySnake, Parasail, and SneakySnake integrated with Parasail using long reads, (a) Set_5 and (b) Set_6, and 40 CPU threads. The y-axis is on a logarithmic scale. For each edit distance threshold value, we provide the rate of accepted pairs (out of 100,000 pairs for Set_5 and out of 74,687 pairs for Set_6)

Long Sequence Filtering (SneakySnake vs KSW2)

10K bp dataset



100K bp dataset



The execution time of SneakySnake, KSW2, and SneakySnake integrated with KSW2 using long reads, (a) Set_5 and (b) Set_6, and a single CPU thread. The y-axis is on a logarithmic scale. For each edit distance threshold value, we provide the rate of accepted pairs (out of 100,000 pairs for Set_5 and out of 74,687 pairs for Set_6) by SneakySnake that are passed to KSW2.

SneakySnake

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

Mohammed Alser^{1,3}, Taha Shahroodi¹, Juan Gómez-Luna¹, Can Alkan³, and Onur Mutlu^{1,2,3}

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

²Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh 15213, PA, USA

³Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey

Download and test for CPU, GPU, and FPGA:

<https://github.com/CMU-SAFARI/SneakySnake>

Alser + "[SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs.](#)" *arXiv preprint* (2019).

Read Mapping & Filtering

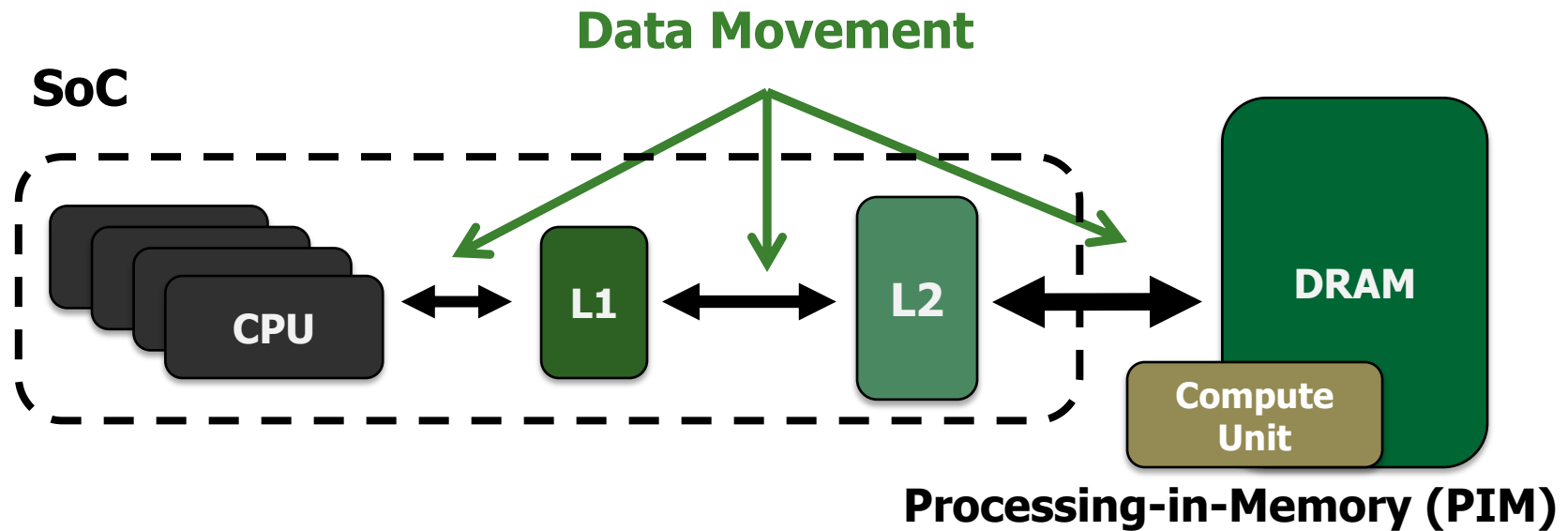
- Problem: Heavily bottlenecked by Data Movement
- Shouji performance limited by DRAM bandwidth [Alser+, Bioinformatics 2019]
- GateKeeper performance limited by DRAM bandwidth [Alser+, Bioinformatics 2017]
- Ditto for SHD [Xin+, Bioinformatics 2015]
- Solution: Processing-in-memory can alleviate the bottleneck

Read Mapping & Filtering in Memory

We need to design
mapping & filtering algorithms
that fit processing-in-memory

Energy Cost of Data Movement

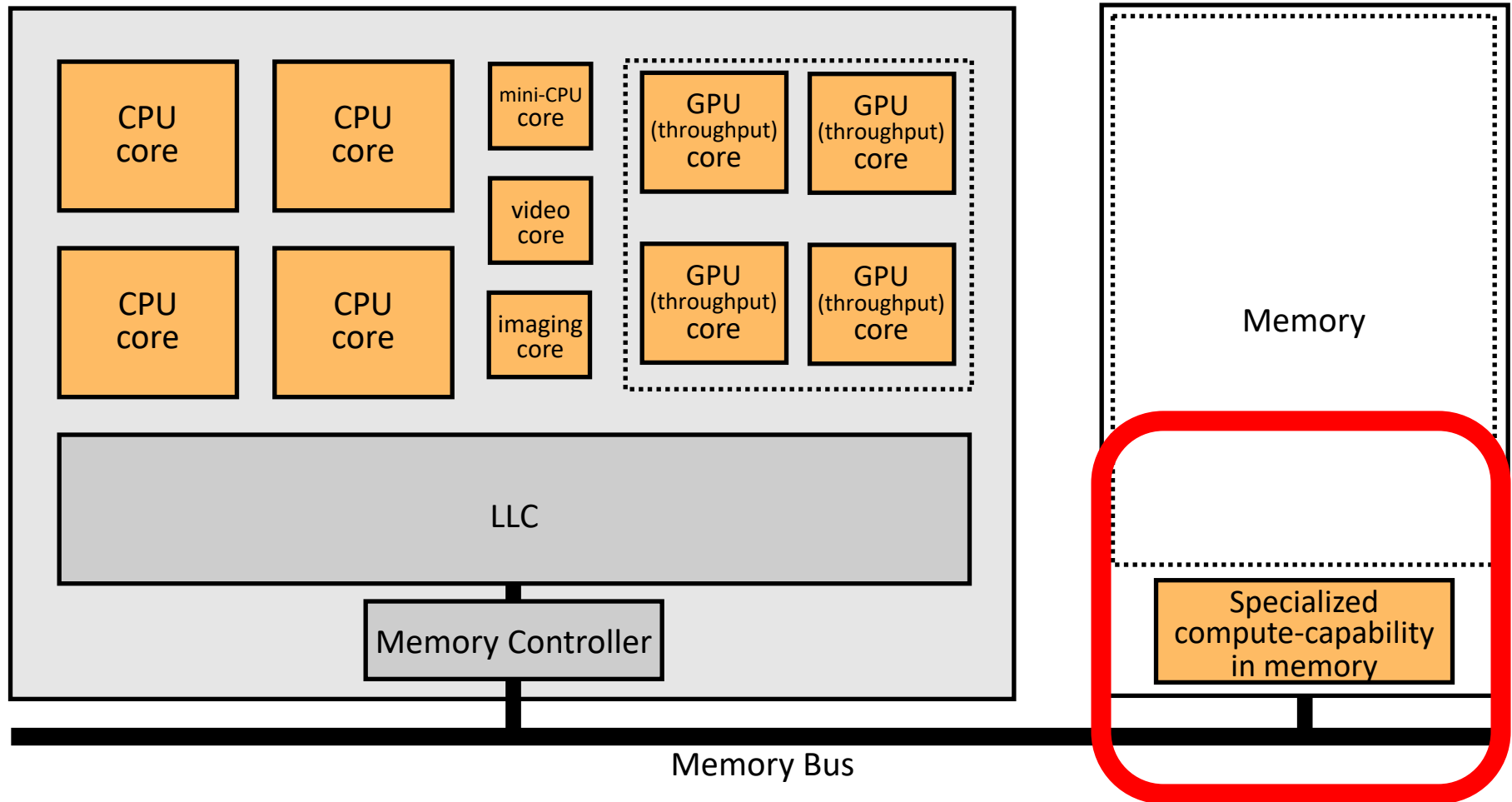
key observation: 62.7% of the total system energy is spent on **data movement**



Potential solution: move computation **close to data**

Challenge: limited area and energy budget

Memory as an Accelerator



Memory similar to a “conventional” accelerator

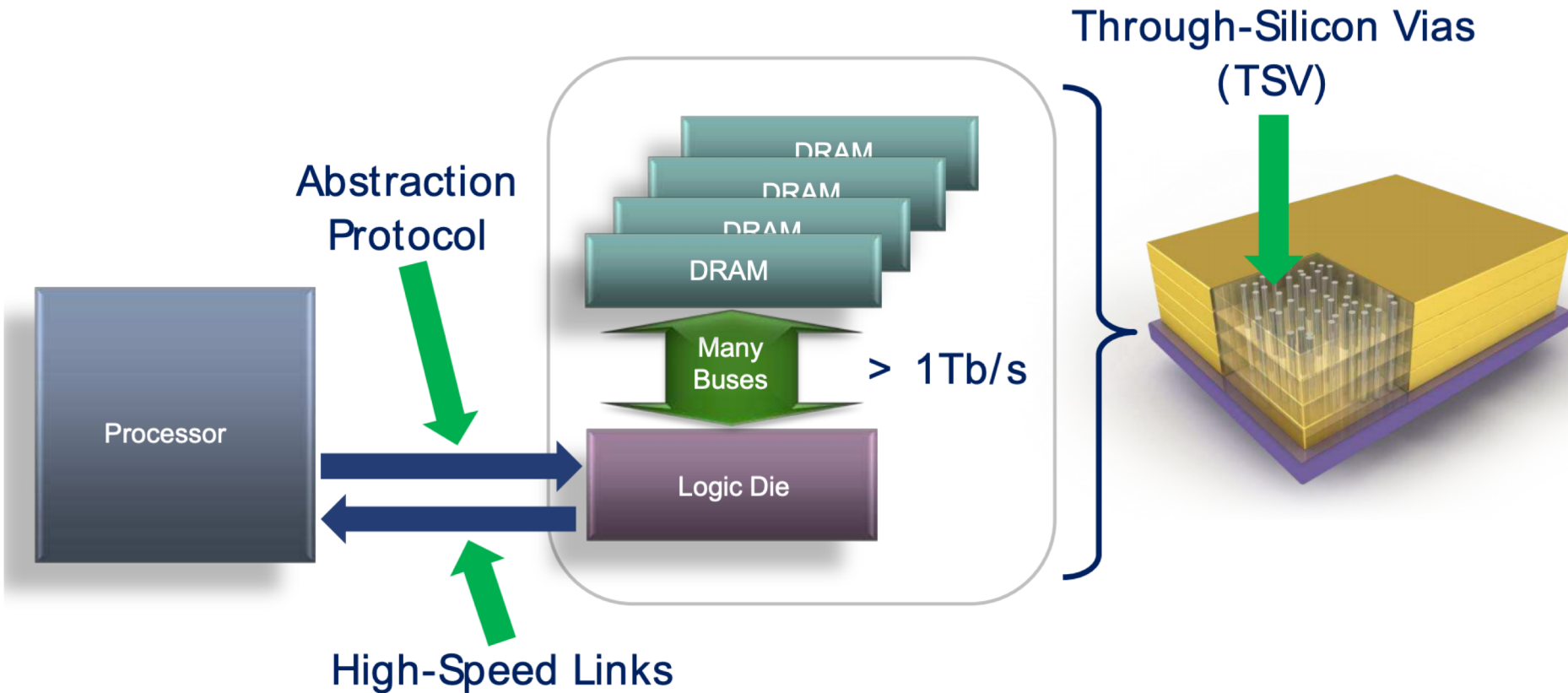
Processing in Memory Approaches

- 1. Minimally changing memory chips
- 2. Exploiting 3D-stacked memory

In-Memory Bulk Bitwise Operations

- We can support in-DRAM COPY, ZERO, AND, OR, NOT, MAJ
- At low cost
- Using analog computation capability of DRAM
 - Idea: activating multiple rows performs computation
- 30-60X performance and energy improvement
 - Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO 2017.
- New memory technologies enable even more opportunities
 - Memristors, resistive RAM, phase change mem, STT-MRAM, ...
 - Can operate on data with minimal movement

