Accelerating Genome Analysis

A Primer on an Ongoing Journey

Onur Mutlu

omutlu@gmail.com

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

26 January 2021

Technion Invited Lecture

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?
    - . . .

Agenda

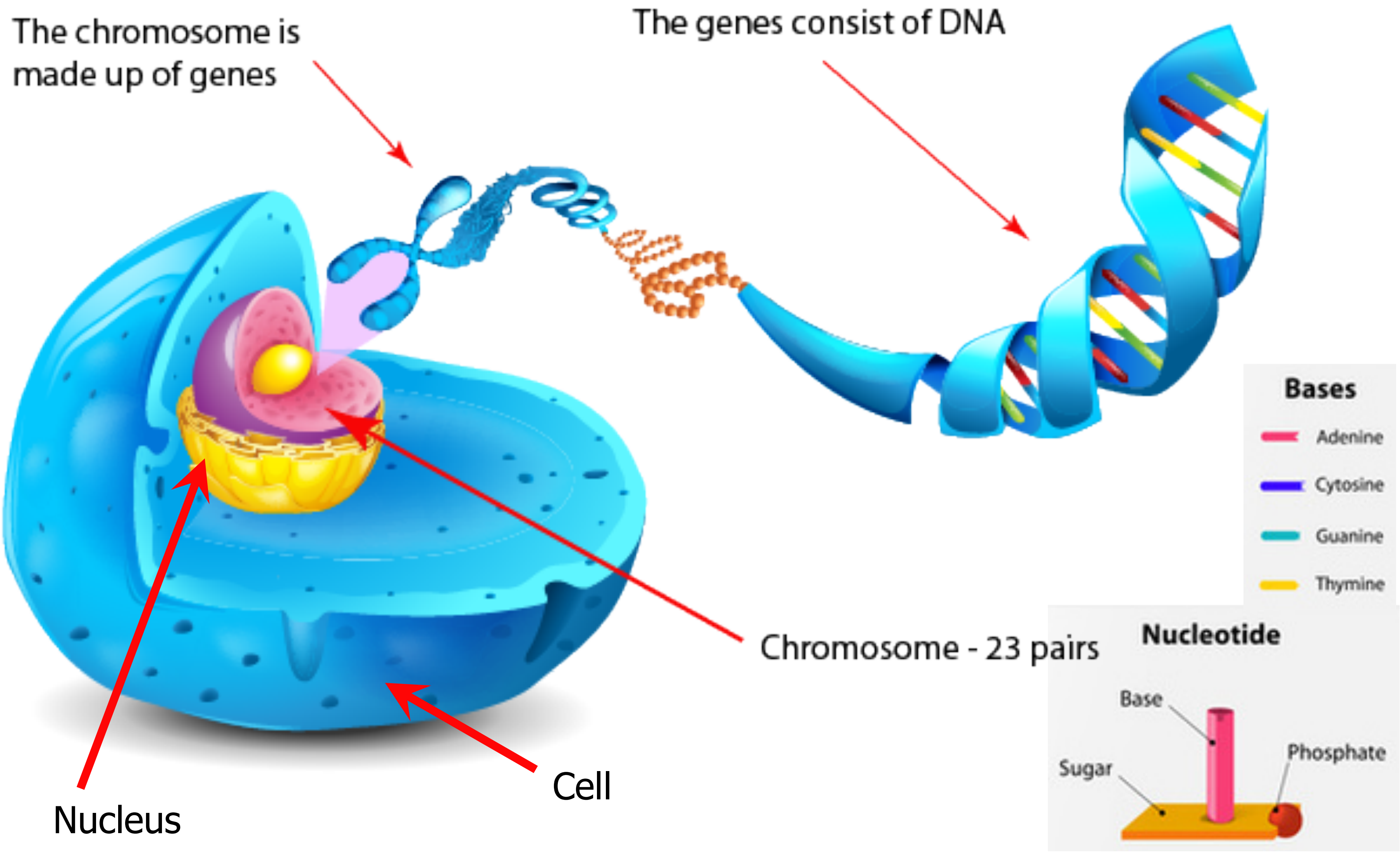
- The Problem: DNA Read Mapping
  - State-of-the-art Read Mapper Design

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

What Is a Genome Made Of?



The chromosome is made up of genes

The genes consist of DNA

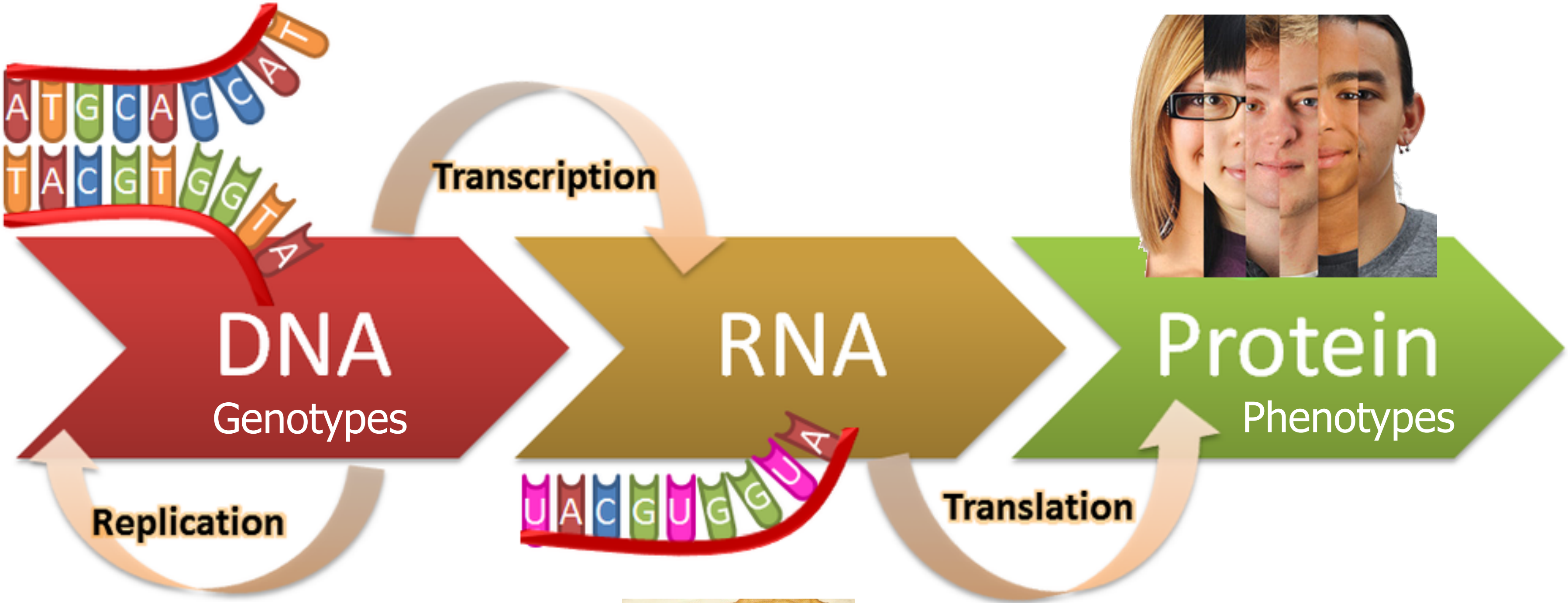
Chromosome - 23 pairs

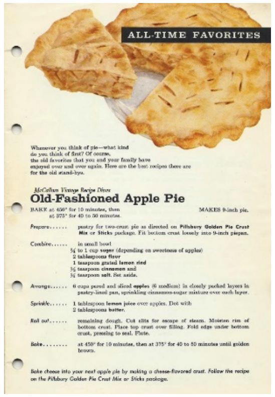
Nucleus

Cell

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

The Central Dogma of Molecular Biology





ALL-TIME FAVORITES

Whenever you think of pie—what kind do you think of first? Of course, the old favorites that you and your family have enjoyed over and over again. Here are the best recipes there are for the old stand-bys.

McCallum Vintage Recipe Diner

Old-Fashioned Apple Pie

BAKE at 450° for 10 minutes, then at 375° for 40 to 50 minutes. MAKES 9-inch pie.

Prepare...... pastry for two-crust pie as directed on **Pillsbury Golden Pie Crust Mix** or **Sticks** package. Fit bottom crust loosely into 9-inch piepan.

Combine...... in small bowl
¾ to 1 cup **sugar** (depending on sweetness of apples)
2 tablespoons **flour**
1 teaspoon grated **lemon rind**
½ teaspoon **cinnamon** and
½ teaspoon **salt**. Set aside.

Arrange...... 6 cups pared and sliced **apples** (6 medium) in closely packed layers in pastry-lined pan, sprinkling cinnamon-sugar mixture over each layer.

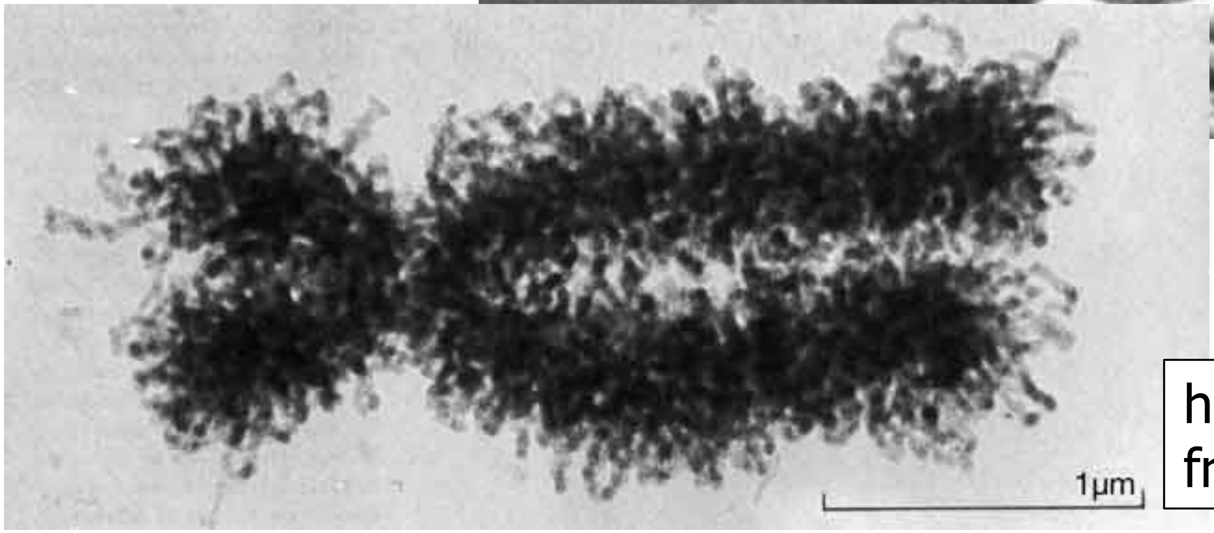
Sprinkle...... 1 tablespoon **lemon juice** over apples. Dot with 2 tablespoons **butter**.

Roll out...... remaining dough. Cut slits for escape of steam. Moisten rim of bottom crust. Place top crust over filling. Fold edge under bottom crust, pressing to seal. Flute.

Bake........ at 450° for 10 minutes, then at 375° for 40 to 50 minutes until golden brown.

Bake cheese into your next apple pie by making a cheese-flavored crust. Follow the recipe on the Pillsbury Golden Pie Crust Mix or Sticks package.

DNA Under Electron Microscope



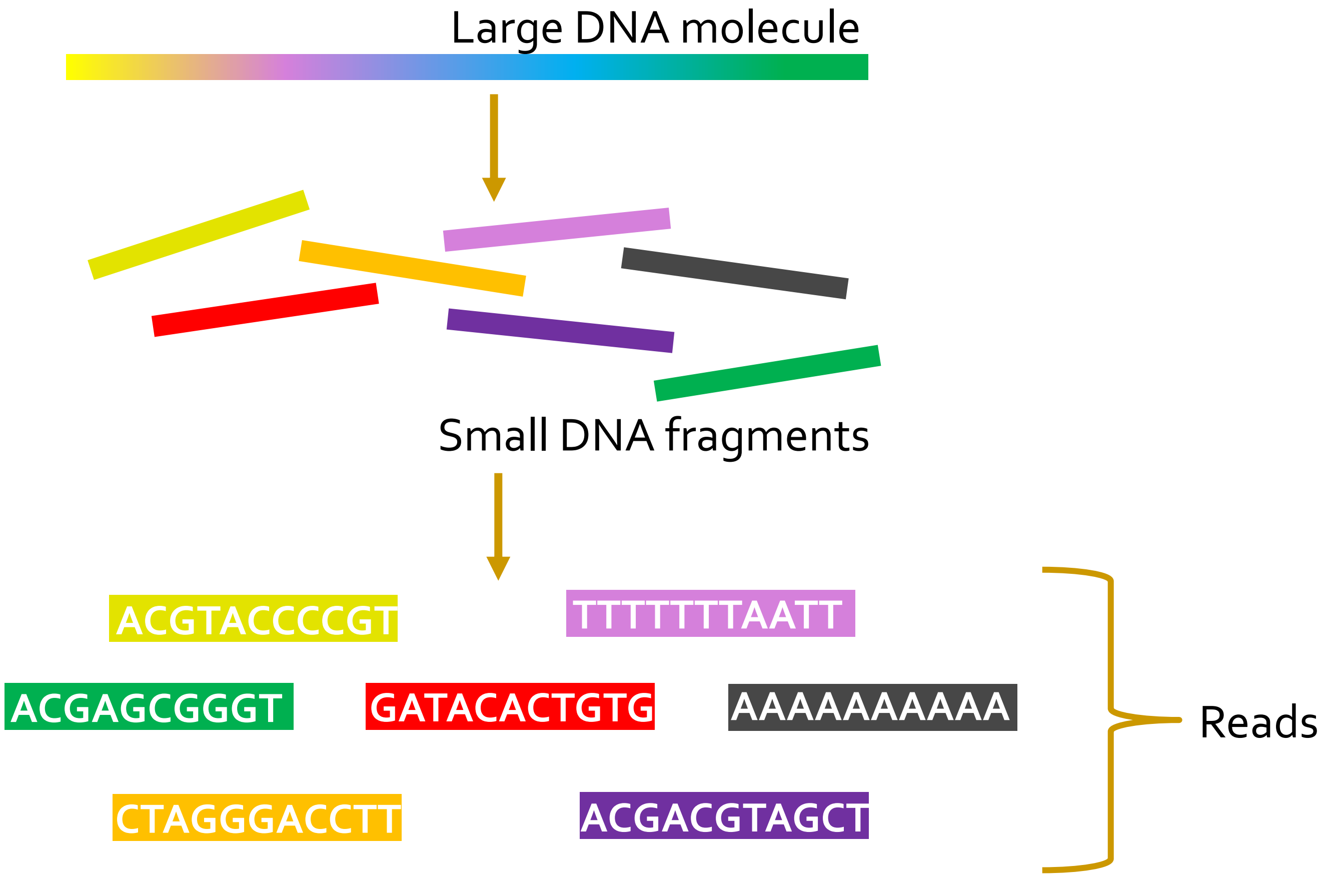
human chromosome #12
from HeLa’s cell

DNA Sequencing

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

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

Genome Sequencing



Large DNA molecule

Small DNA fragments

ACGTACCCCGT

TTTTTTTAATT

ACGAGCGGGT

GATACACTGTG

AAAAAAAAAA

CTAGGGACCTT

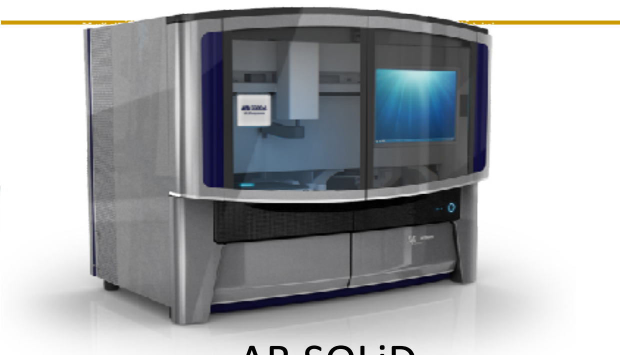
ACGACGTAGCT

Reads

Untangling Yarn Balls & DNA Sequencing

Genome Sequencers

Roche/454



AB SOLiD

Illumina MiSeq

Complete Genomics

Oxford Nanopore MinION

Illumina NovaSeq 6000

Illumina HiSeq2000

Pacific Biosciences RS

Oxford Nanopore GridION

Ion Torrent PGM

Ion Torrent Proton

… and more! All produce data with different properties.

High-Throughput Sequencers

Illumina MiSeq

Pacific Biosciences Sequel II

Oxford Nanopore PromethION

Oxford Nanopore MinION

Illumina NovaSeq 6000

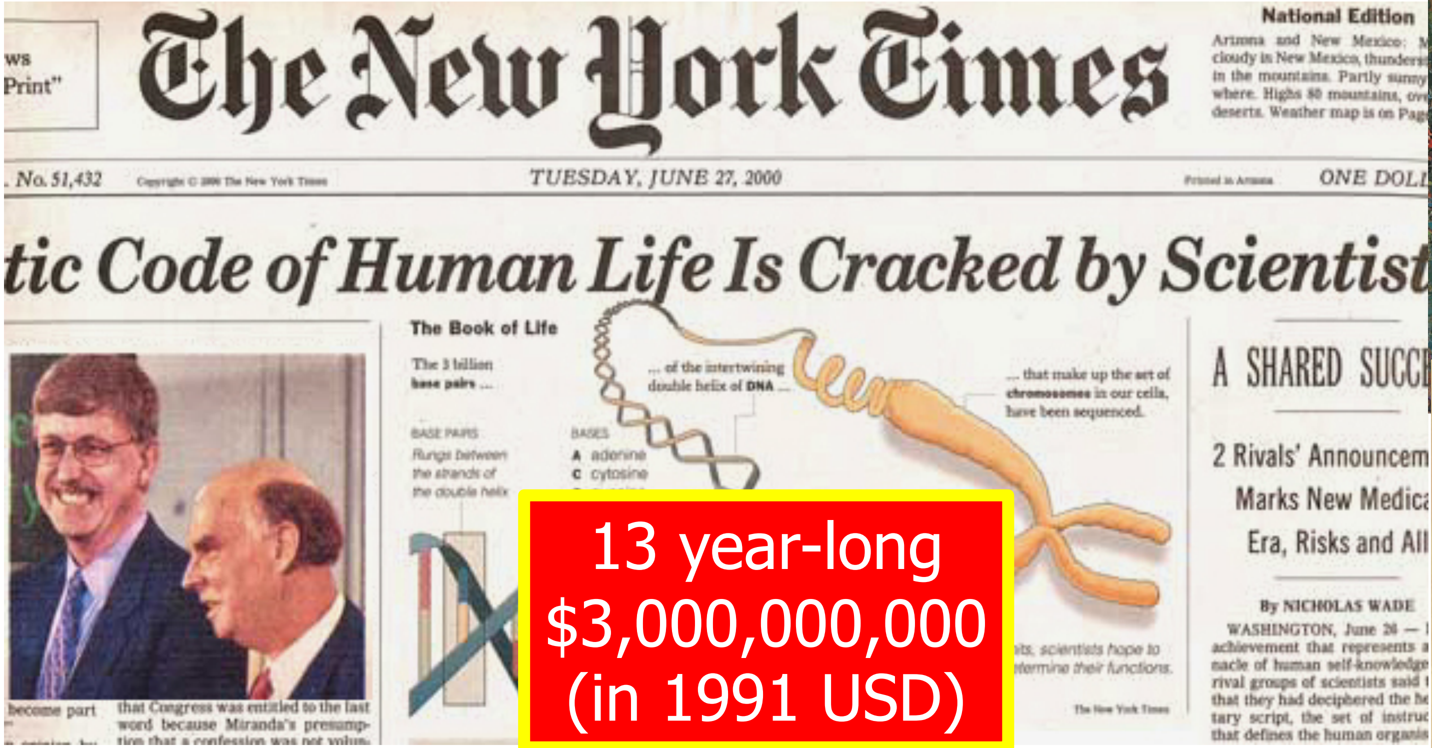
Pacific Biosciences RS II

Oxford Nanopore SmidgION

… and more! All produce data with different properties.

The Genomic Era

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



ws Print"

The New York Times

National Edition

Arizona and New Mexico: M cloudy in New Mexico, thunders in the mountains. Partly sunny where. Highs 80 mountains, ov deserts. Weather map is on Pag

No. 51,432 Copyright © 2000 The New York Times TUESDAY, JUNE 27, 2000 Printed in Arizona ONE DOLL

tic Code of Human Life Is Cracked by Scientist

A SHARED SUCC

2 Rivals' Announcem Marks New Medica Era, Risks and All

By NICHOLAS WADE

WASHINGTON, June 26 — I achievement that represents a nacle of human self-knowledge rival groups of scientists said t that they had deciphered the he tary script, the set of instruc that defines the human organis

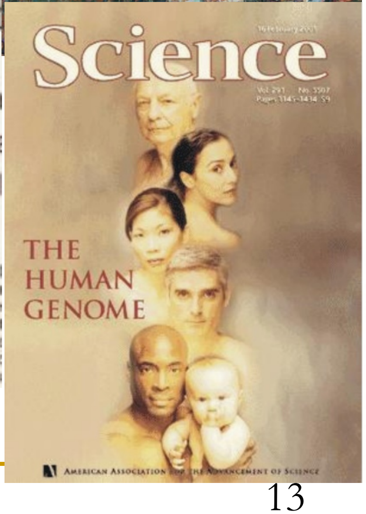
become part

that Congress was entitled to the last word because Miranda's presumption that a confession was not volun-

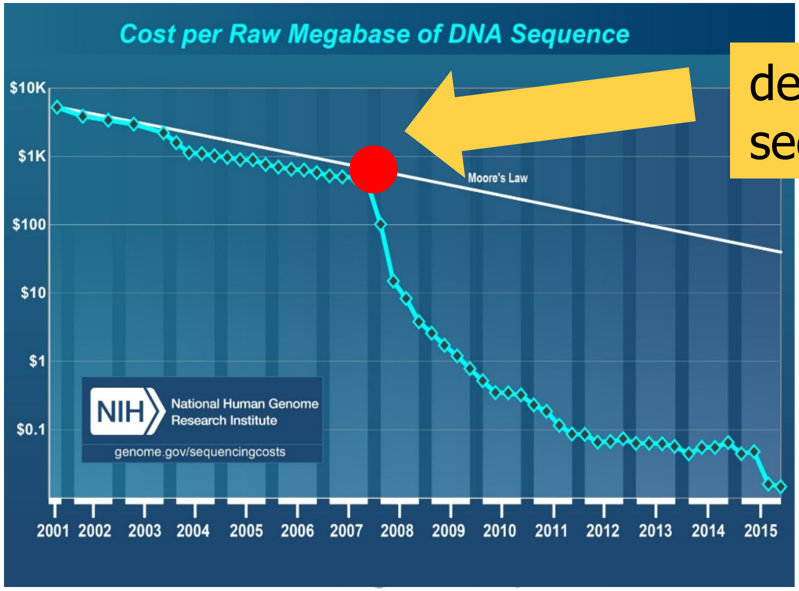
The Book of Life
The 3 billion base pairs ...
... of the intertwining double helix of DNA ...
... that make up the set of chromosomes in our cells, have been sequenced.
BASE PAIRS
Rungs between the strands of the double helix
BASES
A adenine
C cytosine
its, scientists hope to etermine their functions.
The New York Times

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

15 February 2001
nature
www.nature.com
the human genome
Nuclear fission
Five-dimensional energy landscapes
Seafloor spreading
The view from under the Arctic ice
Career prospects
Sequence creates new opportunities

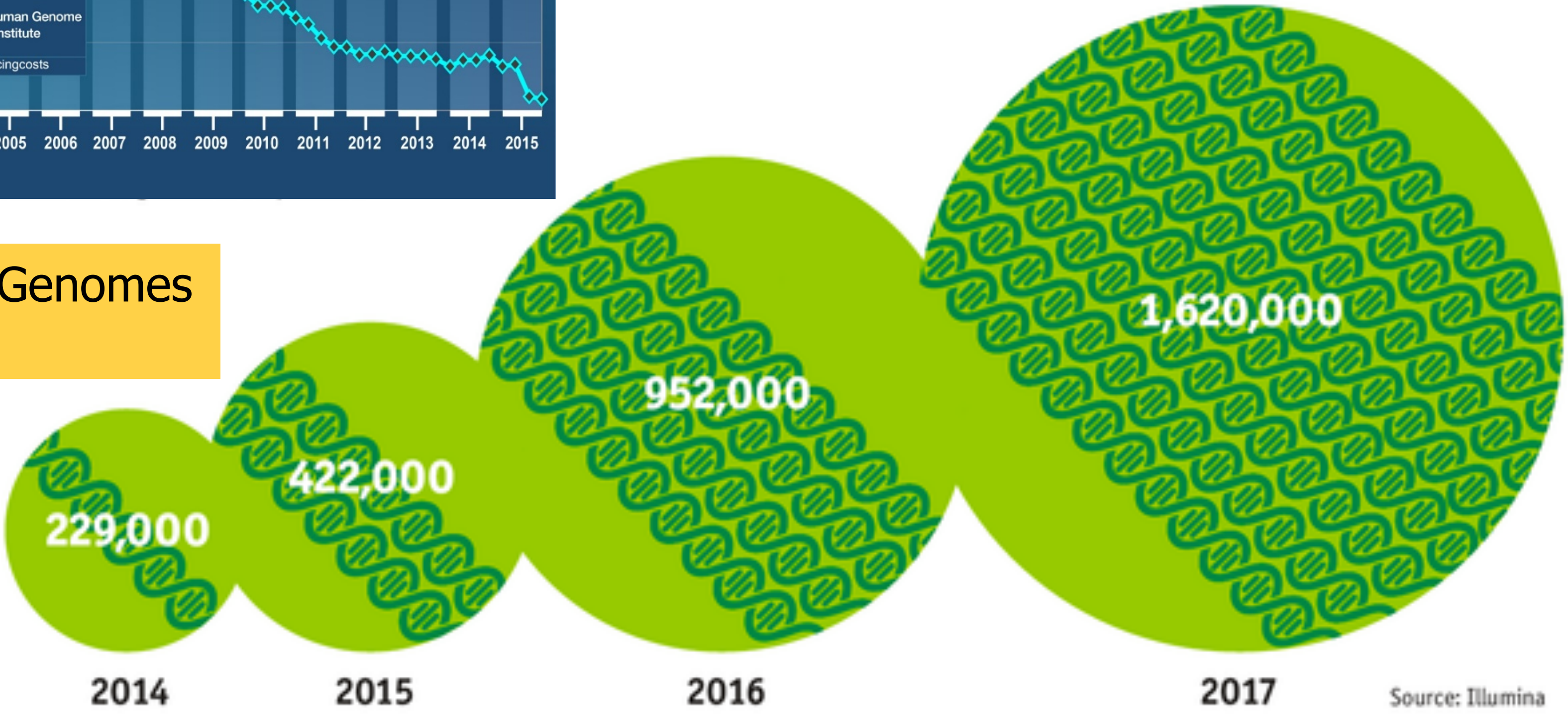

16 February 2001
Science
Vol. 291 No. 5507 Pages 1145–1434 $9
THE HUMAN GENOME
American Association for the Advancement of Science

The Genomic Era (continued)



development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

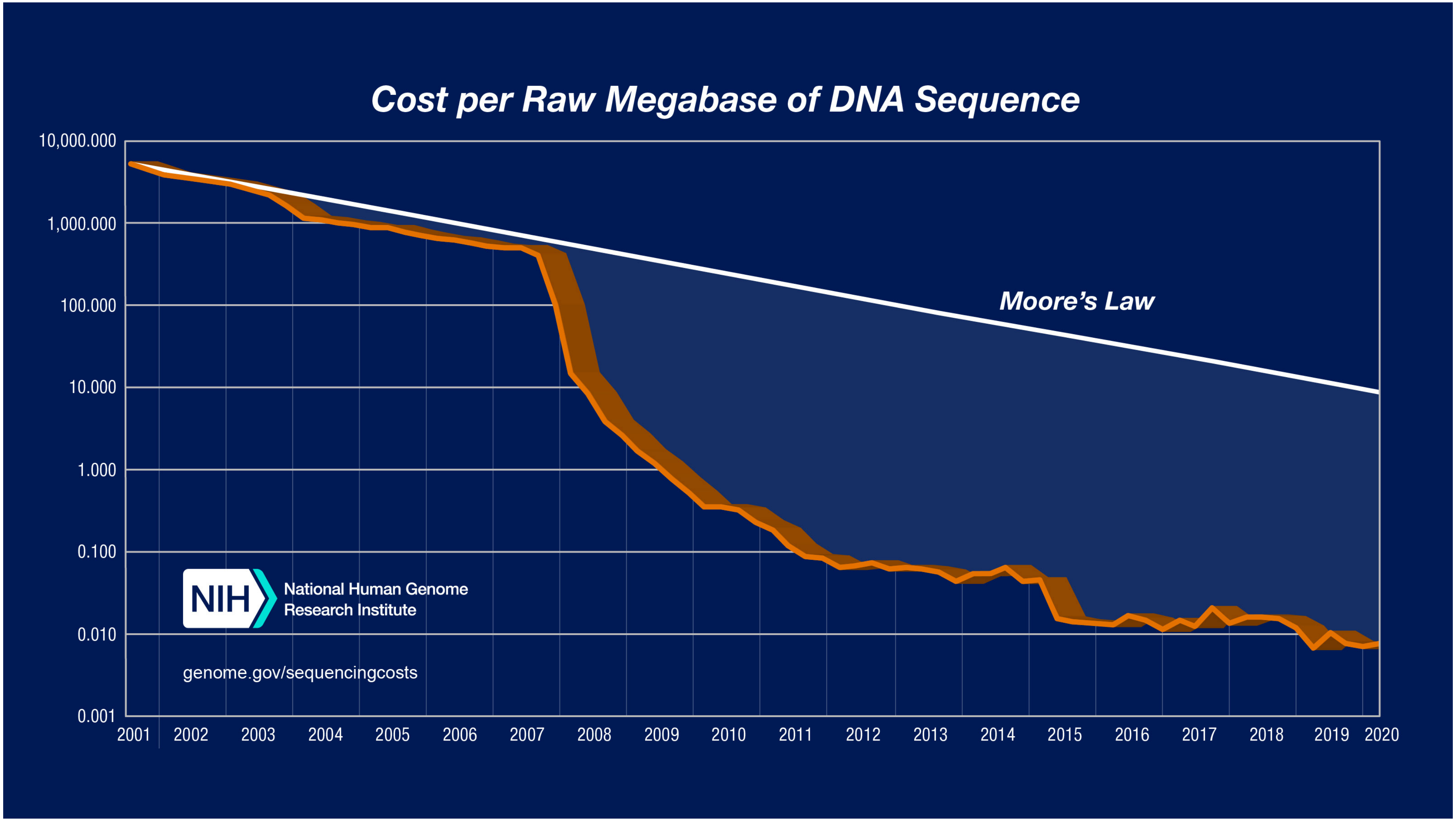


Source: Illumina

The Economist

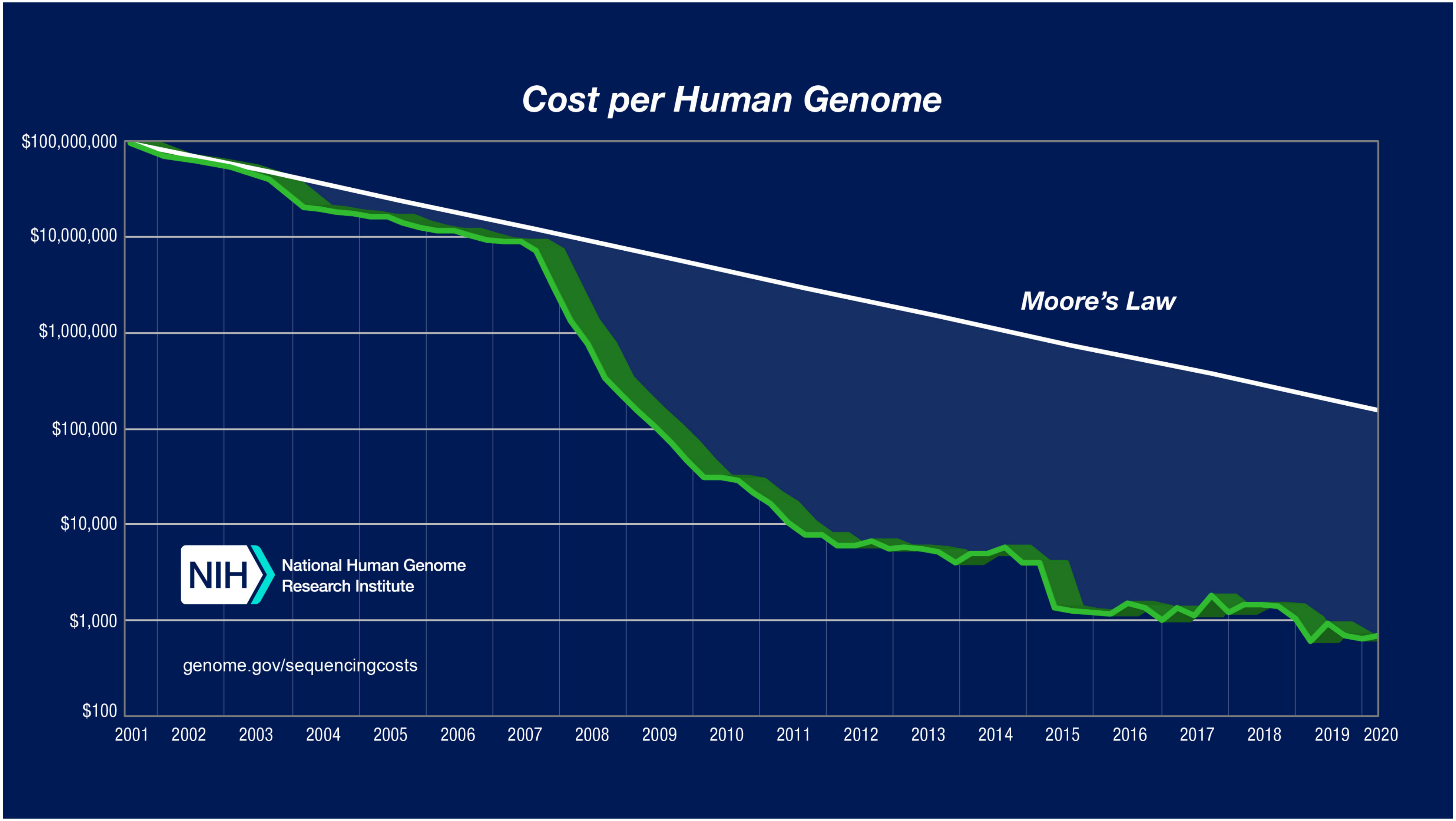
http://www.economist.com/news/21631808-so-much-genetic-data-so-many-uses-genes-unzipped

Cost of Sequencing



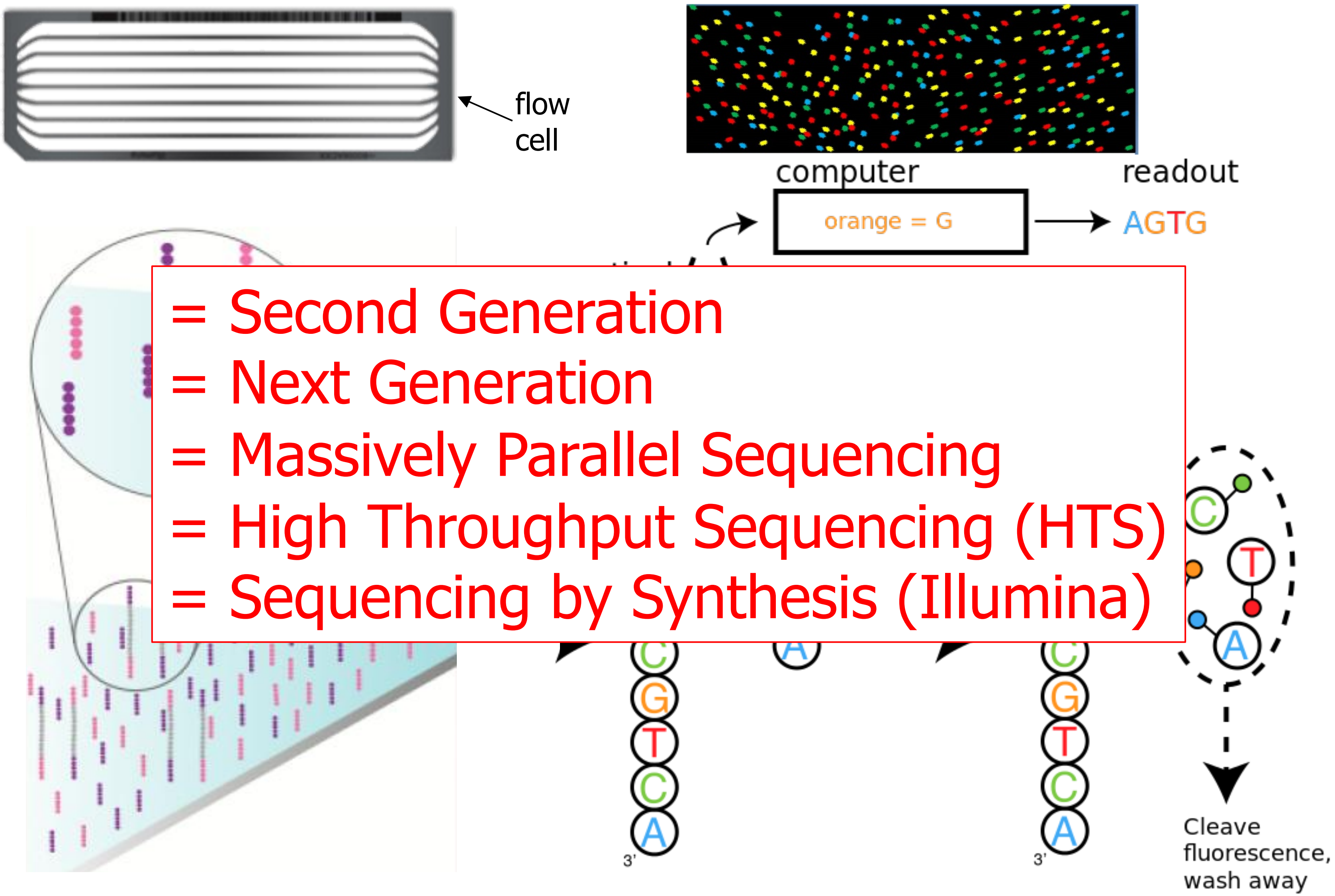
*From NIH (https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data)

Cost of Sequencing (cont.)