Hybrid Memory Cube (HMC)



Notes: Tb/s = Terabits / second
HMC height is exaggerated

More on In-DRAM Bulk AND/OR

- Vivek Seshadri, Kevin Hsieh, Amirali Boroumand, Donghyuk Lee, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,
"Fast Bulk Bitwise AND and OR in DRAM"
IEEE Computer Architecture Letters (***CAL***), April 2015.

Fast Bulk Bitwise AND and OR in DRAM

Vivek Seshadri*, Kevin Hsieh*, Amirali Boroumand*, Donghyuk Lee*,
Michael A. Kozuch†, Onur Mutlu*, Phillip B. Gibbons†, Todd C. Mowry*

*Carnegie Mellon University

†Intel Pittsburgh

More on In-DRAM Bitwise Operations

- Vivek Seshadri et al., “**Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology**,” MICRO 2017.

Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology

Vivek Seshadri^{1,5} Donghyuk Lee^{2,5} Thomas Mullins^{3,5} Hasan Hassan⁴ Amirali Boroumand⁵
Jeremie Kim^{4,5} Michael A. Kozuch³ Onur Mutlu^{4,5} Phillip B. Gibbons⁵ Todd C. Mowry⁵

¹Microsoft Research India ²NVIDIA Research ³Intel ⁴ETH Zürich ⁵Carnegie Mellon University

More on In-DRAM Bulk Bitwise Execution

- Vivek Seshadri and Onur Mutlu,
"In-DRAM Bulk Bitwise Execution Engine"
Invited Book Chapter in Advances in Computers, to appear
in 2020.
[[Preliminary arXiv version](#)]

In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri
Microsoft Research India
`visesha@microsoft.com`

Onur Mutlu
ETH Zürich
`onur.mutlu@inf.ethz.ch`

RowClone & Bitwise Ops in Real DRAM Chips

ComputeDRAM: In-Memory Compute Using Off-the-Shelf DRAMs

Fei Gao

feig@princeton.edu

Department of Electrical Engineering
Princeton University

Georgios Tziantzioulis

georgios.tziantzioulis@princeton.edu

Department of Electrical Engineering
Princeton University

David Wentzlaff

wentzlaf@princeton.edu

Department of Electrical Engineering
Princeton University

Pinatubo: RowClone and Bitwise Ops in PCM

Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-volatile Memories

Shuangchen Li^{1*}, Cong Xu², Qiaosha Zou^{1,5}, Jishen Zhao³, Yu Lu⁴, and Yuan Xie¹

University of California, Santa Barbara¹, Hewlett Packard Labs²

University of California, Santa Cruz³, Qualcomm Inc.⁴, Huawei Technologies Inc.⁵
{shuangchenli, yuanxie}@ece.ucsb.edu¹

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

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"** to appear in ***BMC Genomics***, 2018.
Proceedings of the 16th Asia Pacific Bioinformatics Conference (APBC), Yokohama, Japan, January 2018.
[arxiv.org Version \(pdf\)](#)

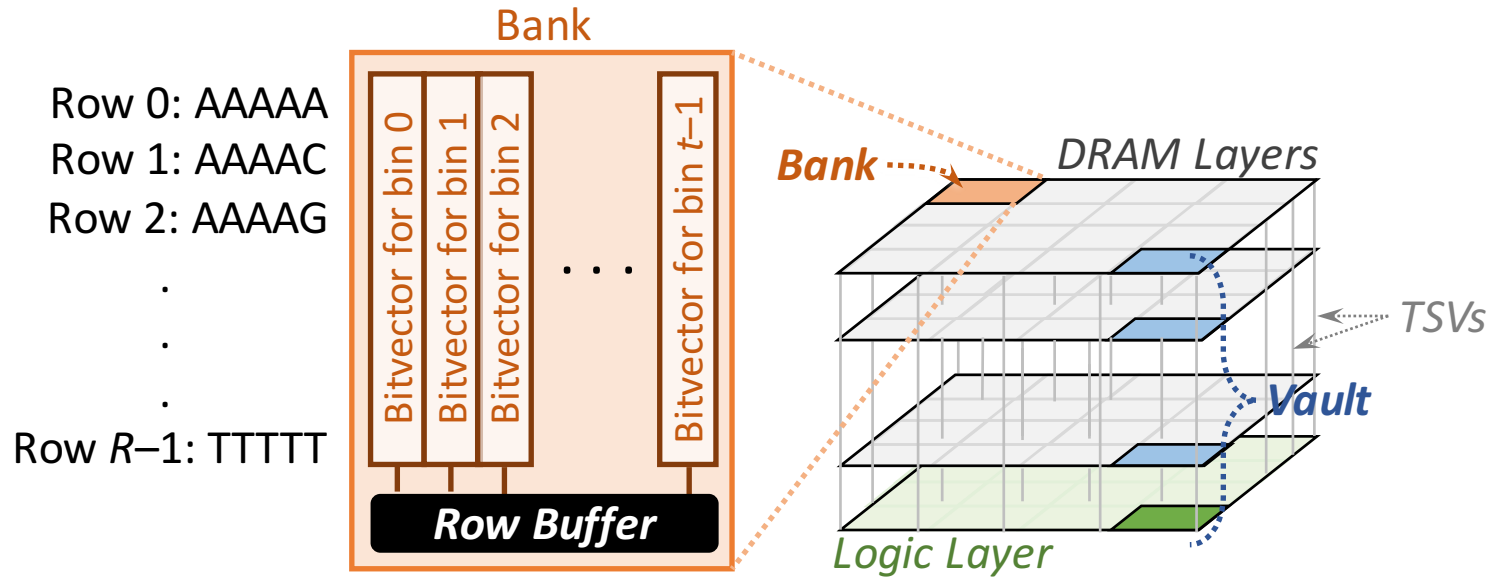
GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies

Jeremie S. Kim^{1,6*}, Damla Senol Cali¹, Hongyi Xin², Donghyuk Lee³, Saugata Ghose¹, Mohammed Alser⁴, Hasan Hassan⁶, Oguz Ergin⁵, Can Alkan^{*4}, and Onur Mutlu^{*6,1}

GRIM-Filter

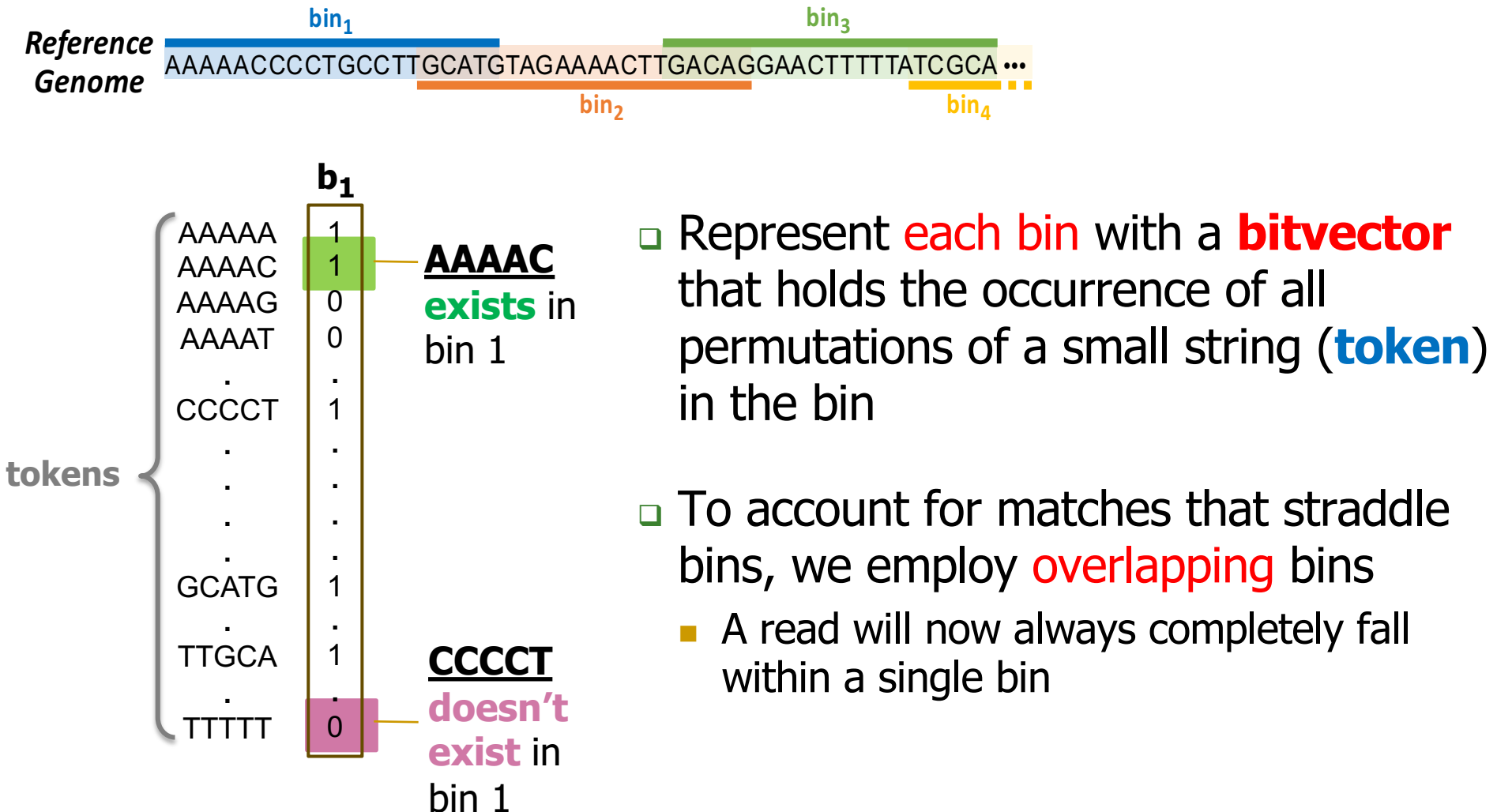
- **Key observation:** FPGA and GPU accelerators are Heavily bottlenecked by **Data Movement**.
- **Key idea:** exploiting the high memory bandwidth and the logic layer of **3D-stacked memory** to perform **highly-parallel filtering** in the DRAM chip itself.
- **Key results:**
 - We propose an algorithm called **GRIM-Filter**
 - GRIM-Filter with processing-in-memory is 1.8x-3.7x (2.1x on average) **faster than FastHASH filter** (BMC Genomics'13) across real data sets.
 - GRIM-Filter has 5.6x-6.4x (6.0x on average) lower falsely accepted pairs than **FastHASH filter** (BMC Genomics'13) across real data sets.

GRIM-Filter in 3D-Stacked DRAM

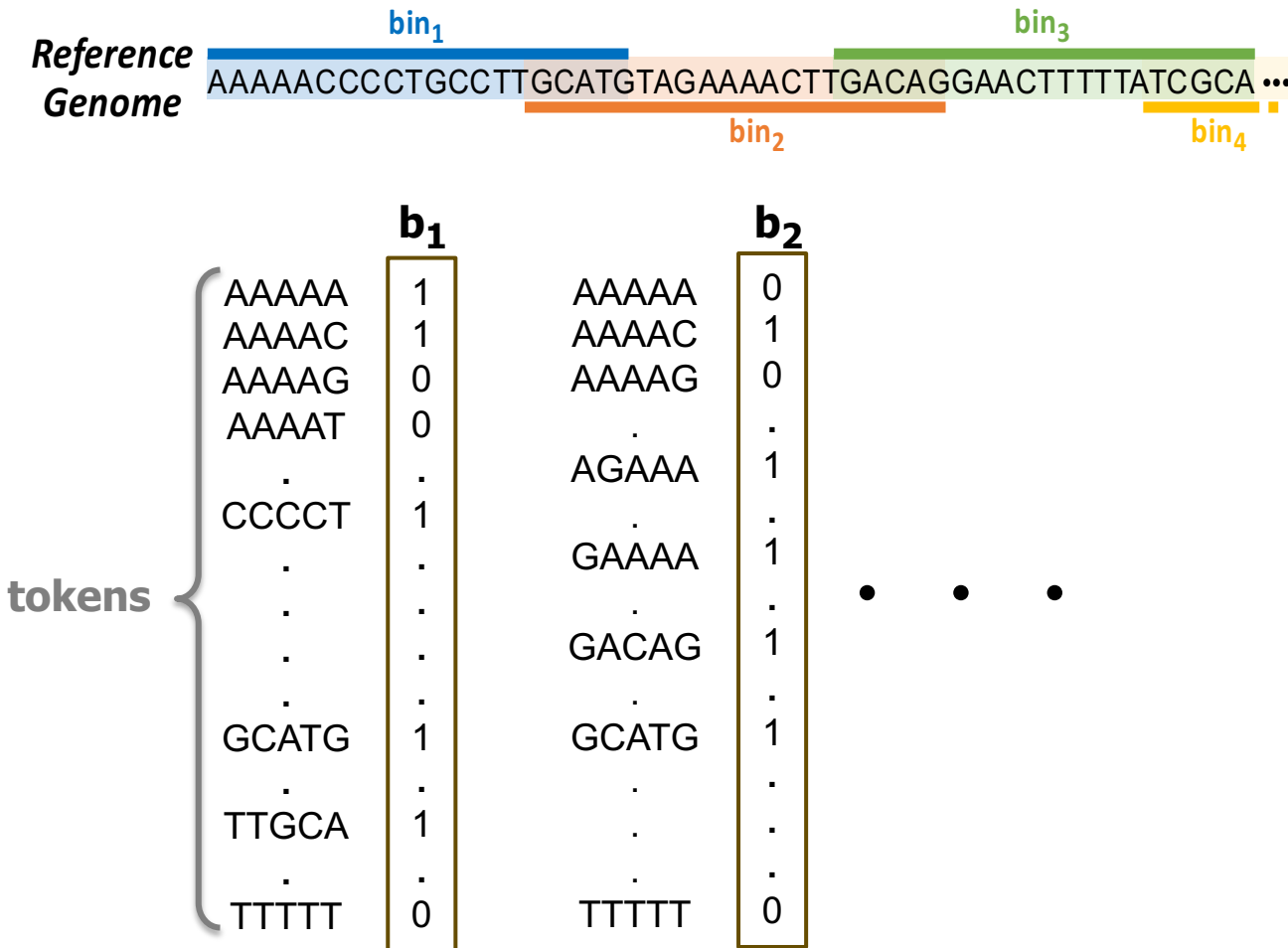


- Each DRAM layer is organized as an array of **banks**
 - A **bank** is an array of cells with a row buffer to transfer data
- The layout of bitvectors in a bank enables filtering many bins in parallel

GRIM-Filter: Bitvectors



GRIM-Filter: Bitvectors

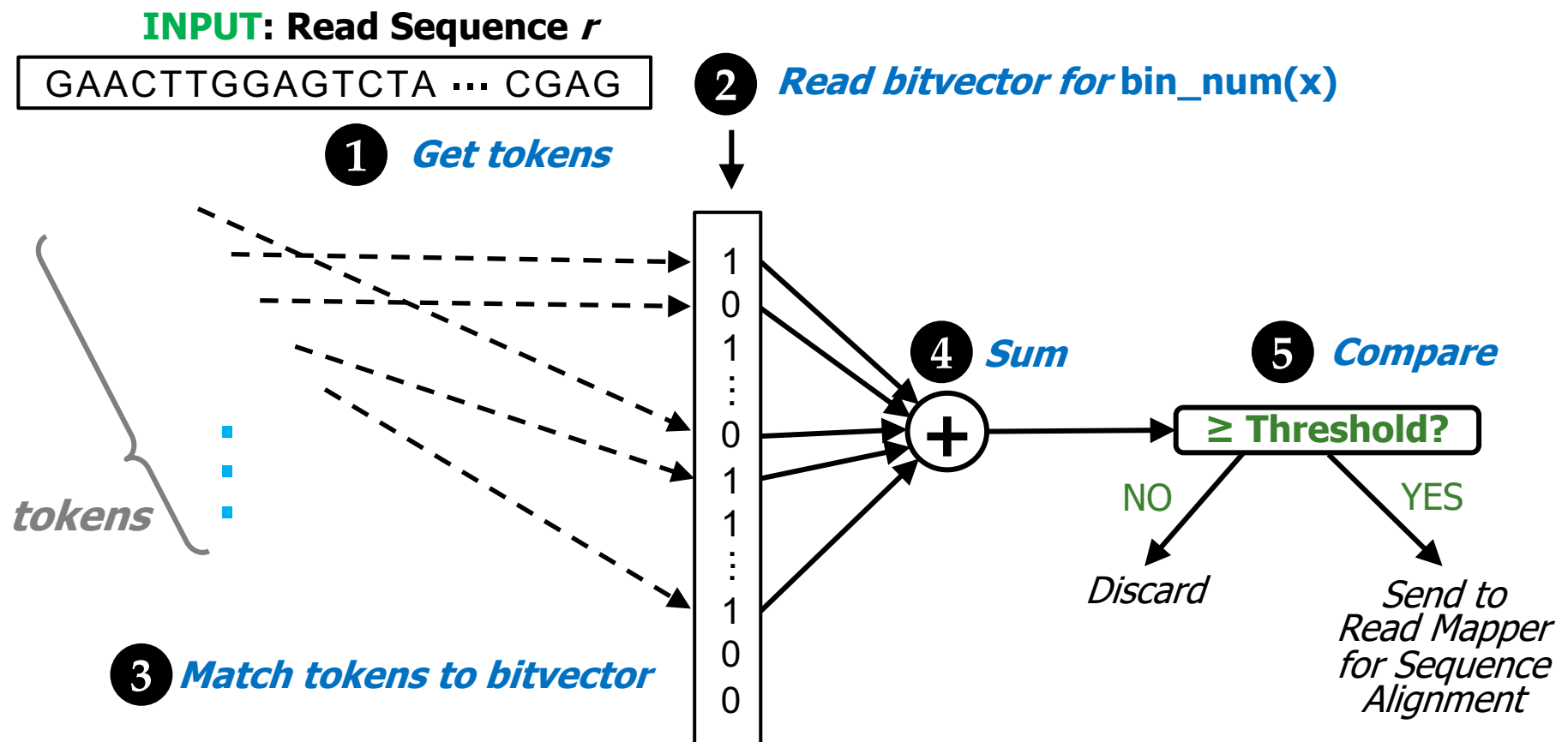


Storing all bitvectors requires $4^n * t$ bits in memory, where
 t = number of bins
 &
 n = token length.

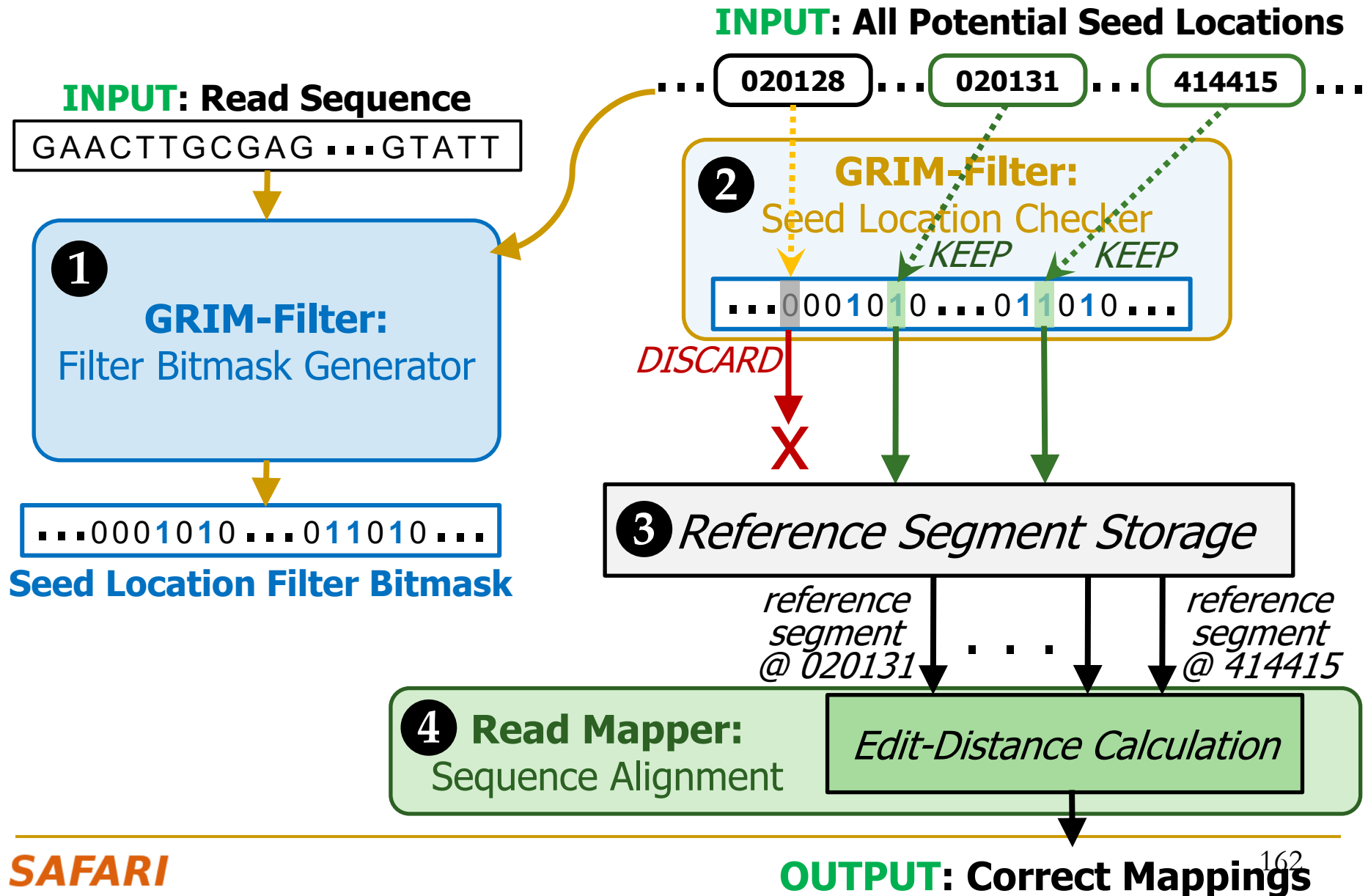
For **bin size** ~200,
 and **n** = 5,
memory footprint
 ~3.8 GB