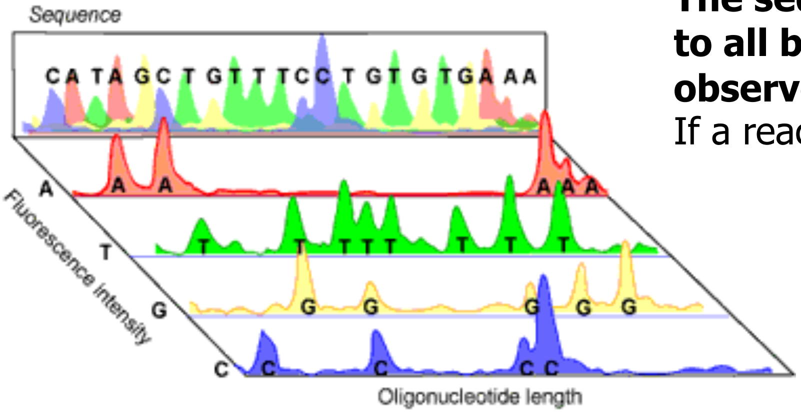
*From NIH (https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data)

High-Throughput Sequencing (HTS)

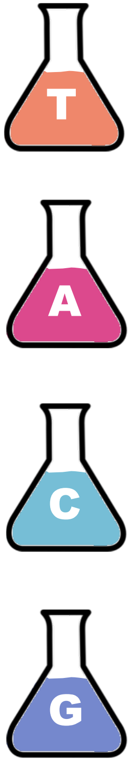


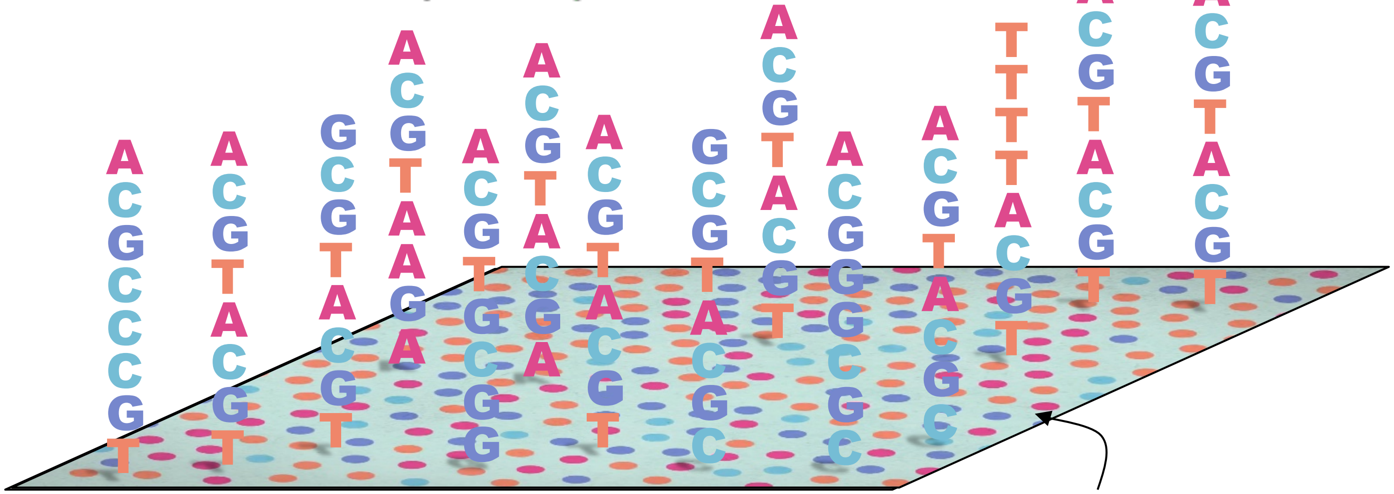
= Second Generation
= Next Generation
= Massively Parallel Sequencing
= High Throughput Sequencing (HTS)
= Sequencing by Synthesis (Illumina)

High-Throughput Sequencing (HTS)