GRIM-Filter: Checking a Bin

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



Integrating GRIM-Filter into a Read Mapper



Key Properties of GRIM-Filter

1. Simple Operations:

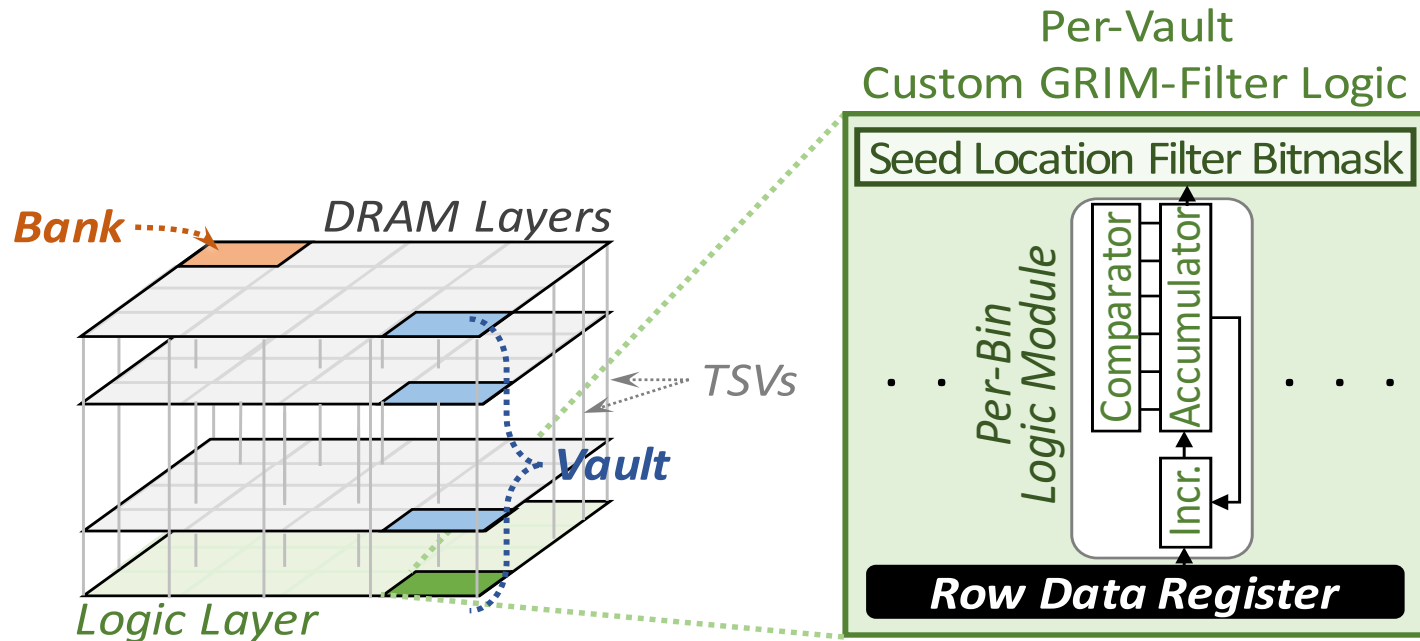
- ❑ To check a given bin, find the **sum** of all bits corresponding to each token in the read
- ❑ **Compare** against threshold to determine whether to align

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

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

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"** to appear in ***BMC Genomics***, 2018.
Proceedings of the 16th Asia Pacific Bioinformatics Conference (APBC), Yokohama, Japan, January 2018.
[arxiv.org Version \(pdf\)](#)

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

Jeremie S. Kim^{1,6*}, Damla Senol Cali¹, Hongyi Xin², Donghyuk Lee³, Saugata Ghose¹, Mohammed Alser⁴, Hasan Hassan⁶, Oguz Ergin⁵, Can Alkan^{*4}, and Onur Mutlu^{*6,1}

GenCache: Leveraging In-Cache Operators for Efficient Sequence Alignment

Anirban Nag
anirban@cs.utah.edu
University of Utah
Salt Lake City, Utah

C. N. Ramachandra
ramgowda@cs.utah.edu
University of Utah
Salt Lake City, Utah

Rajeev Balasubramonian
rajeev@cs.utah.edu
University of Utah
Salt Lake City, Utah

Ryan Stutsman
stutsman@cs.utah.edu
University of Utah
Salt Lake City, Utah

Edouard Giacomin
edouard.giacomin@utah.edu
University of Utah
Salt Lake City, Utah

Hari Kambalasubramanyam
hari.kambalasubramanyam@utah.edu
University of Utah
Salt Lake City, Utah

Pierre-Emmanuel Gaillardon
pierre-
emmanuel.gaillardon@utah.edu
University of Utah
Salt Lake City, Utah

Nag, Anirban, et al. "**GenCache: Leveraging In-Cache Operators for Efficient Sequence Alignment**." *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 52)* , ACM, 2019.

GenCache

- **Key observation:** State-of-the-art alignment accelerators are still **bottlenecked by memory**.
- **Key ideas:**
 - ❑ Performing **in-cache alignment + pre-alignment filtering** by enabling processing-in-cache using previous proposal, ComputeCache (HPCA'17).
 - ❑ Using **different Pre-alignment filters** depending on the selected edit distance threshold.
- **Results:**
 - ❑ GenCache on CPU is 1.36x faster than GenAx (ISCA 2018).
GenCache in cache is 5.26x faster than GenAx.
 - ❑ GenCache chip has 16.4% higher area, 34.7% higher peak power, and 15% higher average power than GenAx.

GenCache's Four Phases

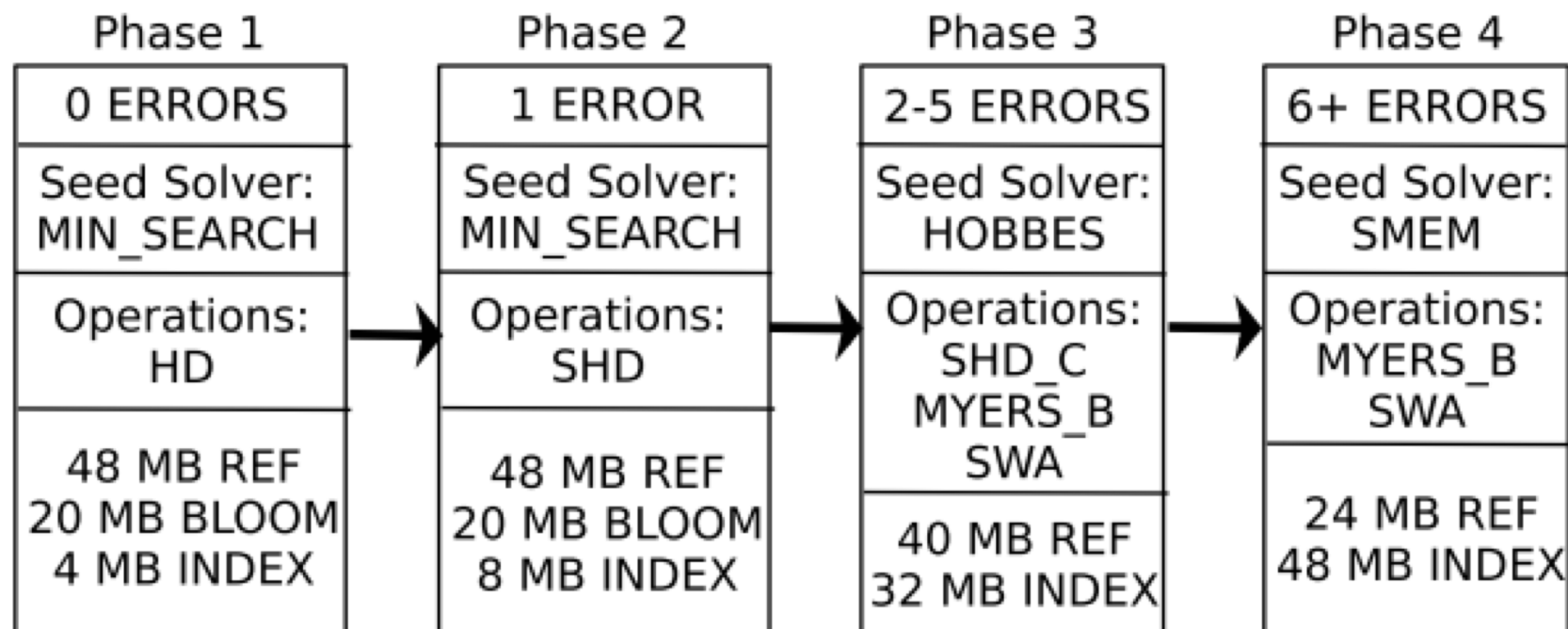


Figure 7: Four phases in the new alignment algorithm that exploits in-cache operators.

Throughput Results

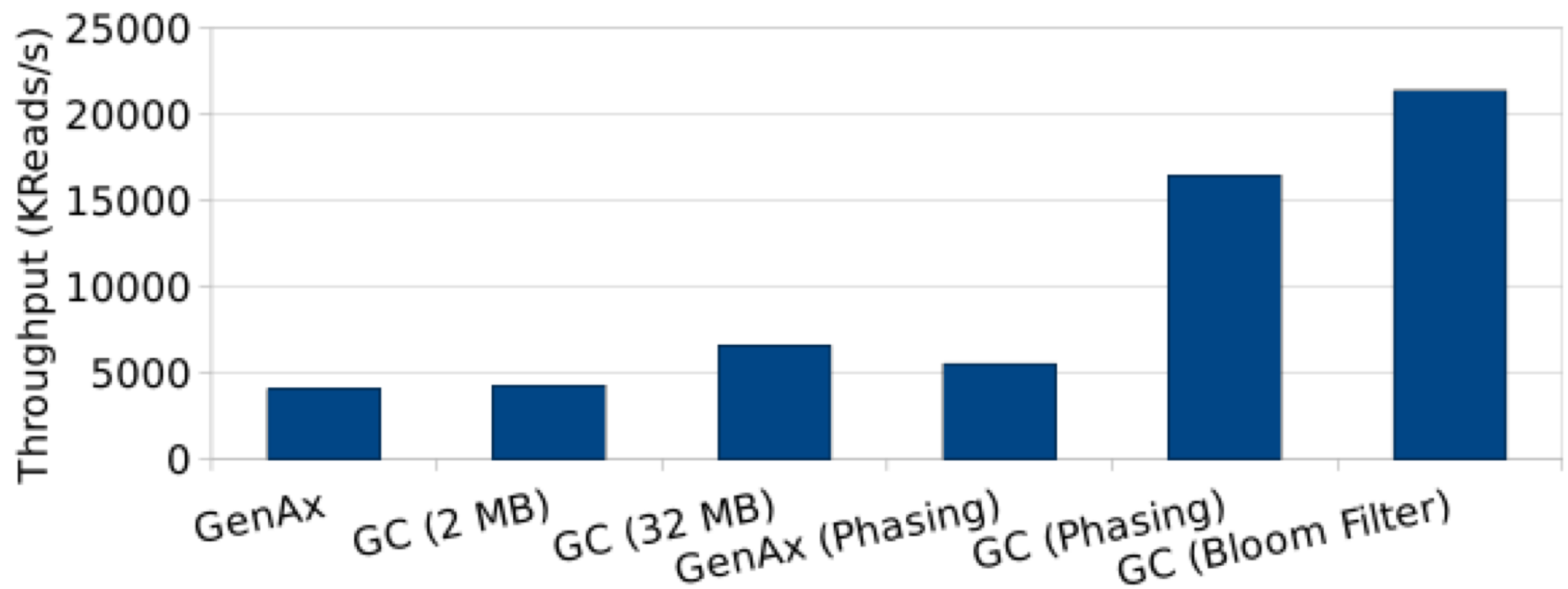


Figure 9: Throughput improvement of GenCache (Hardware & Software).

Ongoing Directions

■ **Seed Filtering Technique:**

- **Goal:** Reducing the number of seed (k-mer) locations.
 - **Heuristic** (limits the number of mapping locations for each seed).
 - Supports **exact** matches only.

■ **Pre-alignment Filtering Technique:**

- **Goal:** Reducing the number of *invalid mappings* ($>E$).
 - Supports both **exact and inexact** matches.
 - Provides some **falsely-accepted** mappings.

■ **Read Alignment Acceleration:**

- **Goal:** Performing read alignment at scale.
 - Limits the **numeric range** of each cell in the DP table and hence supports **limited scoring** function.
 - May not support **backtracking** step due to random memory accesses.

Session 3A: Programmable Devices and Co-processors

ASPLOS'18, March 24–28, 2018, Williamsburg, VA, USA

Darwin: A Genomics Co-processor Provides up to 15,000× acceleration on long read assembly

Yatish Turakhia
Stanford University
yatisht@stanford.edu

Gill Bejerano
Stanford University
bejerano@stanford.edu

William J. Dally
Stanford University
NVIDIA Research
dally@stanford.edu

- Seed filter: **D-Soft**
- Read alignment accelerator: **GACT** ← We will cover this

Yatish+ "Darwin: A genomics co-processor provides up to 15,000x acceleration on long read assembly." *ASPLOS* 2018.

<http://bejerano.stanford.edu/papers/p199-turakhia.pdf>

Darwin: GACT Hardware Acceleration

■ Key observation:

- ❑ **Data Dependencies limit** accelerating the dynamic programming table calculation.

■ Key idea:

- ❑ **Divide** the dynamic programming table into **overlapping *tiles***.
- ❑ Calculate each tile **independently** and in a **systolic array** fashion.
- ❑ Calculate **many** alignments **concurrently**.

■ Key result:

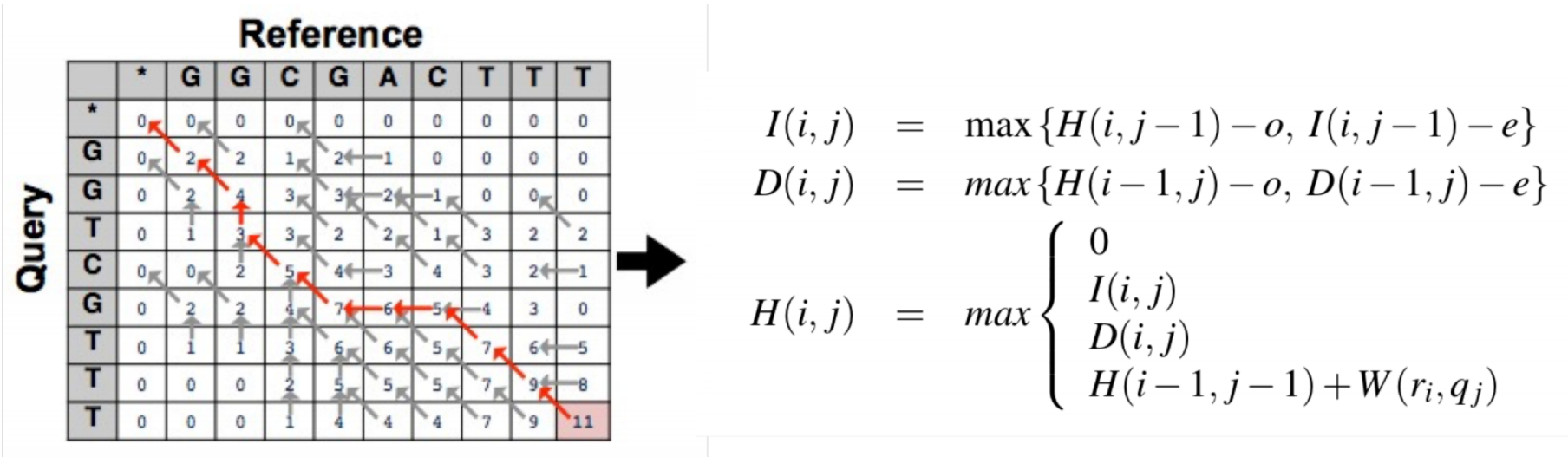
- ❑ It is simulated for TSMC 40nm **CMOS** process.
- ❑ It provides a **speedup** of up to **380x** compared to GACT **software**.
- ❑ It is **three orders** of magnitude **faster** than **Edlib** (best-performing CPU read aligner).

■ Weaknesses:

- ❑ It is not clear if tiling **maintains the same accuracy** as the original dynamic programming algorithm.

Specialized Accelerator for Read Aligner

- Accelerating the read alignment algorithm as-is using specialized hardware (40 nm CMOS) provides a **limited speedup** (37x).



Dynamic programming for gene sequence alignment (Smith-Waterman)

CPU-based read aligner

vs.

Hardware accelerated read aligner

On 14nm CPU

On 40nm Special Unit

35 ALU ops, 15 load/store

1 cycle (37x speedup)

37 cycles

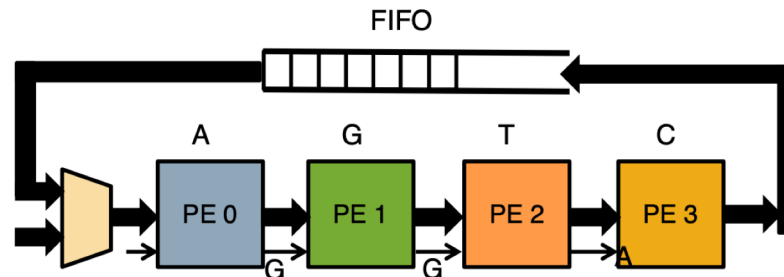
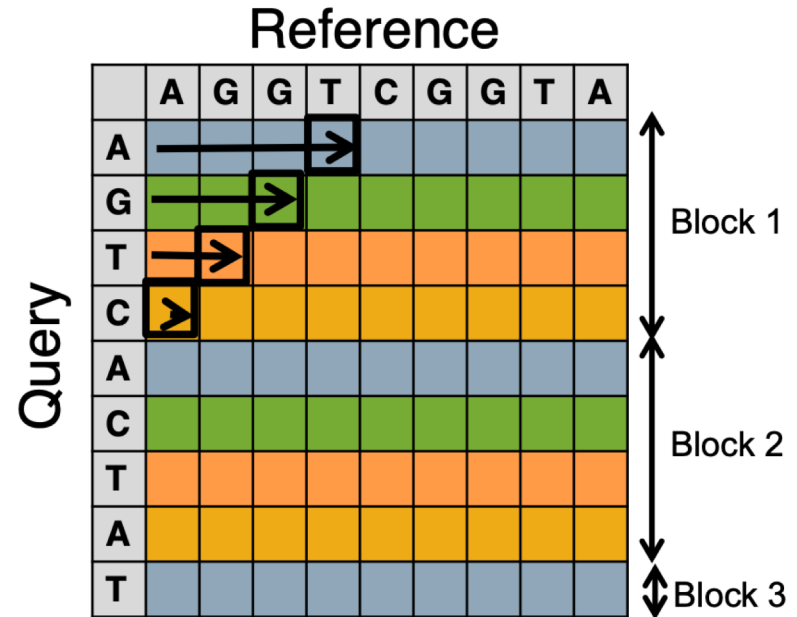
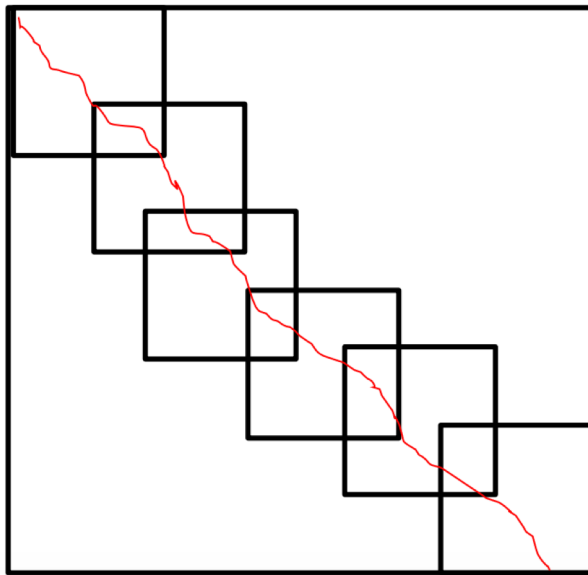
3.1pJ (26,000x efficiency)