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





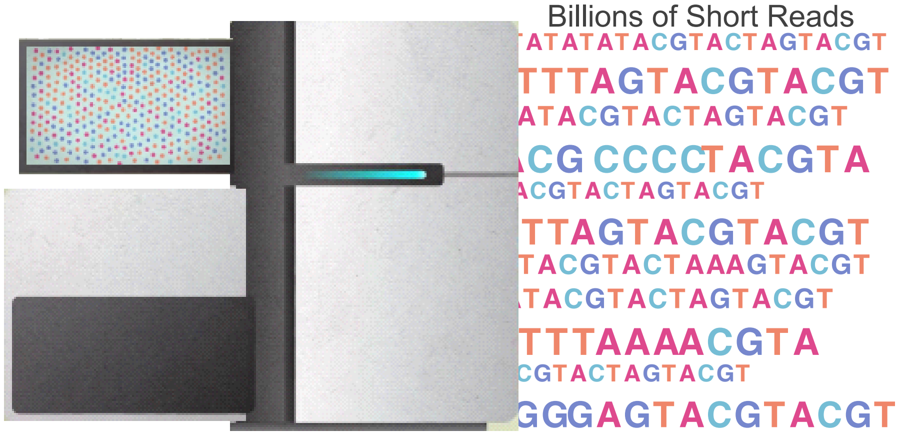
Glass flow cell surface

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

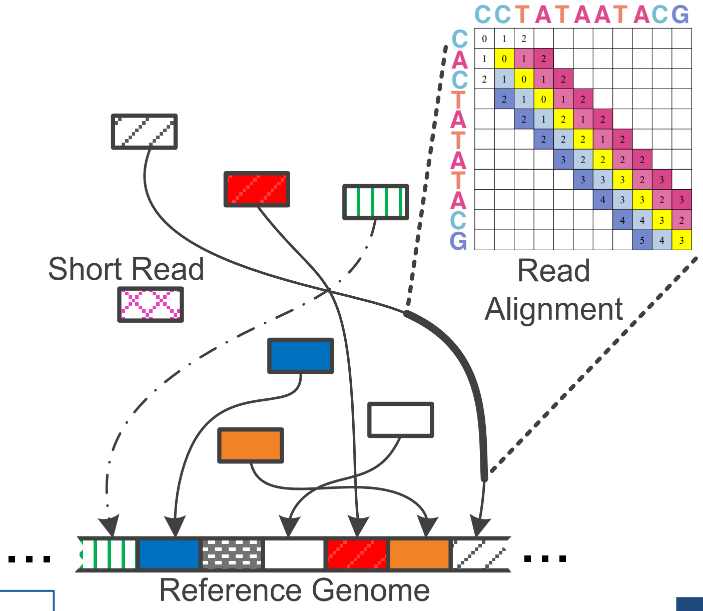
High-Throughput Sequencing

- 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

Genome Analysis



1 Sequencing



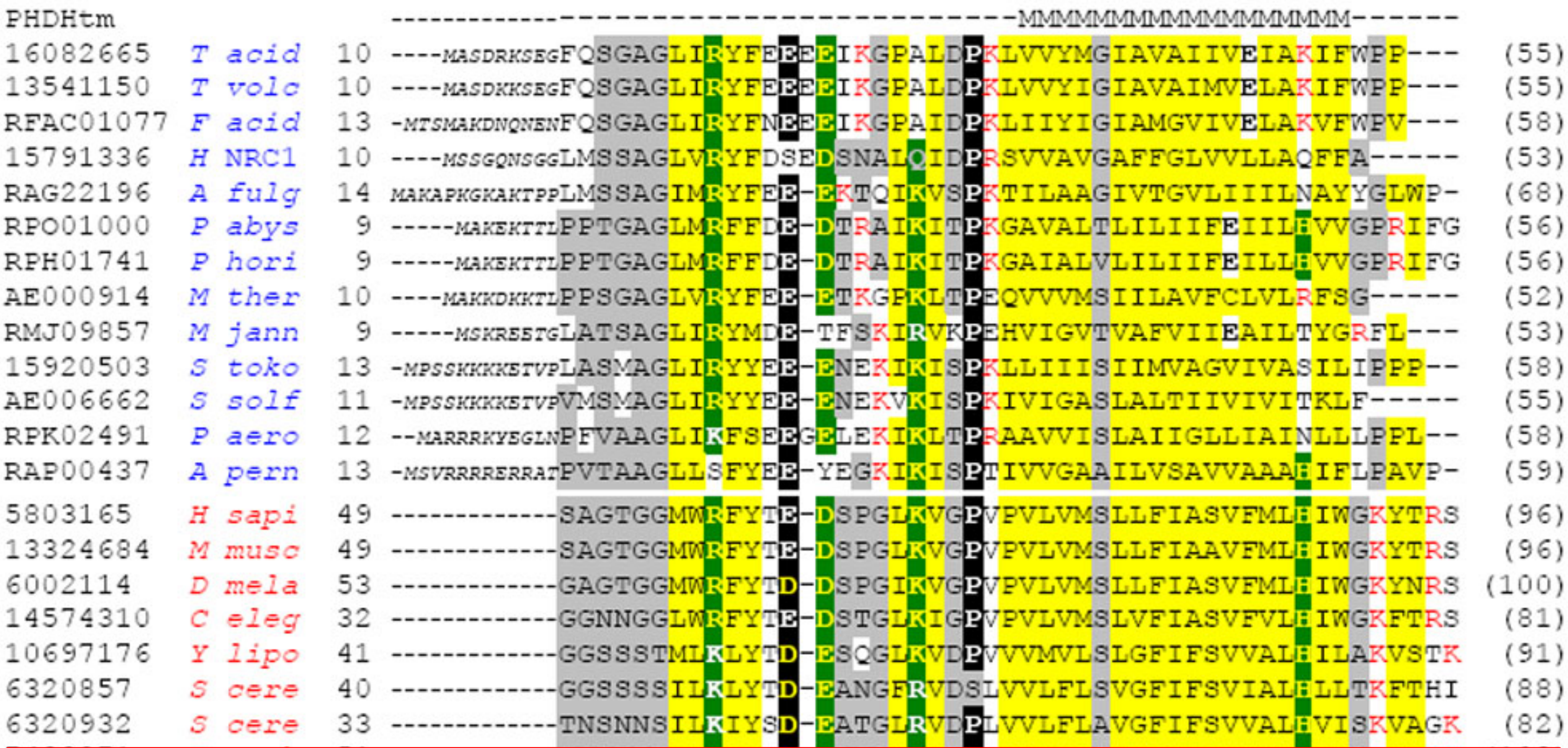
2 Read Mapping

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

3 Variant Calling

4 Scientific Discovery

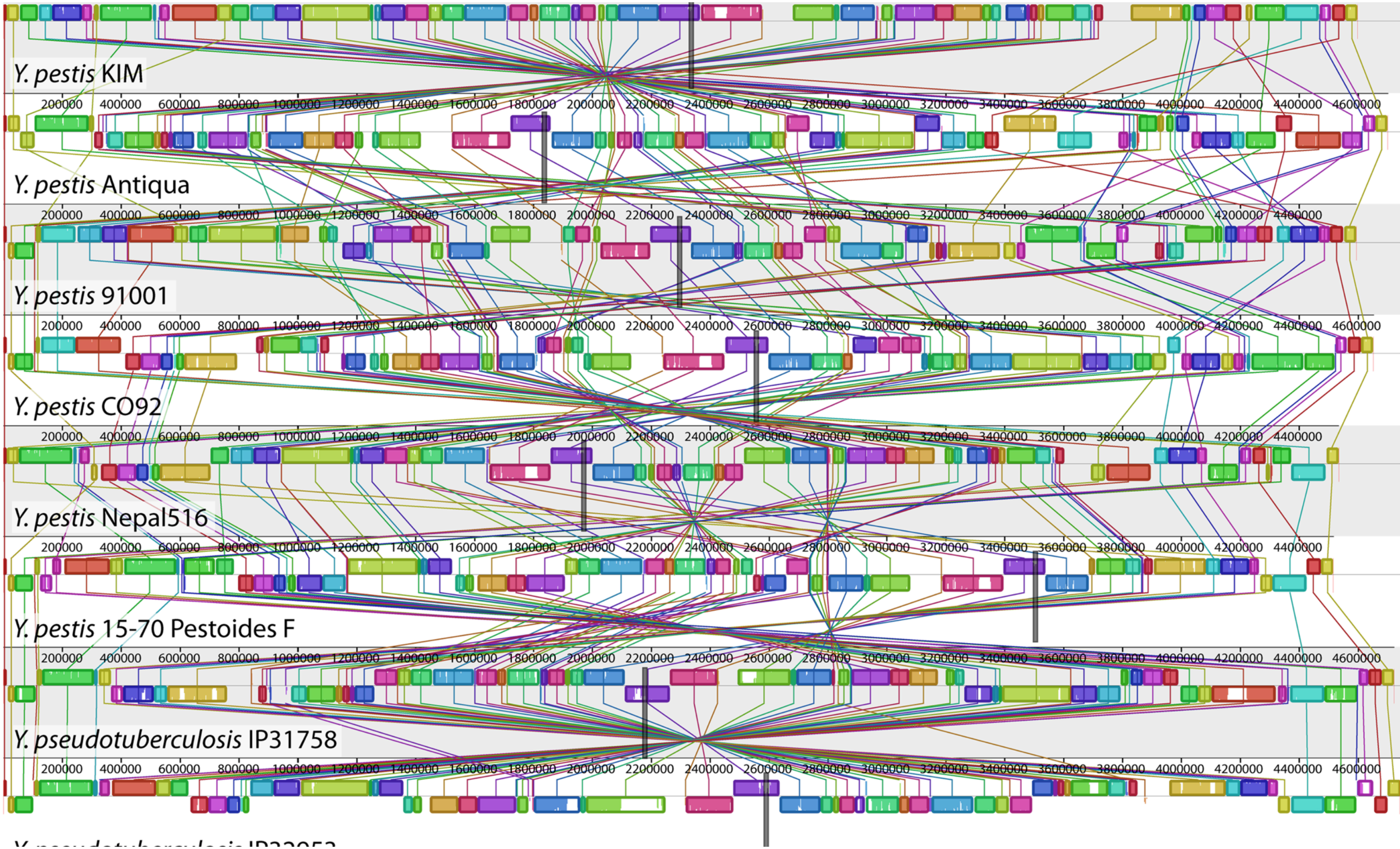
Multiple sequence alignment



```
PHDHtm                              --------------------------------MMMMMMMMMMMMMMMMMMM------
16082665    T acid    10  ----MASDRKSEGFQSGAGLIRYFEEEEIKGPALDPKLVVYMGIAVAIIVEIAKIFWPP---   (55)
13541150    T volc    10  ----MASDKKSEGFQSGAGLIRYFEEEEIKGPALDPKLVVYIGIAVAIMVELAKIFWPP---   (55)
RFAC01077   F acid    13  -MTSMAKDNQNENFQSGAGLIRYFNEEEIKGPAIDPKLIIYIGIAMGVIVELAKVFWPV---   (58)
15791336    H NRC1    10  ----MSSGQNSGGLMSSAGLVRYFDSEDSNALQIDPRSVVAVGAFFGLVVLLAQFFA-----   (53)
RAG22196    A fulg    14  MAKAPKGKAKTPPLMSSAGIMRYFEE-EKTQIKVSPKTILAAGIVTGVLIIILNAYYGLWP-   (68)
RPO01000    P abys     9  -----MAKEKTTLPPTGAGLMRFFDE-DTRAIKITPKGAVALTLILIIFEIILHVVGPRIFG   (56)
RPH01741    P hori     9  -----MAKEKTTLPPTGAGLMRFFDE-DTRAIKITPKGAIALVLILIIFEILLHVVGPRIFG   (56)
AE000914    M ther    10  ----MAKKDKKTLPPSGAGLVRYFEE-ETKGPKLTPEQVVVMSIILAVFCLVLRFSG-----   (52)
RMJ09857    M jann     9  -----MSKREETGLATSAGLIRYMDE-TFSKIRVKPEHVIGVTVAFVIIEAILTYGRFL---   (53)
15920503    S toko    13  -MPSSKKKKETVPLASMAGLIRYYEE-ENEKIKISPKLLIIISIIMVAGVIVASILIPPP--   (58)
AE006662    S solf    11  -MPSSKKKKETVPVMSMAGLIRYYEE-ENEKVKISPKIVIGASLALTIIVIVITKLF-----   (55)
RPK02491    P aero    12  --MARRRKYEGLNPFVAAGLIKFSEEGELEKIKLTPRAAVVISLAIIGLLIAINLLLPPL--   (58)
RAP00437    A pern    13  -MSVRRRRERRATPVTAAGLLSFYEE-YEGKIKISPTIVVGAAILVSAVVAAAHIFLPAVP-   (59)
5803165     H sapi    49  -------------SAGTGGMWRFYTE-DSPGLKVGPVPVLVMSLLFIASVFMLHIWGKYTRS   (96)
13324684    M musc    49  -------------SAGTGGMWRFYTE-DSPGLKVGPVPVLVMSLLFIAAVFMLHIWGKYTRS   (96)
6002114     D mela    53  -------------GAGTGGMWRFYTD-DSPGIKVGPVPVLVMSLLFIASVFMLHIWGKYNRS  (100)
14574310    C eleg    32  -------------GGNNGGLWRFYTE-DSTGLKIGPVPVLVMSLVFIASVFVLHIWGKFTRS   (81)
10697176    Y lipo    41  -------------GGSSSTMLKLYTD-ESQGLKVDPVVVMVLSLGFIFSVVALHILAKVSTK   (91)
6320857     S cere    40  -------------GGSSSSILKLYTD-EANGFRVDSLVVLFLSVGFIFSVIALHLLTKFTHI   (88)
6320932     S cere    33  -------------TNSNNSILKIYSD-EATGLRVDPLVVLFLAVGFIFSVVALHVISKVAGK   (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

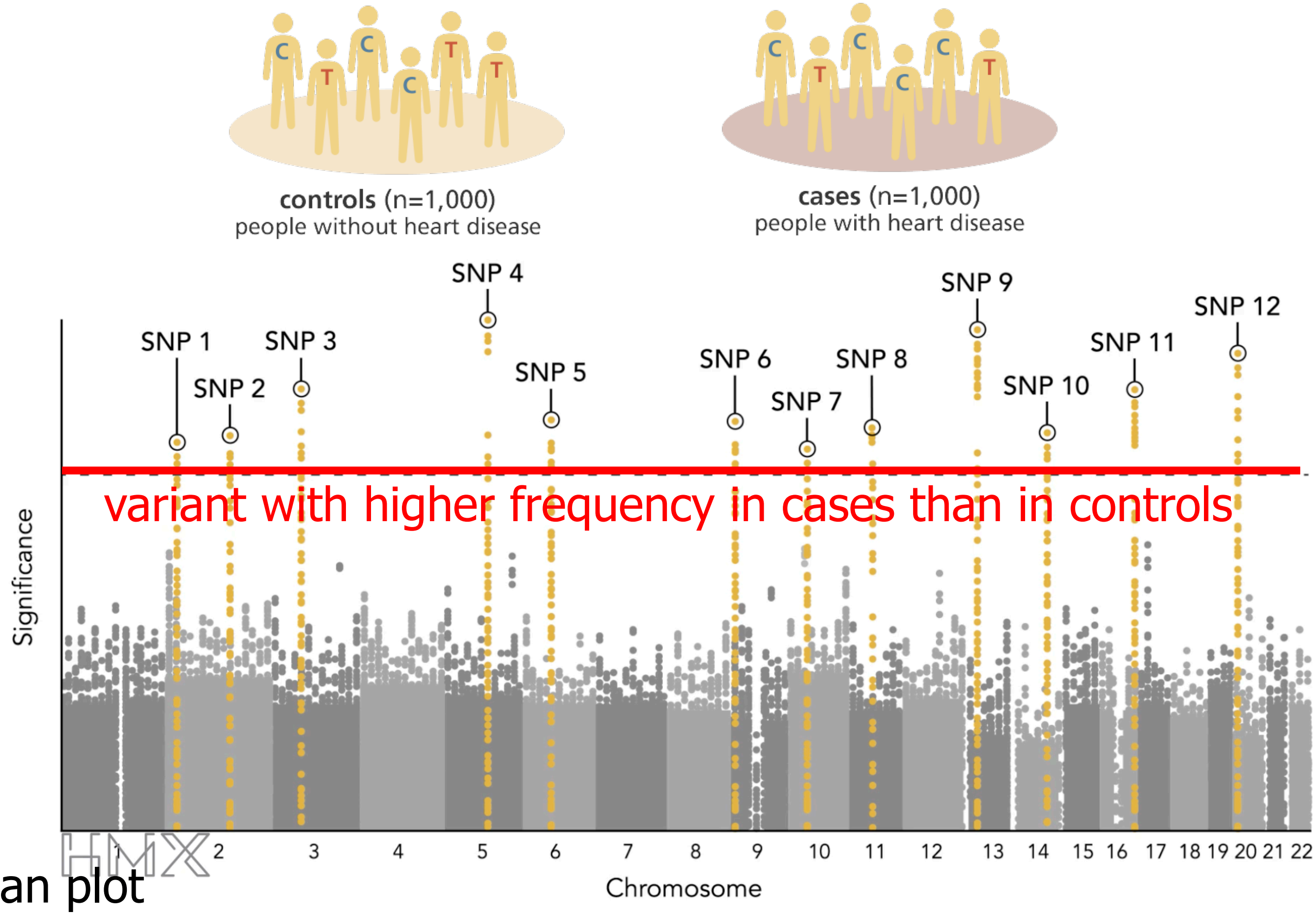
	SNP1 SNP2	Blood Pressure
Individual #1	...ACATGCCGACATTTCATAGGCC...	180
Individual #2	...ACATGCCGACATTTCATAAGCC...	175
Individual #3	...ACATGCCGACATTTCATAGGCC...	170
Individual #4	...ACATGCCGACATTTCATAAGCC...	165
Individual #5	...ACATGCCGACATTTCATAGGCC...	160
Individual #6	...ACATGCCGACATTTCATAGGCC...	145
Individual #7	...ACATGCCGACATTTCATAAGCC...	140
Individual #8	...ACATGCCGACATTTCATAAGCC...	130
Individual #9	...ACATGTCGACATTTCATAGGCC...	120
Individual #10	...ACATGTCGACATTTCATAAGCC...	120
Individual #11	...ACATGTCGACATTTCATAGGCC...	115
Individual #12	...ACATGTCGACATTTCATAAGCC...	110
Individual #13	...ACATGTCGACATTTCATAGGCC...	110
Individual #14	...ACATGTCGACATTTCATAAGCC...	110
Individual #15	...ACATGTCGACATTTCATAGGCC...	105
Individual #16	...ACATGTCGACATTTCATAAGCC...	100

SNP: single nucleotide polymorphism

SAFARI Eleazar Eskin: Discovering the Causal Variants Involved in GWAS Studies, CGSI 2018, UCLA

Genome-Wide Association Studies (GWAS)

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

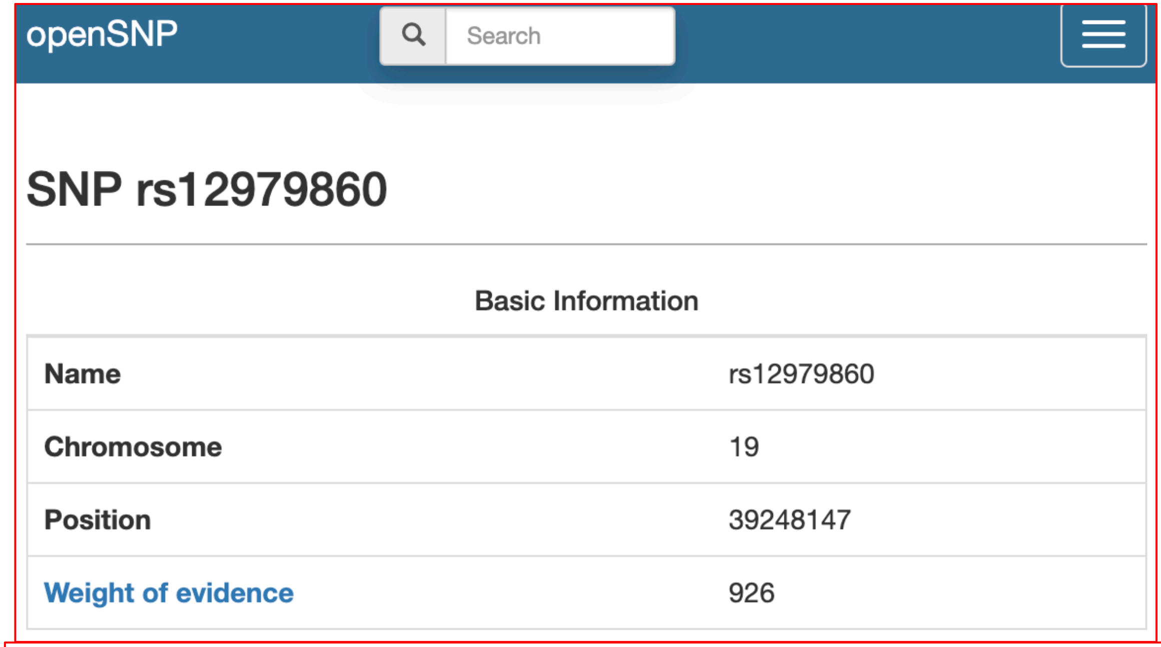


variant with higher frequency in cases than in controls

Manhattan plot

https://onlinelearning.hms.harvard.edu/hmx/courses/genetic-testing/

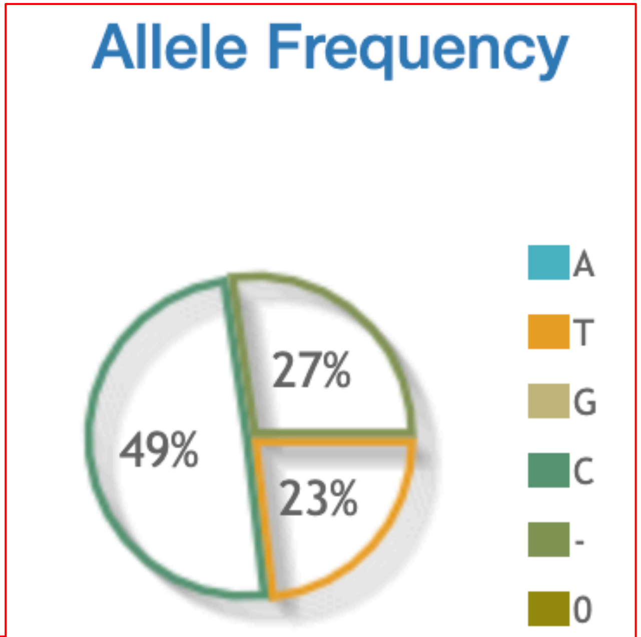
SNPs and Personalized Medicine

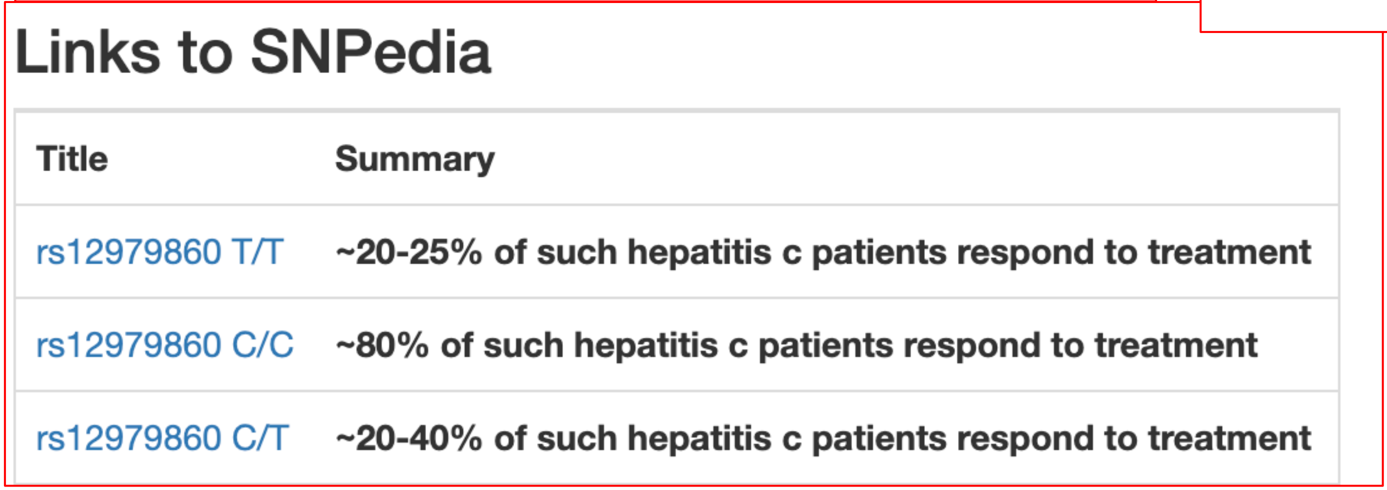


SNP rs12979860

Basic Information

Name	rs12979860
Chromosome	19
Position	39248147
Weight of evidence	926





Links to SNPedia

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

https://opensnp.org/snps/rs12979860

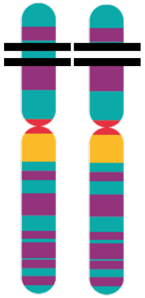
Much Larger Structural Variations

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

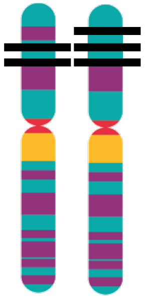
SCHIZOPHRENIA
McCarthy, *Nat Genet* 2009
Duplication of 593 kb

OBESITY
Walters, *Nature* 2010
Deletion of 593 kb

UNDERWEIGHT
Jacquemont, *Nature* 2011
Duplication of 593 kb



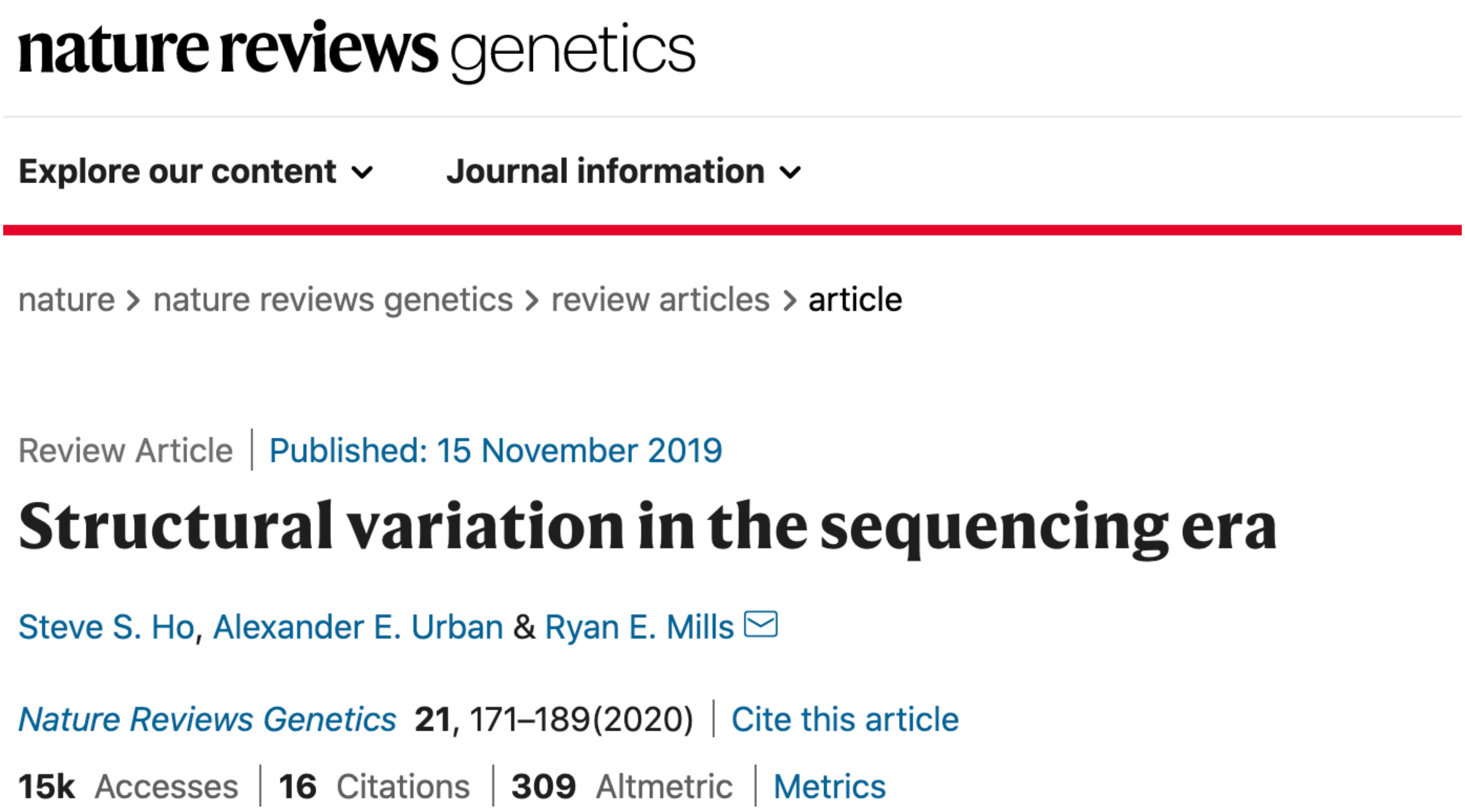
Deletion in the short arm
of chromosome 16 (16p11.2)



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

CNV: copy number variation

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 ✉

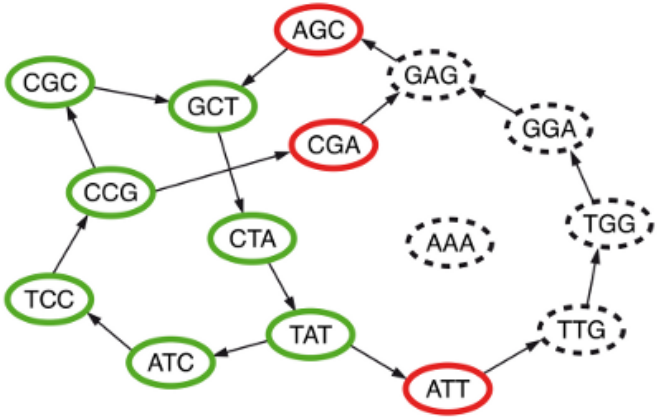
Nature Reviews Genetics **21**, 171–189(2020) | Cite this article

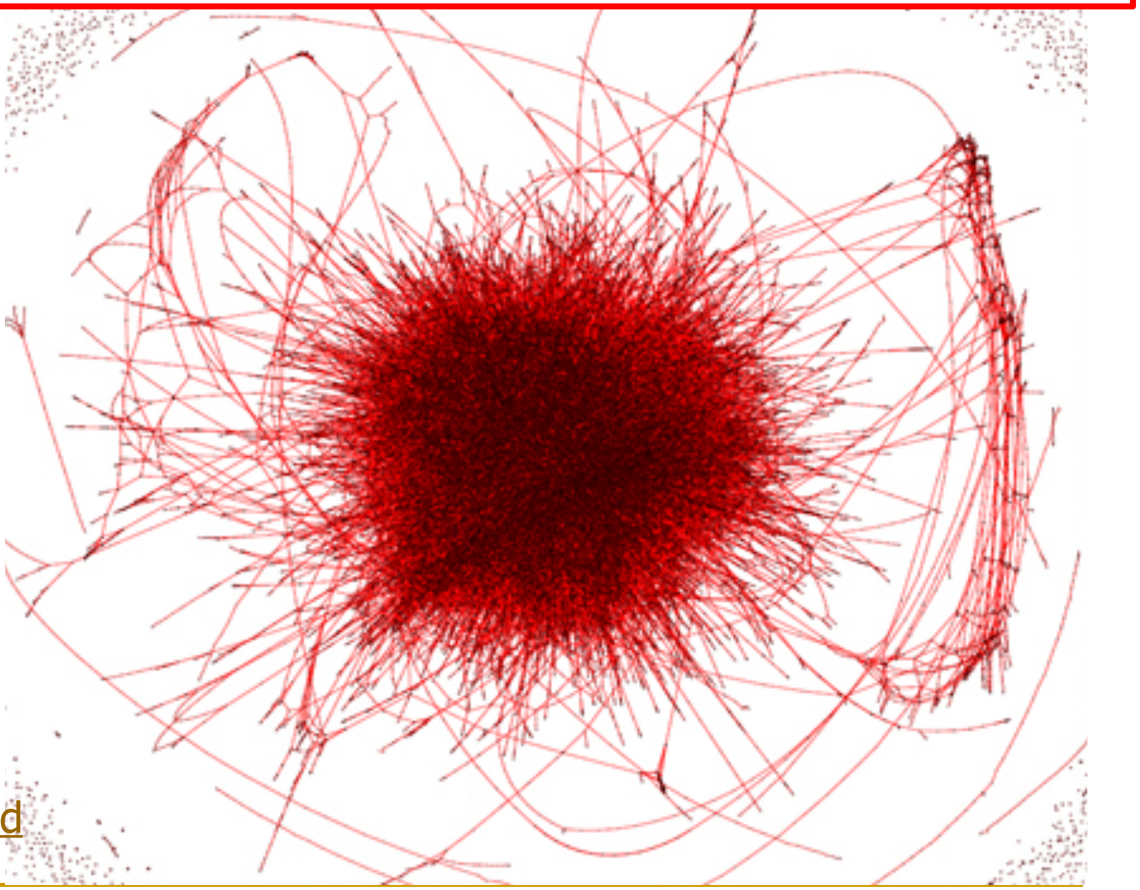
15k Accesses | **16** Citations | **309** Altmetric | Metrics

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

https://blog.wego.com/7-crowded-places-and-events-that-you-will-love/

City-Scale Microbiome Profiling

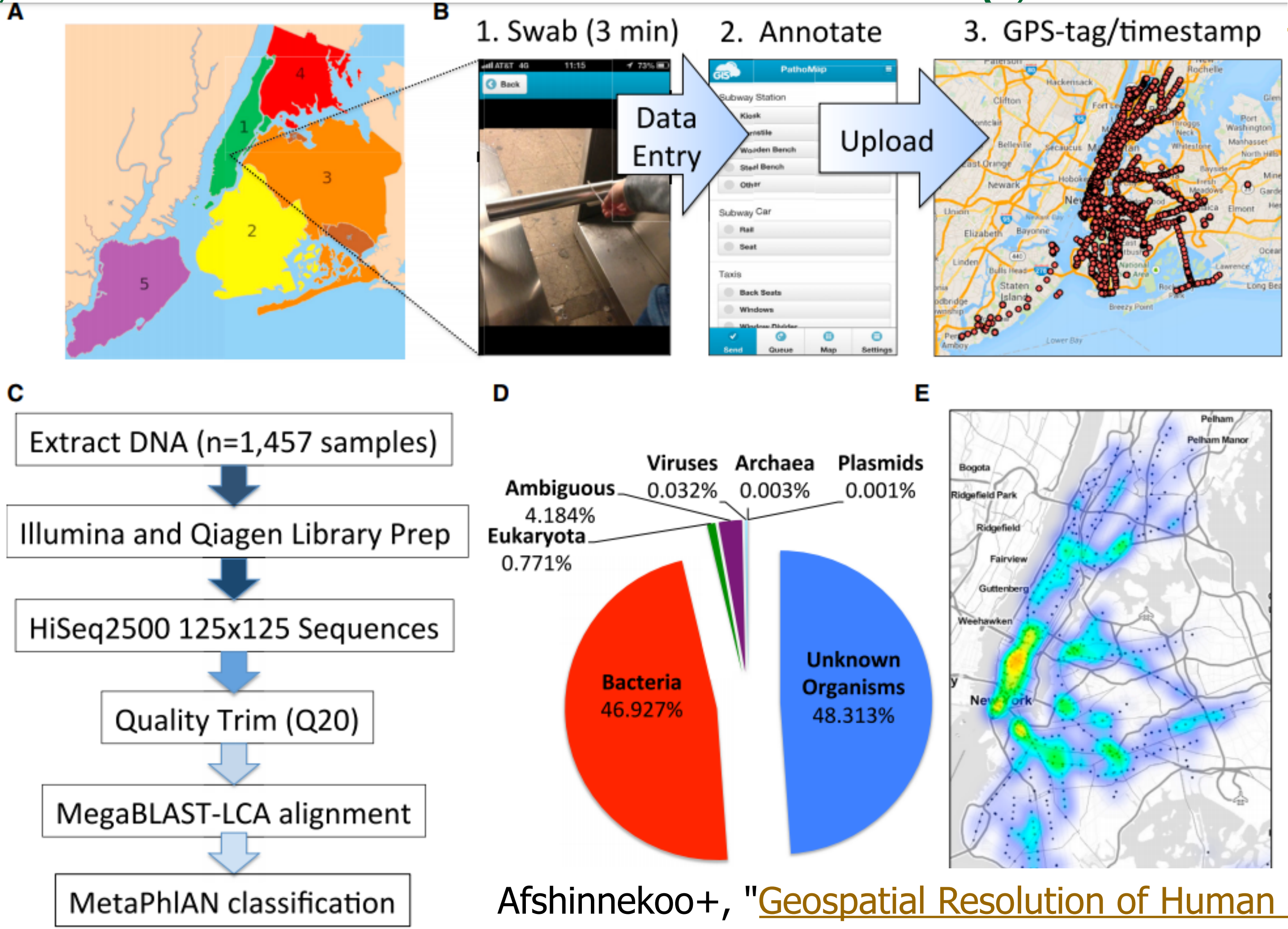


Figure 1. The Metagenome of New York City

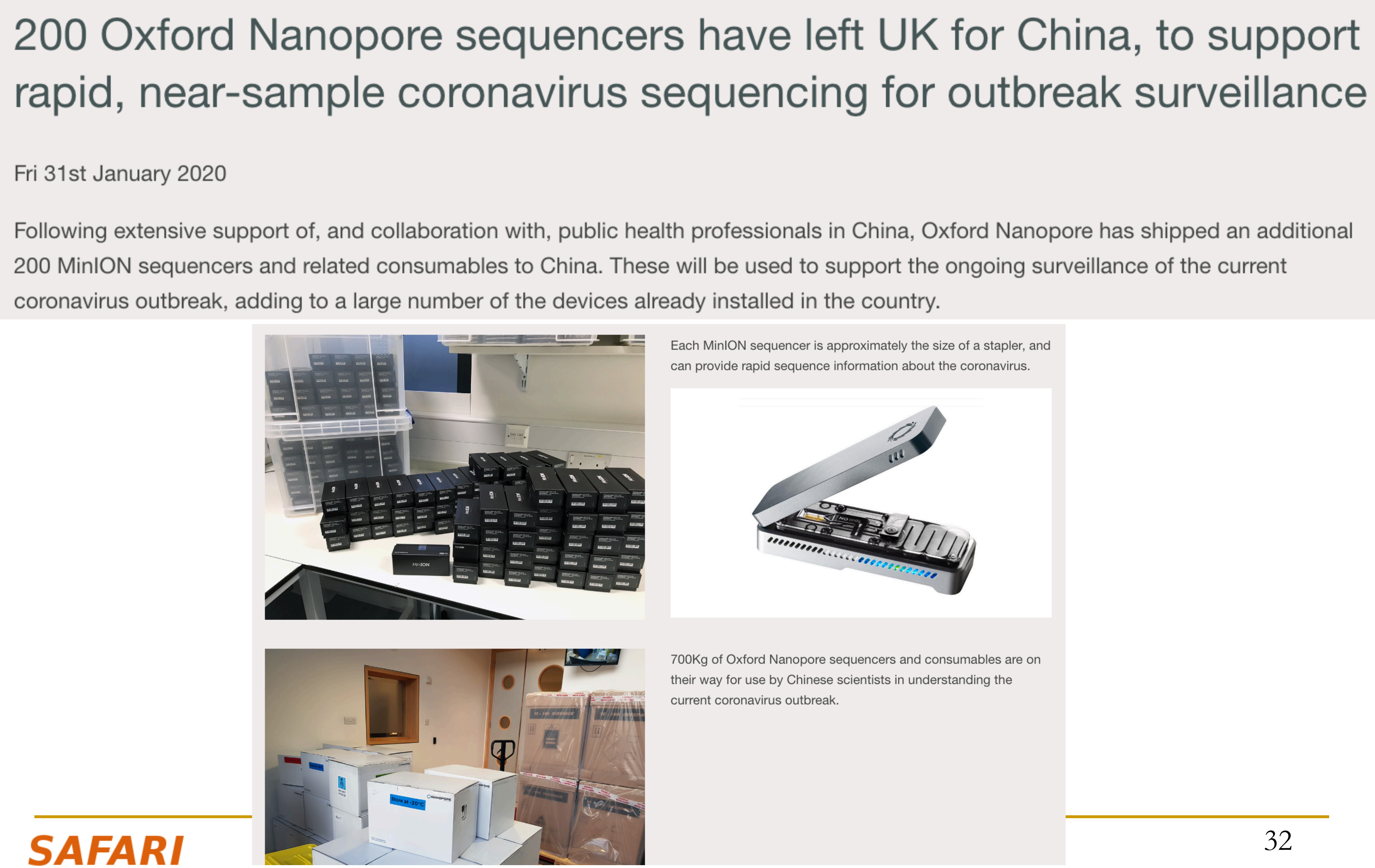
(A) The five boroughs of NYC include (1) Manhattan (green)

(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

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

Another Question: Example from 2020



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

Fri 31st January 2020

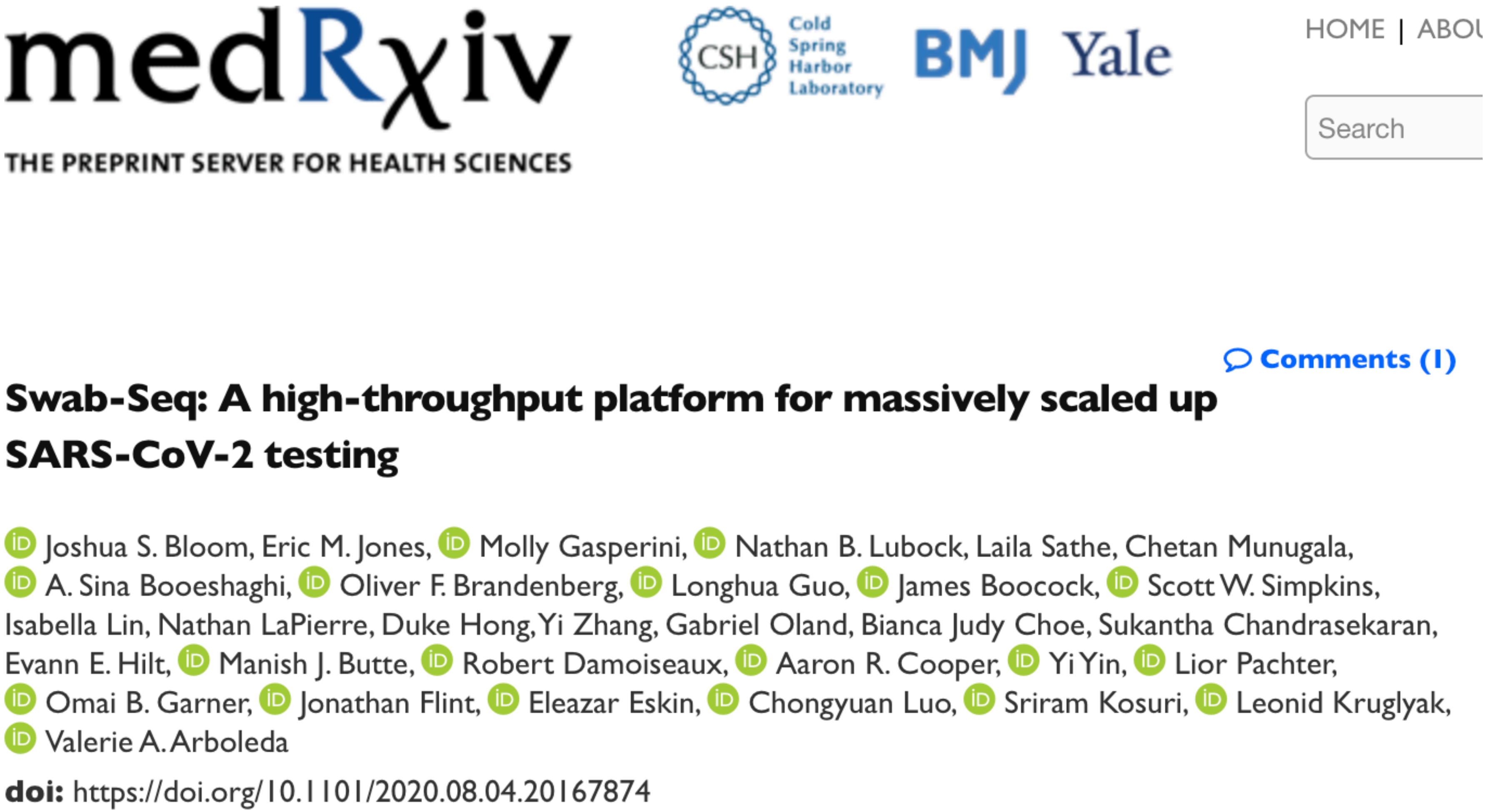
Following extensive support of, and collaboration with, public health professionals in China, Oxford Nanopore has shipped an additional 200 MinION sequencers and related consumables to China. These will be used to support the ongoing surveillance of the current coronavirus outbreak, adding to a large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

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

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

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

Example: Scalable SARS-CoV-2 Testing


medRxiv
THE PREPRINT SERVER FOR HEALTH SCIENCES
CSH Cold Spring Harbor Laboratory
BMJ
Yale
HOME | ABOU
Search
Comments (1)

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

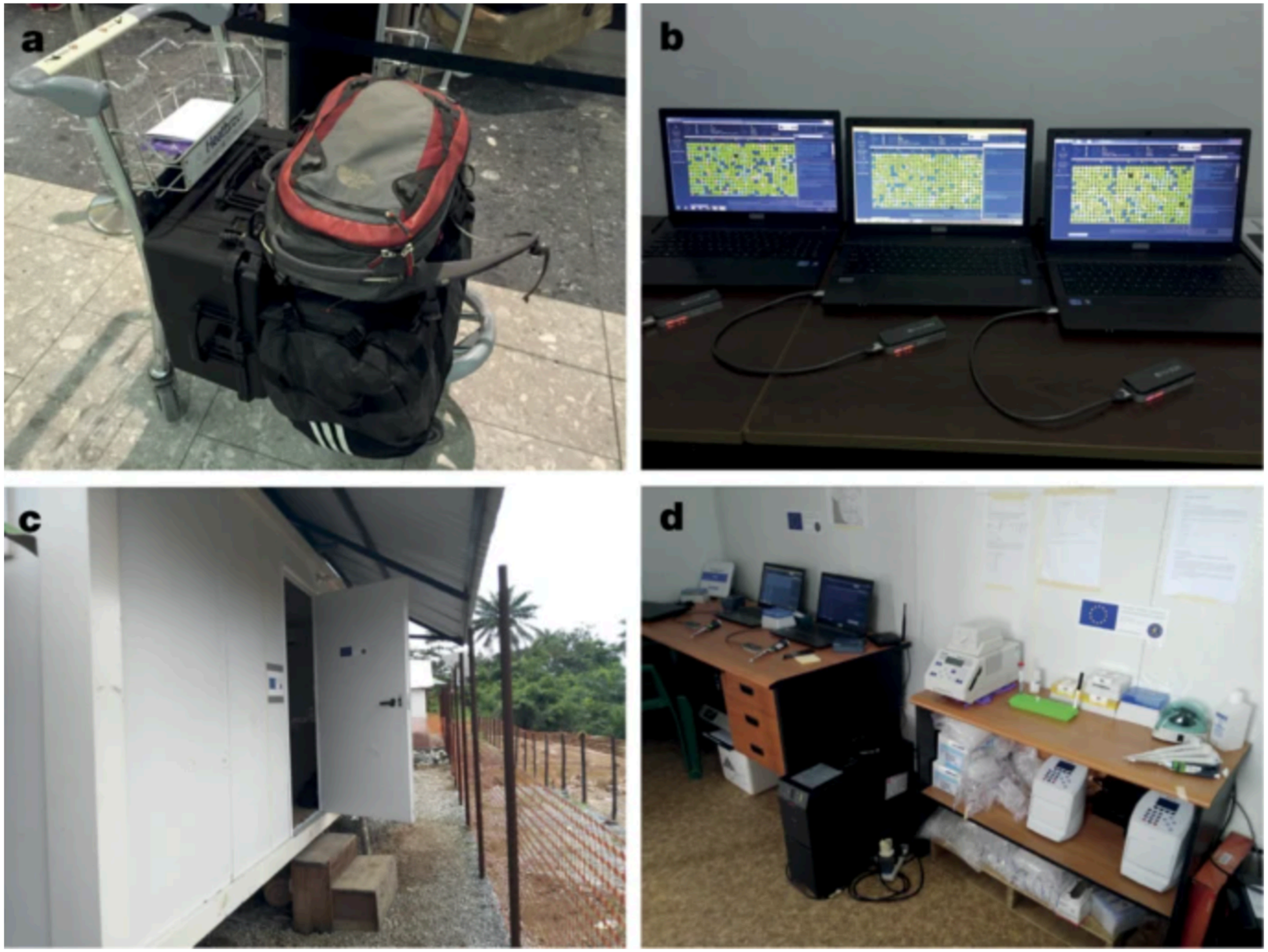
Joshua S. Bloom, Eric M. Jones, Molly Gasperini, Nathan B. Lubock, Laila Sathe, Chetan Munugala, A. Sina Booeshaghi, Oliver F. Brandenberg, Longhua Guo, James Boocock, Scott W. Simpkins, Isabella Lin, Nathan LaPierre, Duke Hong, Yi Zhang, Gabriel Oland, Bianca Judy Choe, Sukantha Chandrasekaran, Evann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

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

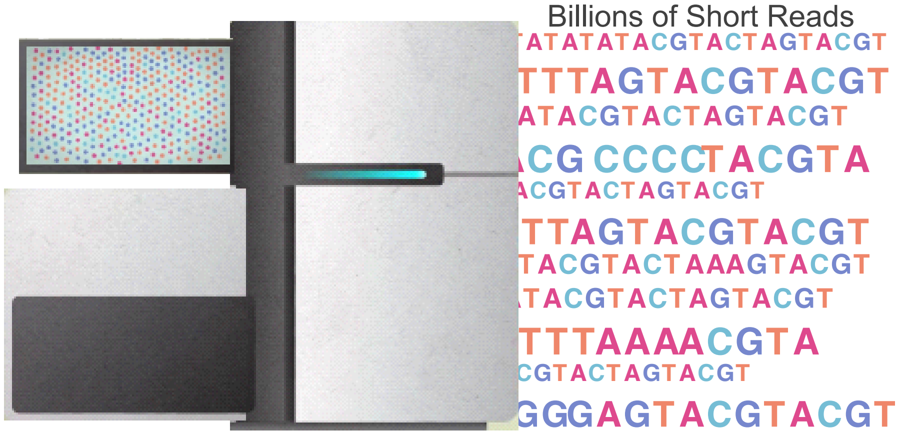
Bloom+, "Swab-Seq: A high-throughput platform for massively scaled up SARS-CoV-2 testing", *medRxiv*, 2020

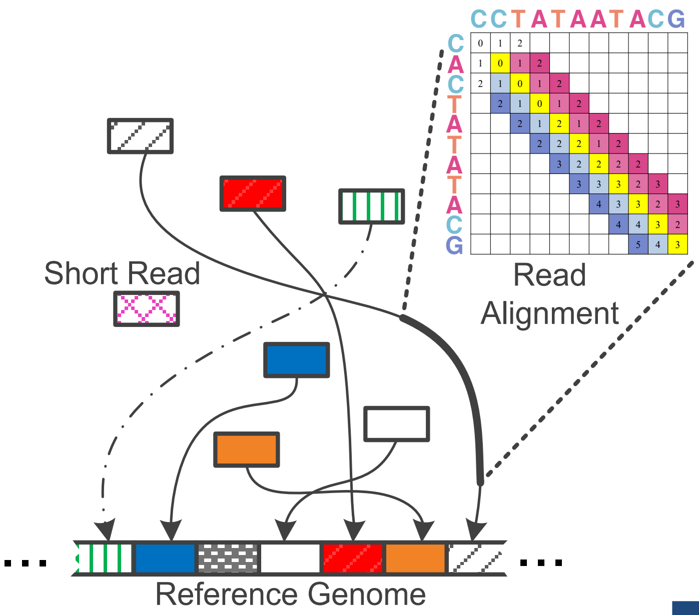
Example: Rapid Surveillance of Ebola Outbreak

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



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



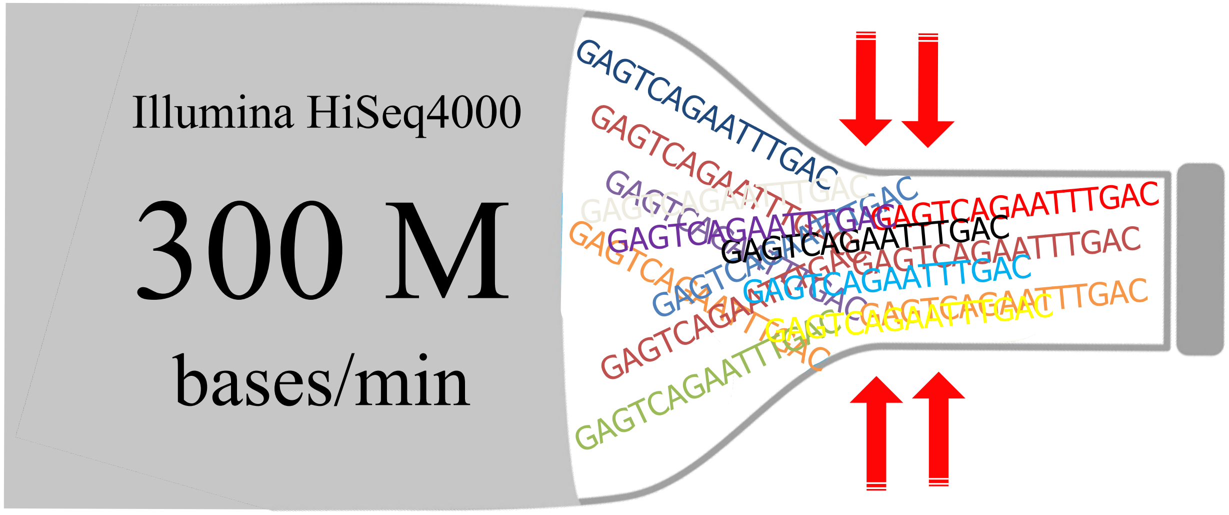


	C	C	T	A	T	A	A	T	A	C	G
C	0	1	2								
A	1	0	1	2							
C	2	1	0	1	2						
T		2	1	0	1	2					
A			2	1	2	1	2				
T				2	2	2	1	2			
A					3	2	2	2	2		
T						3	3	3	2	3	
A							4	3	3	2	3
C								4	4	3	2
G									5	4	3

1 **Sequencing**

Read Mapping 2

Bottlenecked in Mapping!!



Illumina HiSeq4000

300 M

bases/min

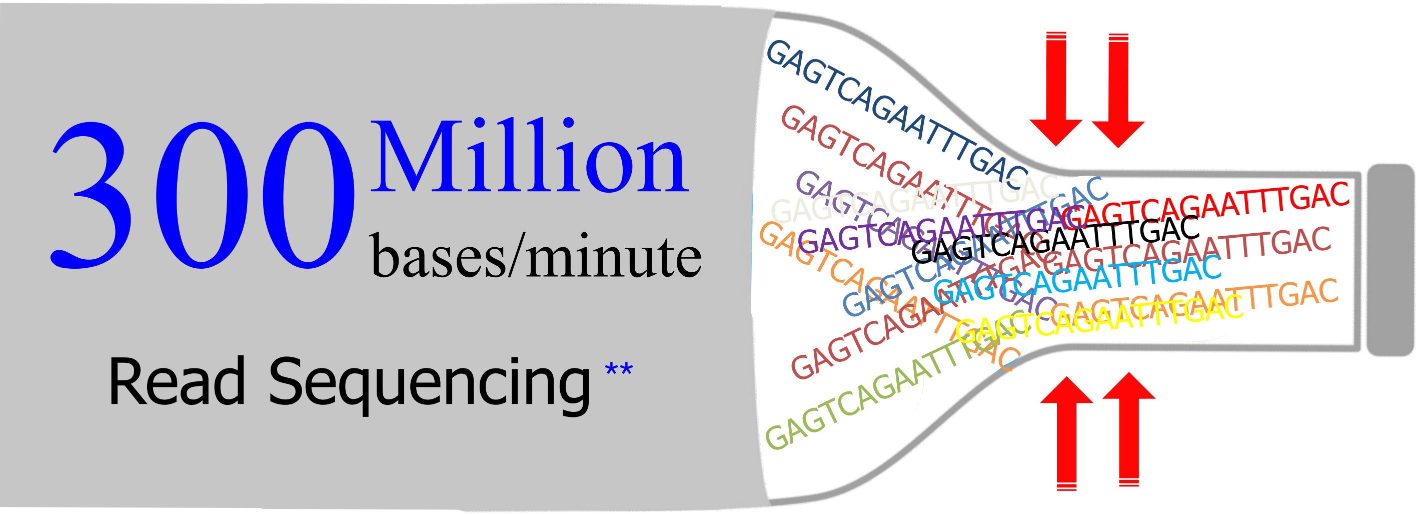
on average

2 M

bases/min

(0.6%)

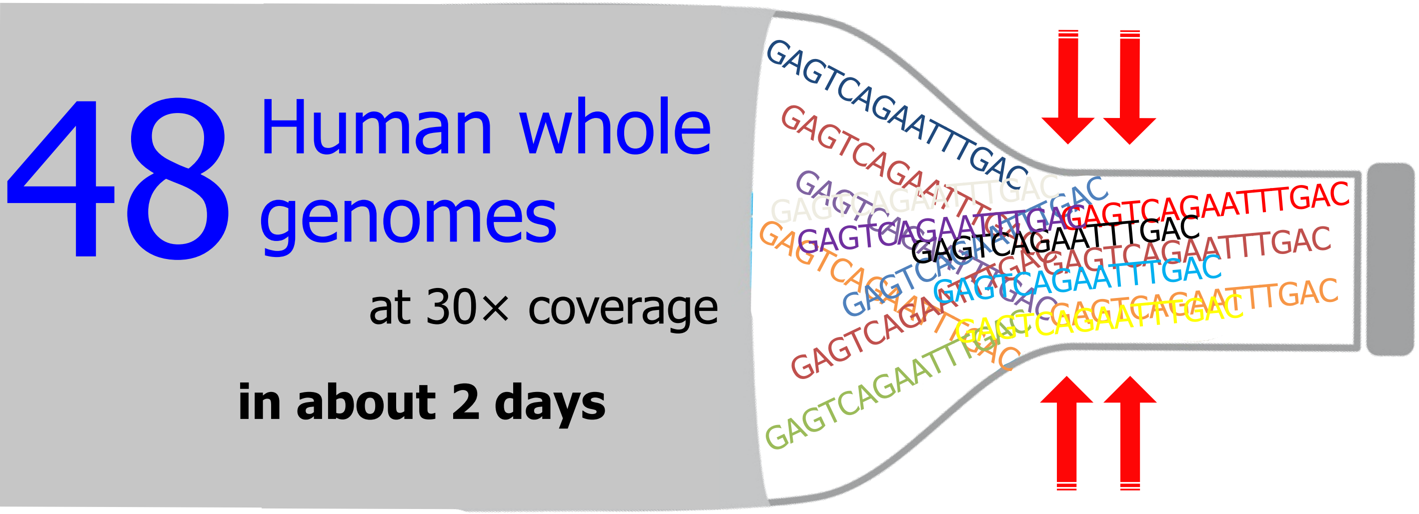
The Read Mapping Bottleneck



150x slower

* BWA-MEM
** HiSeqX10, MinION

The Read Mapping Bottleneck



48 Human whole genomes

at 30× coverage

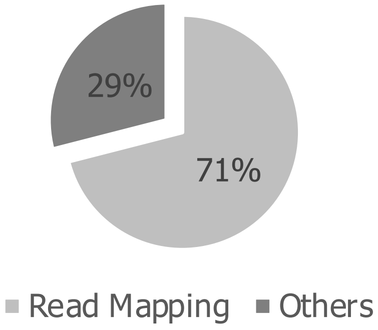
in about 2 days

Illumina NovaSeq 6000

1 Human genome

32 CPU hours

on a 48-core processor

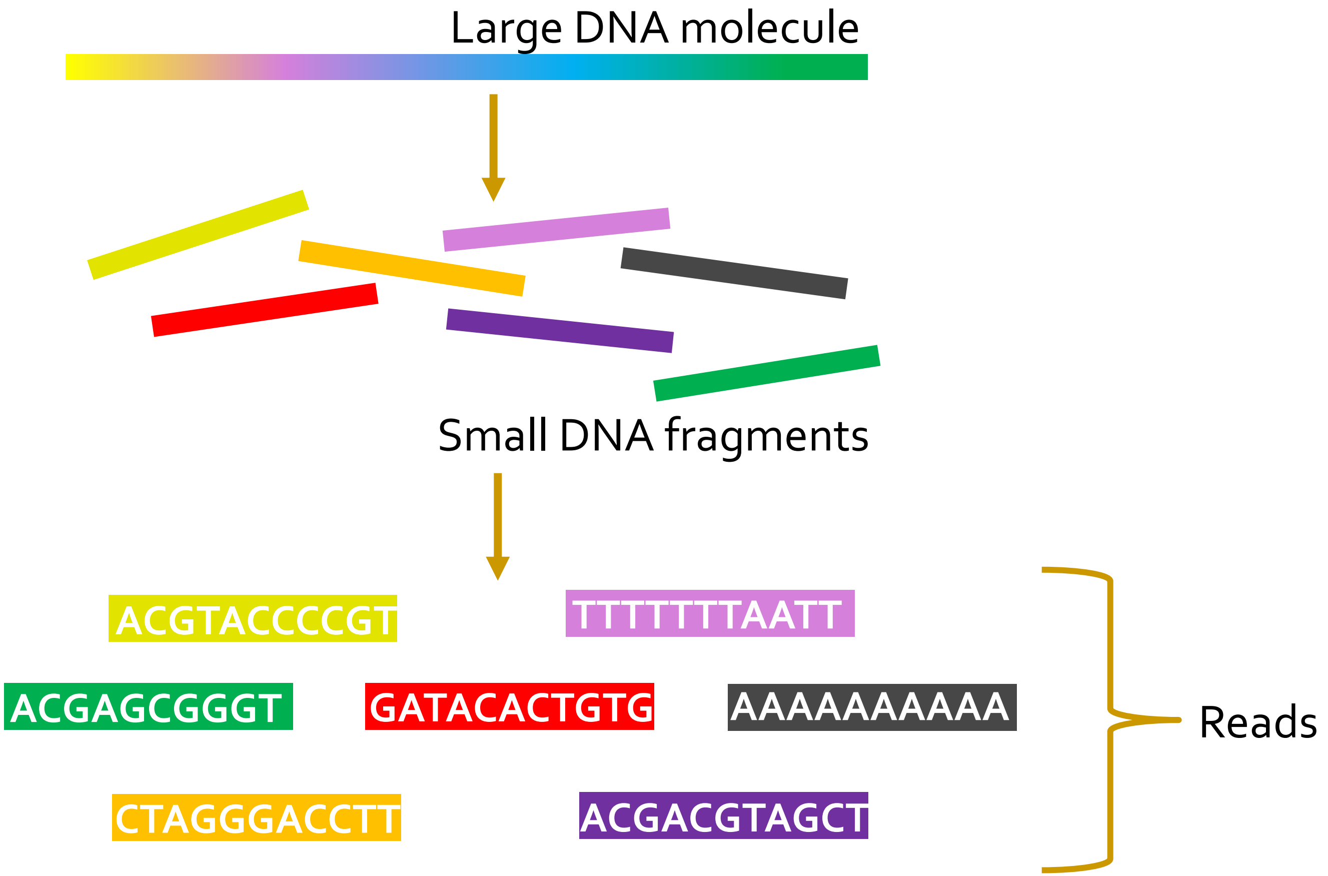


Goyal+, "Ultra-fast next generation human genome sequencing data processing using DRAGENTM bio-IT processor for precision medicine", *Open Journal of Genetics*, 2017.

Problem

Need to construct the entire genome from many reads

Genome Sequencing



Large DNA molecule

Small DNA fragments

ACGTACCCCGT

TTTTTTTAATT

ACGAGCGGGT

GATACACTGTG

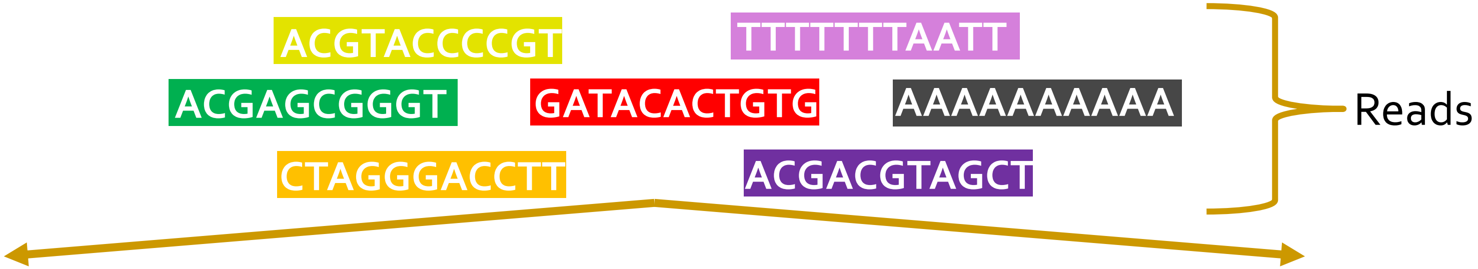
AAAAAAAAAA

CTAGGGACCTT

ACGACGTAGCT

Reads

Genome Sequence Analysis



ACGTACCCCGT

TTTTTTTAATT

ACGAGCGGGT

GATACACTGTG

AAAAAAAAAA

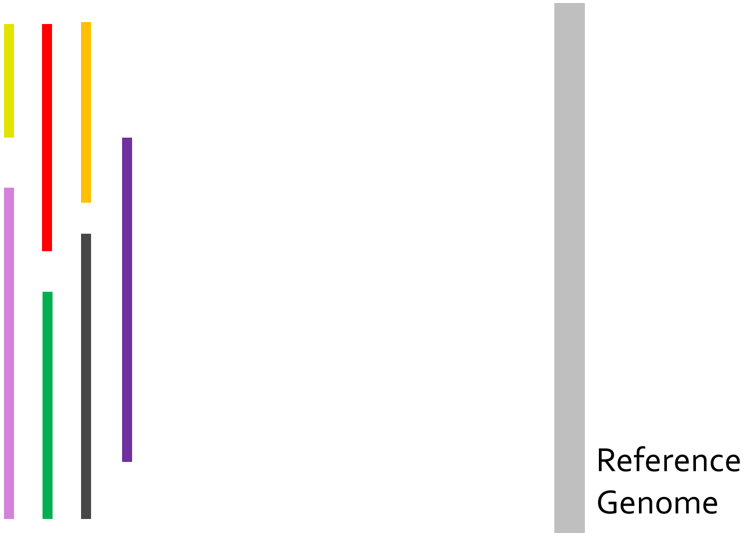
CTAGGGACCTT

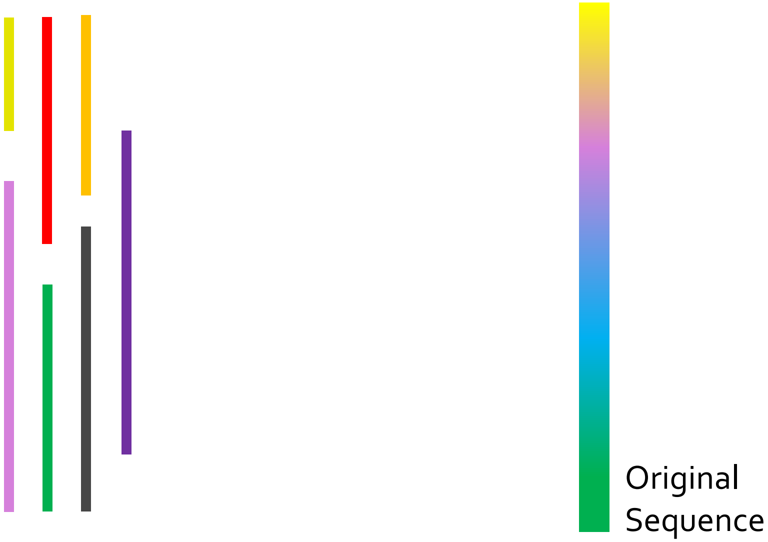
ACGACGTAGCT

Reads

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

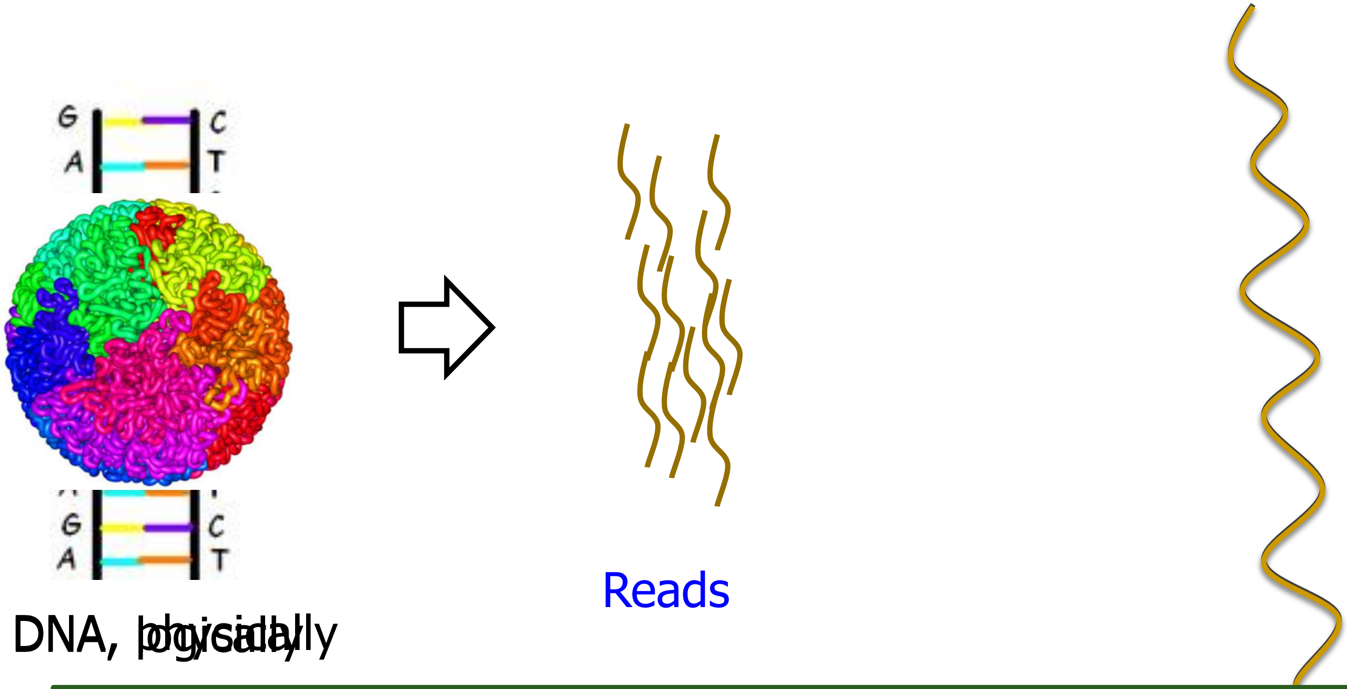
***De novo* Assembly,** method of merging the reads in order to **construct** the original sequence.





Read Mapping

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



Reference genome

DNA, logically

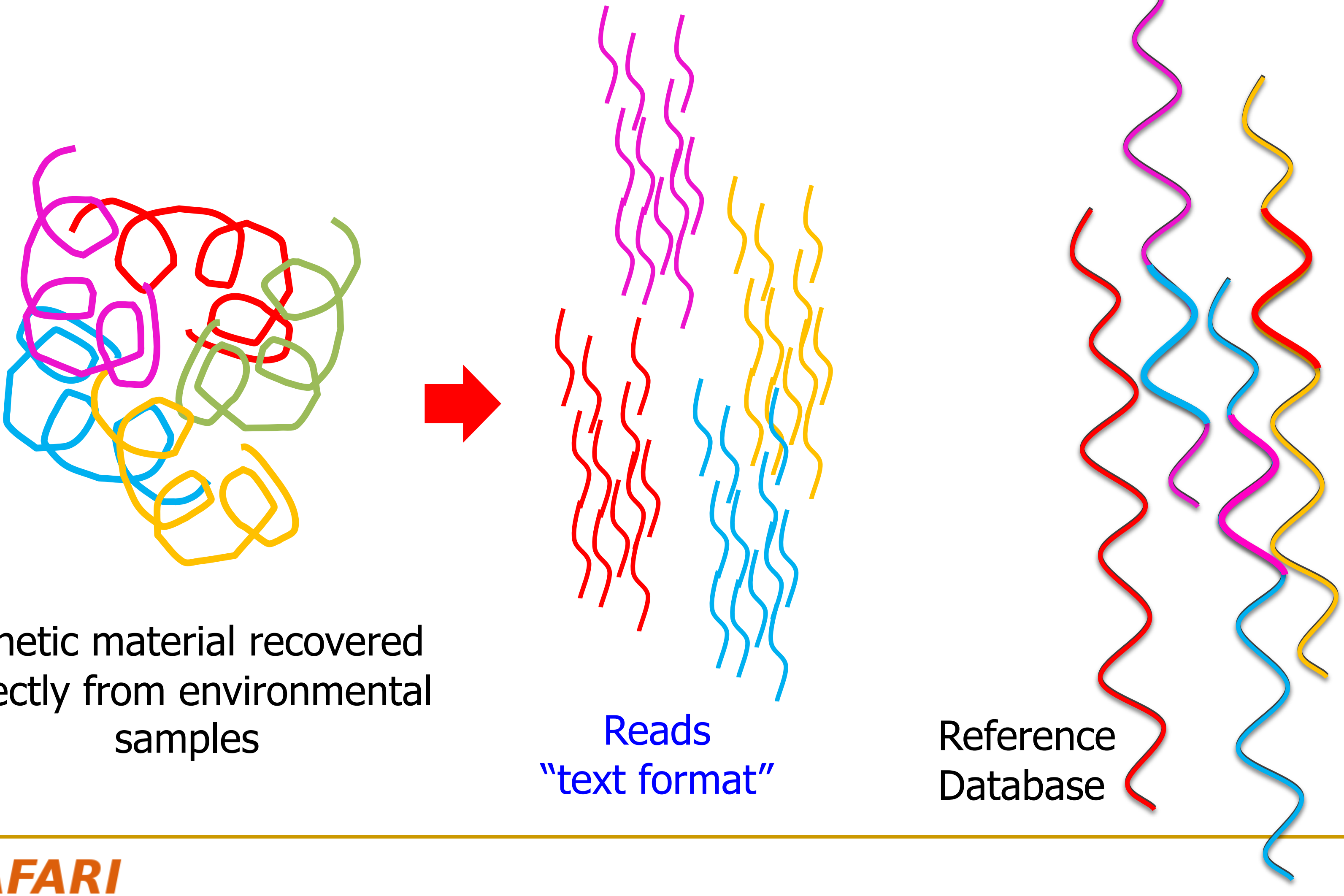
DNA, physically

Reads

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

Read Mapping for Metagenomic Analysis

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

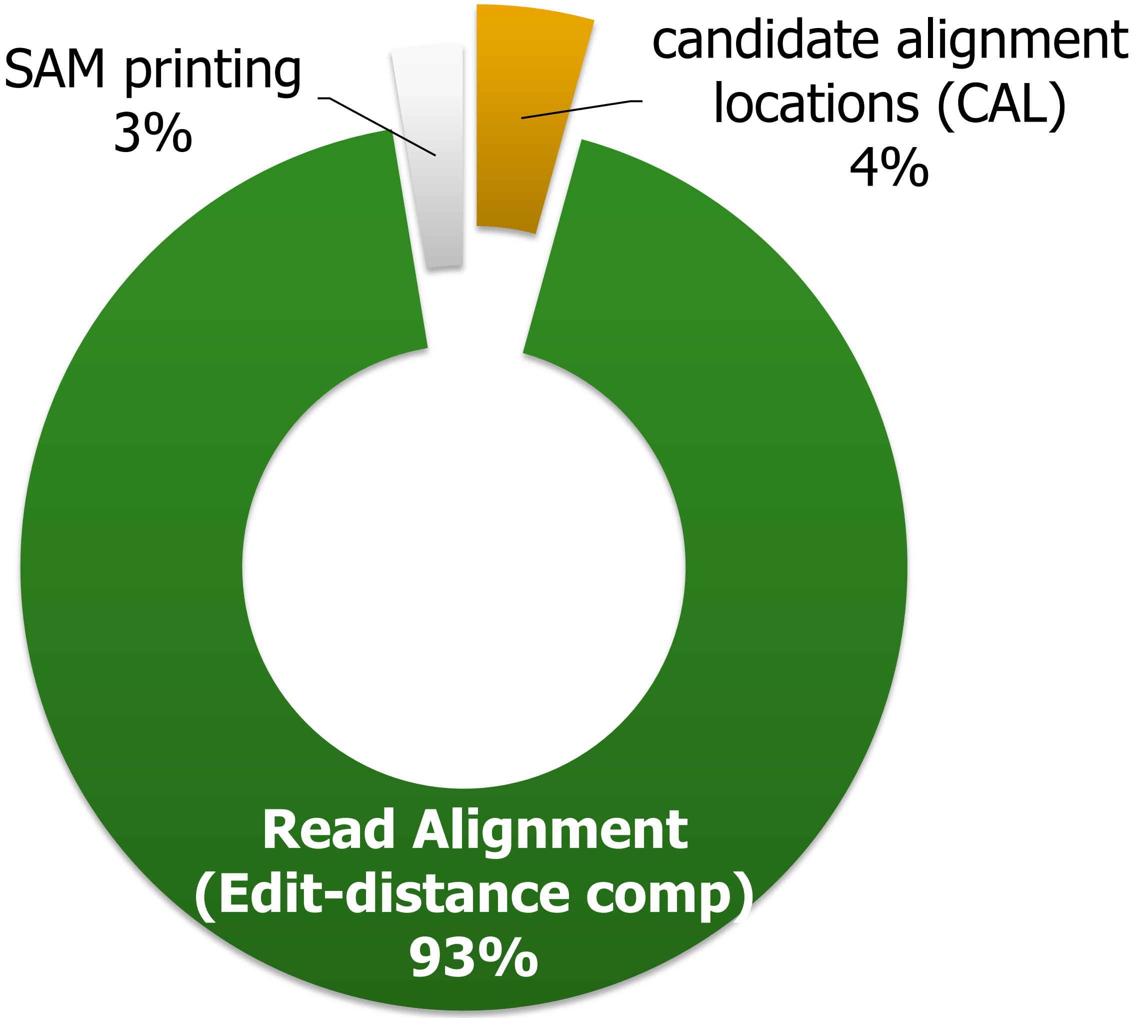


genetic material recovered directly from environmental samples

Reads
“text format”

Reference Database

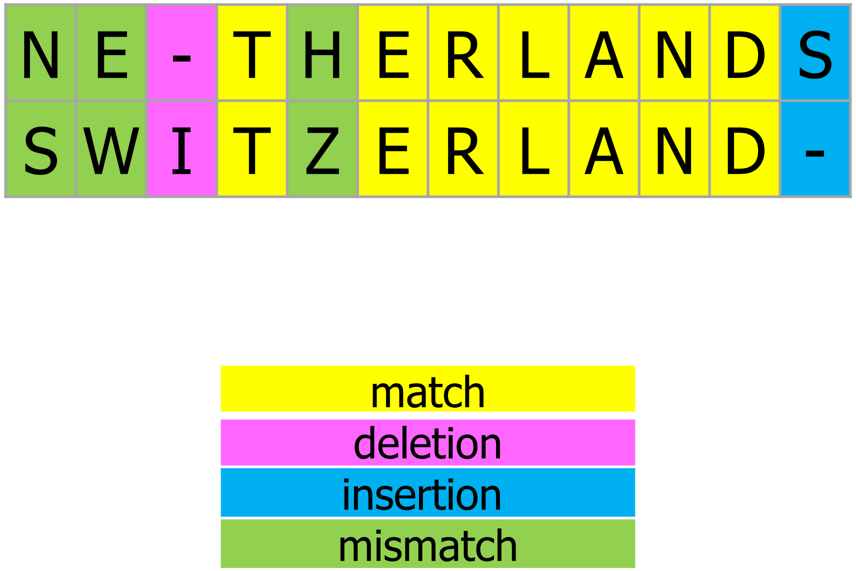
Read Mapping Execution Time Breakdown



Read Alignment/Verification

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

NETHERLANDS x SWITZERLAND



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

🟨 match
🟪 deletion
🟦 insertion
🟩 mismatch

Challenges in Read Mapping

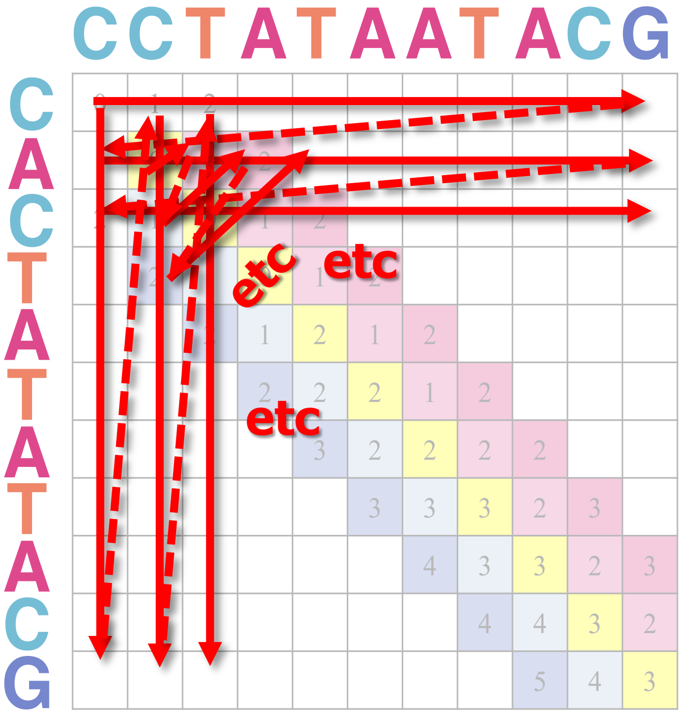
- Need to find many mappings of each read
  - A short read may map to many locations, especially with High-Throughput DNA Sequencing technologies
  - How can we find all mappings efficiently?

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

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

Why Is Read Alignment Slow?

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

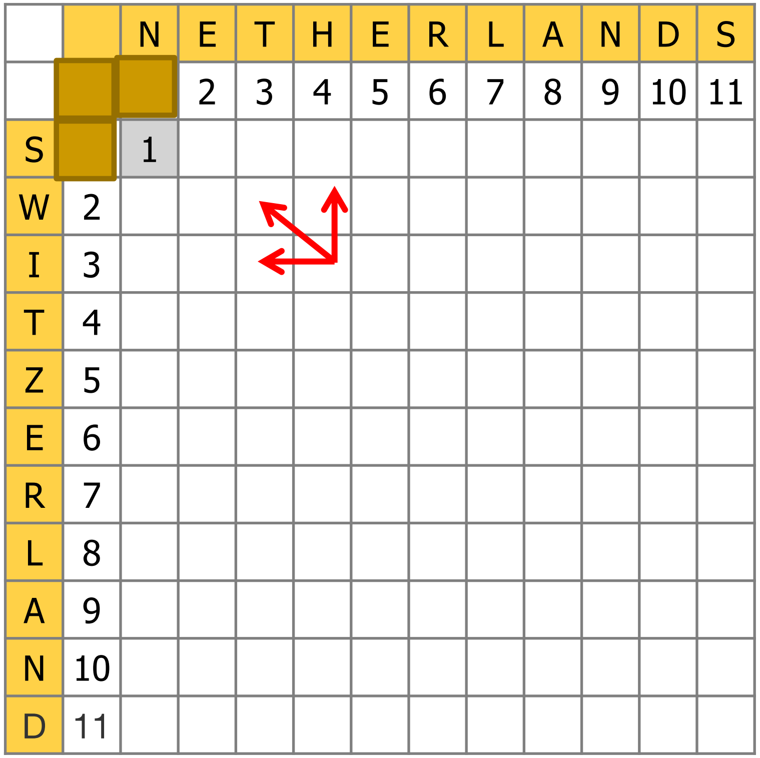


Read Alignment

Example: Dynamic Programming Table

immediate left,
upper left,
upper entries of its own

NETHERLANDS x SWITZERLAND



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

Example: Dynamic Programming Table

NETHERLANDS x SWITZERLAND

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

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

Example: Dynamic Programming

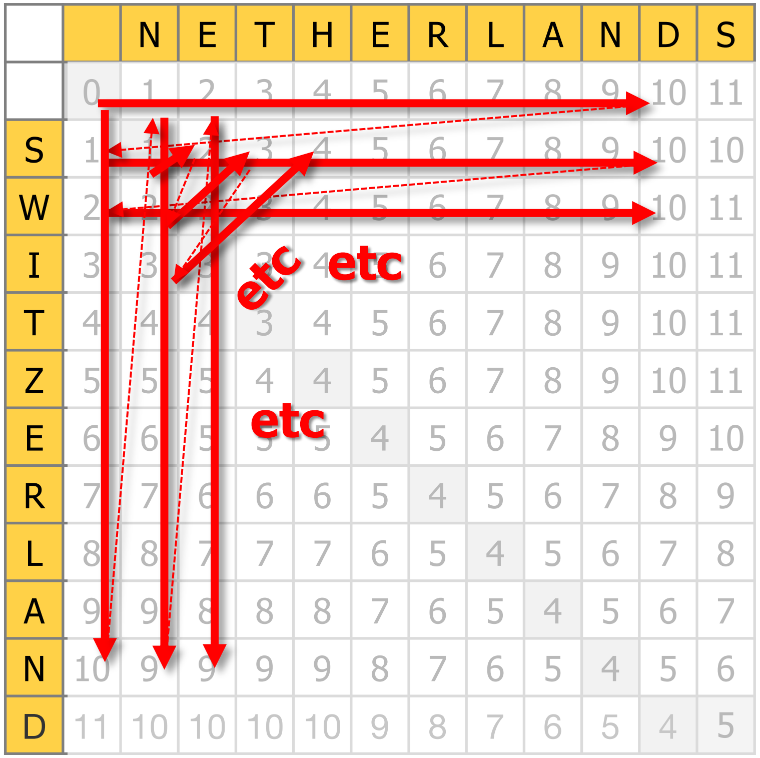
- Quadratic-time dynamic-programming algorithm

WHY?!

Enumerate all possible prefixes

NETHERLANDS x SWITZERLAND

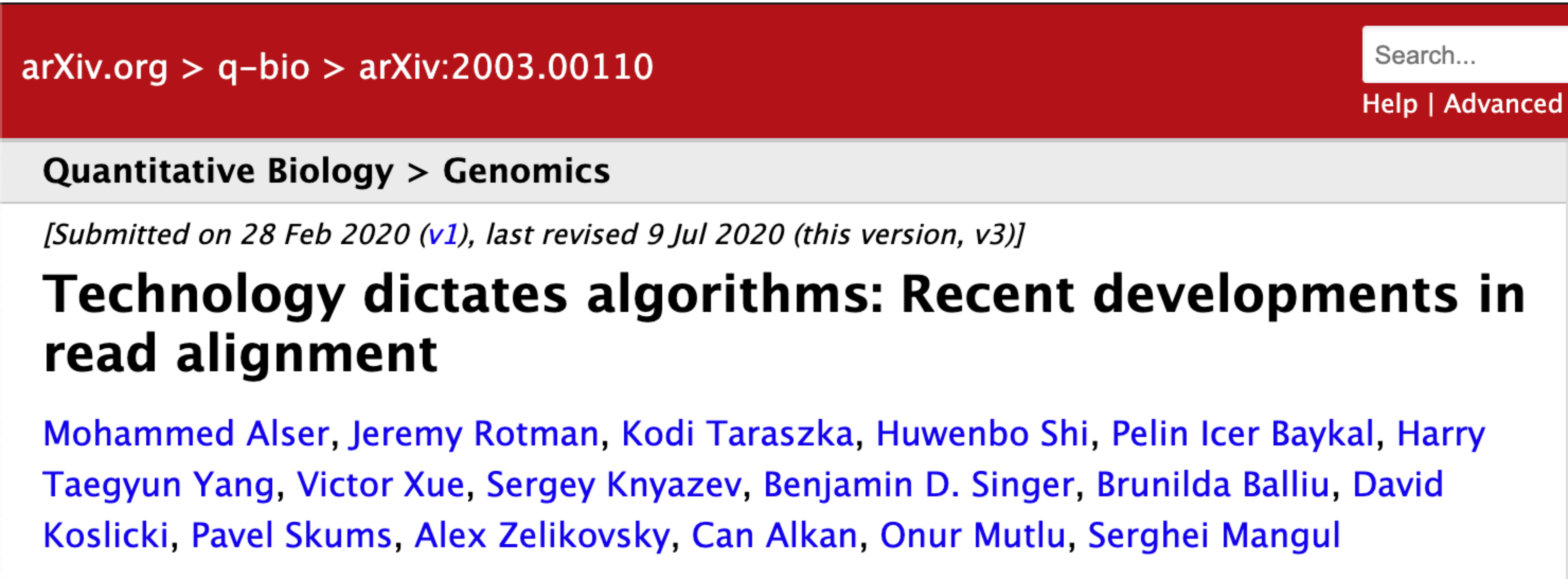
NETHERLANDS x S
NETHERLANDS x SW
NETHERLANDS x SWI
NETERLANDS x SWIT
NETHERLANDS x SWITZ
NETHERLANDS x SWITZE
NETHERLANDS x SWITZER
NETHERLANDS x SWITZERL
NETHERLANDS x SWITZERLA
NETHERLANDS x SWITZERLAN
NETHERLANDS x SWITZERLAND



		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

Read Mapping Survey in 111 Pages

In-depth analysis of 107 read mapping techniques (1988-2020)



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

SAFARI

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

- Future Opportunities: New Sequencing Technologies

Read Mapping Algorithms: Two Styles

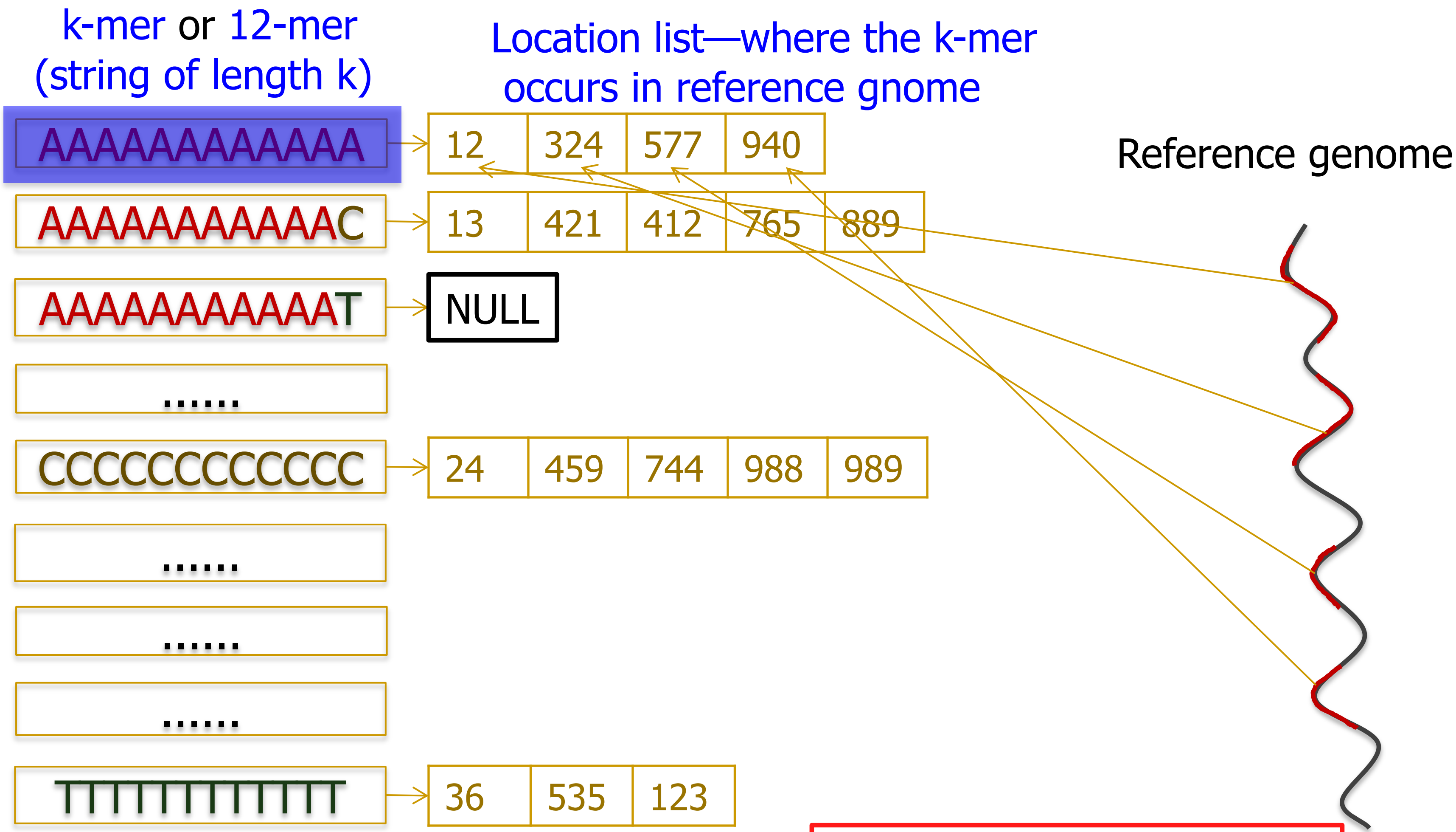
- Hash based seed-and-extend (hash table, suffix array, suffix tree)
  - Index the "k-mers" in the genome into a hash table (pre-processing)
  - When searching a read, find the location of a k-mer in the read; then extend through alignment
  - More sensitive (can find all mapping locations), but slow
  - 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]



k-mer or 12-mer (string of length k)

Location list—where the k-mer occurs in reference gnome

Reference genome

AAAAAAAAAAAA → 12 | 324 | 577 | 940

AAAAAAAAAAAC → 13 | 421 | 412 | 765 | 889

AAAAAAAAAAAT → NULL

......

CCCCCCCCCCCC → 24 | 459 | 744 | 988 | 989

......

......

......

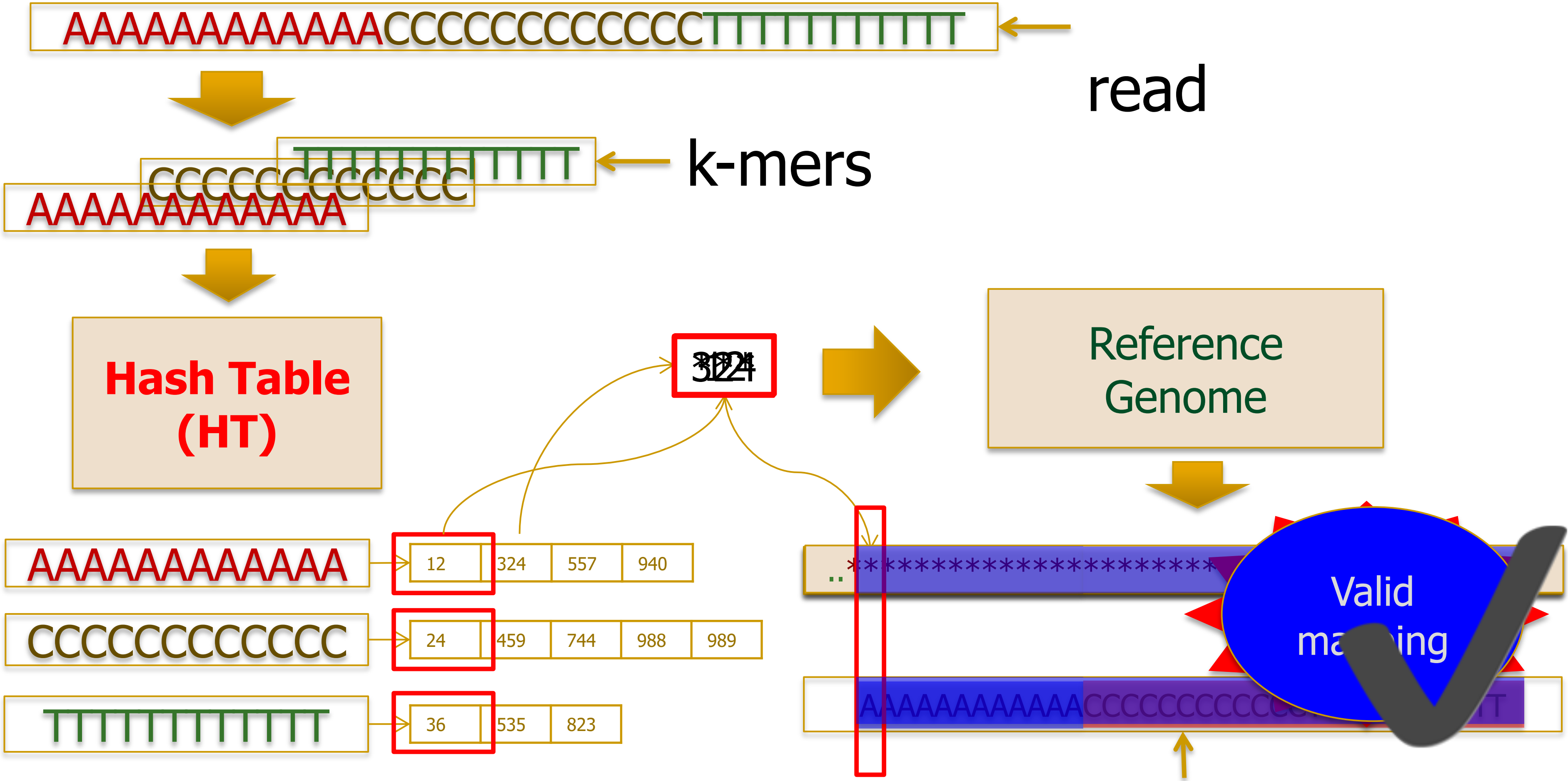
TTTTTTTTTTTT → 36 | 535 | 123

Once for a reference

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]



AAAAAAAAAAAACCCCCCCCCCCCTTTTTTTTTTTT

read

TTTTTTTTTTTT

CCCCCCCCCCCC

AAAAAAAAAAAA

k-mers

Hash Table (HT)

324

Reference Genome

AAAAAAAAAAAA → 12 | 324 | 557 | 940

CCCCCCCCCCCC → 24 | 459 | 744 | 988 | 989

TTTTTTTTTTTT → 36 | 535 | 823

..**********************

AAAAAAAAAAAACCCCCCCCCCCC...TT

Valid mapping

read

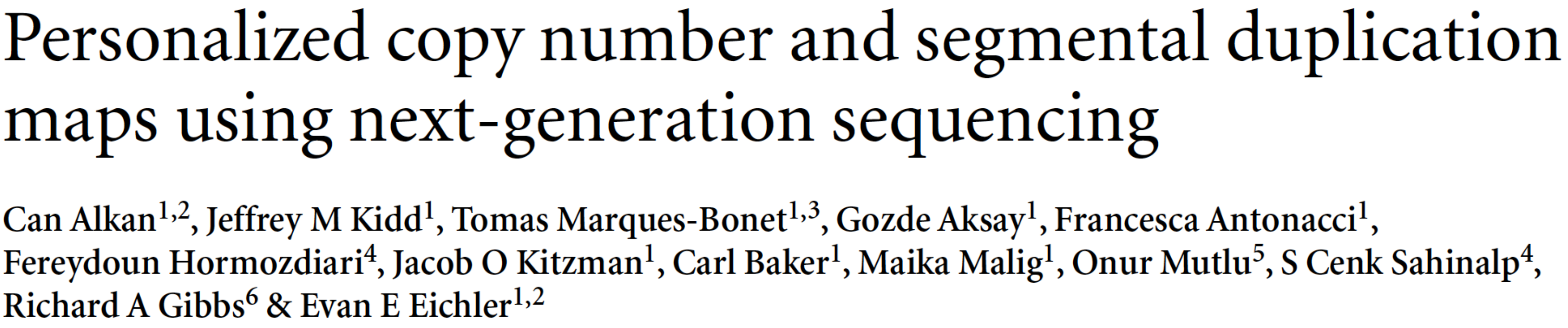
Verification/Local Alignment

Our First Step: Comprehensive Mapping

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

nature genetics

http://mrfast.sourceforge.net/



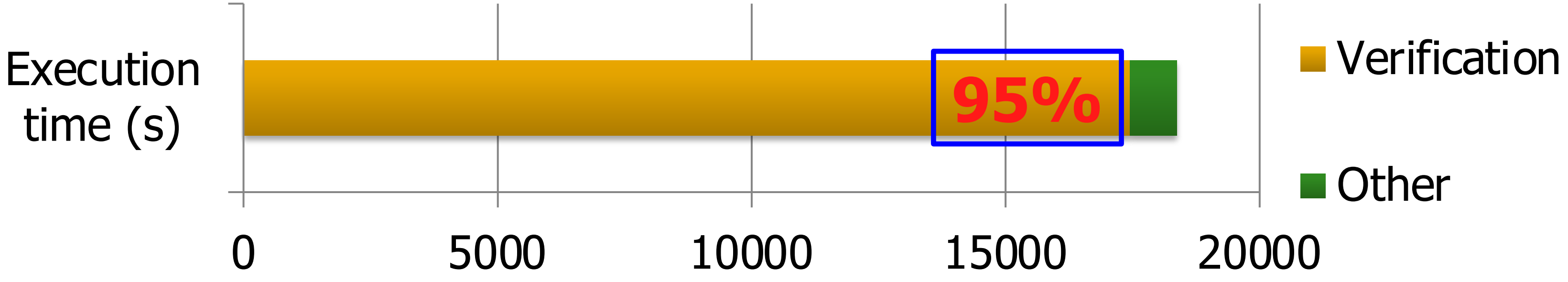
Personalized copy number and segmental duplication maps using next-generation sequencing

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

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

Problem and Goal

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



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

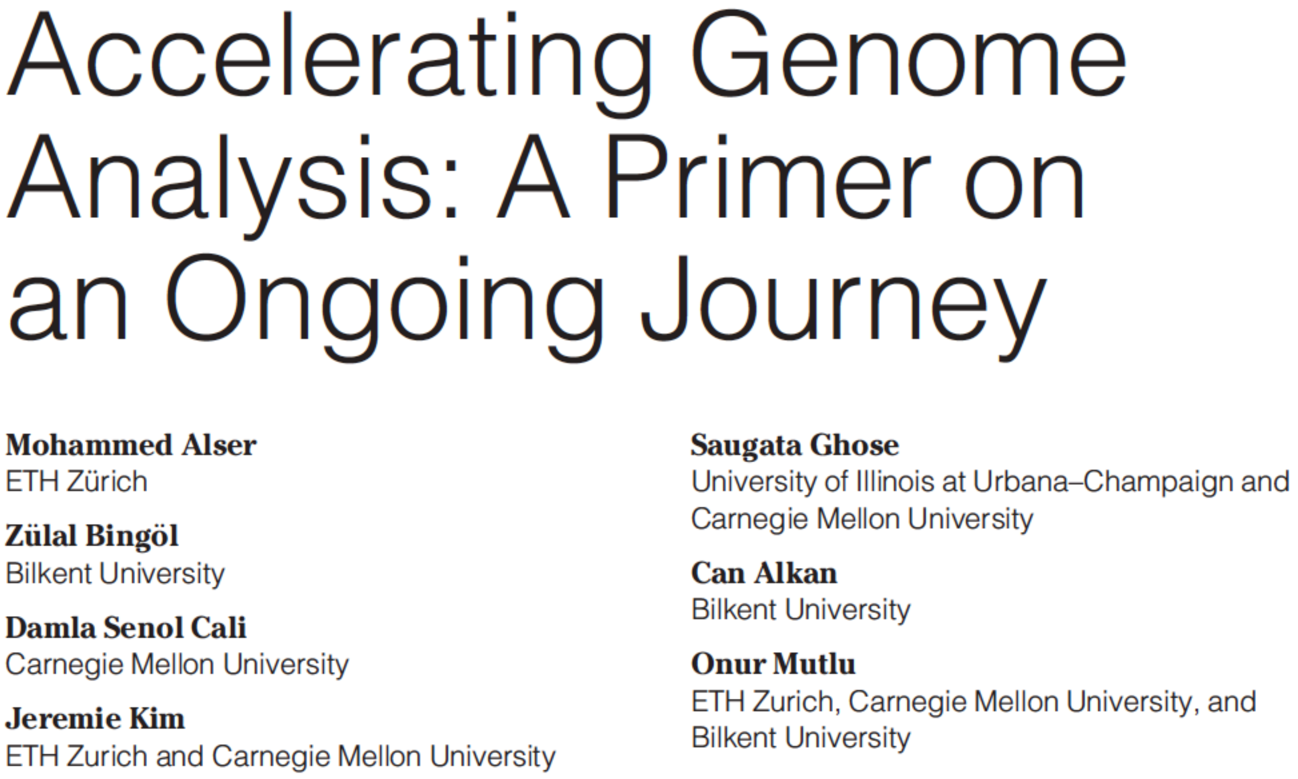
Overarching Key Idea

Filter fast before you align

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

Accelerating Genome Analysis: Overview

- Mohammed Alser, Zulal Bingol, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
  "Accelerating Genome Analysis: A Primer on an Ongoing Journey"
  IEEE Micro (***IEEE MICRO***), Vol. 40, No. 5, pages 65-75, September/October 2020.
  [Slides (pptx)(pdf)]
  [Talk Video (1 hour 2 minutes)]



Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser
ETH Zürich

Zülal Bingöl
Bilkent University

Damla Senol Cali
Carnegie Mellon University

Jeremie Kim
ETH Zurich and Carnegie Mellon University

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

Can Alkan
Bilkent University

Onur Mutlu
ETH Zurich, Carnegie Mellon University, and Bilkent University

Agenda

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

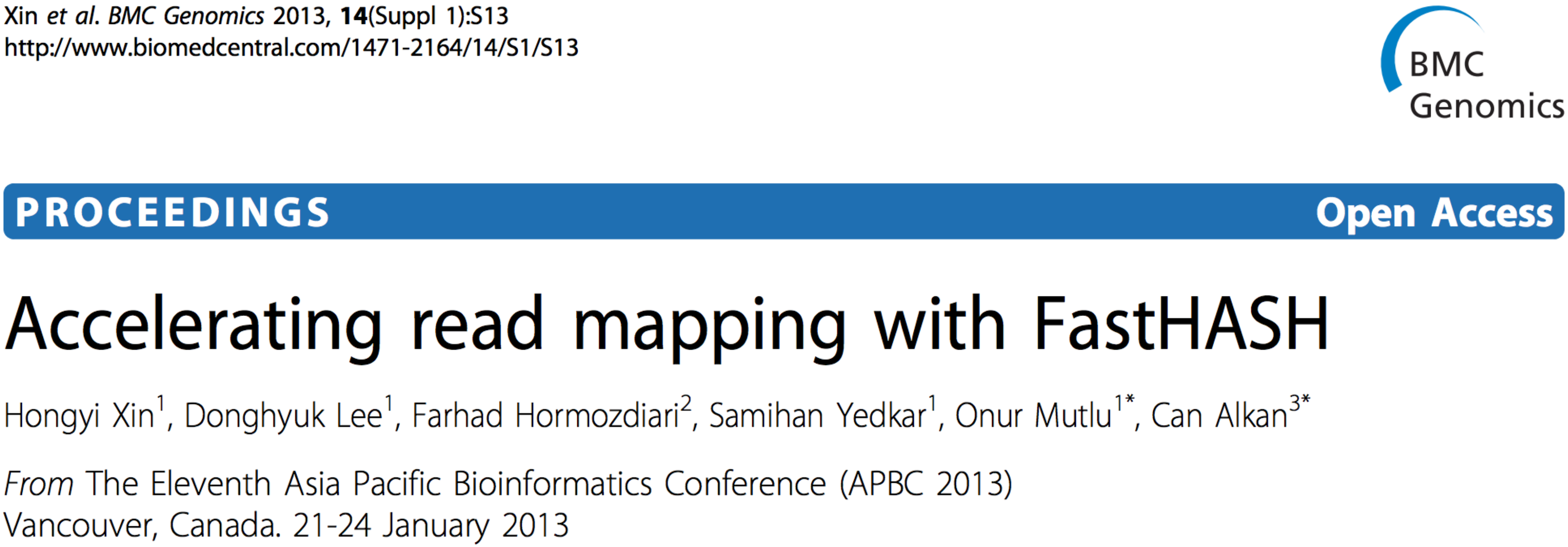
- Algorithmic Acceleration
  - Exploiting Structure of the Genome
  - Exploiting SIMD Instructions

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

Our First Filter: Pure Software Approach

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



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

BMC Genomics

PROCEEDINGS Open Access

Accelerating read mapping with FastHASH

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

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

Reducing the Cost of Verification

- We observe that most verification (edit distance computation) calculations are unnecessary
    - 1 out of 1000 potential locations passes the verification process

- We observe that 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

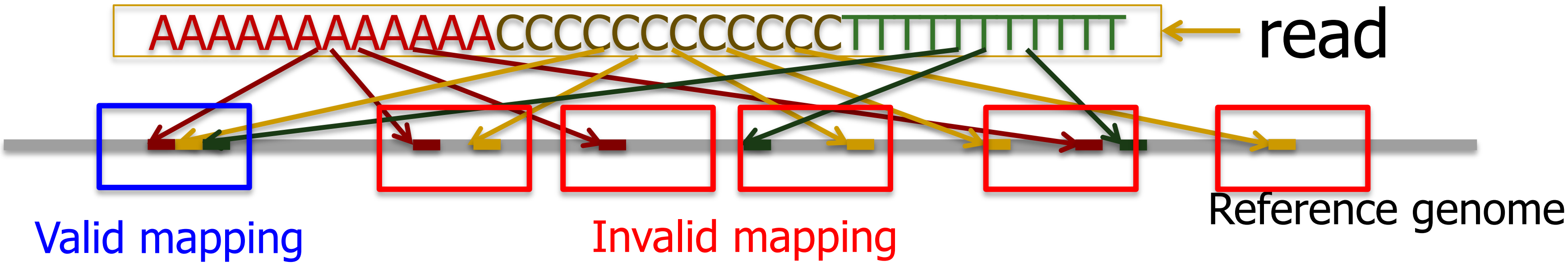
FastHASH Mechanisms [Xin+, BMC Genomics 2013]

- **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

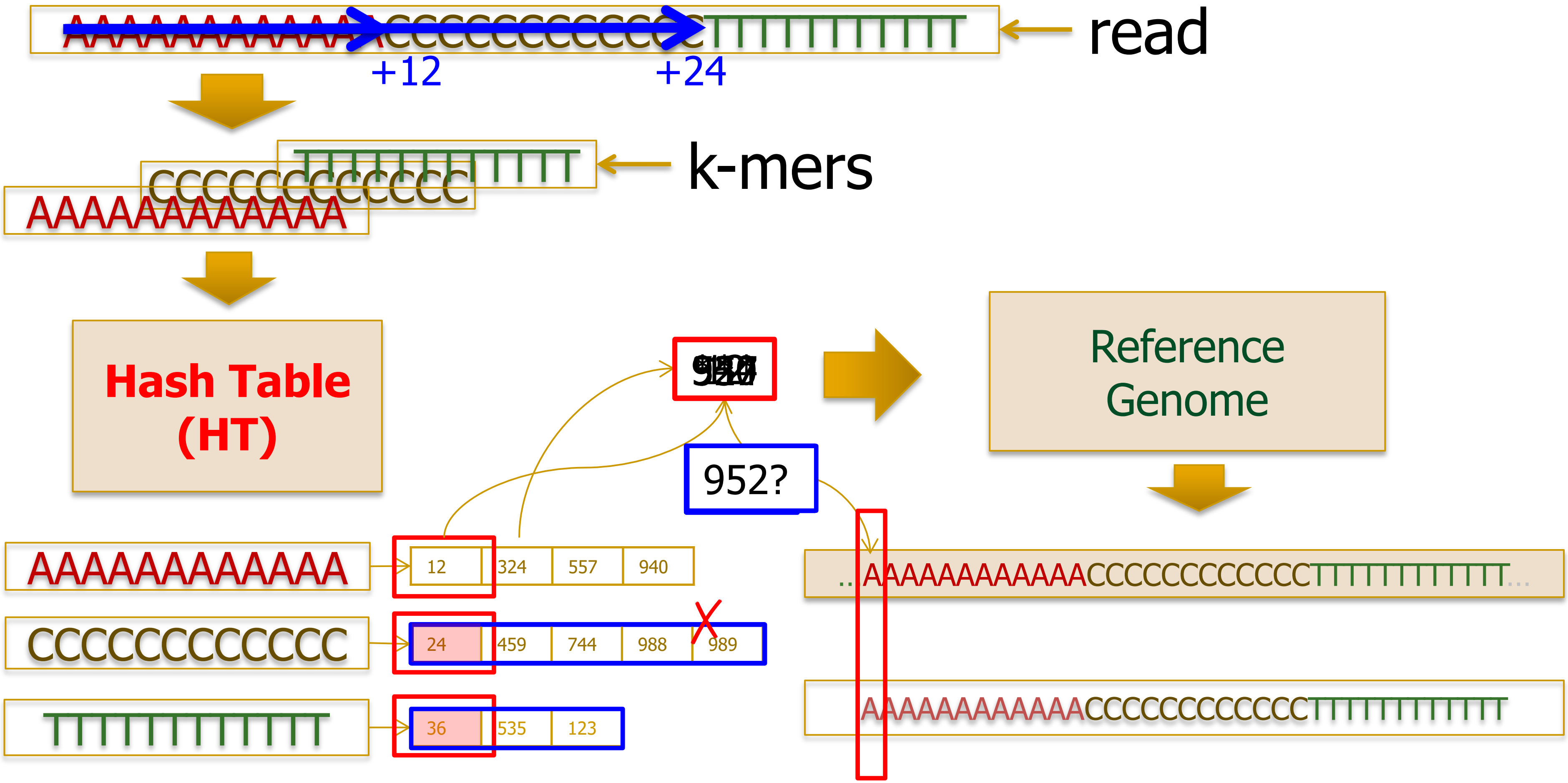
Adjacency Filtering (AF)

- **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 → there must be more than e errors → invalid mapping

Adjacency Filtering (AF)



FastHASH Mechanisms [Xin+, BMC Genomics 2013]

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

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

Cheap K-mer Selection (CKS)

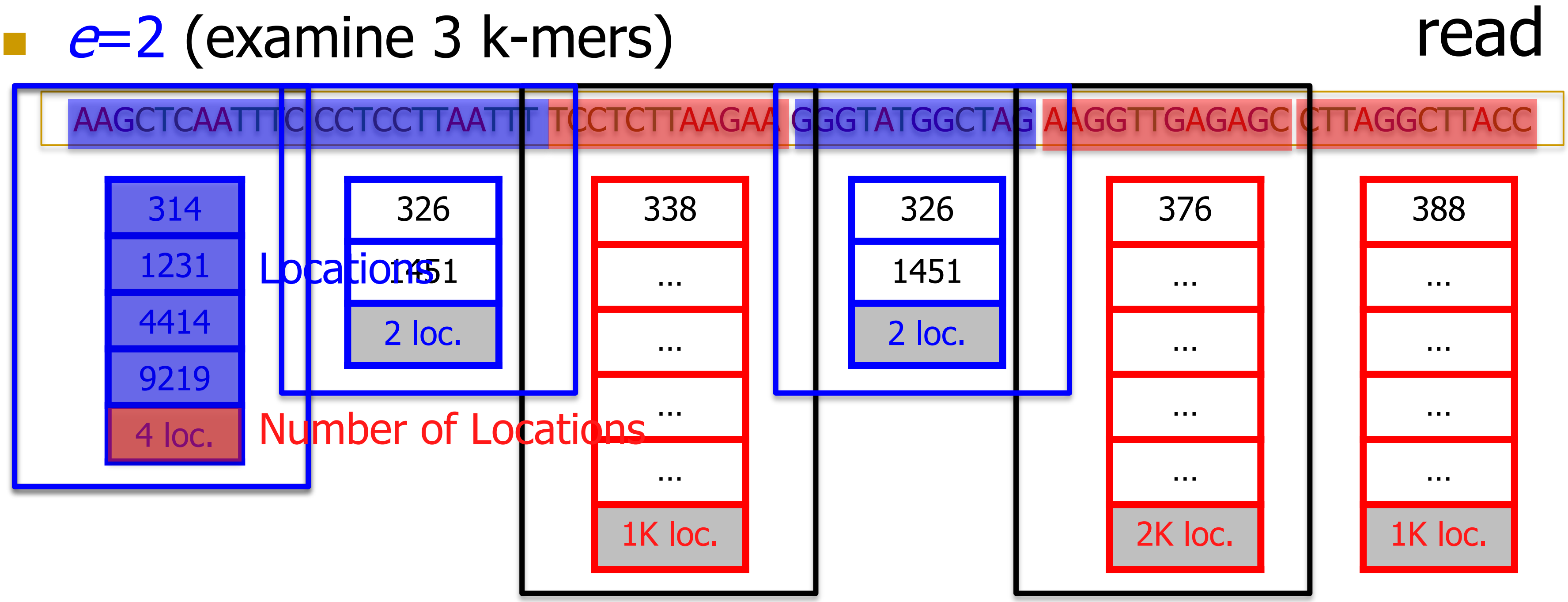
- **Goal:** Reduce the number of potential mappings to examine

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

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

Cheap K-mer Selection

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



Expensive 3 k-mers

Cheapest 3 k-mers

Previous work needs to verify:

3004 locations

FastHASH verifies only:

8 locations

Methodology

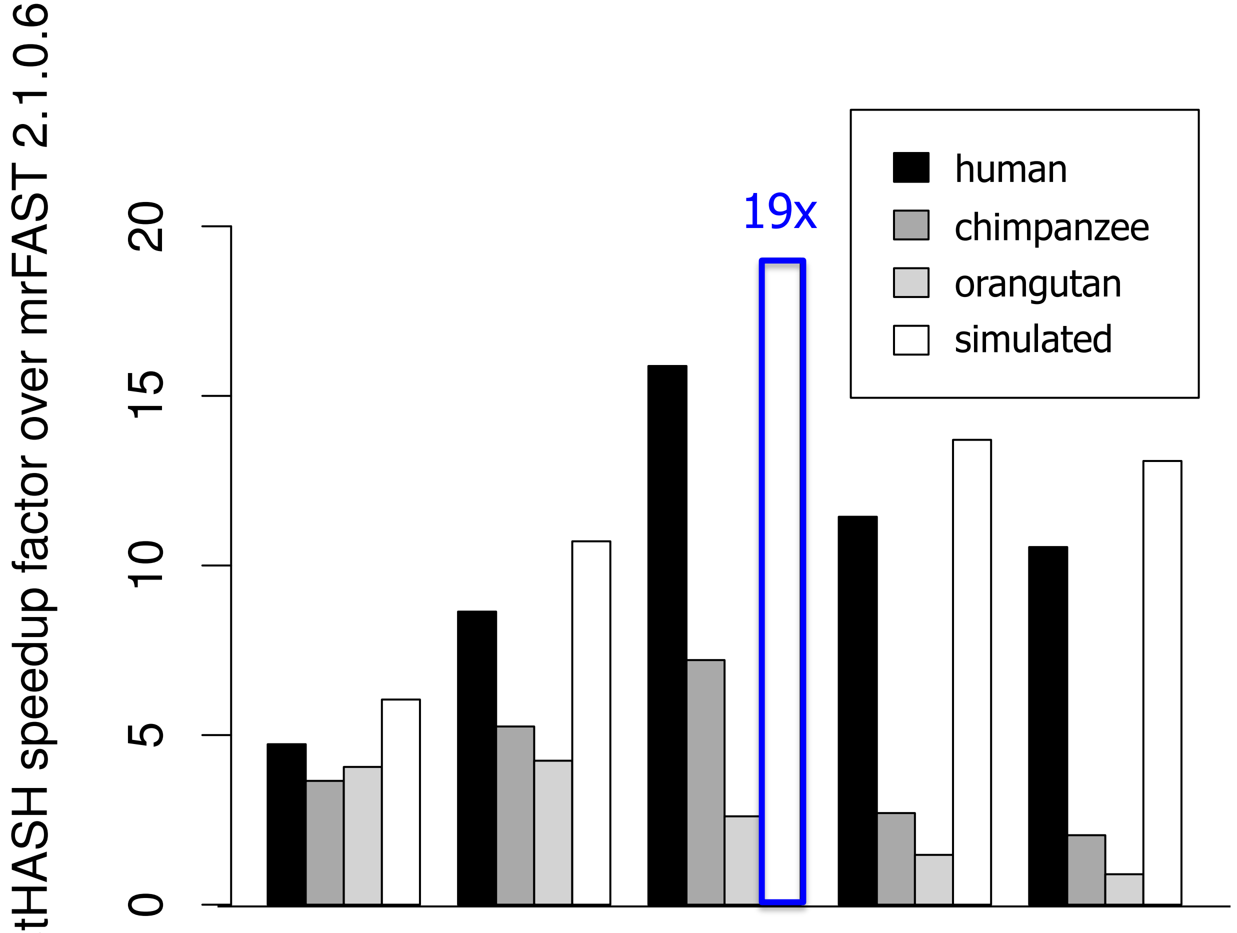
- Implemented FastHASH on top of state-of-the-art mapper: mrFAST
  - New version mrFAST-2.5.0.0 over mrFAST-2.1.0.6

- 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

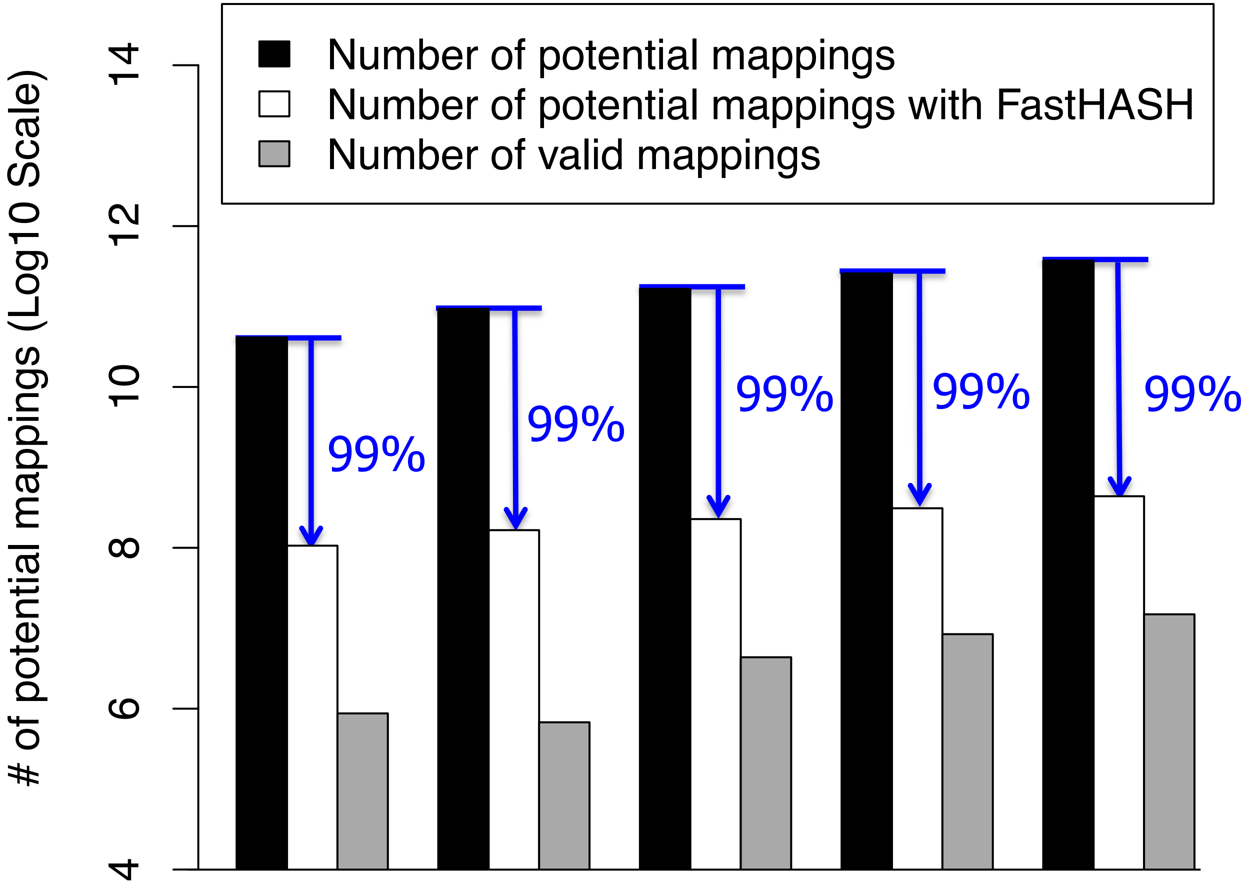
FastHASH Speedup: Entire Read Mapper



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 Conclusion

- Problem: Existing read mappers perform poorly in mapping millions of short reads to the reference genome, in the presence of errors

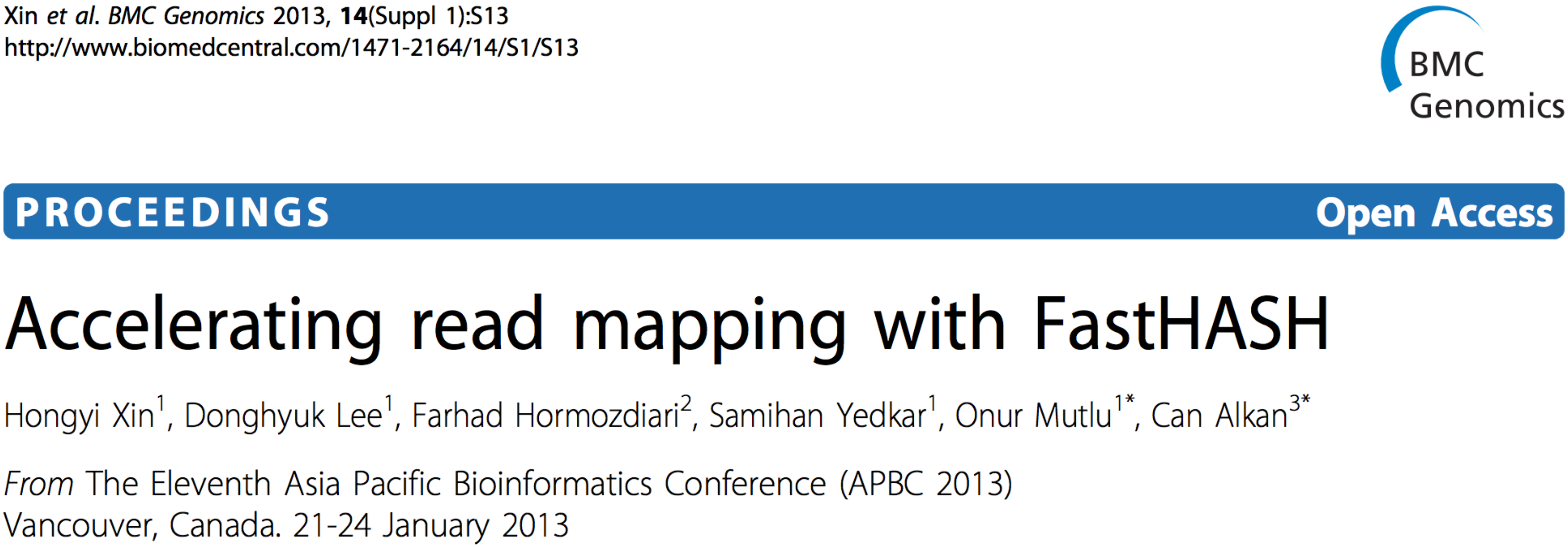
- Observation: Most of the verification calculations are unnecessary → filter them out

- Key Idea: Exploit the structure of the genome to
  - Reject invalid mappings early (Adjacency Filtering)
  - Reduce the number of possible mappings to examine (Cheap K-mer Selection)

- Key Result: FastHASH obtains up to 19x speedup over the state-of-the-art mapper without losing valid mappings

More on FastHASH

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



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

BMC
Genomics

PROCEEDINGS **Open Access**

Accelerating read mapping with FastHASH

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

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

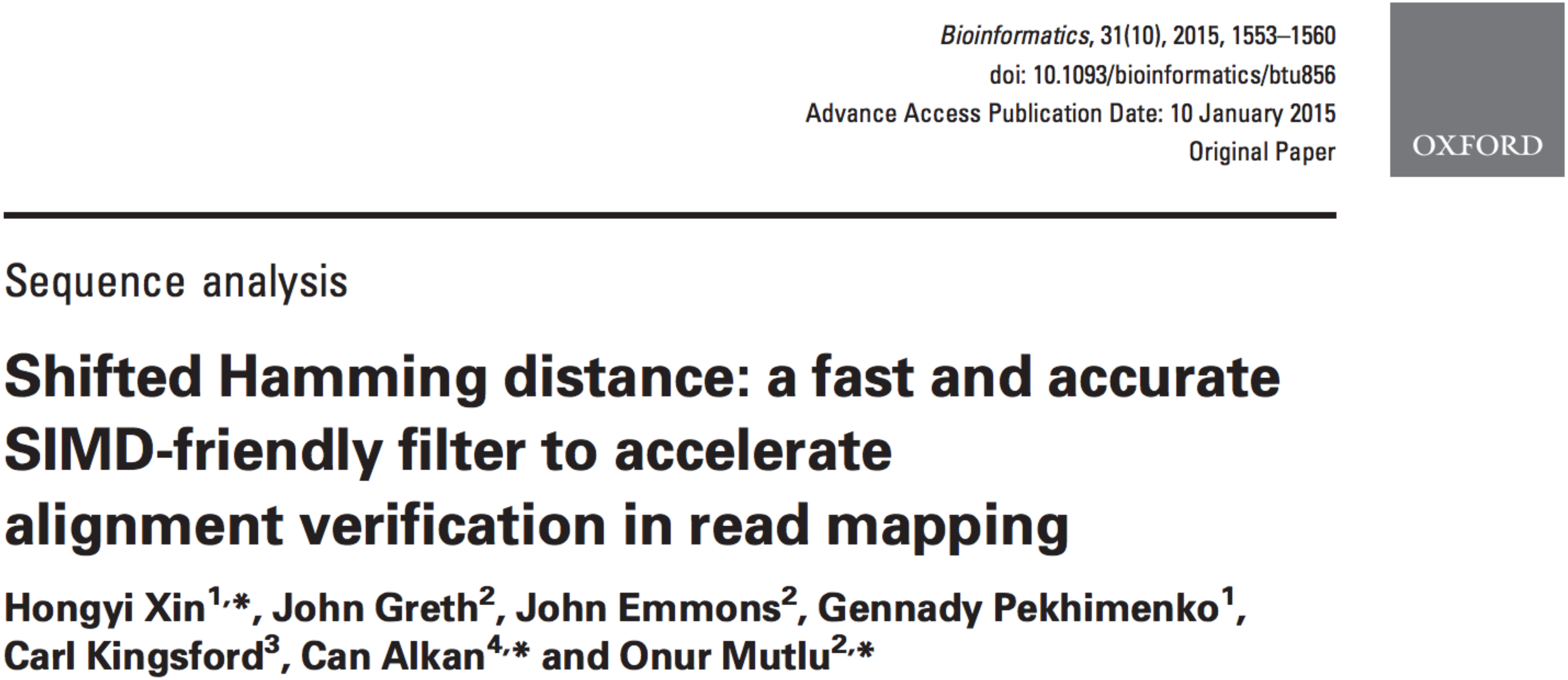
Xin+, **"Accelerating Read Mapping with FastHASH"**, BMC Genomics 2013.

Agenda

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

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

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

Shifted Hamming Distance

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

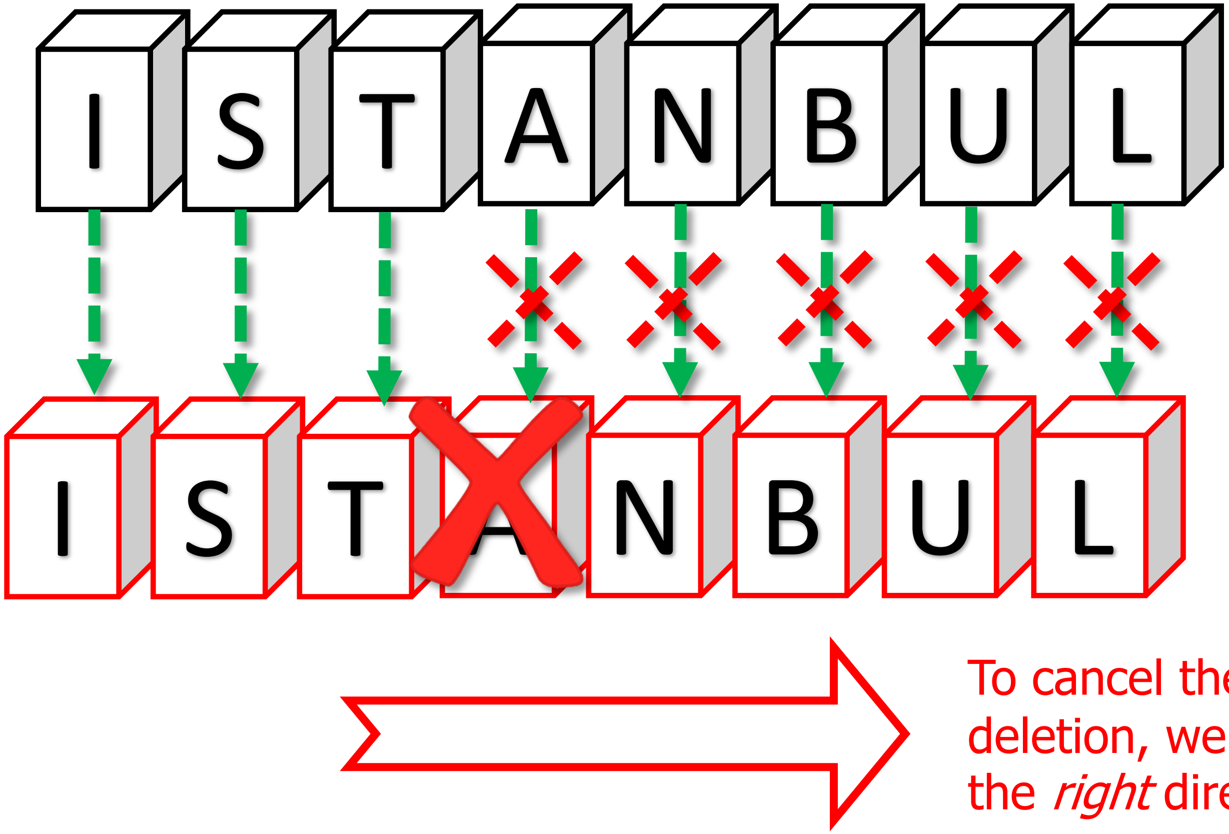
- **Key idea:**
  - Compute "Shifted Hamming Distance": AND of 2E Hamming Distances of two strings, to filter out invalid mappings
    - Uses bit-parallel operations that nicely map to SIMD instructions

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

Hamming Distance (∑⊕)

3 matches 5 mismatches

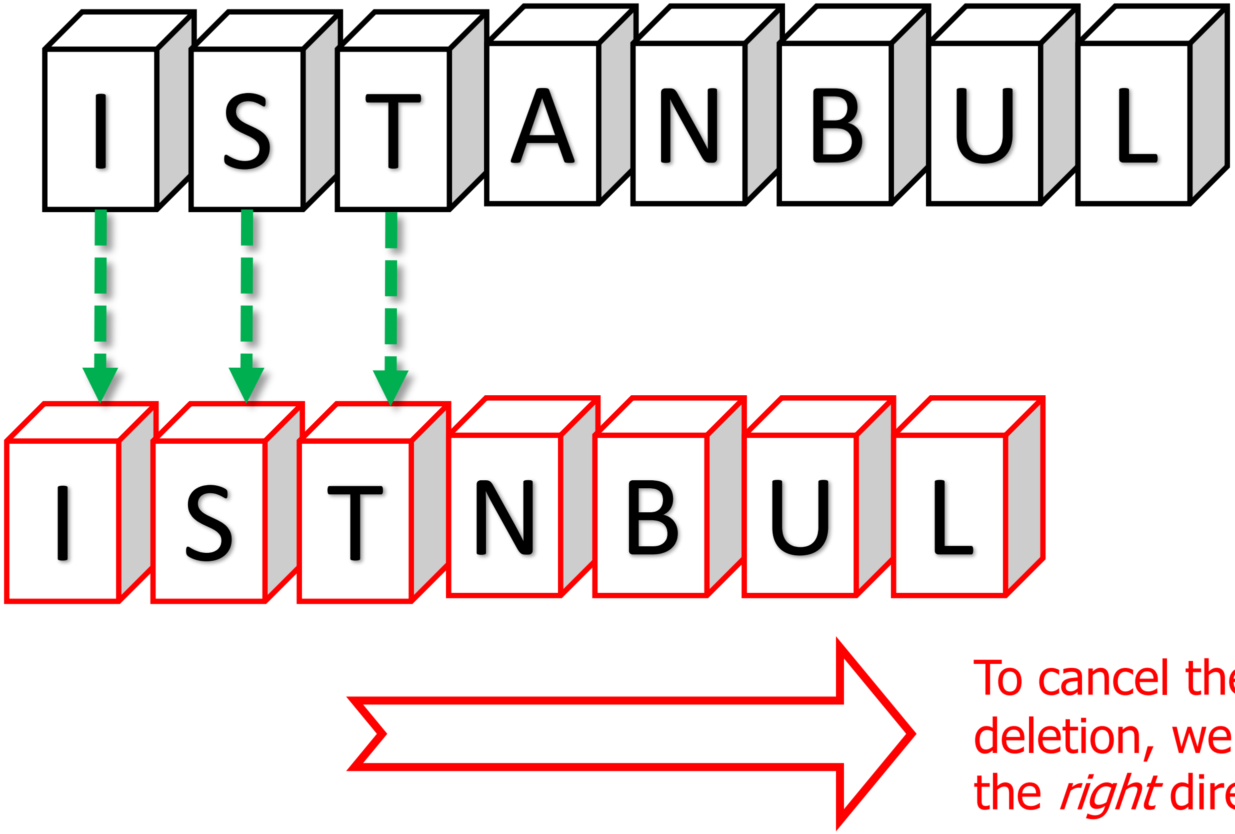
Edit = 1 Deletion



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

Insight: Shifting a String Helps Similarity Search

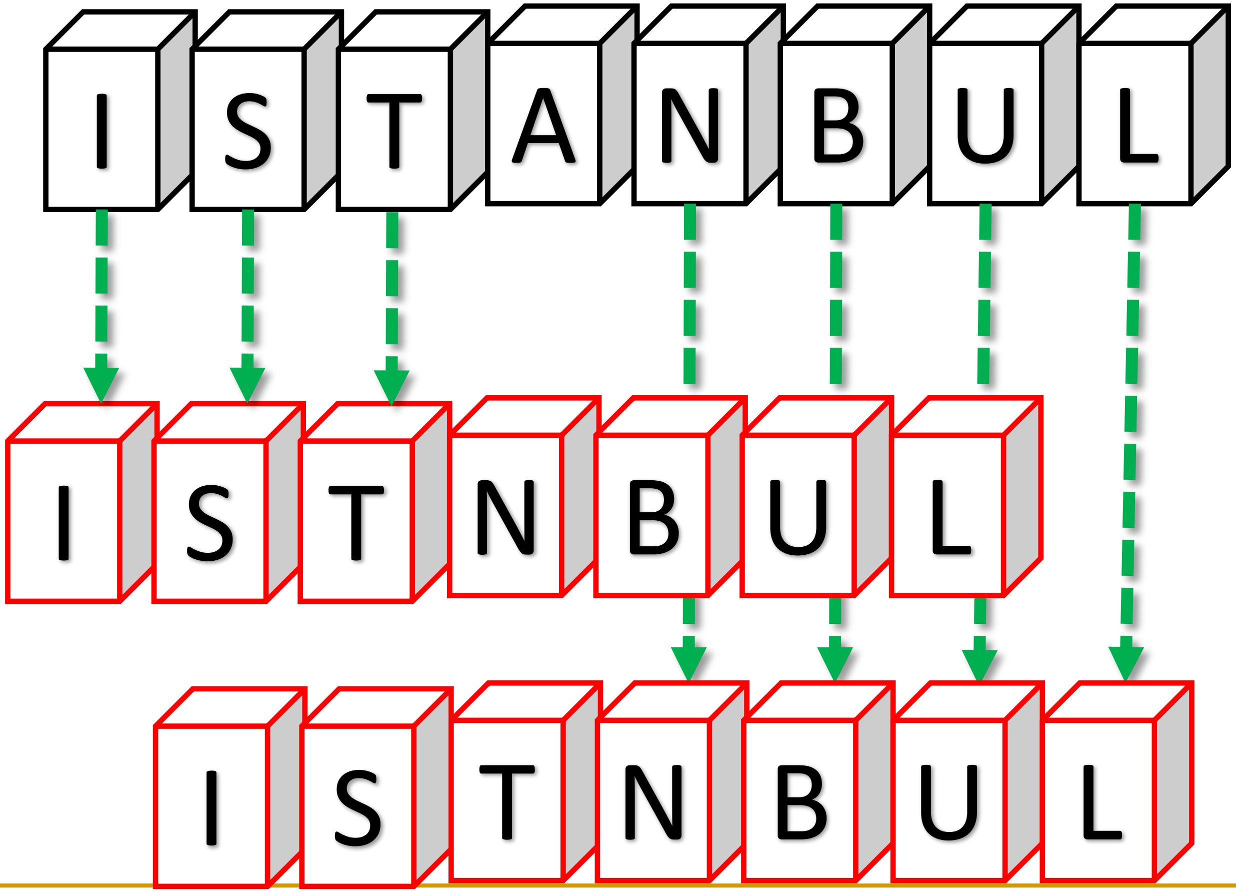
3 matches 5 mismatches



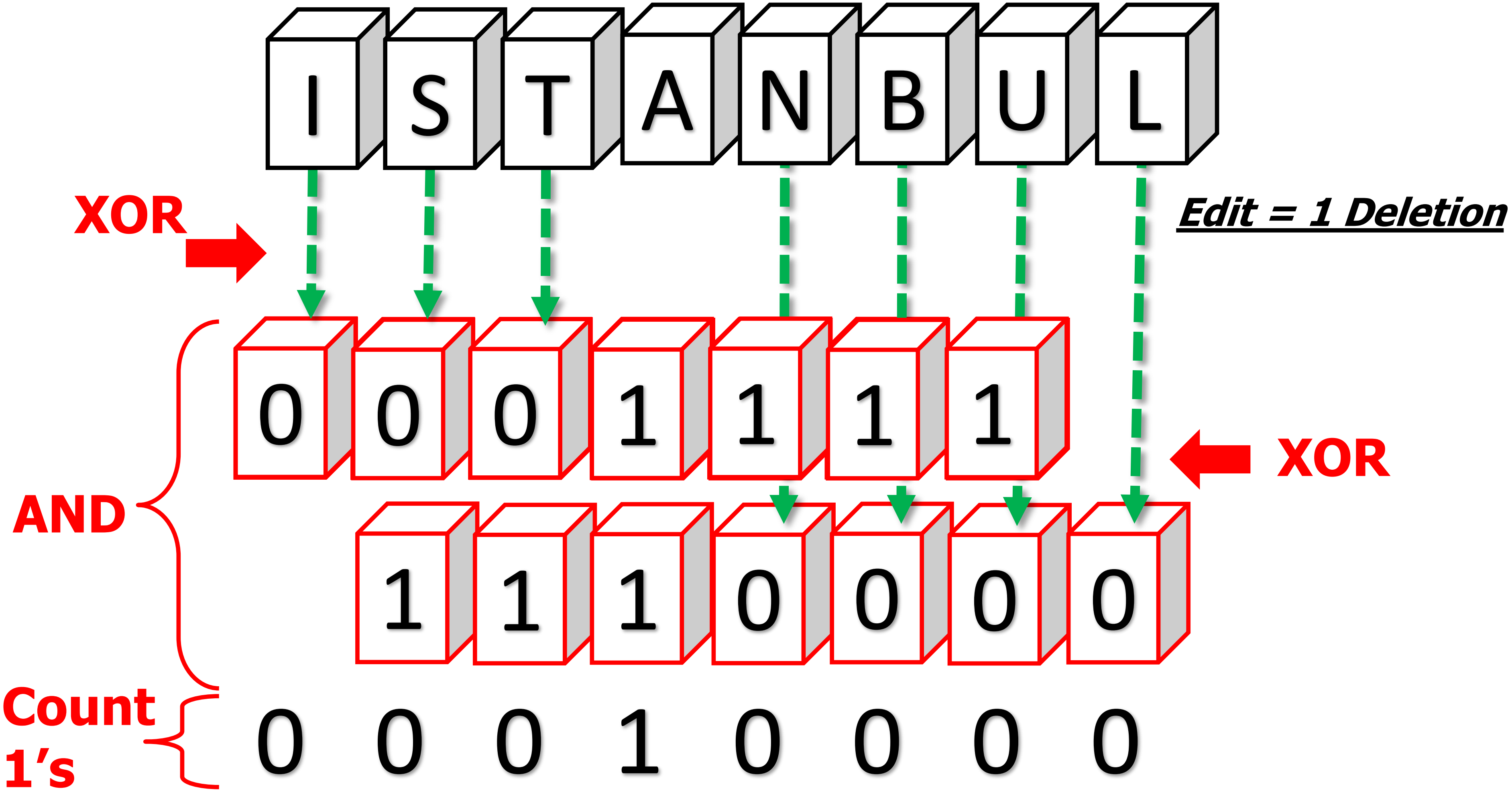
To cancel the effect of the deletion, we need to shift in the *right* direction

Insight: Shifting a String Helps Similarity Search

7 matches 1 mismatch



Shifted Hamming Distance



I S T A N B U L

Edit = 1 Deletion

XOR

0 0 0 1 1 1 1

XOR

1 1 1 0 0 0 0

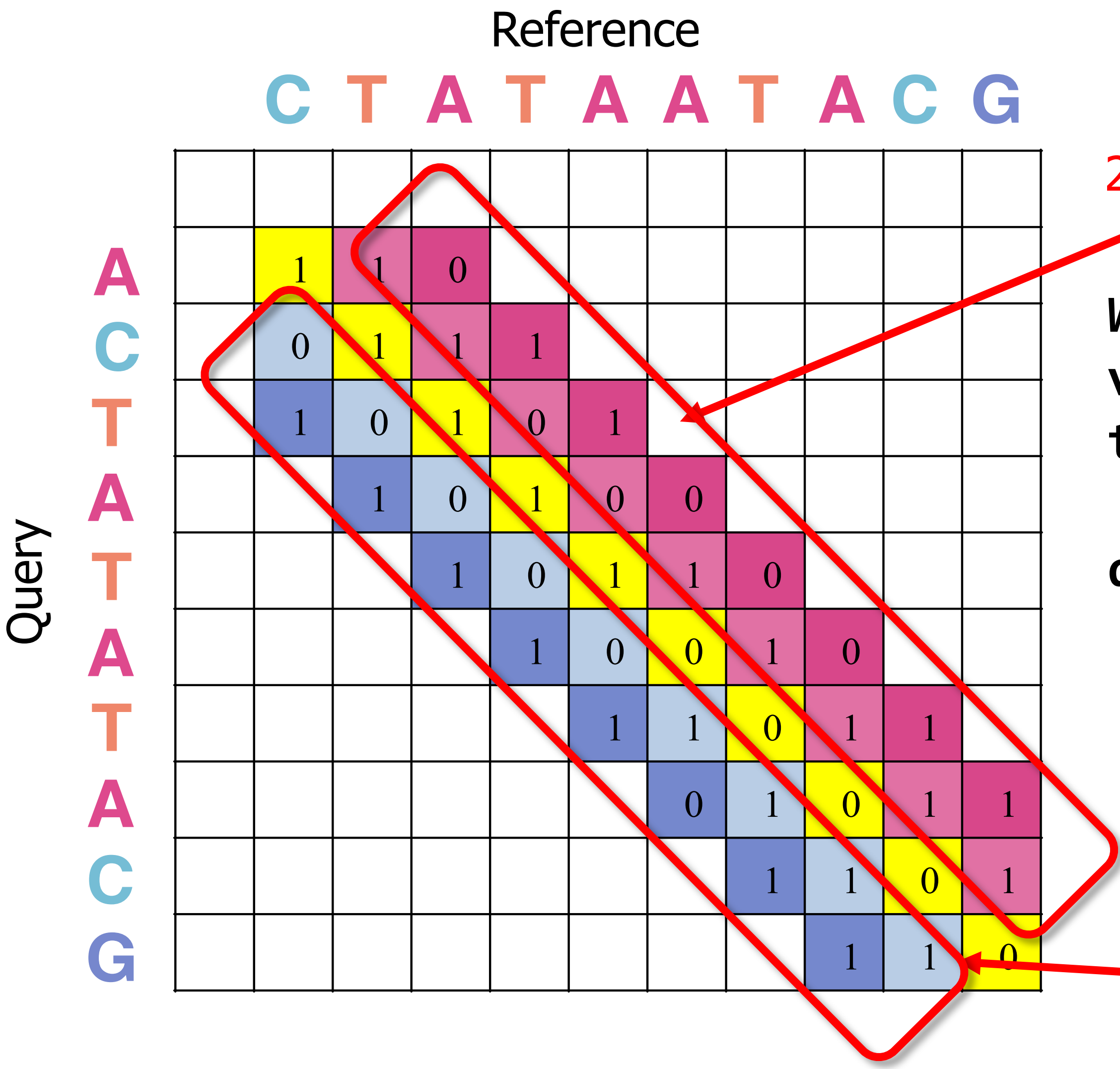
AND

Count 1's

0 0 0 1 0 0 0 0

7 matches 1 mismatch

Highly Parallel Matrix Computation



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

2 Deletion Hamming masks

2 Insertion Hamming masks

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

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

No data dependencies!

Key Idea of SHD Filtering

Generate 2E+1 masks

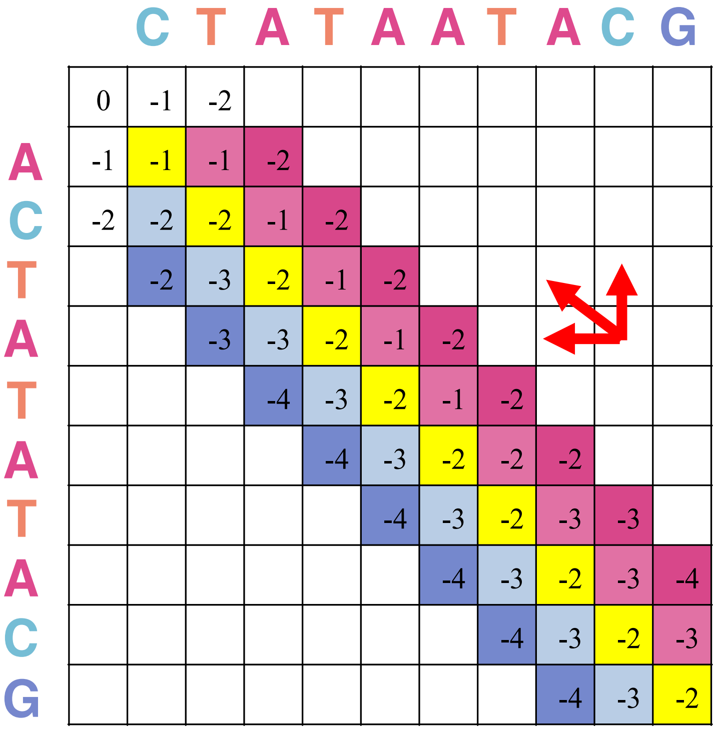
Amend random zeros:
101 → 111 & 1001 → 1111

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