81nJ

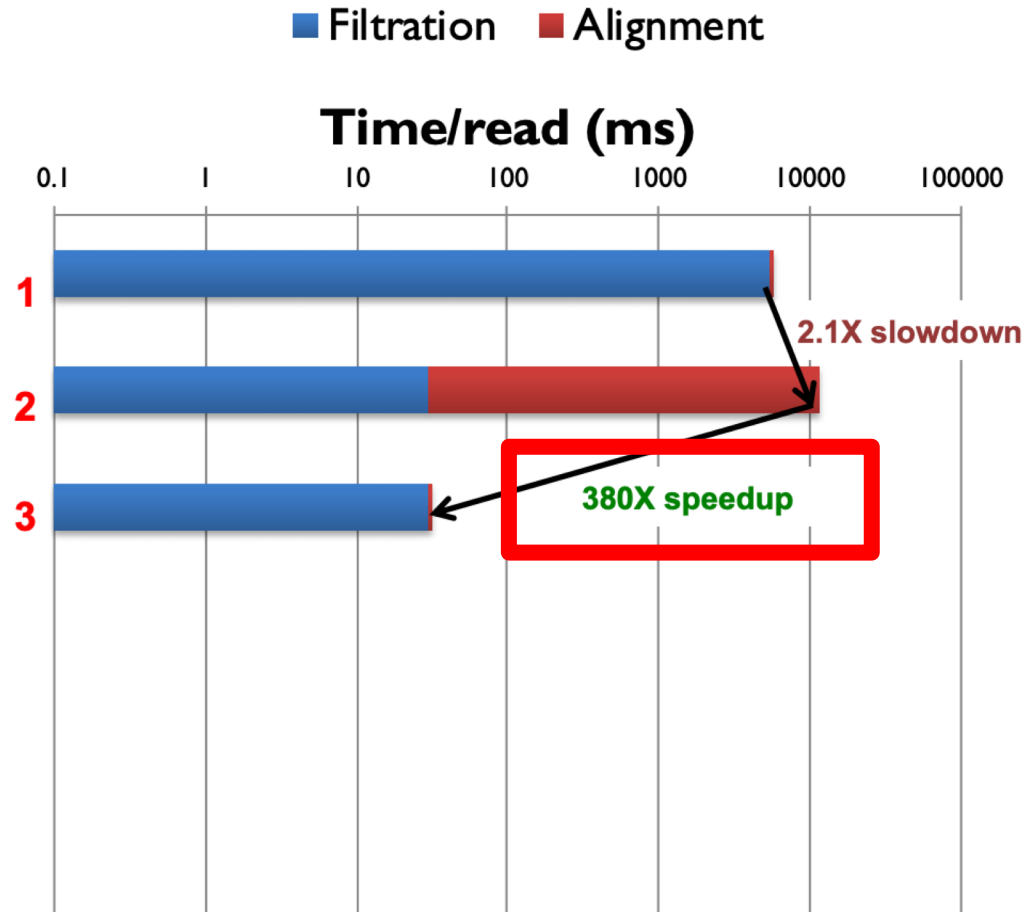
300fJ for logic (remainder is memory)

GACT Alignment

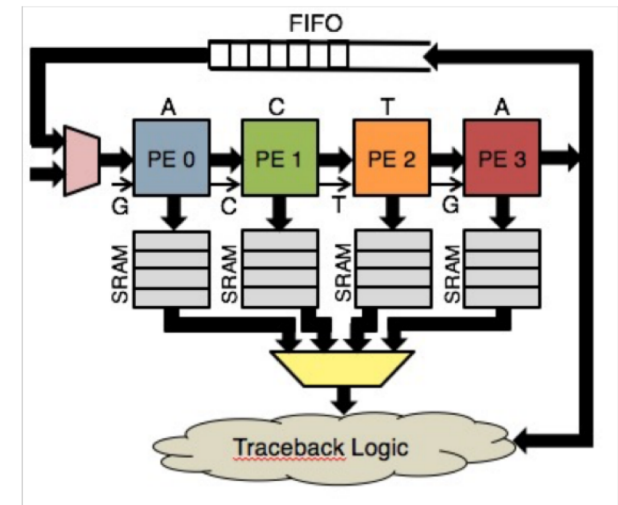
- **Solution:** Divide the table into **overlapping tiles** and compute them all **independently** using **systolic arrays**.
- Store the **trace** of each cell in an SRAM for **traceback**.



GACT Hardware vs. Software Speedup



1. Graphmap (software)
2. Replace by D-SOFT and GACT (software)
3. GACT hardware-acceleration



GACT Hardware vs. Edlib

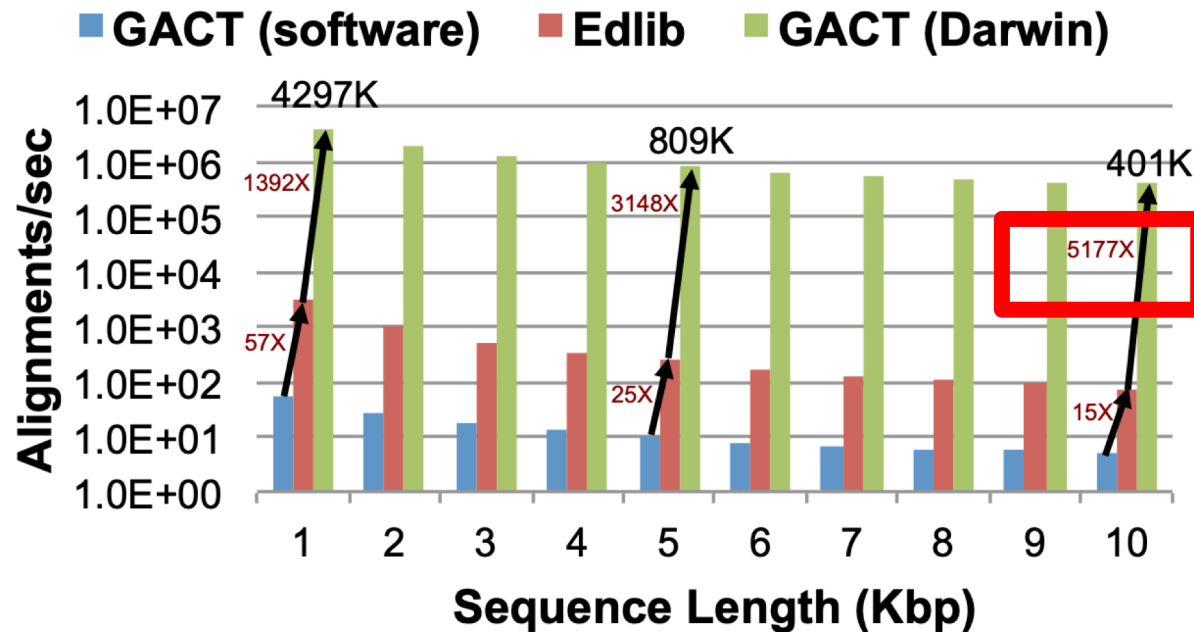


Figure 10: Throughput (alignments/second) comparison for different sequence lengths between a software implementation of GACT, Edlib library and the hardware-acceleration of GACT in Darwin.

More on Darwin

<https://github.com/gsneha26/Darwin-WGA>

Session 3A: Programmable Devices and Co-processors

ASPLOS'18, March 24–28, 2018, Williamsburg, VA, USA

Darwin: A Genomics Co-processor Provides up to 15,000× acceleration on long read assembly

Yatish Turakhia
Stanford University
yatisht@stanford.edu

Gill Bejerano
Stanford University
bejerano@stanford.edu

William J. Dally
Stanford University
NVIDIA Research
dally@stanford.edu

Yatish+ "Darwin: A genomics co-processor provides up to 15,000 x acceleration on long read assembly." *ASPLOS* 2018.

<http://bejerano.stanford.edu/papers/p199-turakhia.pdf>

Disclaimer on Darwin

- Darwin is NOT developed in **SAFARI group**, but we developed GenASM that is published in MICRO 2020.
- GenASM = new read alignment algorithm + PIM specialized accelerator.
- GenASM provides 6.6x better throughput per unit area and 10.5x better throughput per unit power when compared with GACT of Darwin.

Damla will present GenASM during tomorrow lecture 16 October 2020!

arXiv.org > cs > arXiv:2009.07692

Search...

Help | Advanced

Computer Science > Hardware Architecture

[Submitted on 16 Sep 2020]

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

Senol Cali+, "[GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis](#)", MICRO 2020

Conclusion on Ongoing Directions

- Read alignment can be **substantially accelerated** using **computationally inexpensive** and **accurate pre-alignment filtering** algorithms designed for specialized hardware.
- All the **three directions are used** by mappers today, but **filtering has replaced alignment as the bottleneck**.
- **Pre-alignment filtering** does *not* sacrifice any of the aligner capabilities, as it **does not modify or replace the alignment step**.

What **Else** can be **Done**?

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

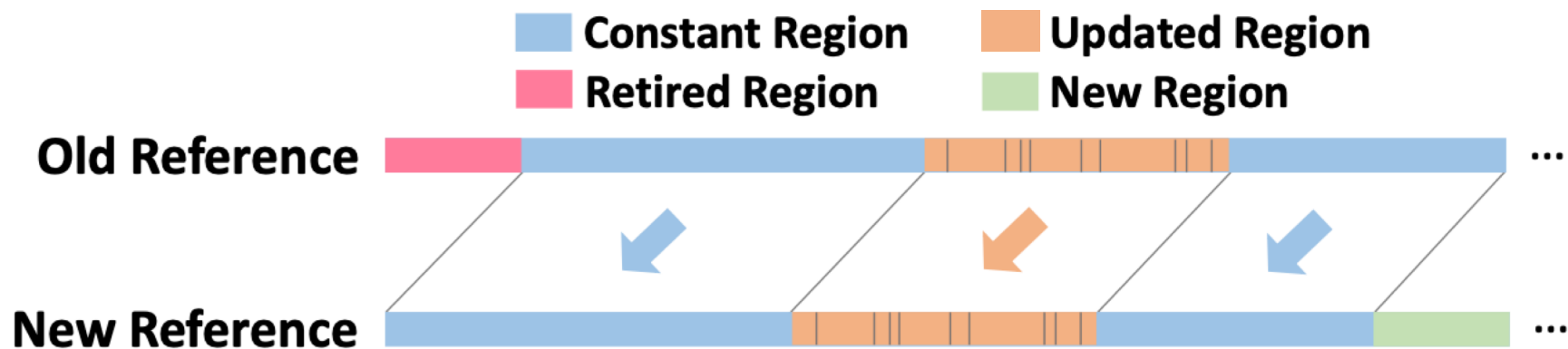


Fig. 2. Reference Genome Regions.

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

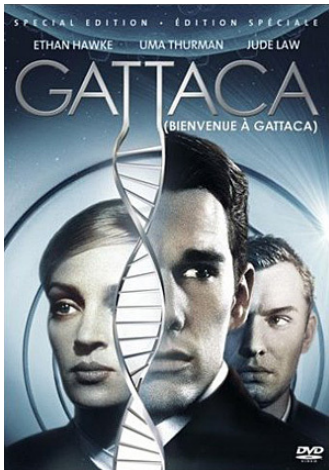
Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- What Makes Read Mapper Slow?
- Algorithmic & Hardware Acceleration
 - Seed Filtering Technique
 - Pre-alignment Filtering Technique
 - Read Alignment Acceleration
- **Where is Read Mapping Going Next?**

Did we Achieve Our Goal?

- **Fast** genome analysis in mere seconds using **limited computational resources** (i.e., personal computer or small hardware).

1997



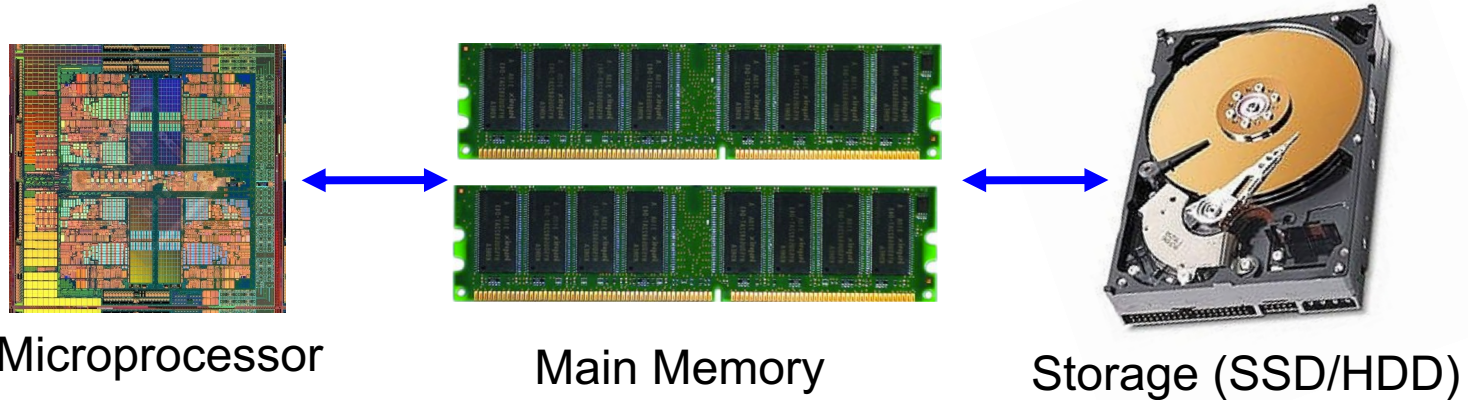
2015



Open Questions

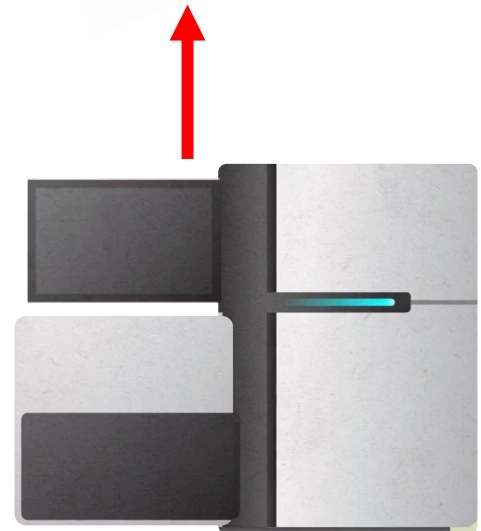
How and where to enable
fast, accurate, cheap,
privacy-preserving, and exabyte scale
analysis of genomic data?

Pushing Towards New Architectures

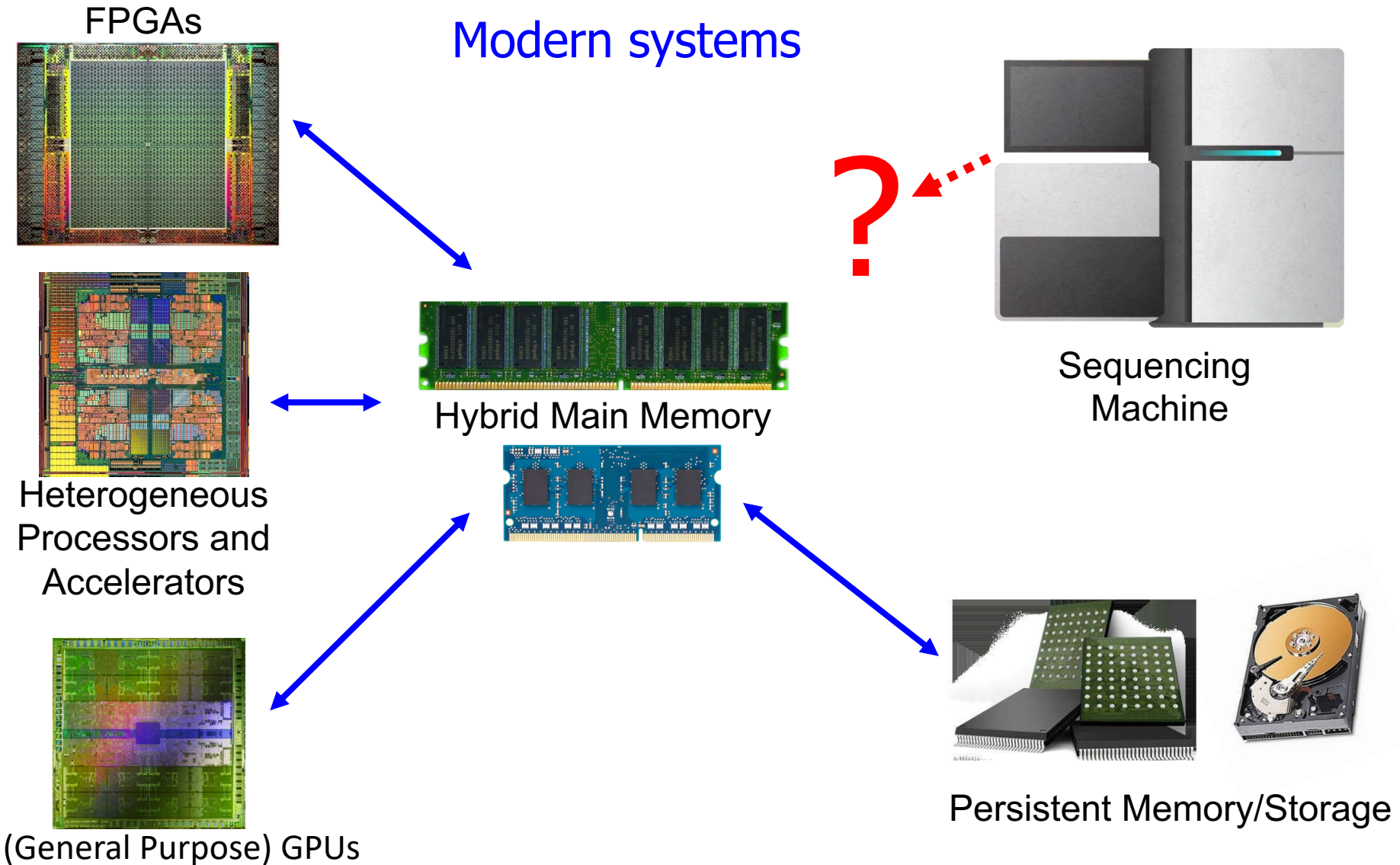


Single **memory** request **consumes**
>160x-800x **more energy** compared to
performing a **complex add operation**

Sequencing
Machine

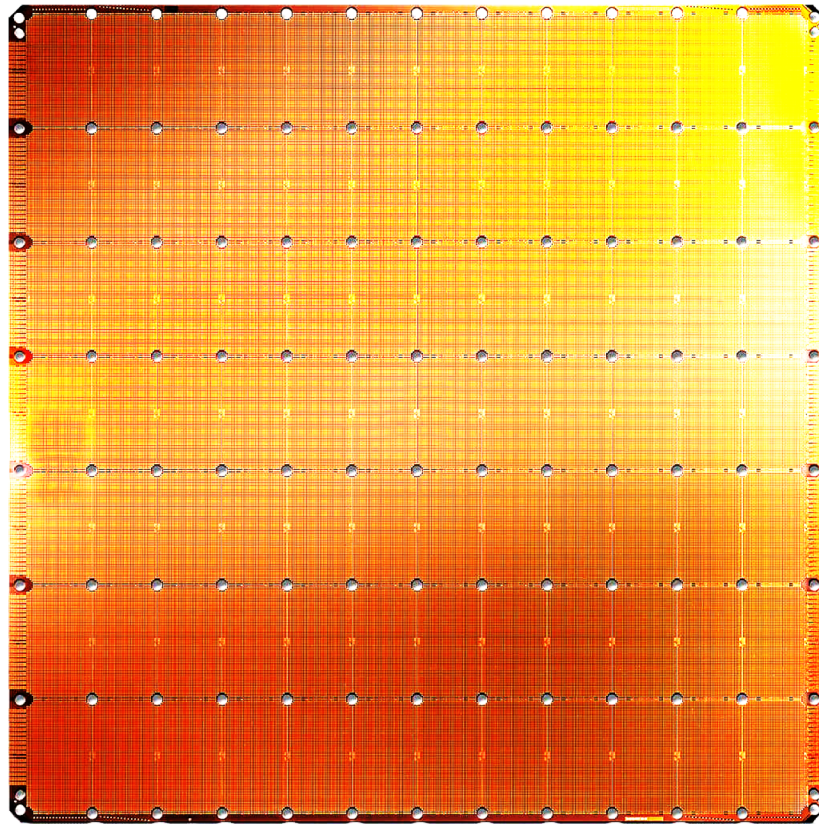


Processing Genomic Data Where it Makes Sense



Most speedup comes from **parallelism** enabled
by **novel architectures** and **algorithms**

Cerebras's Wafer Scale Engine (2019)