```
             Query :GAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAACATTGTTGGGCCGGA
         Reference :GAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGGCCGG

      Hamming Mask :0000000000100000000000011111110111100011101101011011111111110001000001111011010010101
   1-Deletion Mask :1111111111100111110111110000000000000000000000000000000000000000000011000000000000000
   2-Deletion Mask :0000000010110111001111111111111011110001110110101101111111111000100010011101101001010
   3-Deletion Mask :1111111111101110110011011101110110001001001111111111111100101100110010110111011101111
  1-Insertion Mask :1111111111101111101111110111011000100100111111111111110010110011000101011101110111110
  2-Insertion Mask :0000001001111110011111111110010001101010011010111111111111101110011111111000111101100
  3-Insertion Mask :1111111011101100110001111111111010110111111100110010111011111111011101111010111001000

                          --- Masks after amendment ---

      Hamming Mask :0000000000100000000000011111111111100011111111011111111111110001000001111111111111111
   1-Deletion Mask :1111111111111111111111110000000000000000000000000000000000000000000011000000000000000
   2-Deletion Mask :0000000011111111111111111111111111110001111111111111111111111100010001111111111111110
   3-Deletion Mask :1111111111111111111111111111111110001111111111111111111111111111111111111111111111111
  1-Insertion Mask :1111111111111111111111111111111000111111111111111111111111111111000111111111111111110
  2-Insertion Mask :0000001111111111111111111111110001111111111111111111111111111111111111111000111111100
  3-Insertion Mask :1111111111111111110001111111111111111111111111111111111111111111111111111111111111000

          AND Mask :0000000000100000000000010000000000000000000000000000000000000000000001000000000000000

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

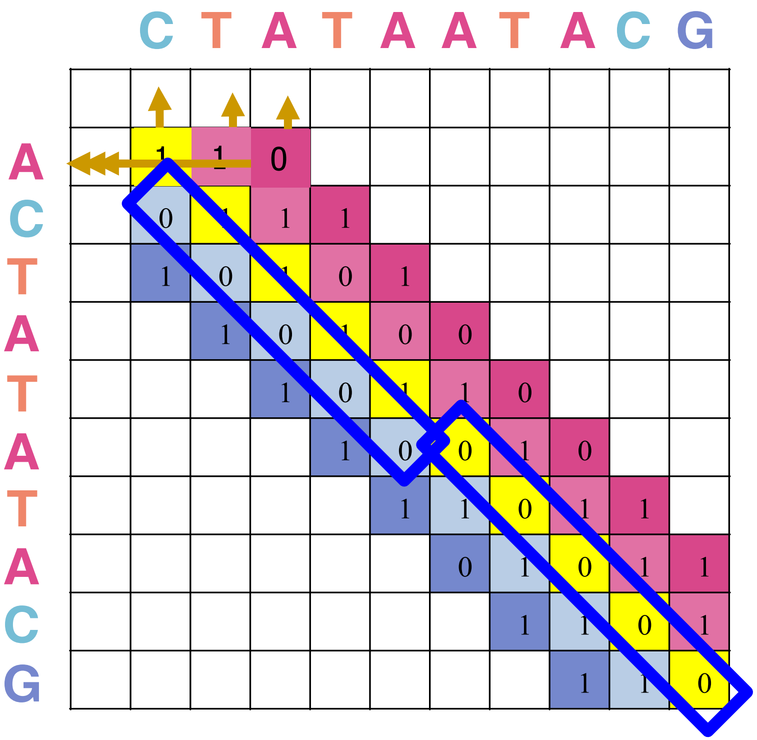
Alignment vs. Pre-alignment (Filtering)

Needleman-Wunsch



		C	T	A	T	A	A	T	A	C	G
	0	-1	-2								
A	-1	-1	-1	-2							
C	-2	-2	-2	-1	-2						
T		-2	-3	-2	-1	-2					
A			-3	-3	-2	-1	-2				
T				-4	-3	-2	-1	-2			
A					-4	-3	-2	-2	-2		
T						-4	-3	-2	-3	-3	
A							-4	-3	-2	-3	-4
C								-4	-3	-2	-3
G									-4	-3	-2

Neighborhood Map



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

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

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

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

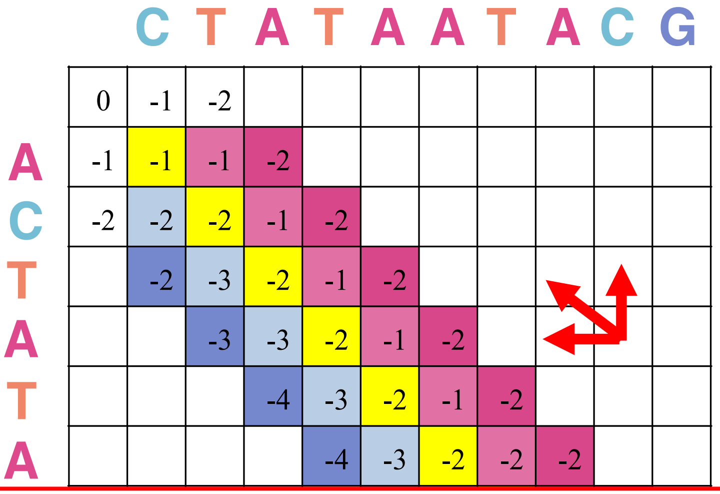
Each cell depends on three pre-computed cells!

No data dependencies!

SAFARI

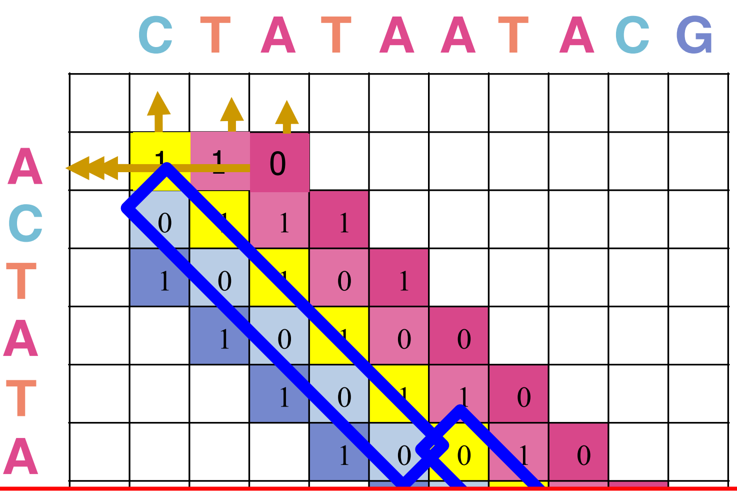
Alignment Matrix vs. Neighborhood Map

Needleman-Wunsch



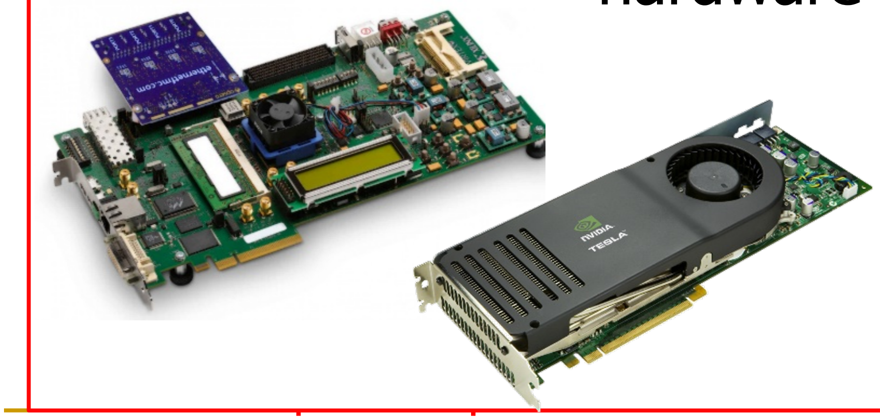
		C	T	A	T	A	A	T	A	C	G
	0	-1	-2								
A	-1	-1	-1	-2							
C	-2	-2	-2	-1	-2						
T		-2	-3	-2	-1	-2					
A			-3	-3	-2	-1	-2				
T				-4	-3	-2	-1	-2			
A					-4	-3	-2	-2	-2		

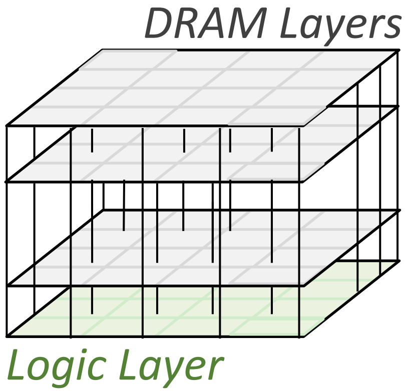
Neighborhood Map



		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		

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

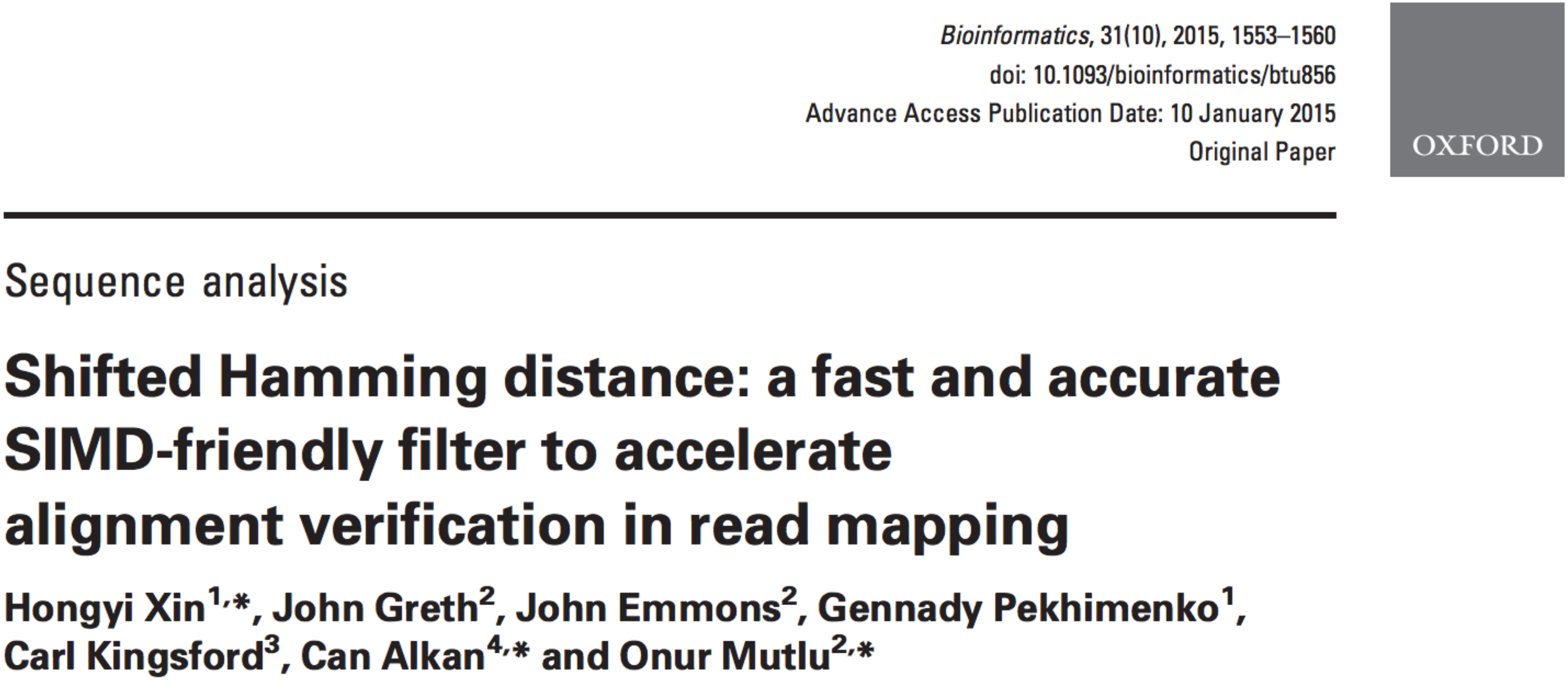
→

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

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

Agenda

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

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

- Hardware Acceleration
    - Specialized Architectures
    - Processing in Memory

- Future Opportunities: New Sequencing Technologies

Location Filtering

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

- M t
  f
  1
  1
  2
  out mismatches quickly

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

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

Ideal Filtering Algorithm

Minimal False Accept Rate

Maximal True Reject Rate

Filter out all incorrect mappings

Zero False Reject Rate

Do not filter out any correct mappings

Faster Than Mapper

Alignment vs. Pre-alignment (Filtering)

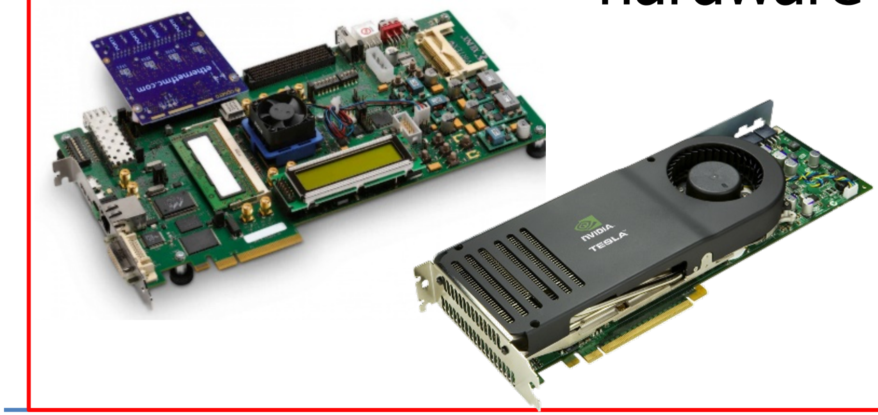
Needleman-Wunsch

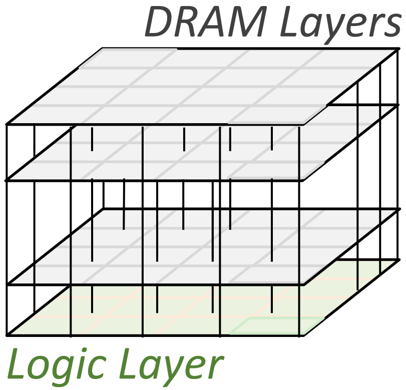
	C	T	A	T	A	A	T	A	C	G
0	1	2								
A	1	0	1	2						
C	2	1	0	1	2					
T		2	1	0	1	2				
A			2	1	2	1	2			
T				2	2	2	1	2		
A					3	2	2	2	2	

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		

Independent vectors can be processed in parallel using hardware technologies





GateKeeper: FPGA-Based Alignment Filtering

Alignment Filter

+

=

1st

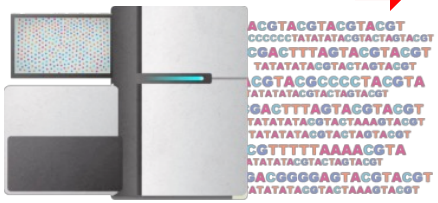
FPGA-based Alignment Filter.

Low Speed & High Accuracy

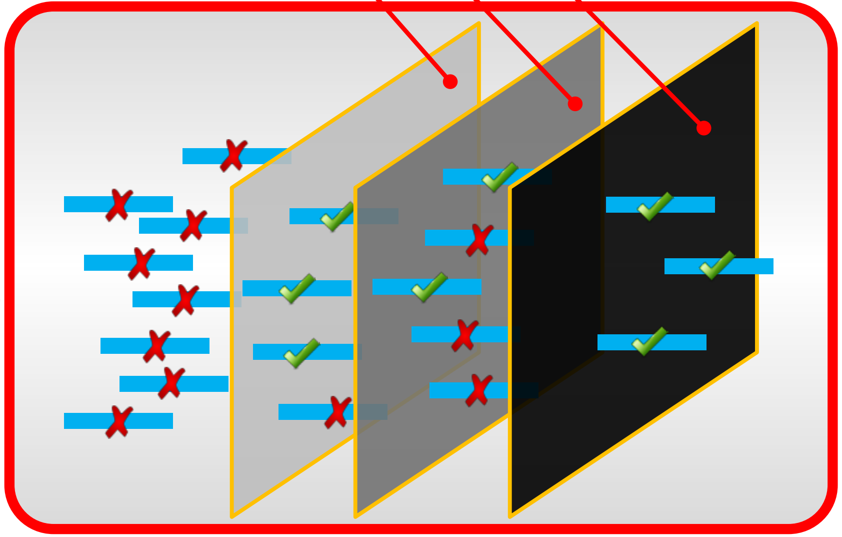
Medium Speed, Medium Accuracy

High Speed, Low Accuracy

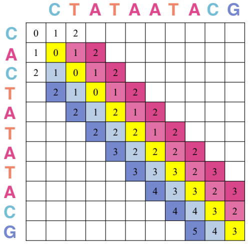
x10^12 mappings



Billions of Short Reads



x10^3 mappings



		C	T	A	T	A	A	T	A	C	G
C	0	1	2								
A	1	0	1	2							
C	2	1	0	1	2						
T		2	1	0	1	2					
A			2	1	2	1	2				
T				2	2	2	1	2			
A					3	2	2	2	2		
T						3	3	3	2	3	
A							4	3	3	2	3
C								4	4	3	2
G									5	4	3

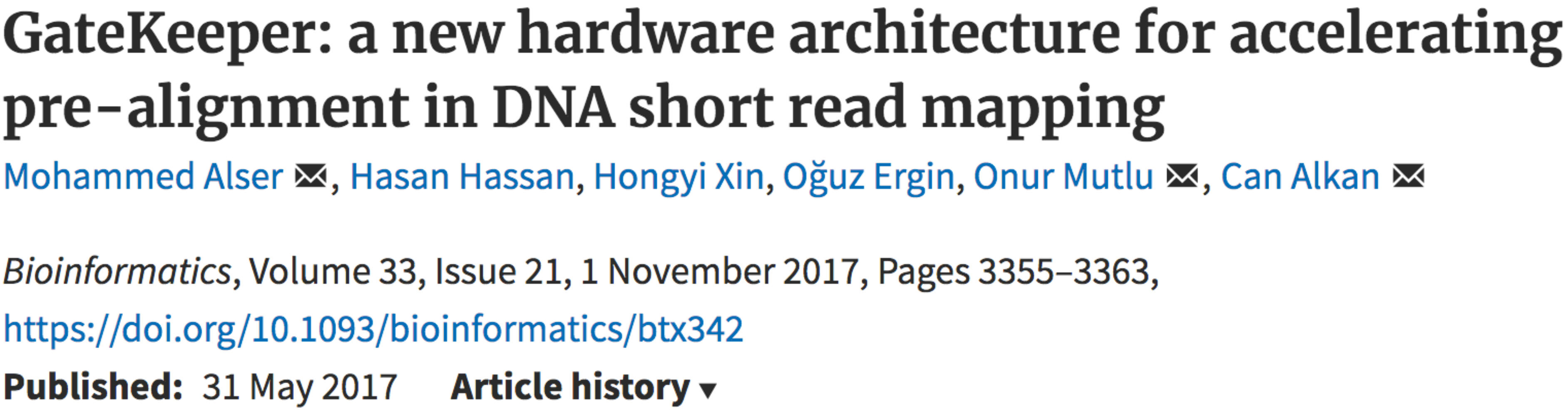
1 **High throughput DNA sequencing (HTS) technologies**

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

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

GateKeeper: FPGA-Based Alignment Filtering

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



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

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

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

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

GateKeeper Walkthrough

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

```
            Query :GAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAACATTGTTGGCCGGA
        Reference :GAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGAGACATTGTTGGCCGG

     Hamming Mask :000000000010000000000000111111011110001110110101101111111110001000001111011010010101
  1-Deletion Mask :111111111110011111011111100000000000000000000000000000000000000000011000000000000000
  2-Deletion Mask :000000001011011100111111111111101111000111011010110111111111000100010011101101001010
  3-Deletion Mask :111111111110111011001101110111011000100100111111111111100101100110010110111011101111
 1-Insertion Mask :111111111110111110111111011101100010010011111111111100101100110001010111011101111110
 2-Insertion Mask :000000100111110011111111100100011010101001101011111111111110111001111111000111101100
 3-Insertion Mask :11111110111011001100011111111101011011111100110010111011111111011101111010111001000

                        --- Masks after amendment ---

     Hamming Mask :000000000010000000000000111111111111000111111101111111111111000100000111111111111111
  1-Deletion Mask :111111111111111111111111100000000000000000000000000000000000000000011000000000000000
  2-Deletion Mask :000000001111111111111111111111111111000111111111111111111111000100011111111111111110
  3-Deletion Mask :111111111111111111111111111111111000111111111111111111111111111111111111111111111111
 1-Insertion Mask :111111111111111111111111111111100011111111111111111111111111100011111111111111111110
 2-Insertion Mask :000000111111111111111111111100011111111111111111111111111111111111111111000111111100
 3-Insertion Mask :111111111111111111000111111111111111111111111111111111111111111111111111111111111000

         AND Mask :000000000010000000000000100000000000000000000000000000000000000000001000000000000000


                   GAGAGAGATATTTAGTGTTGCAG-CACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAACATTGTTGGCCGG
Needleman-Wunsch
        Alignment :|||||||||| |||||||||||| ||||||||||||||||||||||||||||||||||||||||||::|||||||||||||||
                   GAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGAGACATTGTTGGCCGG
```

GateKeeper Walkthrough (cont'd)

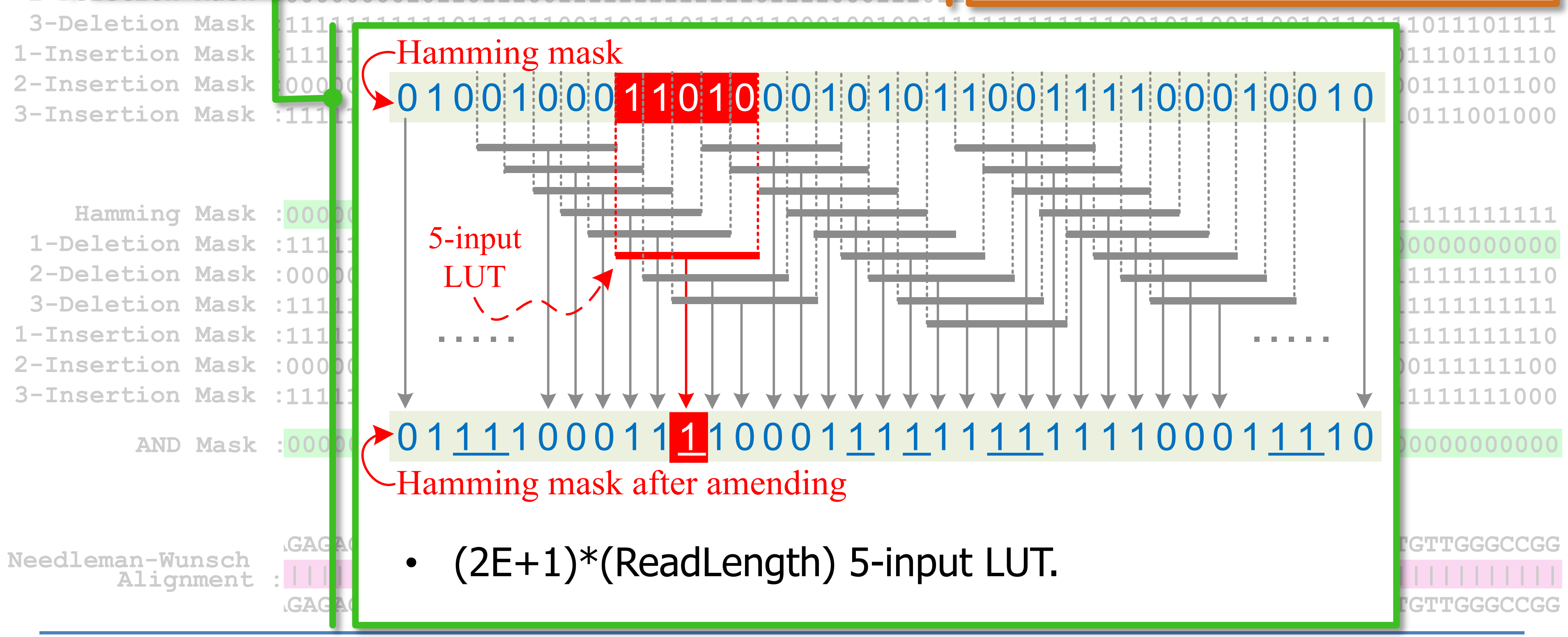
Generate 2E+1 masks

Amend random zeros: 101 → 111 & 1001 → 1111

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

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

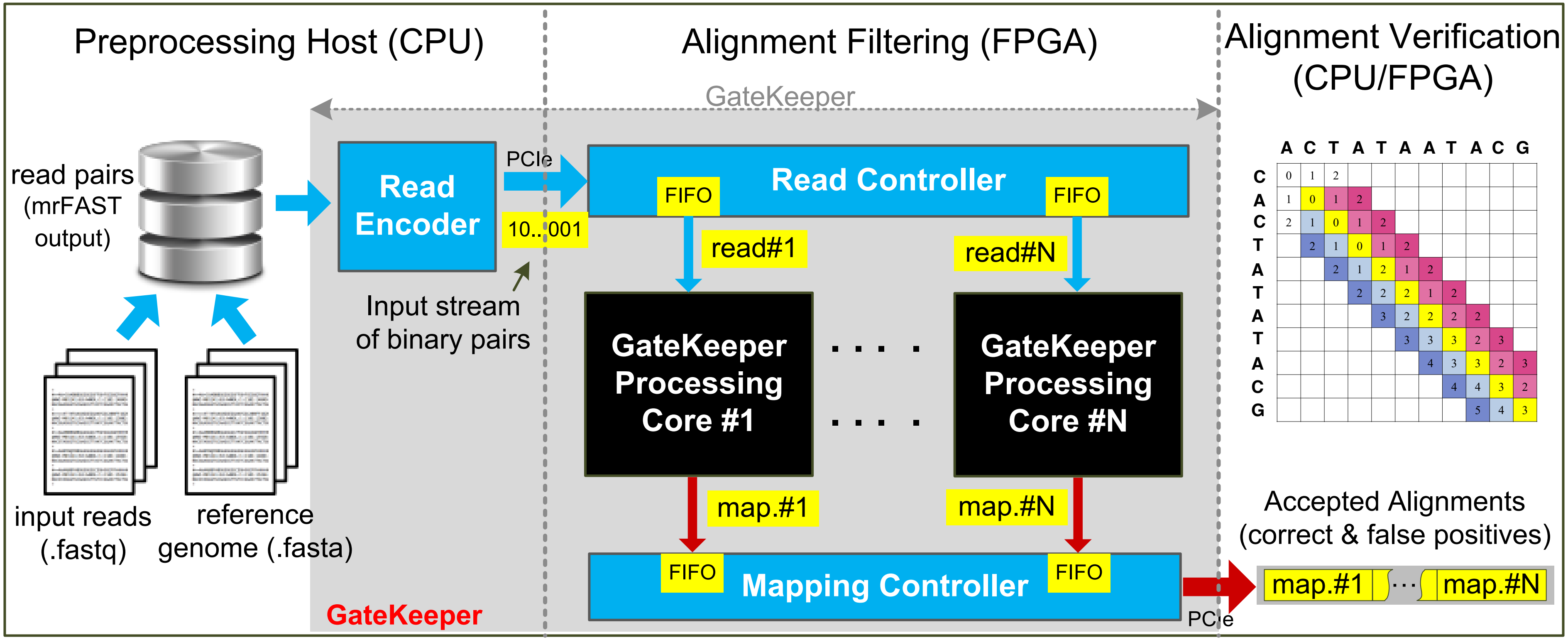
- (2E)*(ReadLength) 2-AND operations.
- (ReadLength/4) 5-input LUT.
- log_2ReadLength-bit counter.



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

GateKeeper Accelerator Architecture

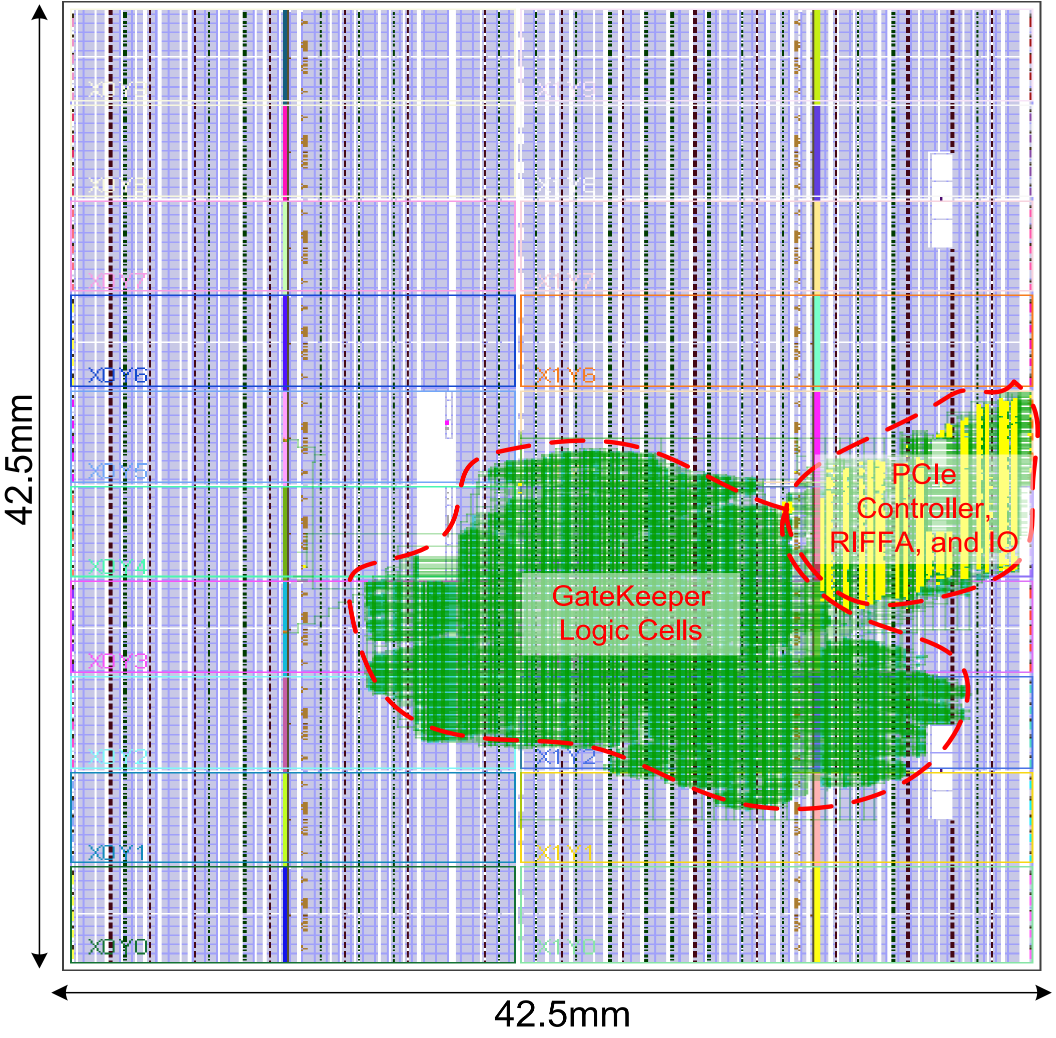
- **Maximum data throughput** =~13.3 billion bases/sec
- Can examine **8 (300 bp) or 16 (100 bp) mappings concurrently** at 250 MHz
- **Occupies 50%** (100 bp) to **91%** (300 bp) of the FPGA slice LUTs and registers


Preprocessing Host (CPU)
Alignment Filtering (FPGA)
Alignment Verification (CPU/FPGA)
GateKeeper
read pairs (mrFAST output)
input reads (.fastq)
reference genome (.fasta)
Read Encoder
PCIe
10...001
Input stream of binary pairs
Read Controller
FIFO
FIFO
read#1
read#N
GateKeeper Processing Core #1
. . . .
. . . .
GateKeeper Processing Core #N
map.#1
map.#N
FIFO
FIFO
Mapping Controller
PCIe
GateKeeper
Accepted Alignments (correct & false positives)
map.#1
...
map.#N

	A	C	T	A	T	A	A	T	A	C	G
C	0	1	2								
A	1	0	1	2							
C	2	1	0	1	2						
T		2	1	0	1	2					
A			2	1	2	1	2				
T				2	2	2	1	2			
A					3	2	2	2	2		
T						3	3	3	2	3	
A							4	3	3	2	3
C								4	4	3	2
G									5	4	3

FPGA Chip Layout

<ins>GateKeeper</ins>: 17.6%, <ins>PCIe Controller, RIFFA, and IO</ins>: 5%



<ins>300 bp</ins>

<ins>E=15</ins>

GateKeeper vs. SHD

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

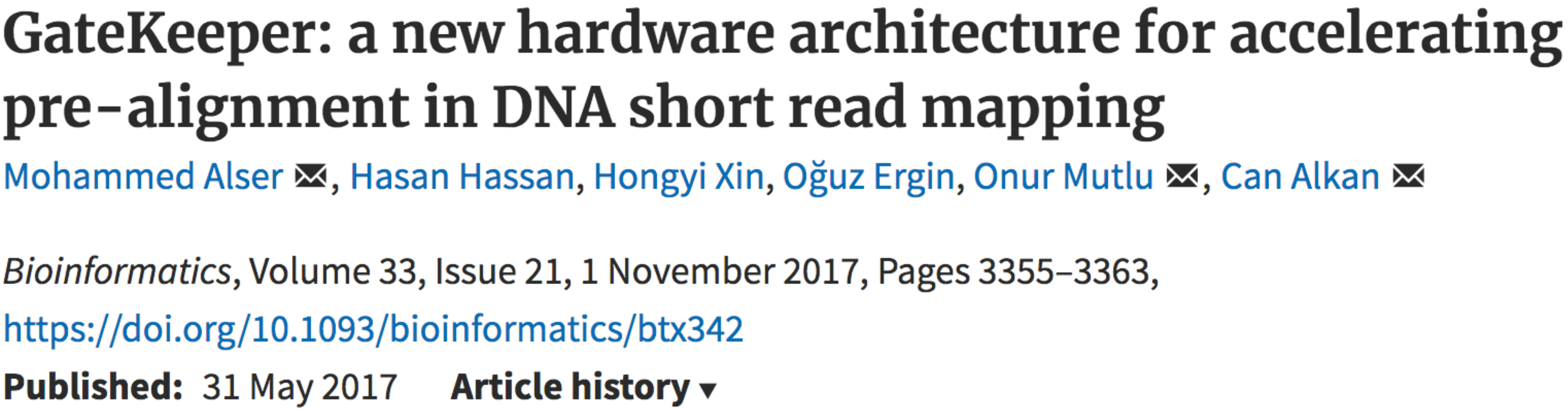
GateKeeper Conclusions

- 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]



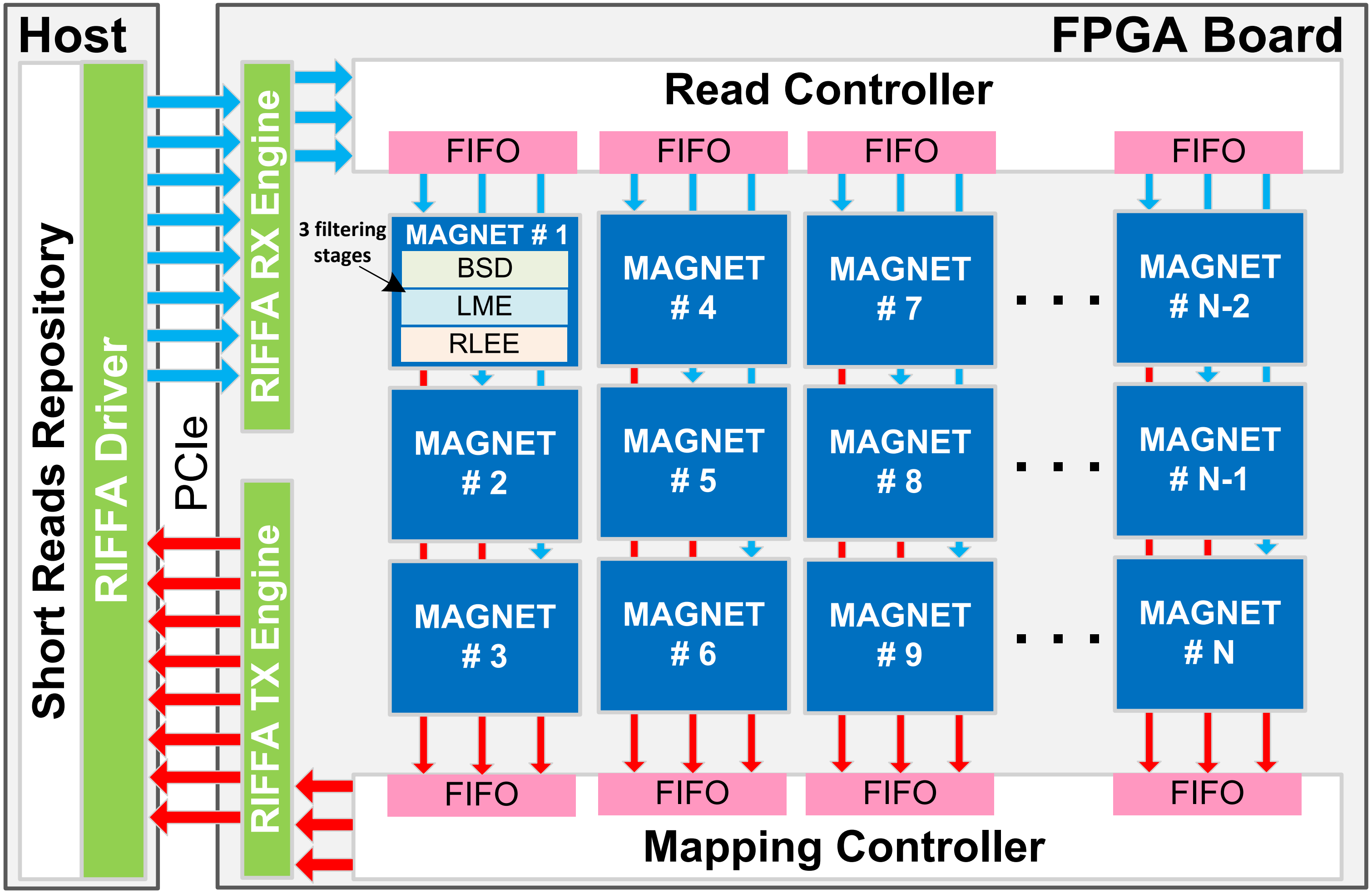
GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping

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]



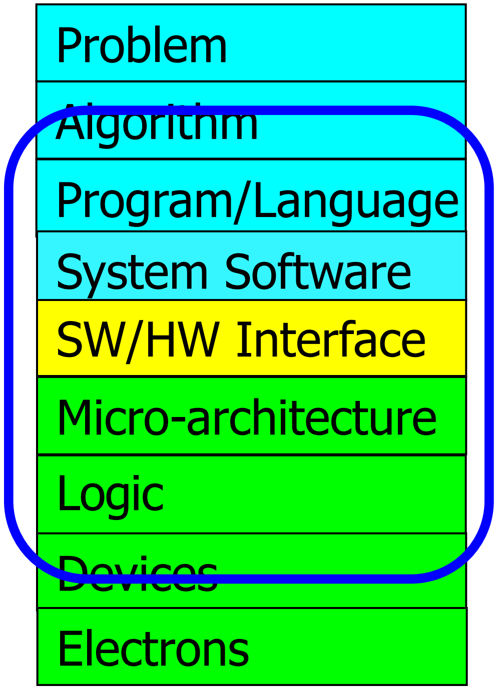
Can We Do Better?

Faster, More Accurate, More Scalable

Pre-Alignment Filtering

Algorithm-Arch-Device Co-Design is Critical

Computer Architecture (expanded view)



Problem

Algorithm

Program/Language

System Software

SW/HW Interface

Micro-architecture

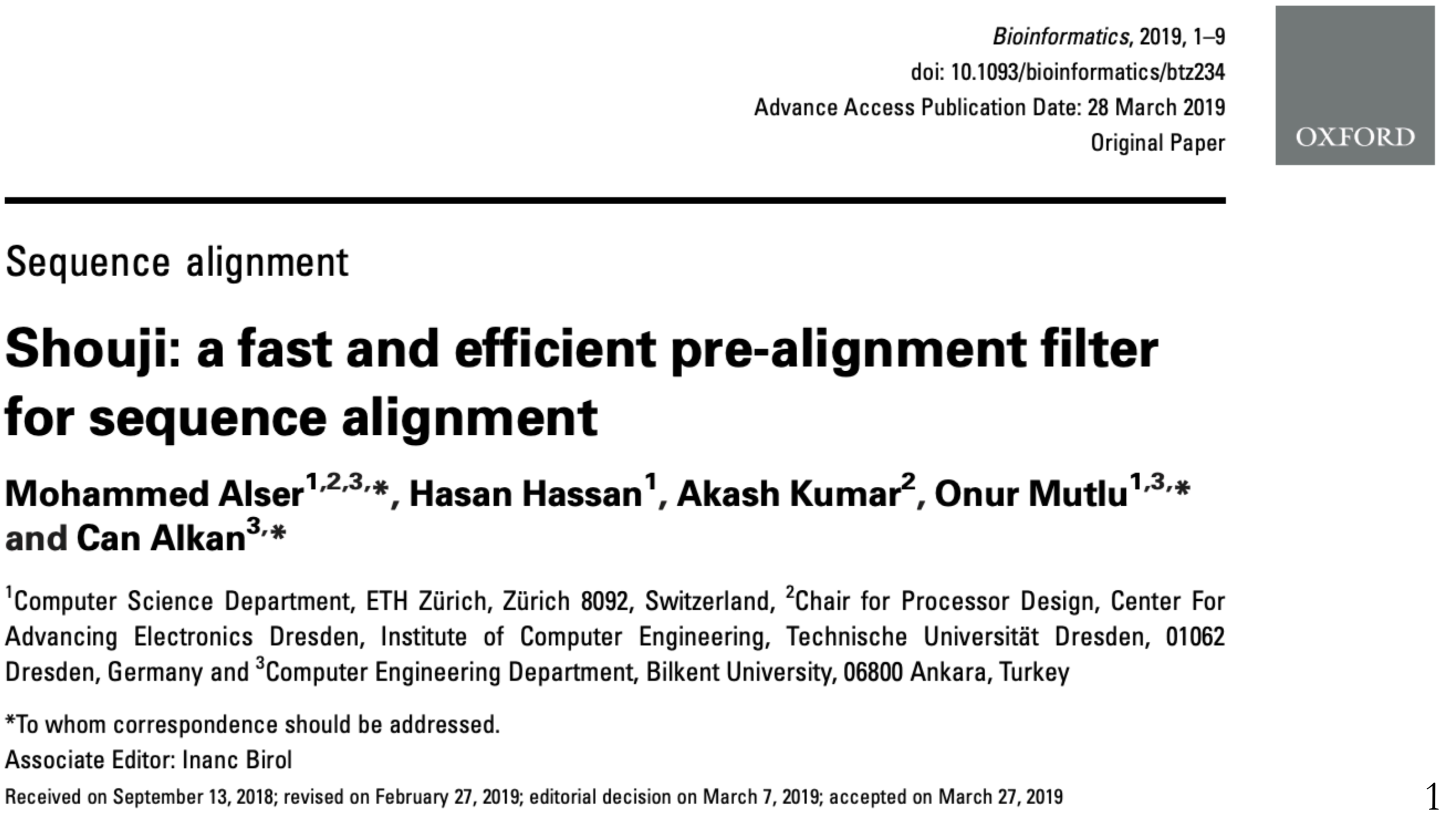
Logic

Devices

Electrons

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

Sequence alignment

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

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

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

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

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

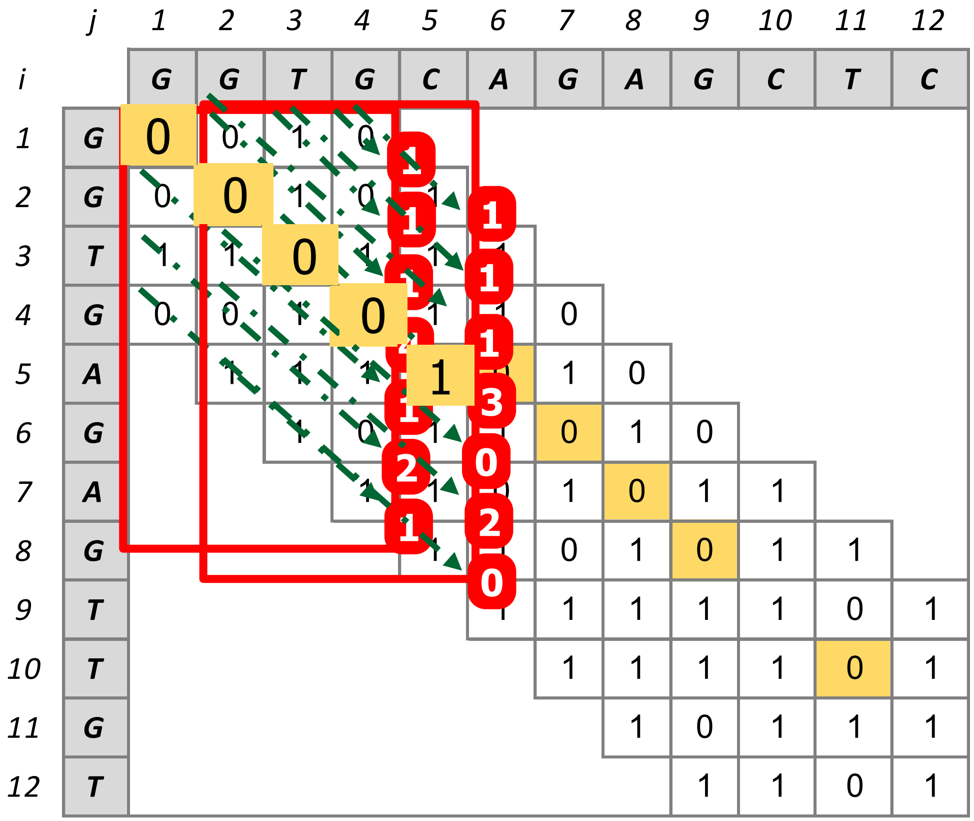
Shouji

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

Shouji Walkthrough

Build the Neighborhood Map

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



Store longest subsequence in Shouji Bit-vector

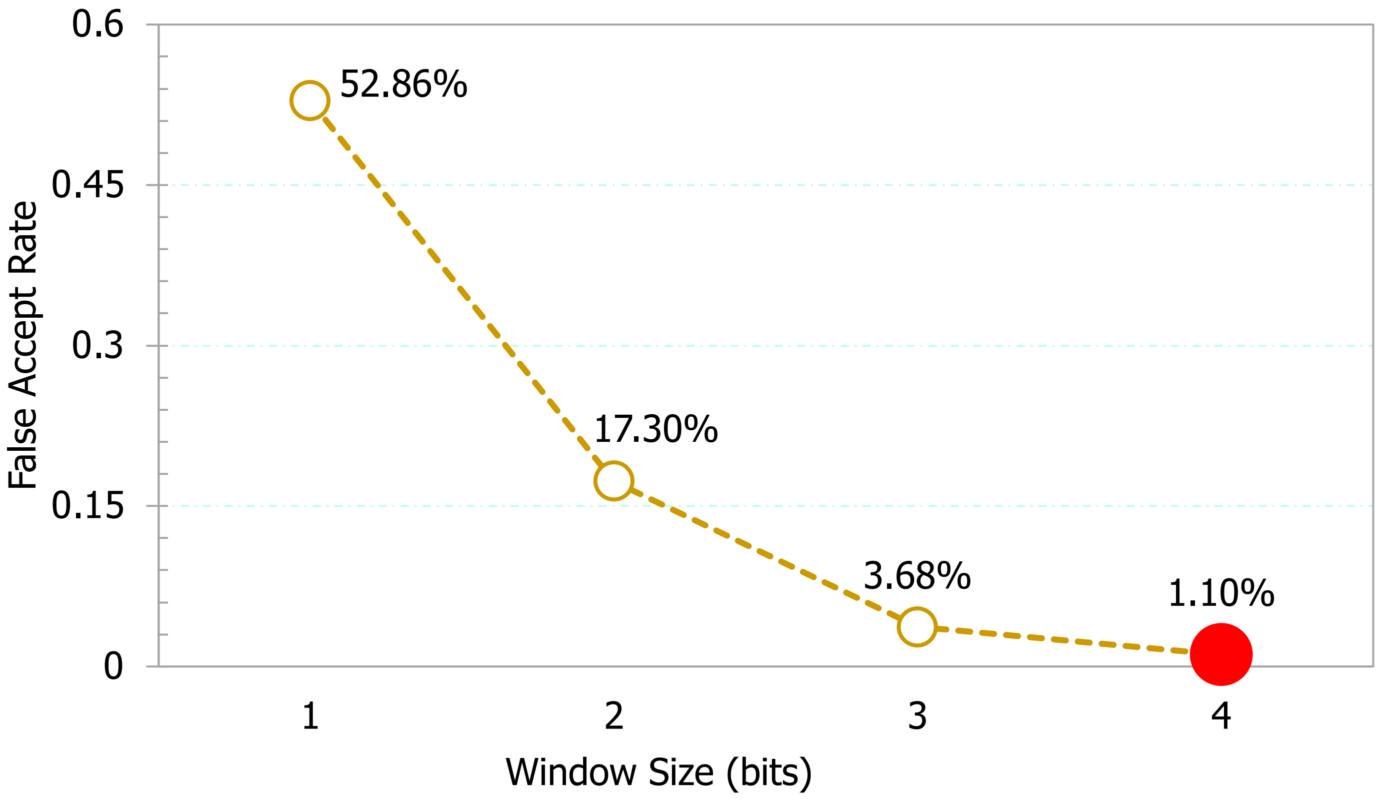
0	0	0	0	1	0	0	0	0	1	0	1

ACCEPT iff number of '1's ≤ Threshold

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

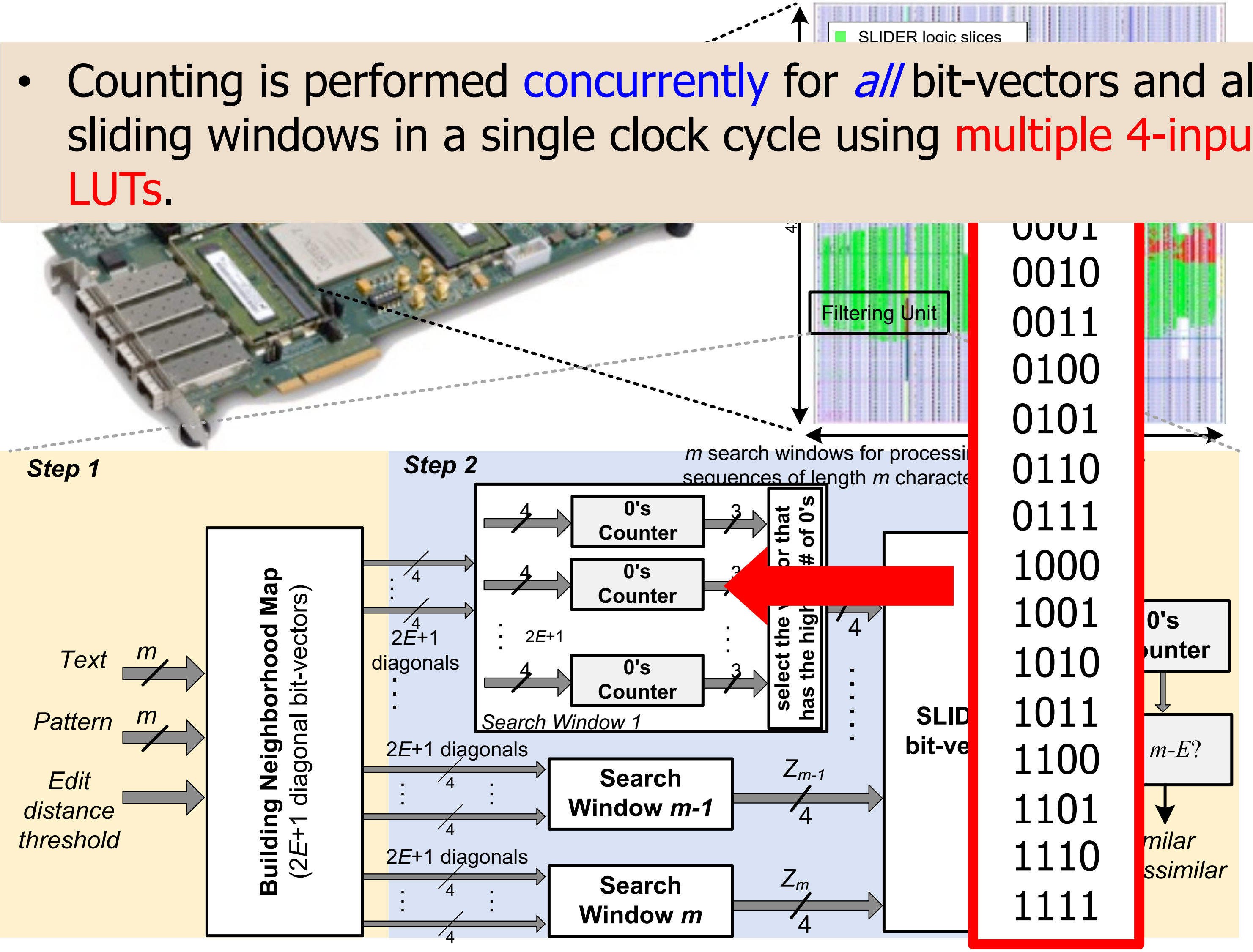
Effect of Sliding Window Size

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



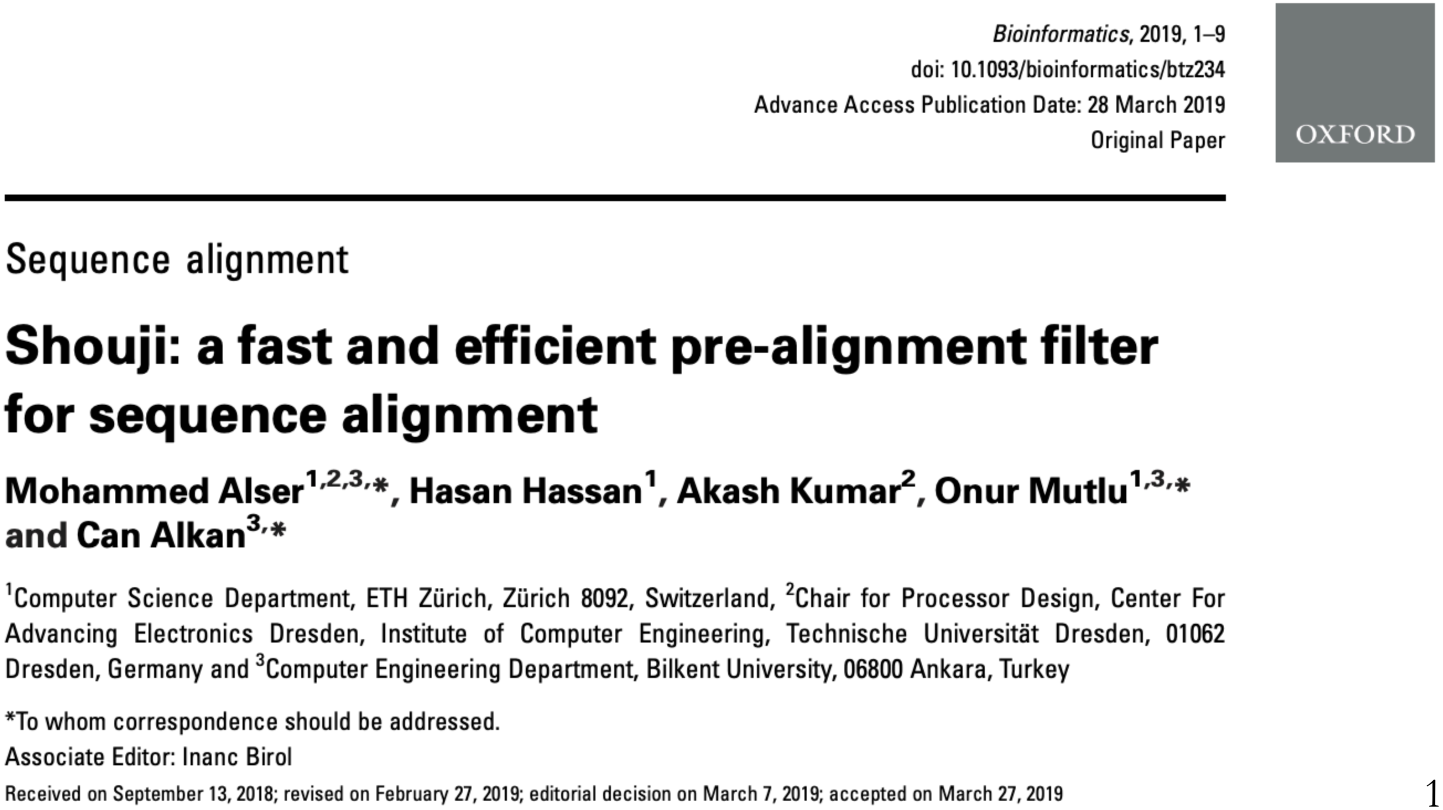
Hardware Implementation

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



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

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



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

Sequence alignment

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

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

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

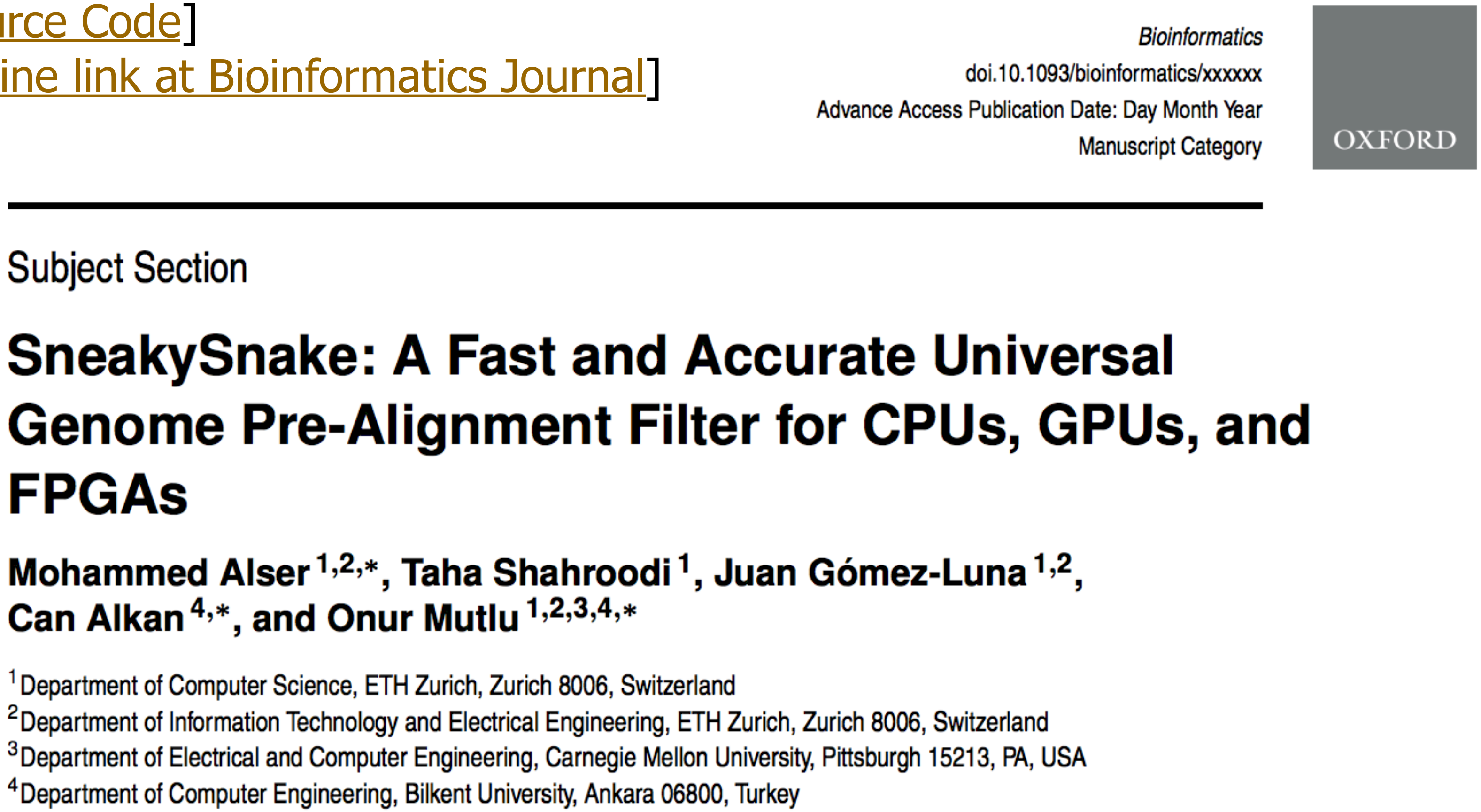
*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

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

SneakySnake [Alser+, Bioinformatics 2020]

Mohammed Alser, Taha Shahroodi, Juan-Gomez Luna, Can Alkan, and Onur Mutlu, **"SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs"**
Bioinformatics, to appear in 2020.
[Source Code]
[Online link at Bioinformatics Journal]



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

OXFORD

Subject Section

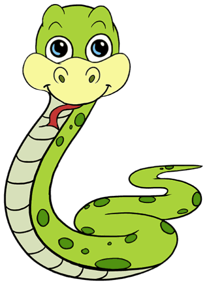
SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs

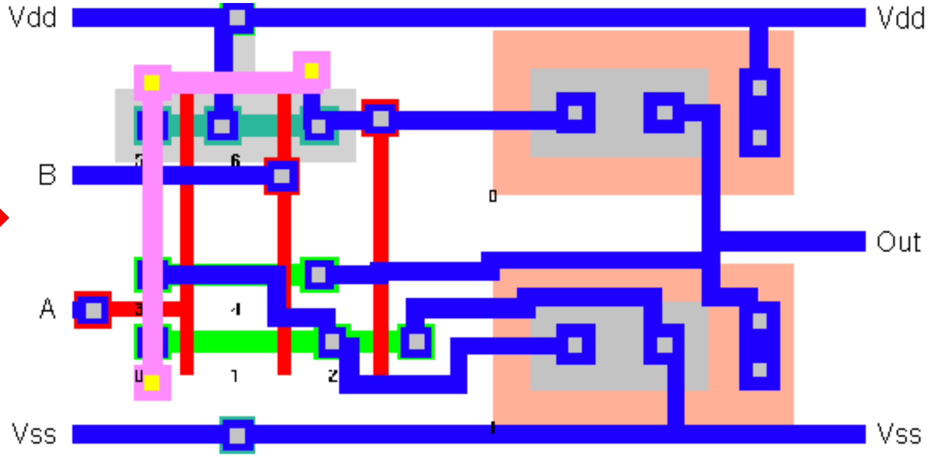
Mohammed Alser[1,2,*], Taha Shahroodi[1], Juan Gómez-Luna[1,2], Can Alkan[4,*], and Onur Mutlu[1,2,3,4,*]

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

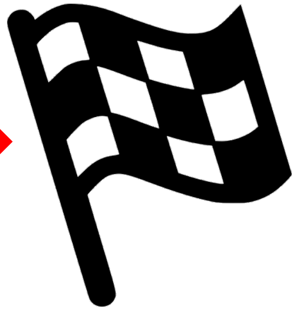
SneakySnake

- **Key observation:**
  - Correct alignment is a sequence of non-overlapping long matches.
- **Key idea:**
  - Reduce the approximate string matching problem to the Single Net Routing problem in VLSI chip layout.