Cerebras WSE

1.2 Trillion transistors

46,225 mm²

- The largest ML accelerator chip
- 400,000 cores

NVIDIA TITAN V



Largest GPU

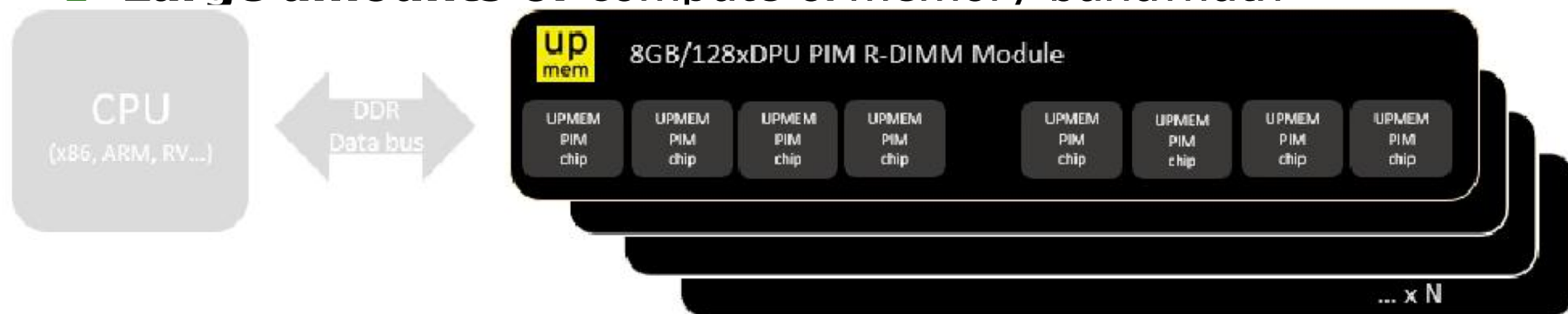
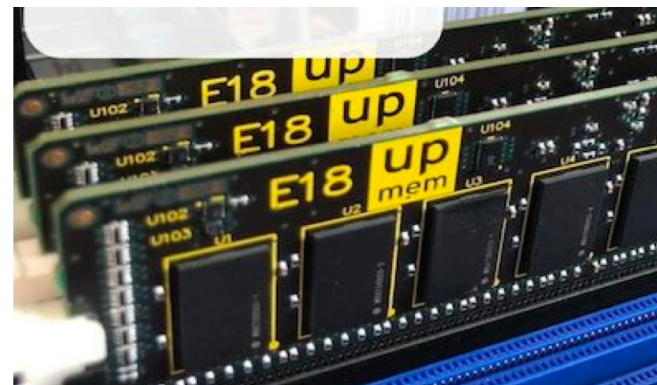
21.1 Billion transistors

815 mm²

<https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/>

UPMEM Processing-in-DRAM Engine (2019)

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



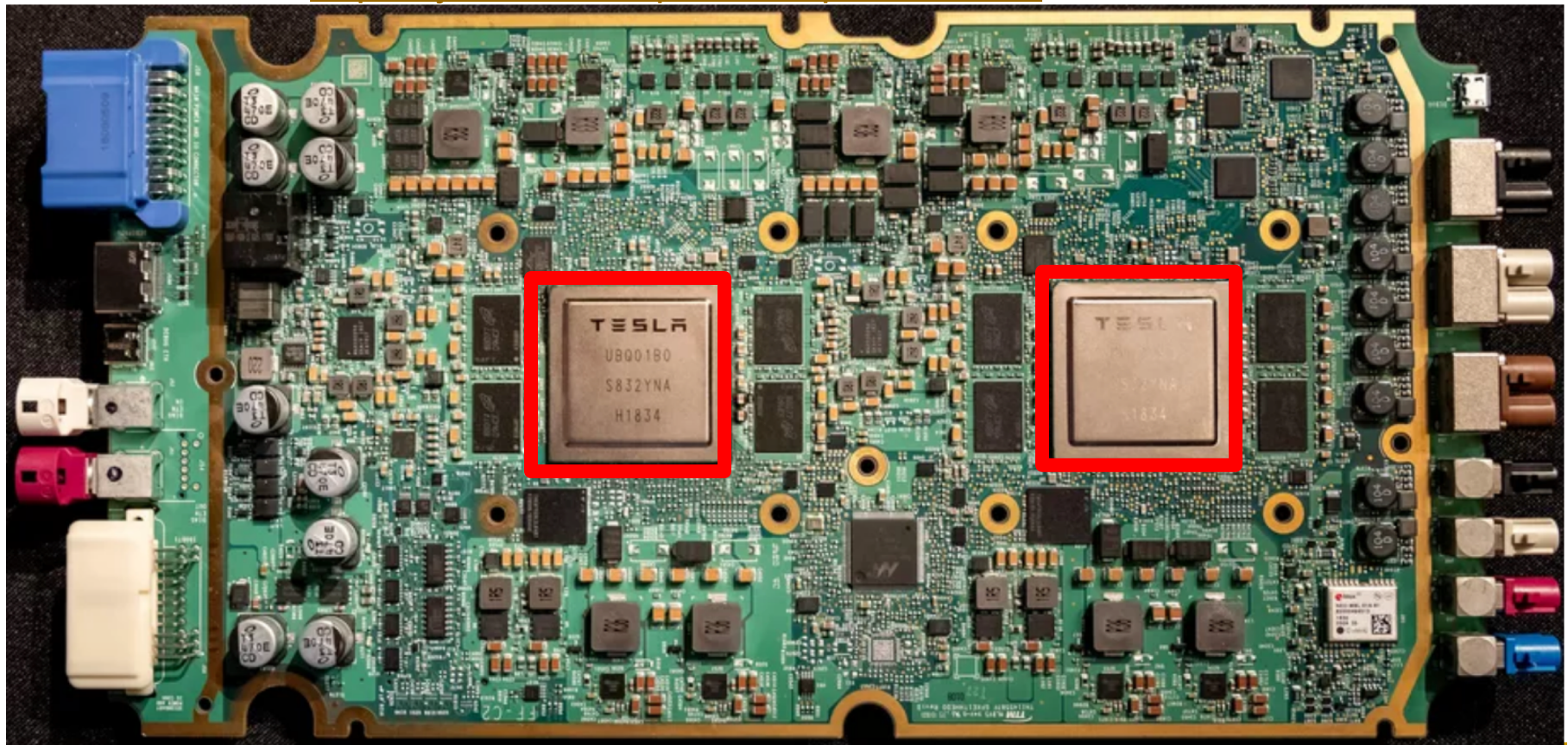
<https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem>

<https://www.upmem.com/video-upmem-presenting-its-true-processing-in-memory-solution-hot-chips-2019/>

TESLA Full Self-Driving Computer (2019)

- ML accelerator: 260 mm², 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.

<https://youtu.be/Ucp0TTmvqOE?t=4236>



Where is Read Mapping Going Next?

Will 100% accurate genome-long reads alleviate/eliminate the need for read mapping?

Think about metagenomics, pan-genomics, ...

Lecture Conclusion

- System design for bioinformatics is a critical problem
 - It has large scientific, medical, societal, personal implications
- This lecture 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 cover various recent ideas to accelerate read mapping
 - A journey since September 2006

Acknowledgments

- Prof. Onur Mutlu, ETH Zurich
- Prof. Can Alkan, Bilkent University

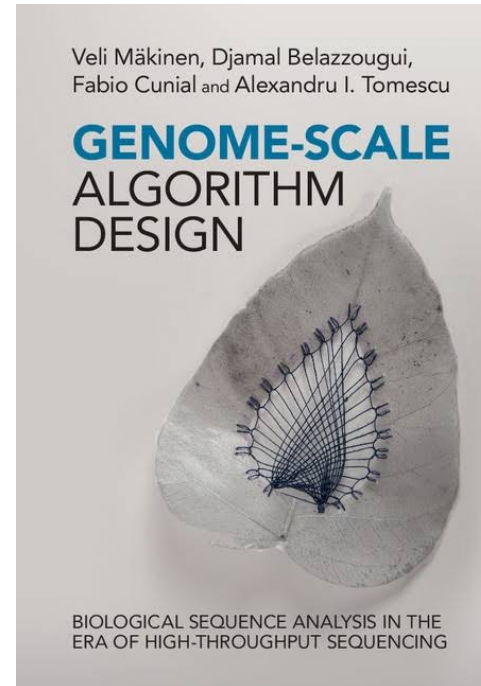
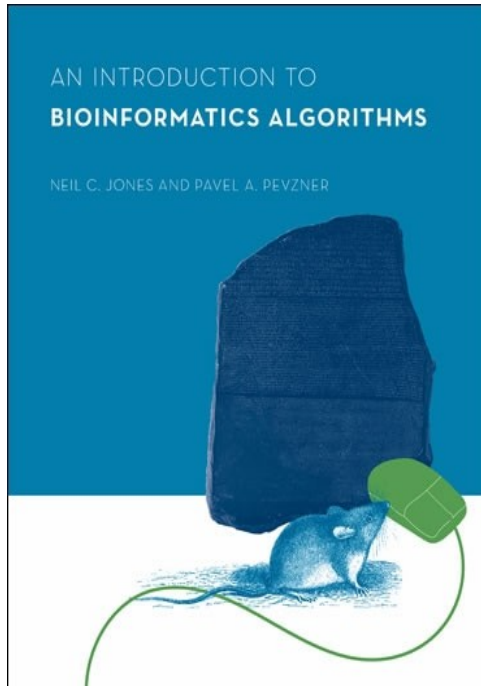
- Many colleagues and collaborators
 - Damla Senol Cali, Jeremie Kim, Hasan Hassan, Can Firtina, Juan Gómez Luna, Donghyuk Lee, Hongyi Xin, ...

- Funders:
 - NIH and Industrial Partners (Alibaba, AMD, Google, Facebook, HP Labs, Huawei, IBM, Intel, Microsoft, Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, VMware)

- All papers, source code, and more are at:
 - <https://people.inf.ethz.ch/omutlu/projects.htm>

Recommended Readings

- Jones, Neil C. and Pavel Pevzner. “[An introduction to bioinformatics algorithms](#),” MIT press, 2004.
- Mäkinen, Veli, Djamel Belazzougui, Fabio Cunial, and Alexandru I. Tomescu. “[Genome-scale algorithm design](#),” Cambridge University Press, 2015.

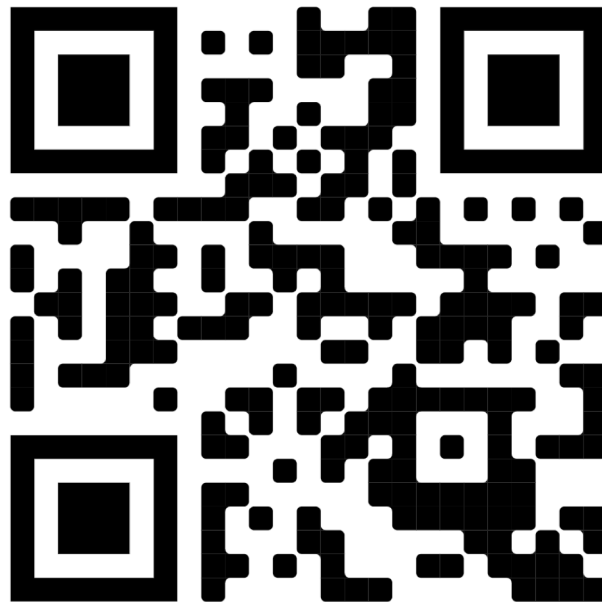


Work With Us

- If you are already a student at ETH and are interested in doing research with SAFARI research group on similar topics, **Talk to me:**
 - **ALSERM @ safari . ethz . ch**

Openings @ SAFARI

- We are **hiring** enthusiastic and motivated students and researchers at all levels.
- Join us now: safari.ethz.ch/apply



Computer Architecture

Lecture 8:

Intelligent Genome Analysis

Dr. Mohammed Alser

ALSERM@safari.ethz.ch

ETH Zurich

Fall 2020

15 October 2020