VLSI chip layout



SneakySnake

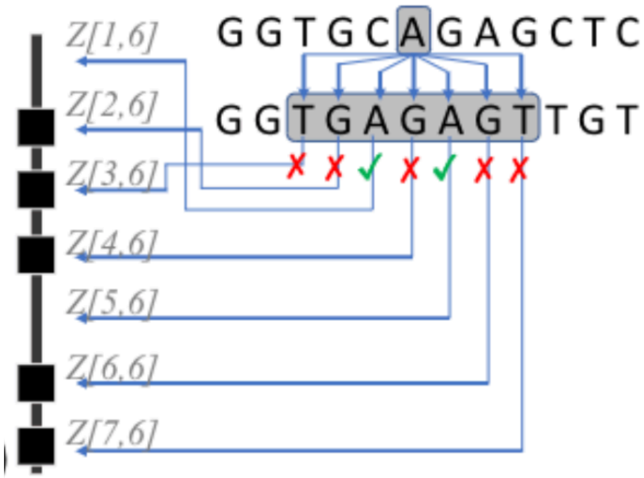
- **Key observation:**
  - Correct alignment is a sequence of non-overlapping long matches.

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

- **Key result:**
  - SneakySnake is up to four orders of magnitude more accurate than Shouji (Bioinformatics'19) and GateKeeper (Bioinformatics'17).
  - SneakySnake greatly accelerates state-of-the-art CPU sequence aligners, Edlib (Bioinformatics'17) and Parasail (BMC Bioinformatics'16)
    - by up to 37.7× and 43.9× (>12× on average), on CPUs
    - by up to 413× and 689× (>400× on average) *with FPGA acceleration*

SneakySnake Walkthrough

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



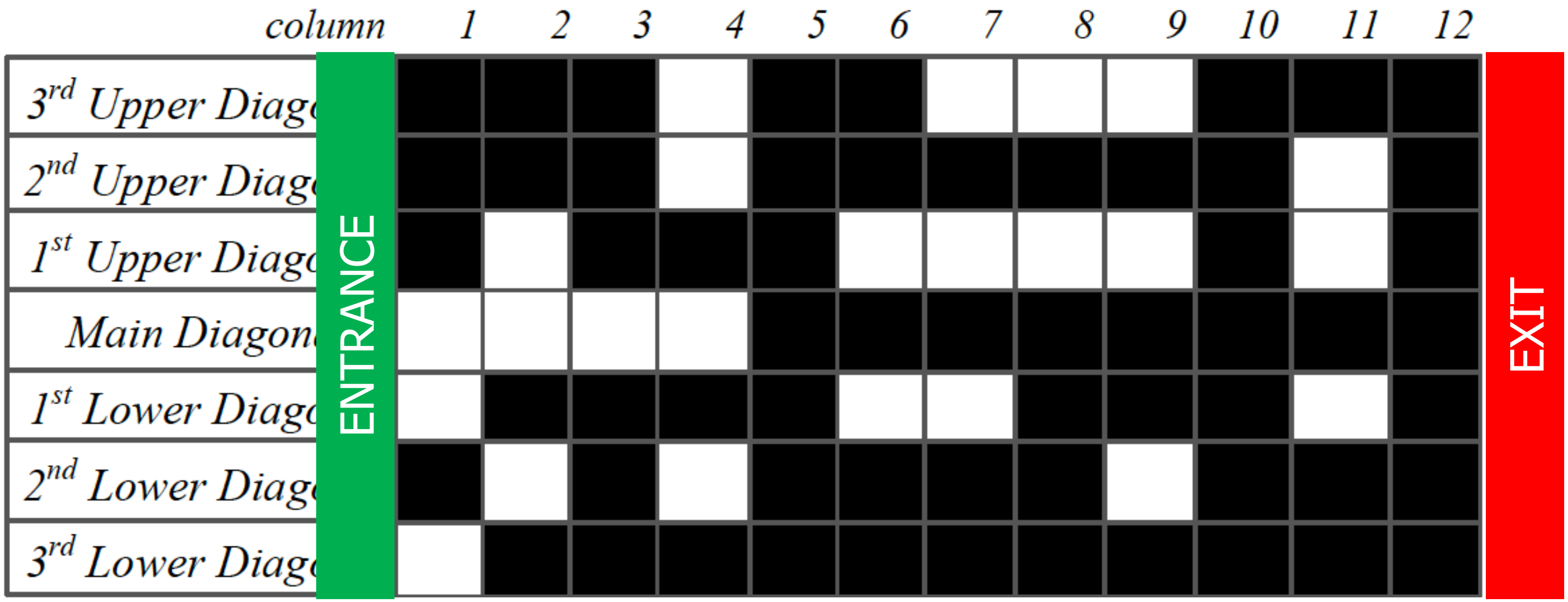
E = 3

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

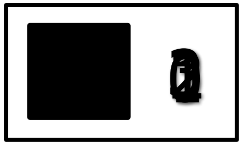
$E = 3$

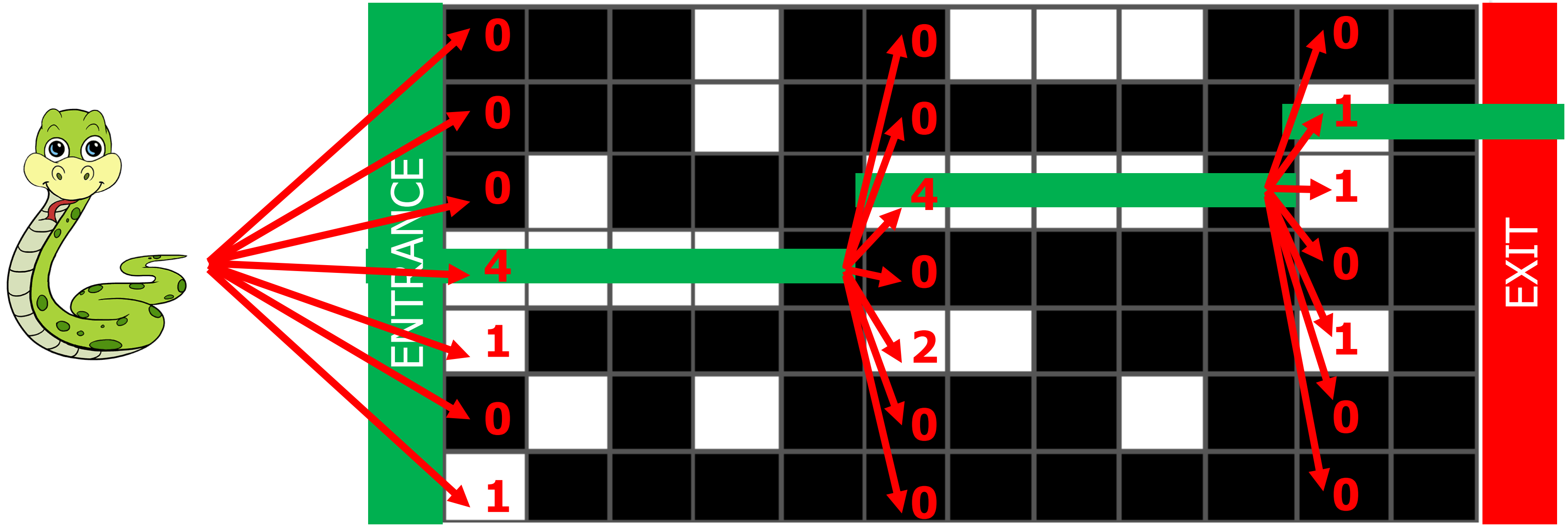


column	1	2	3	4	5	6	7	8	9	10	11	12
3rd Upper Diag	■	■	■	□	■	■	□	□	□	■	■	■
2nd Upper Diag	■	■	■	□	■	■	■	■	■	■	□	■
1st Upper Diag	■	□	■	■	■	□	□	□	□	■	□	■
Main Diagon	□	□	□	□	■	■	■	■	■	■	■	■
1st Lower Diag	□	■	■	■	■	□	□	■	■	■	□	■
2nd Lower Diag	■	□	■	□	■	■	■	■	□	■	■	■
3rd Lower Diag	□	■	■	■	■	■	■	■	■	■	■	■

SneakySnake Walkthrough

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

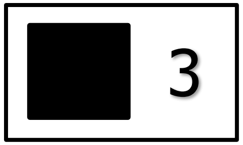


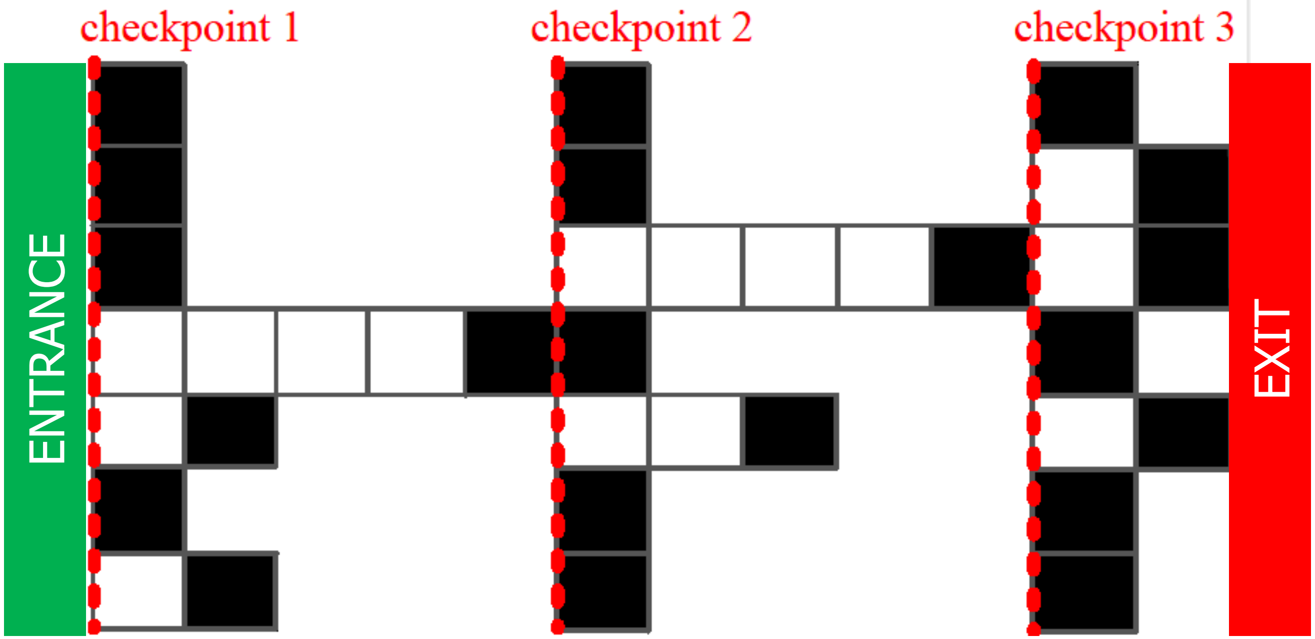


SneakySnake Walkthrough

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

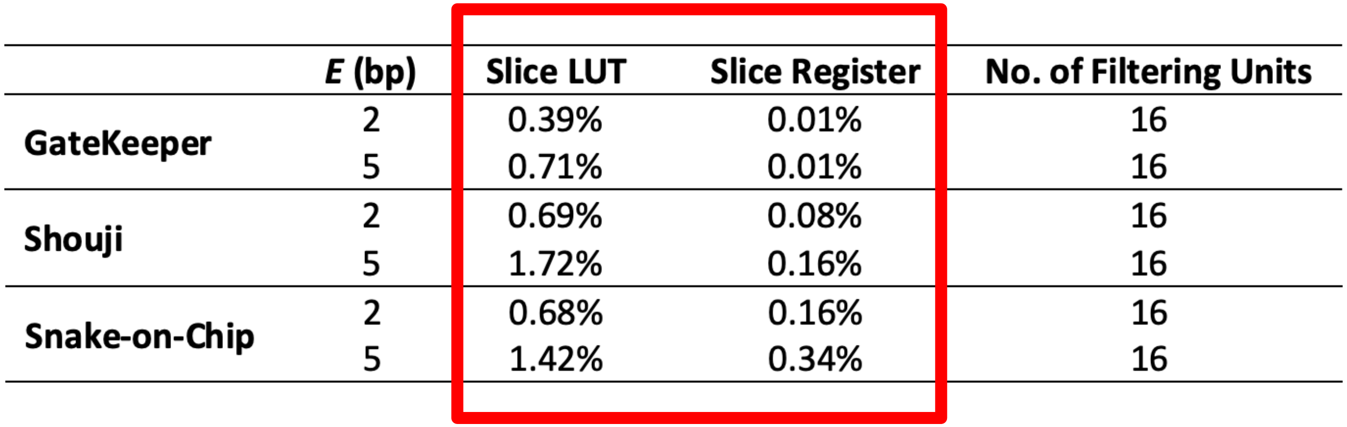
This is what you actually need to build and it can be done on-the-fly!





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 Read Mapping (SneakySnake vs Parasail)

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

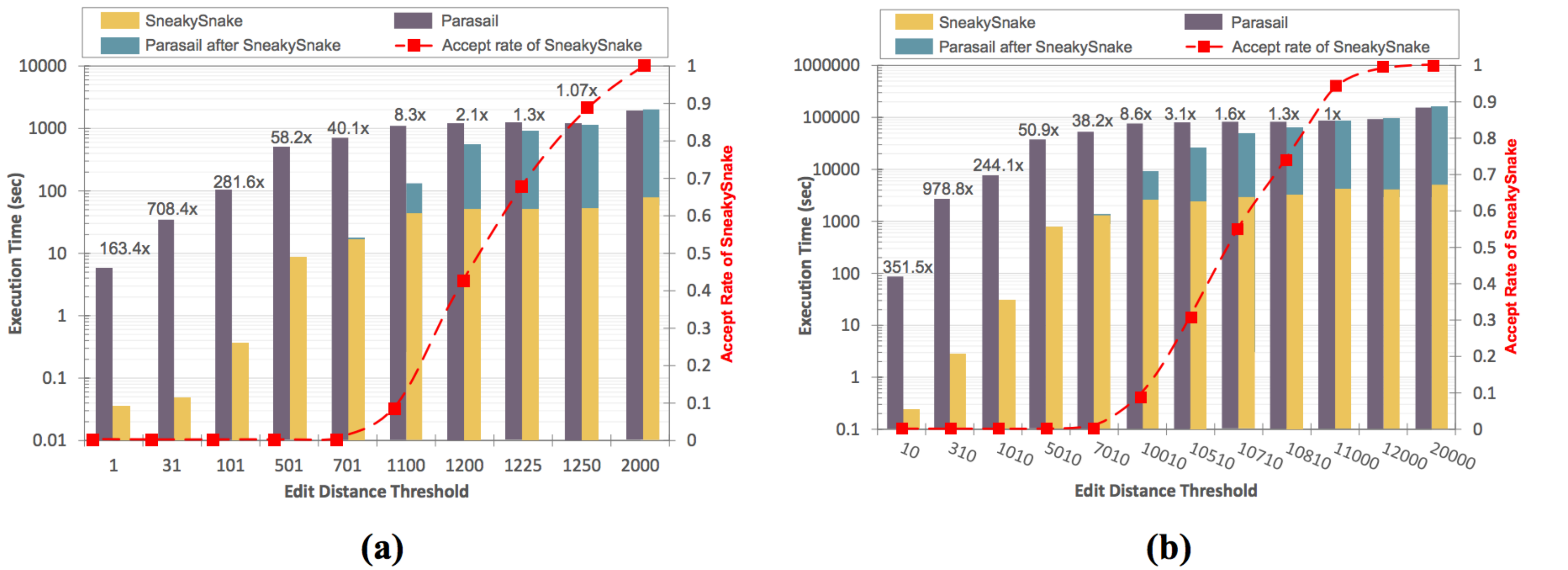
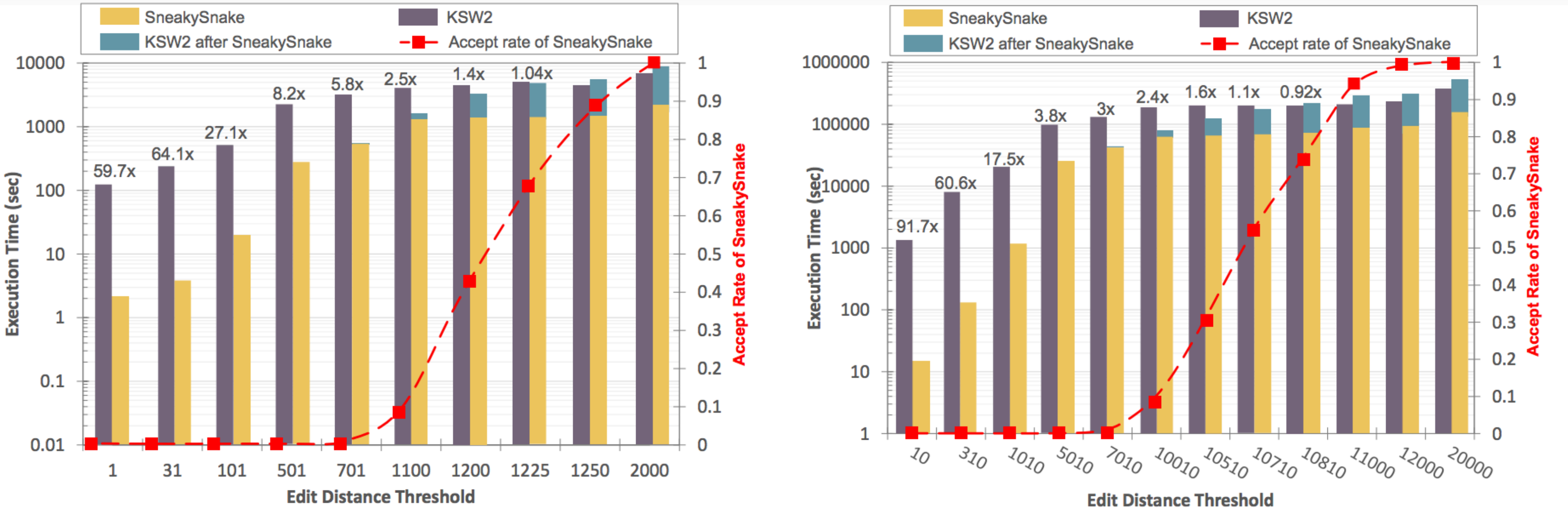


Fig. 10: The execution time of SneakySnake, Parasail, and SneakySnake integrated with Parasail using long sequences, (a) 10Kbp and (b) 100Kbp, and 40 CPU threads. The left y-axes of (a) and (b) are on a logarithmic scale. For each edit distance threshold value, we provide in the right y-axes of (a) and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

Long Read Mapping (SneakySnake vs KSW2)

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

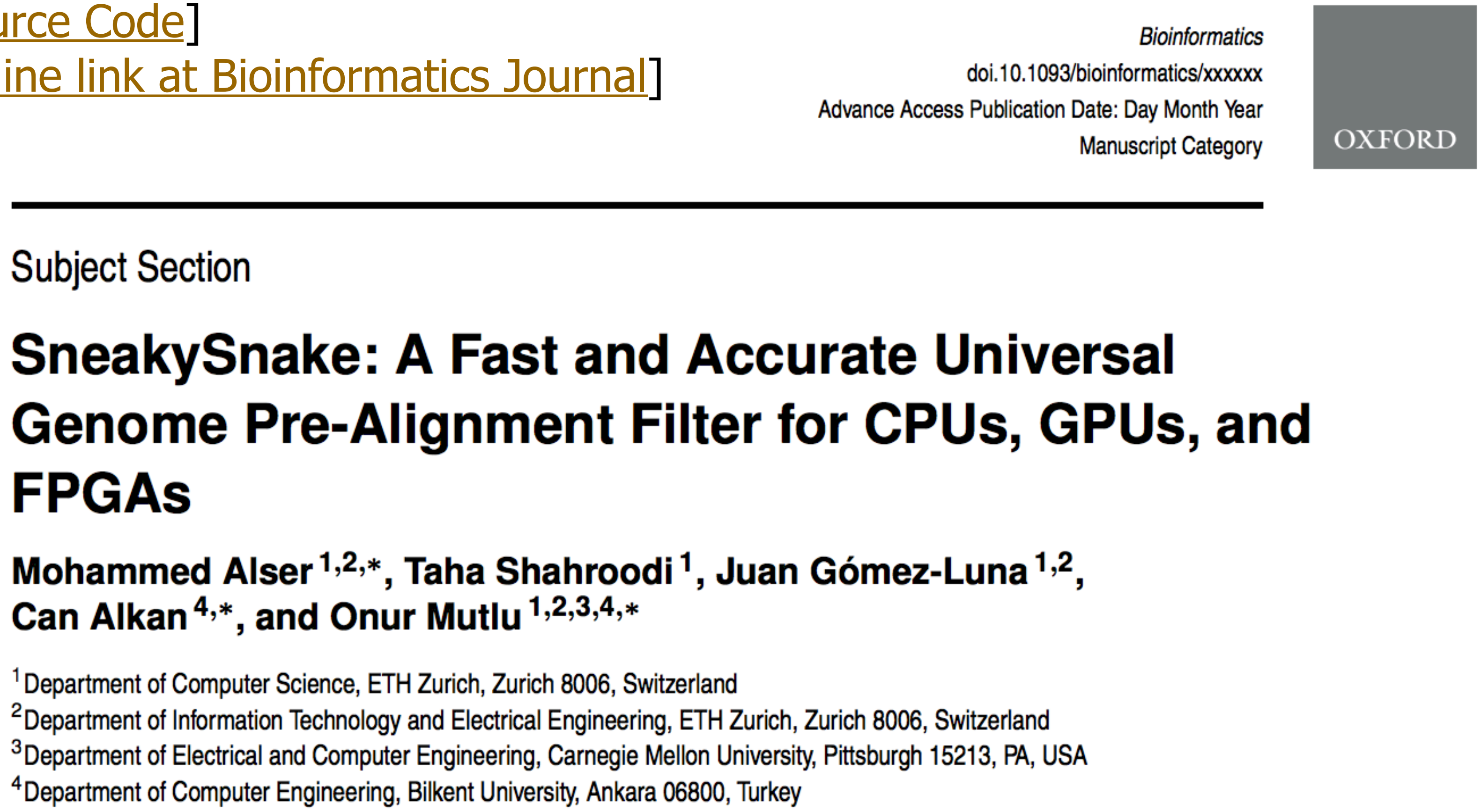


(a) **(b)**

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

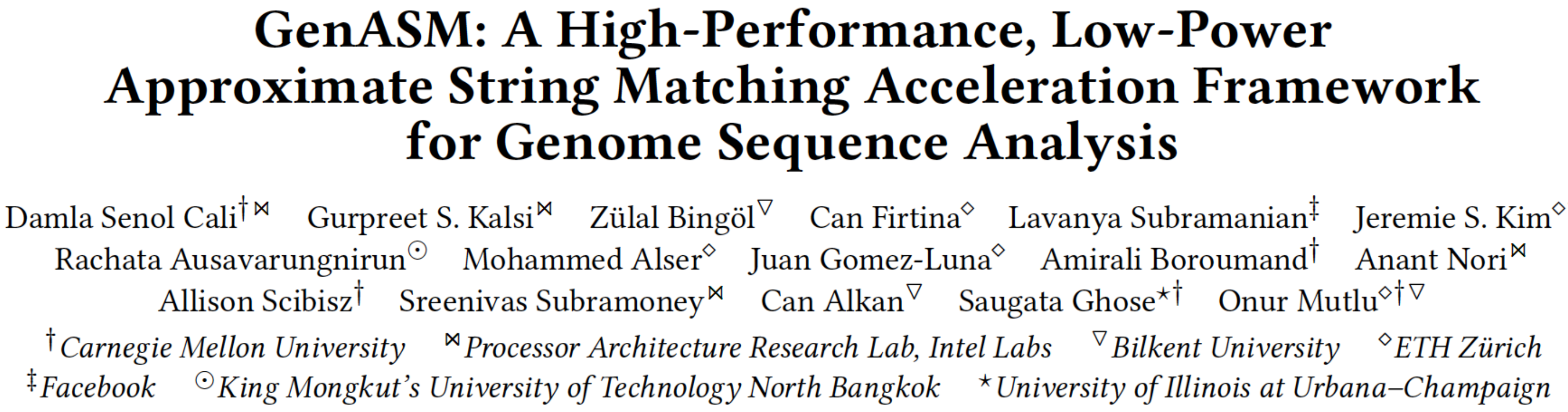
SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs

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

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

GenASM Framework **[MICRO 2020]**

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,
  "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"
  Proceedings of the 53rd International Symposium on Microarchitecture (***MICRO***), Virtual, October 2020.
  [Lighting Talk Video (1.5 minutes)]
  [Lightning Talk Slides (pptx) (pdf)]
  [Talk Video (18 minutes)]
  [Slides (pptx) (pdf)]



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

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

[†]*Carnegie Mellon University* [⋈]*Processor Architecture Research Lab, Intel Labs* [▽]*Bilkent University* [◇]*ETH Zürich*
[‡]*Facebook* [⊙]*King Mongkut's University of Technology North Bangkok* [⋆]*University of Illinois at Urbana–Champaign*

Problem & Our Goal

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

Our Goal:

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

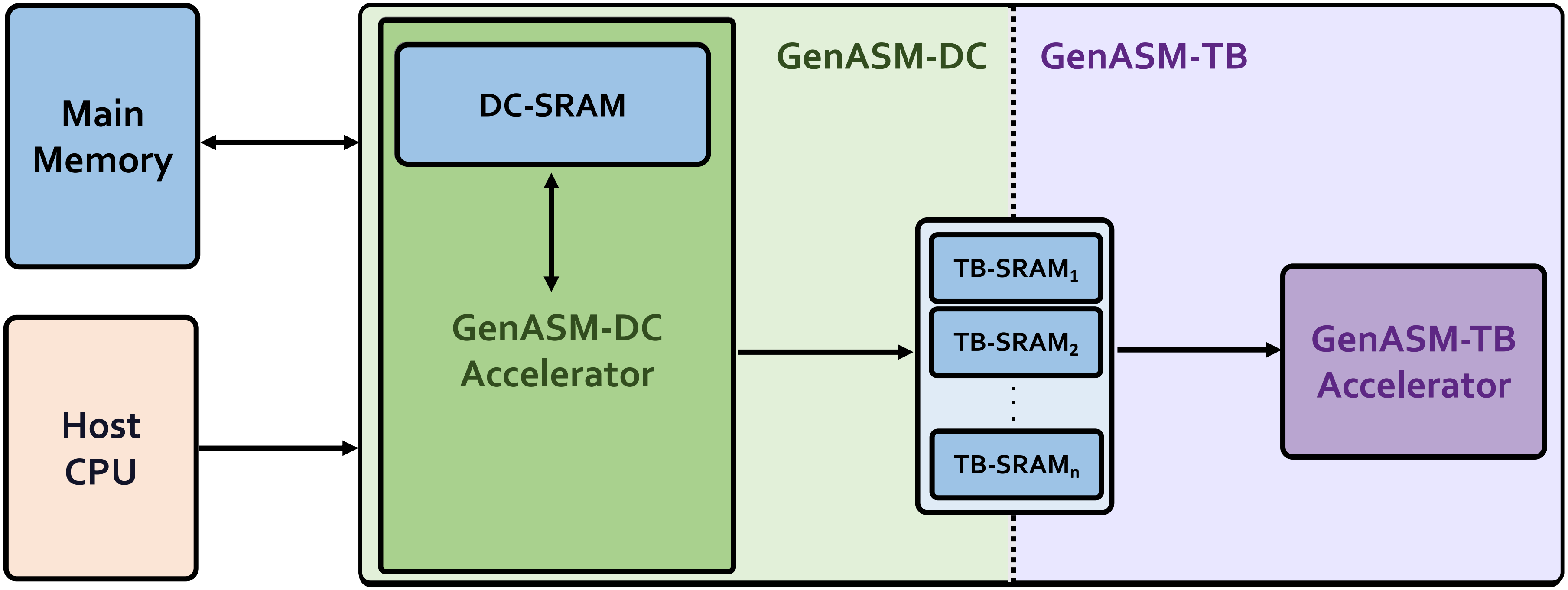
GenASM: ASM Framework for GSA

Our Goal:

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

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

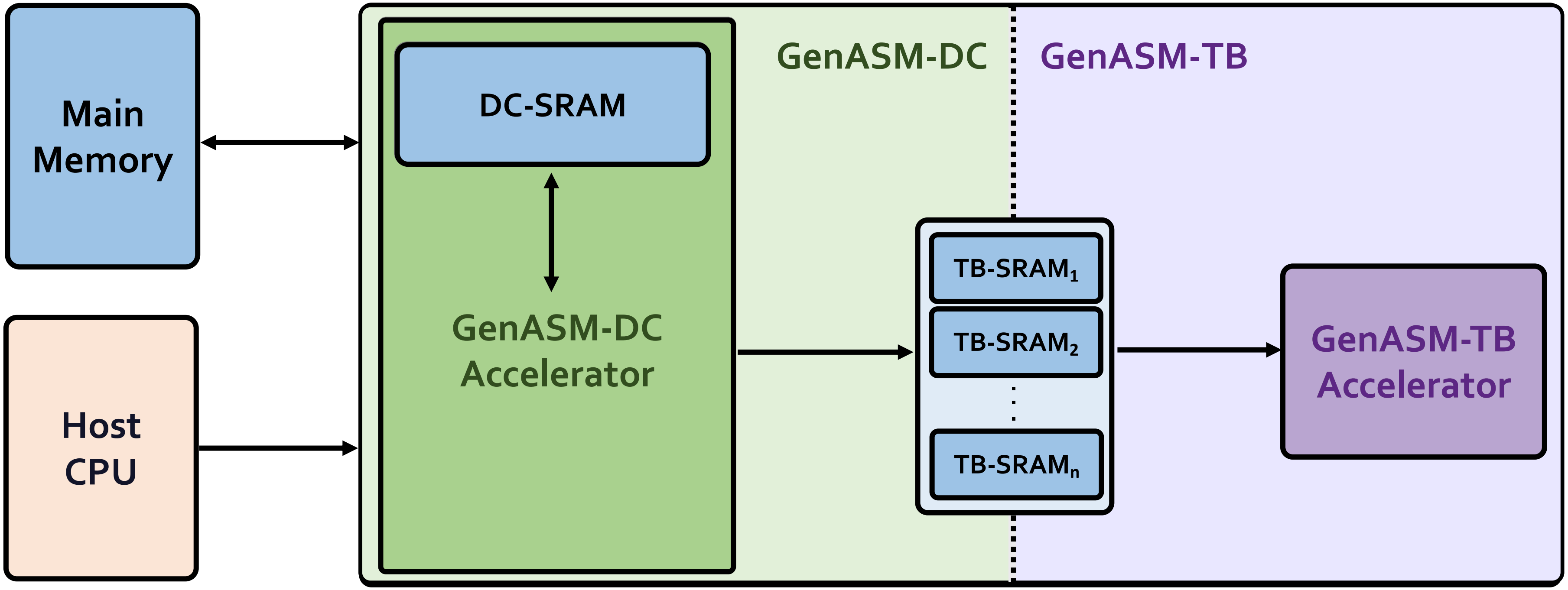
GenASM: Hardware Design



GenASM-DC:
generates bitvectors
and performs edit
Distance Calculation

GenASM-TB:
performs TraceBack
and assembles the
optimal alignment

GenASM: Hardware Design



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

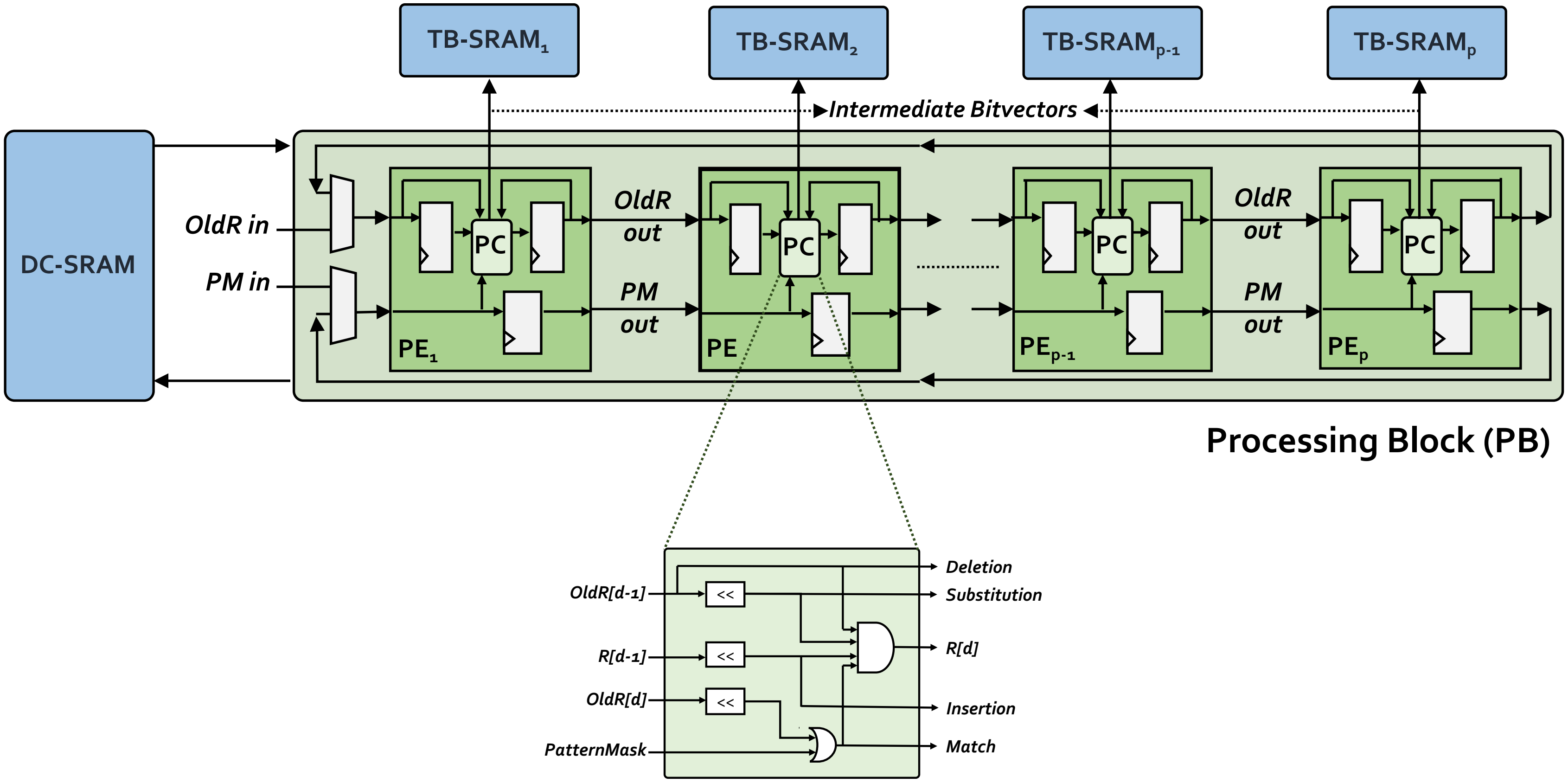
→ Match the rate of computation with memory capacity and bandwidth

→ Achieve high performance and power efficiency

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

GenASM-DC: Hardware Design

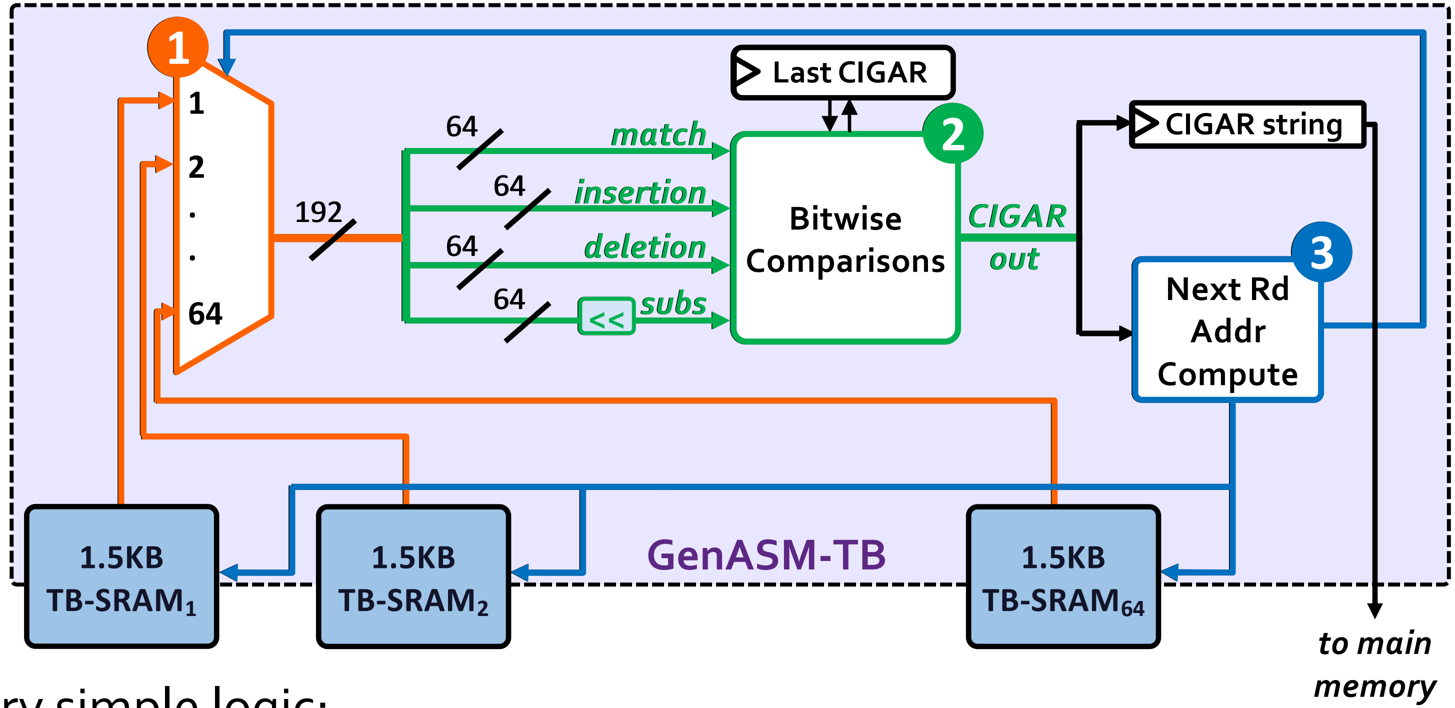
- **Linear cyclic systolic array** based accelerator
  - Designed to maximize parallelism and minimize memory bandwidth and memory footprint



Processing Block (PB)

Processing Core (PC)

GenASM-TB: Hardware Design



- Very simple logic:

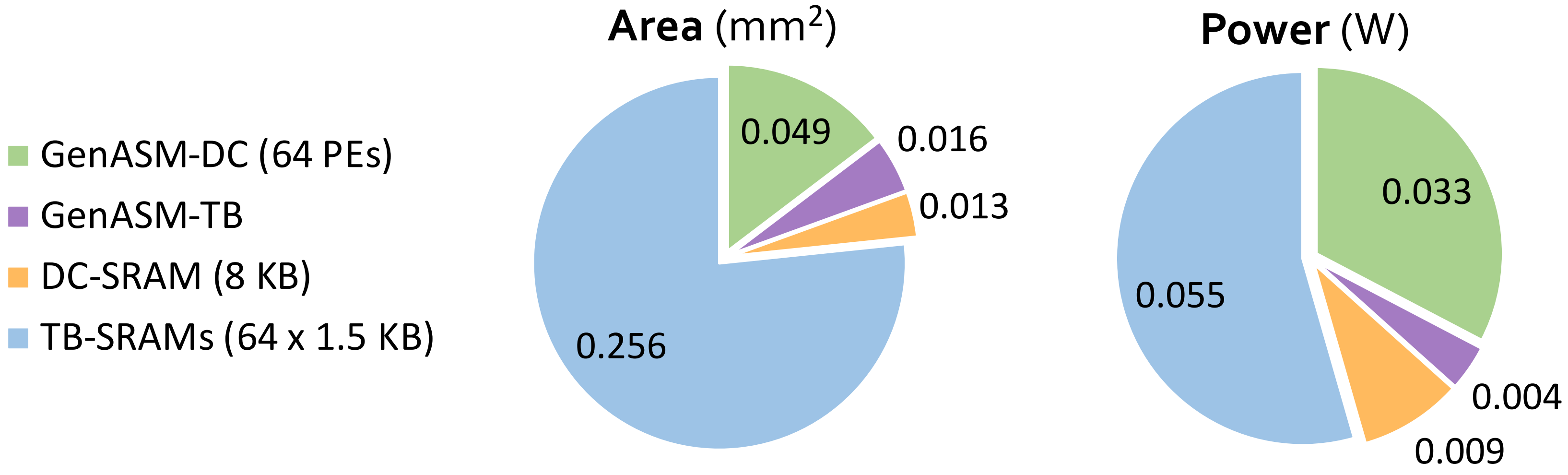
  1 Reads the bitvectors from one of the TB-SRAMs using the computed address

  2 Performs the required bitwise comparisons to find the traceback output for the current position

  3 Computes the next TB-SRAM address to read the new set of bitvectors

Key Results – Area and Power

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



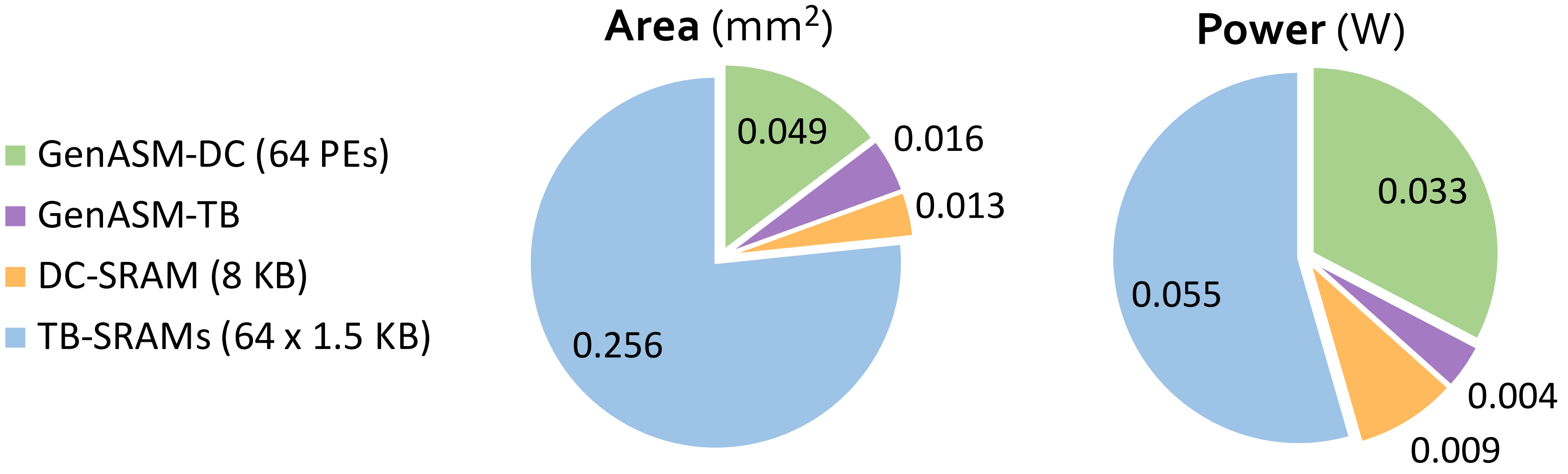
Area (mm^2)

Power (W)

Total (1 vault):	0.334 mm^2	0.101 W
Total (32 vaults):	10.69 mm^2	3.23 W
% of a Xeon CPU core:	**1%**	**1%**

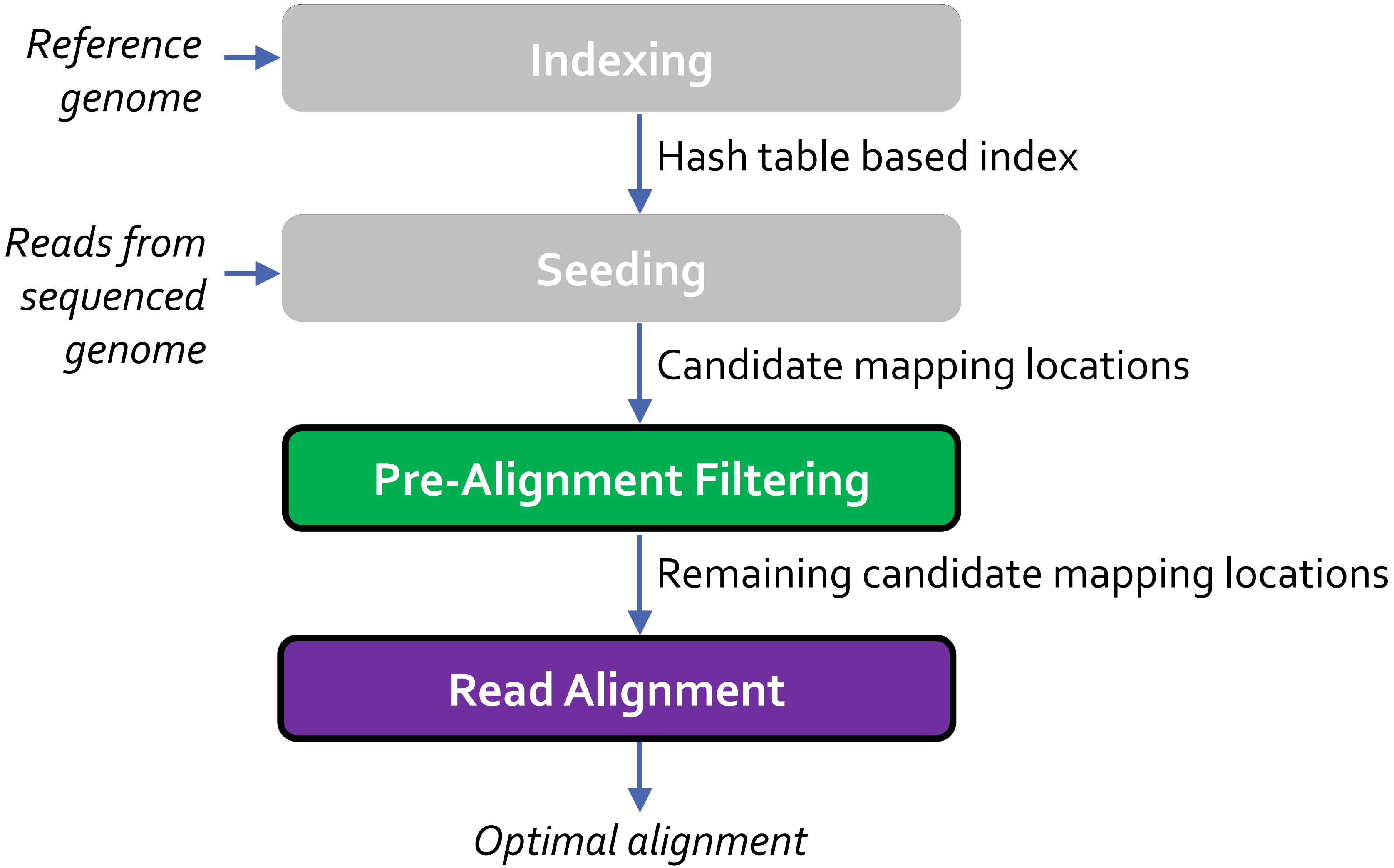
Key Results – Area and Power

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



GenASM has low area and power overheads

Use Cases of GenASM



Use Cases of GenASM (cont'd.)

(1) Read Alignment Step of Read Mapping

- ○ Find the optimal alignment of how reads map to candidate reference regions

(2) Pre-Alignment Filtering for Short Reads

- ○ Quickly identify and filter out the unlikely candidate reference regions for each read

(3) Edit Distance Calculation

- ○ Measure the similarity or distance between two sequences

❑ We also discuss other possible use cases of GenASM in our paper:

- ○ Read-to-read overlap finding, hash-table based indexing, whole genome alignment, generic text search

Key Results

(1) Read Alignment

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

(2) Pre-Alignment Filtering

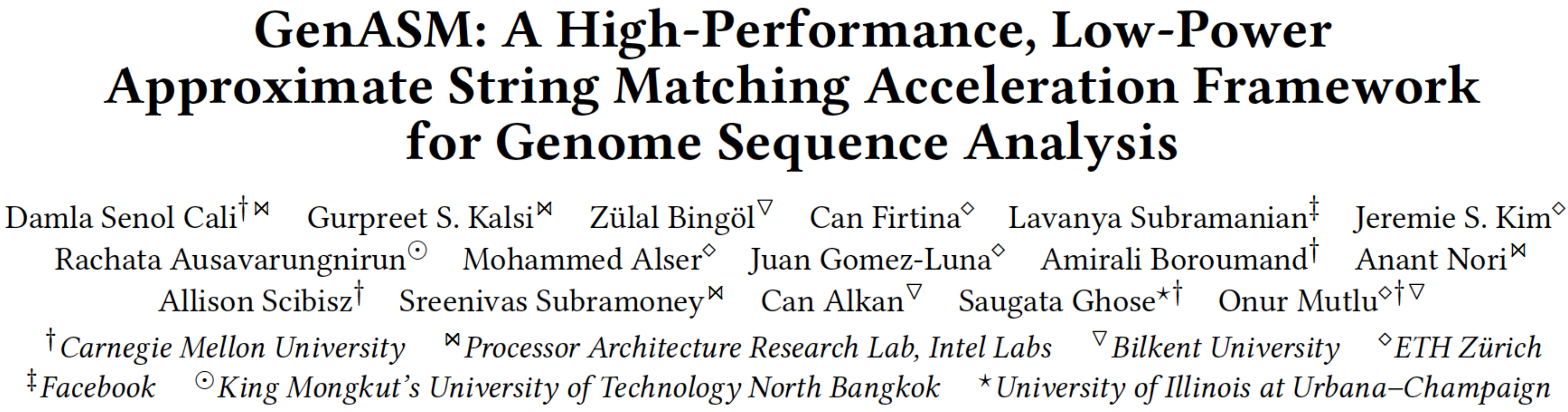
- ❑ **3.7×** speedup, **1.7×** less power than **Shouji** (state-of-the-art **HW**)

(3) Edit Distance Calculation

- ❑ **22–12501×** speedup, **548–582×** less power than **Edlib** (state-of-the-art **SW**)
- ❑ **9.3–400×** speedup, **67×** less power than **ASAP** (state-of-the-art **HW**)

More on GenASM Framework [MICRO 2020]

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,
"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"
Proceedings of the 53rd International Symposium on Microarchitecture (***MICRO***), Virtual, October 2020.
[Lighting Talk Video (1.5 minutes)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (18 minutes)]
[Slides (pptx) (pdf)]



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

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

[†]*Carnegie Mellon University* [⋈]*Processor Architecture Research Lab, Intel Labs* [▽]*Bilkent University* [◇]*ETH Zürich*
[‡]*Facebook* [⊙]*King Mongkut's University of Technology North Bangkok* [⋆]*University of Illinois at Urbana–Champaign*

Agenda

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

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

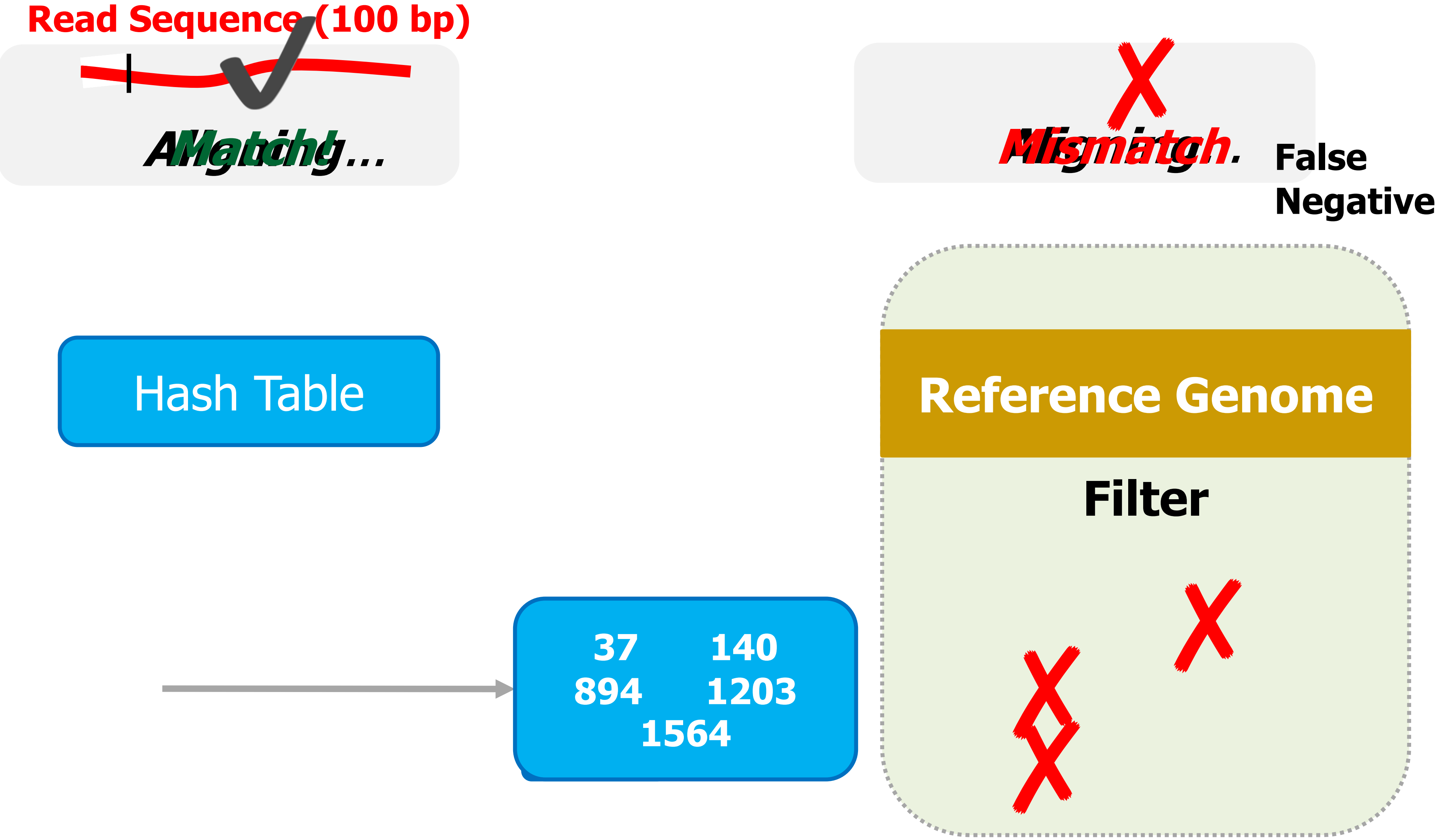
- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

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

Hash Tables in Read Mapping



Read Sequence (100 bp)

Aligning…

Match!

Aligning…

Mismatch.

False Negative

Hash Table

37 140
894 1203
1564

Reference Genome

Filter

SAFARI

Read Mapping & Filtering in Memory

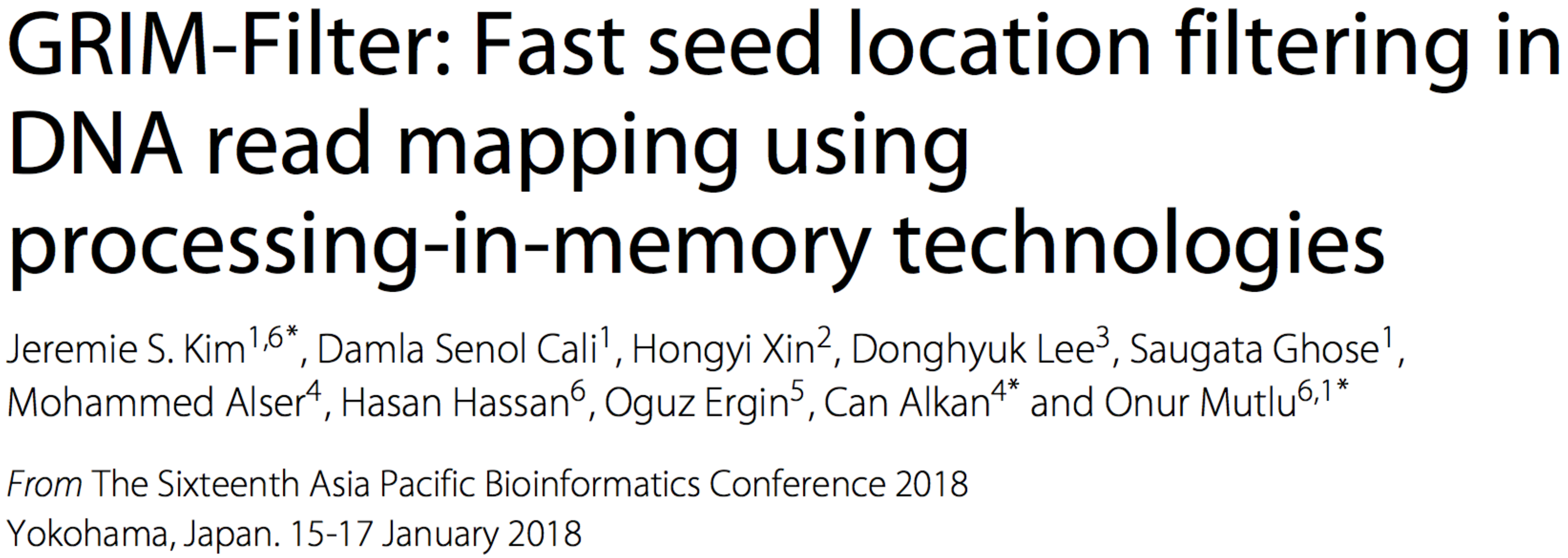
We need to design

mapping & filtering algorithms

that fit processing-in-memory

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]



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

Jeremie S. Kim[1,6*], Damla Senol Cali[1], Hongyi Xin[2], Donghyuk Lee[3], Saugata Ghose[1], Mohammed Alser[4], Hasan Hassan[6], Oguz Ergin[5], Can Alkan[4*] and Onur Mutlu[6,1*]

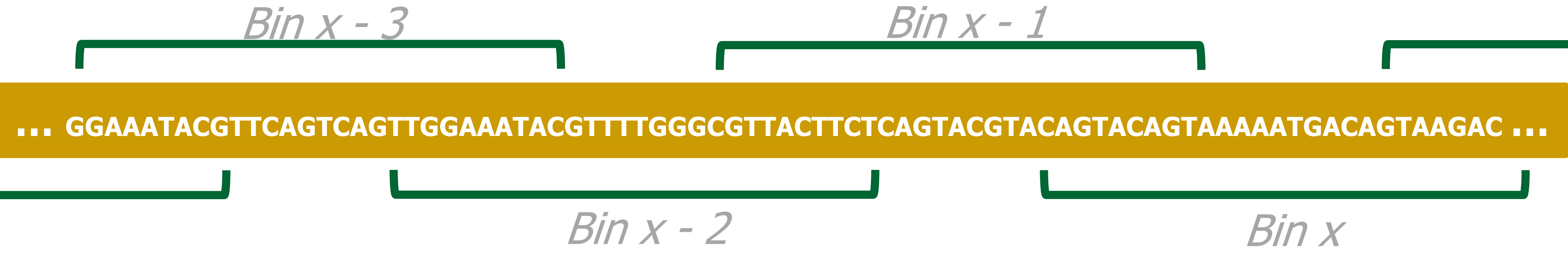
From The Sixteenth Asia Pacific Bioinformatics Conference 2018
Yokohama, Japan. 15-17 January 2018

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**).



Bin x - 3

Bin x - 1

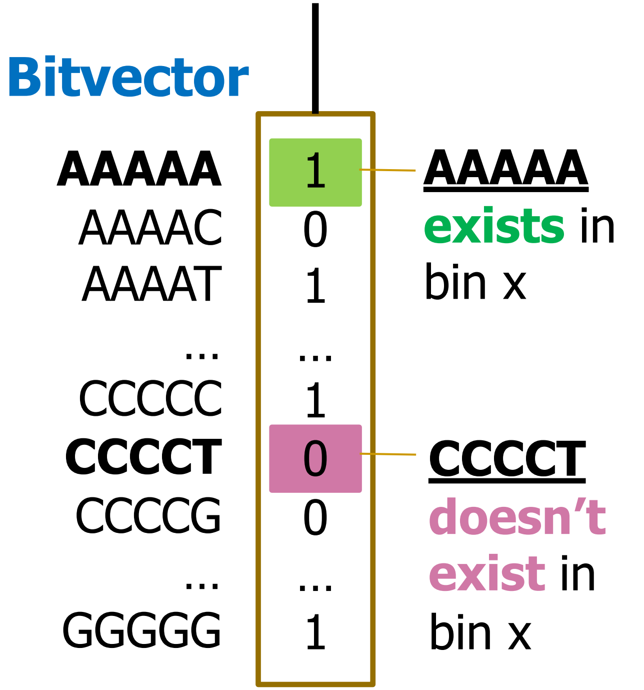
… GGAAATACGTTCAGTCAGTTGGAAATACGTTTTGGGCGTTACTTCTCAGTACGTACAGTACAGTAAAAATGACAGTAAGAC …

Bin x - 2

Bin x

- 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



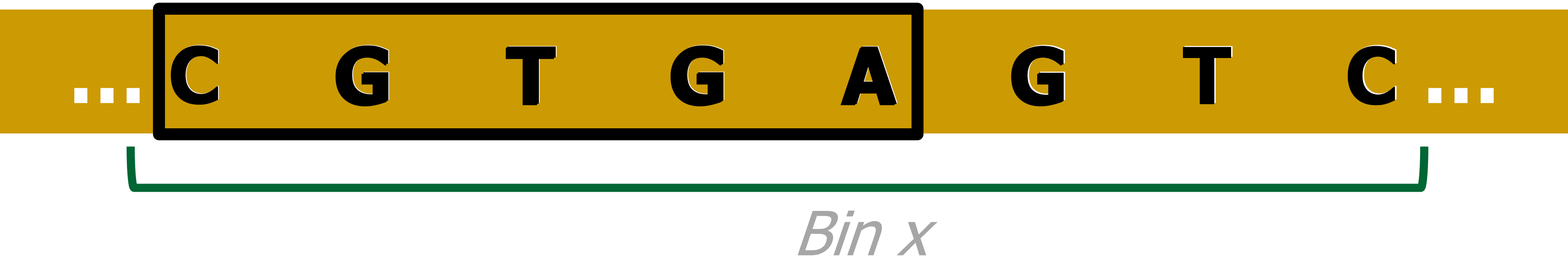
Bitvector

AAAAA	1
AAAAC	0
AAAAT	1
…	…
CCCCC	1
CCCCT	0
CCCCG	0
…	…
GGGGG	1

AAAAA **exists** in bin x

CCCCT **doesn't exist** in bin x

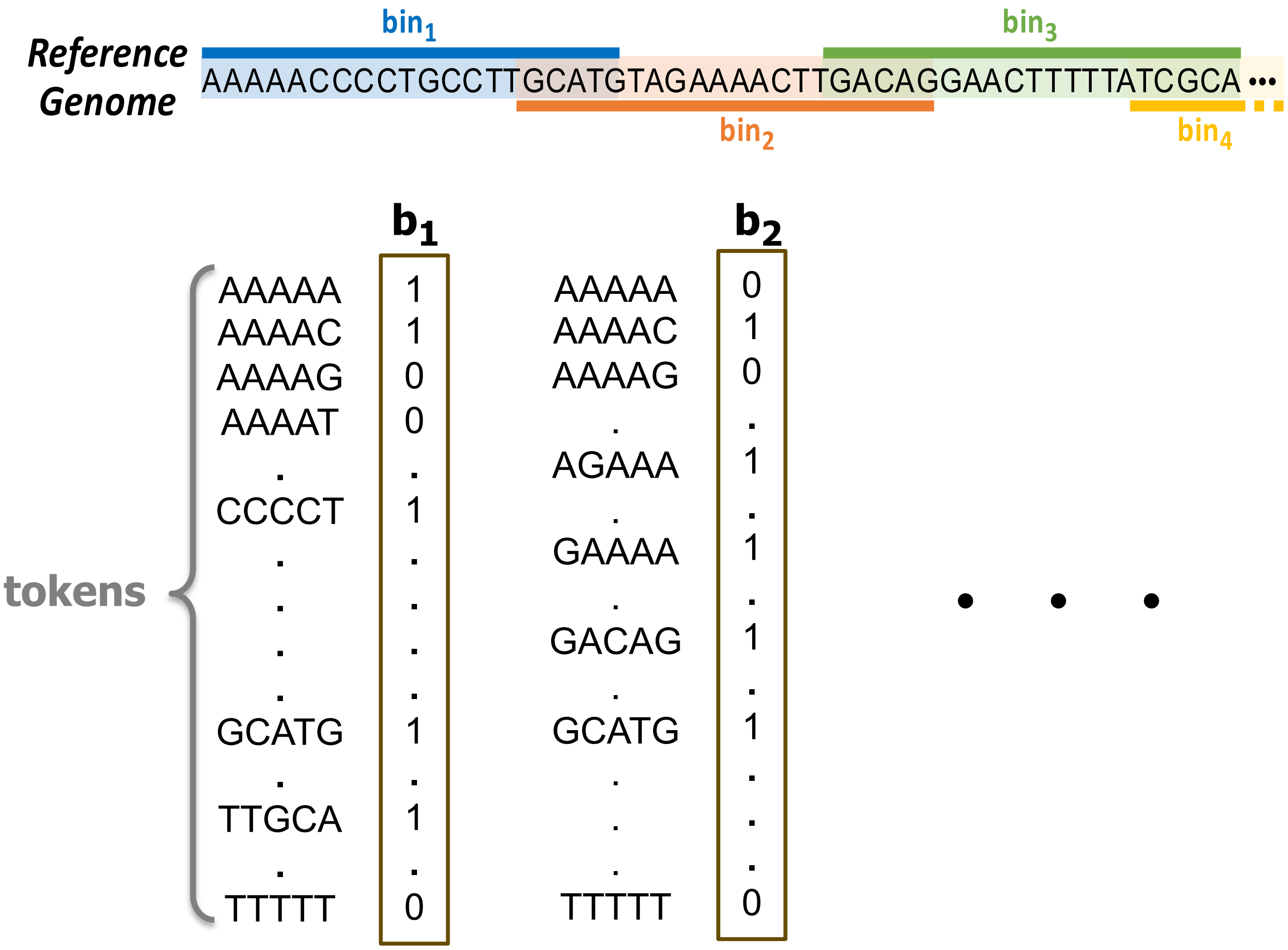
GRIM-Filter: Bitvectors



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.

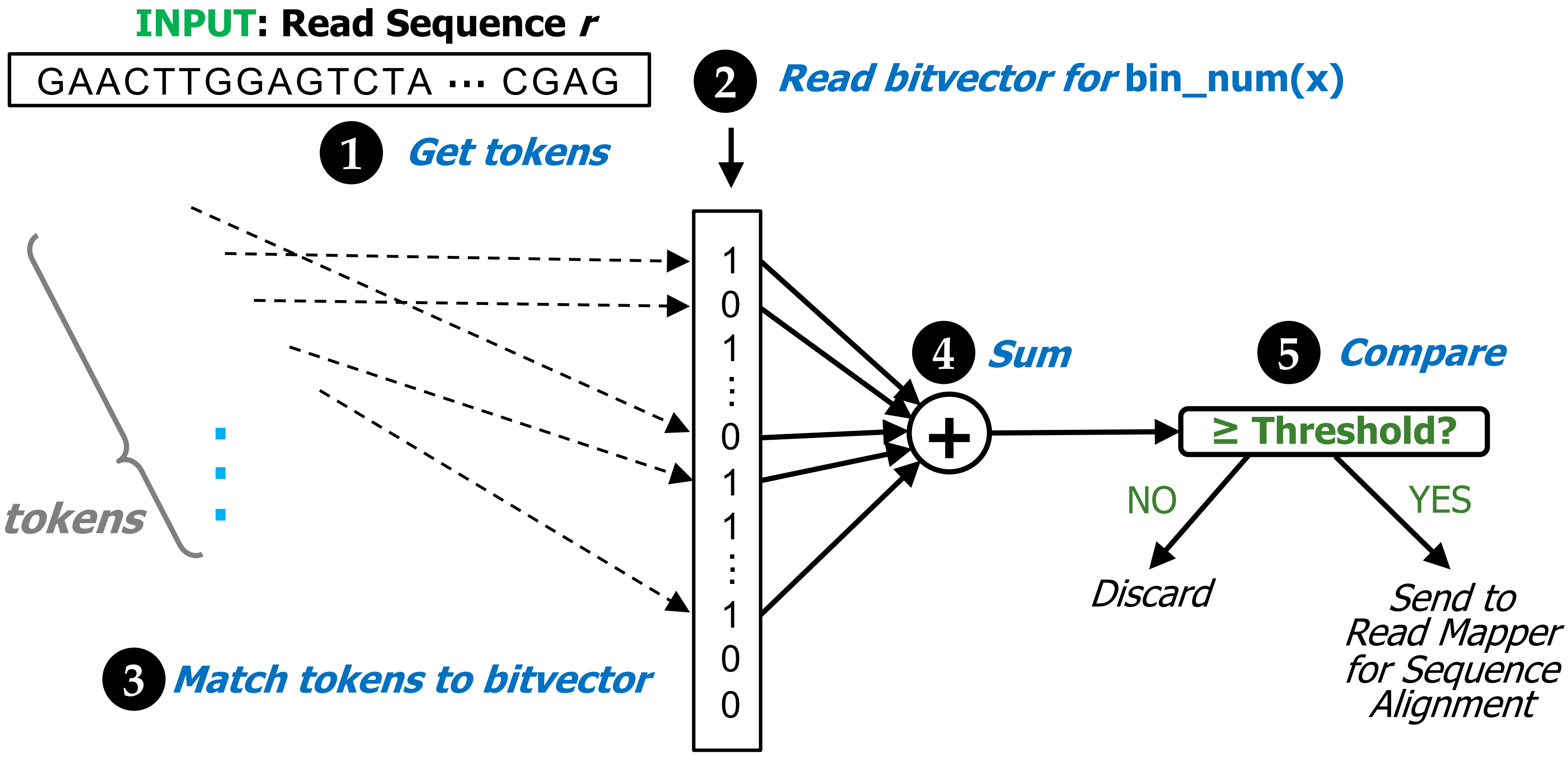
For **bin size** ~200, and **n** = 5, **memory footprint** ~3.8 GB

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

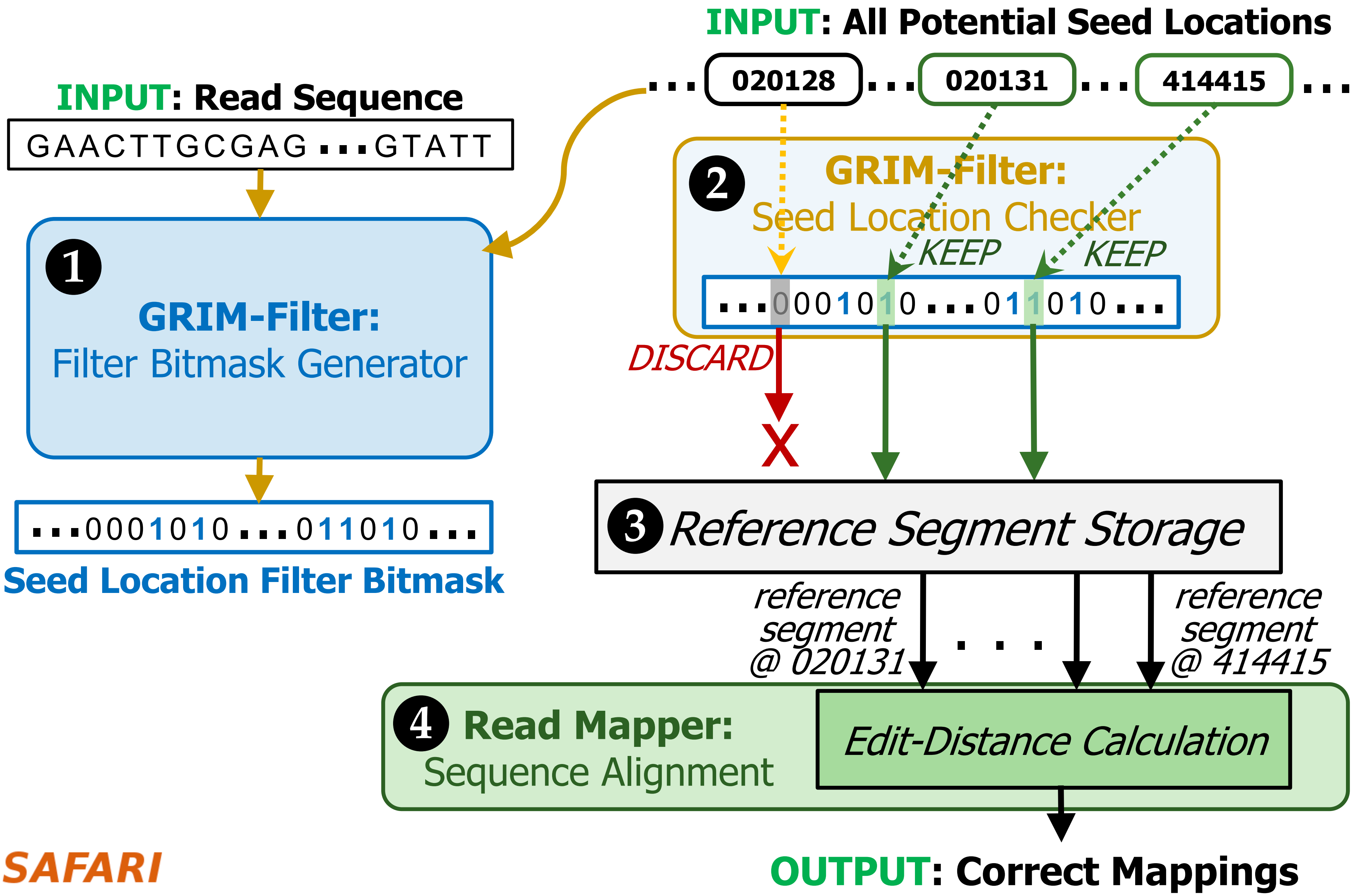


SAFARI

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



INPUT: All Potential Seed Locations

... 020128 ... 020131 ... 414415 ...

INPUT: Read Sequence

GAACTTGCGAG ... GTATT

1 GRIM-Filter: Filter Bitmask Generator

...0001010...011010...

Seed Location Filter Bitmask

2 GRIM-Filter: Seed Location Checker

KEEP KEEP

...0001010...011010...

DISCARD

X

3 *Reference Segment Storage*

reference segment @ 020131

. . .

reference segment @ 414415

4 Read Mapper: Sequence Alignment

Edit-Distance Calculation

OUTPUT: Correct Mappings

Key Properties of GRIM-Filter

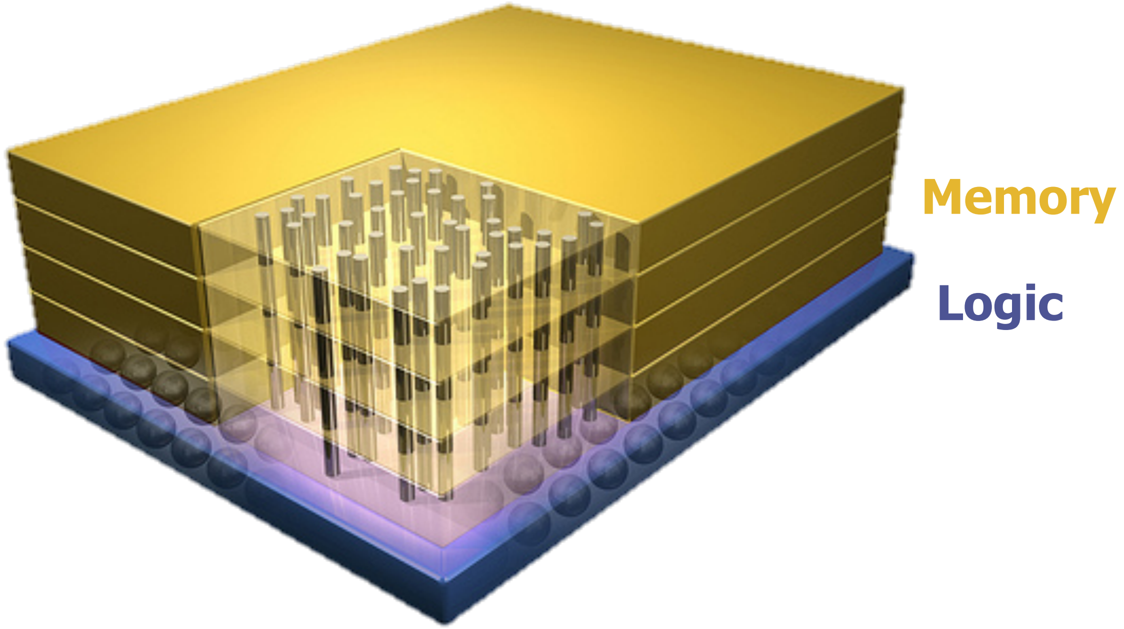
1. **Simple Operations**:
   - To check a given bin, find the **sum** of all bits corresponding to each token in the read
   - **Compare** against threshold to determine whether to align

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

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

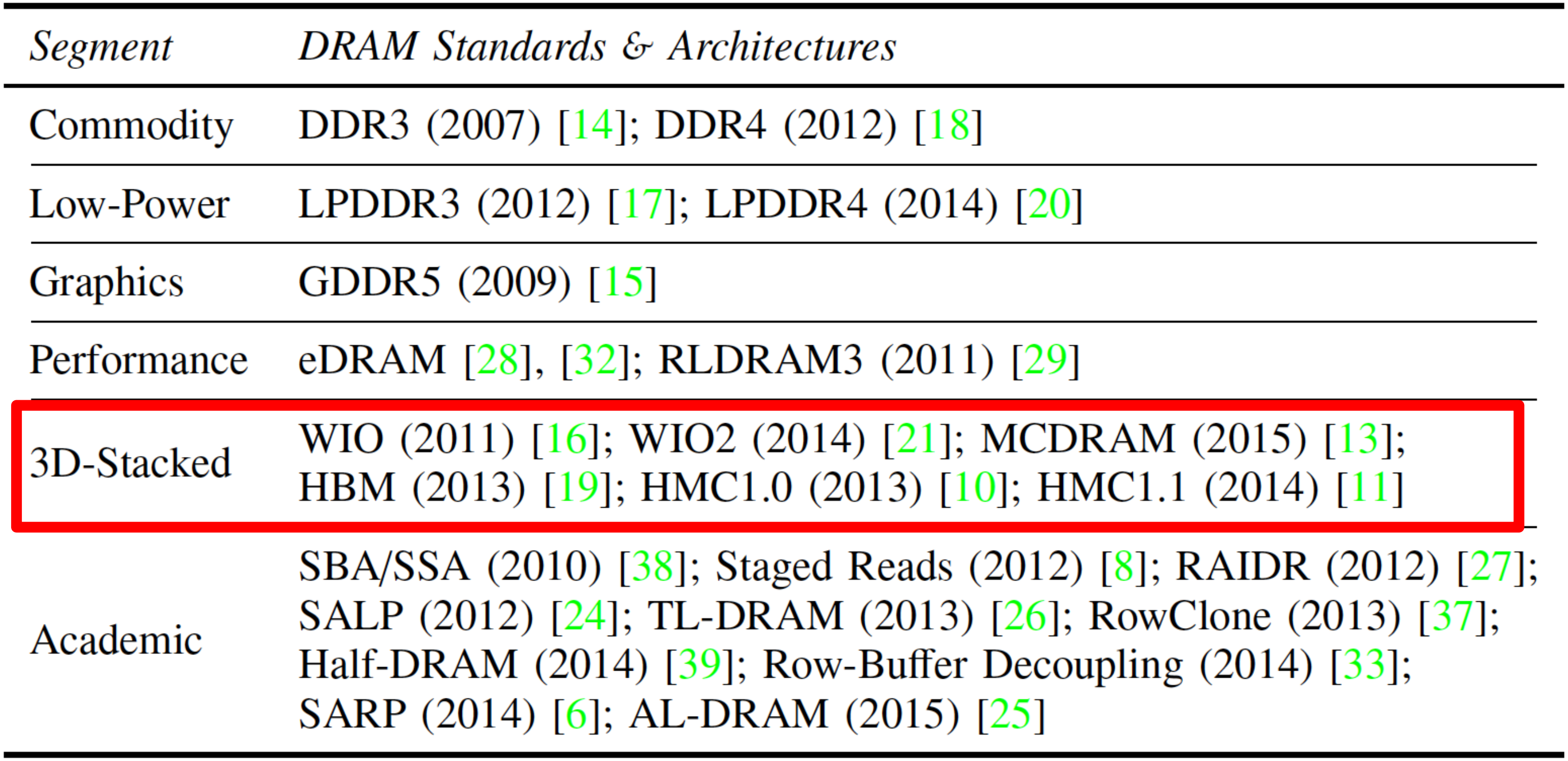
These properties together make GRIM-Filter a good algorithm to be run in 3D-Stacked DRAM

Opportunity: 3D-Stacked Logic+Memory



Other "True 3D" technologies under development

DRAM Landscape (circa 2015)

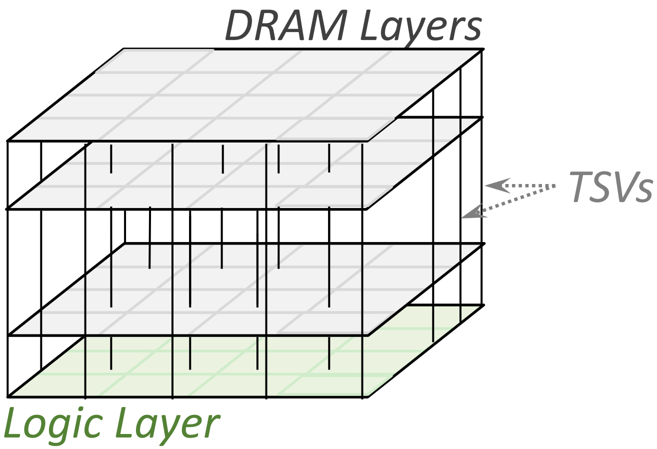


Segment	*DRAM Standards & Architectures*
Commodity	DDR3 (2007) [14]; DDR4 (2012) [18]
Low-Power	LPDDR3 (2012) [17]; LPDDR4 (2014) [20]
Graphics	GDDR5 (2009) [15]
Performance	eDRAM [28], [32]; RLDRAM3 (2011) [29]
3D-Stacked	WIO (2011) [16]; WIO2 (2014) [21]; MCDRAM (2015) [13]; HBM (2013) [19]; HMC1.0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

Table 1. Landscape of DRAM-based memory

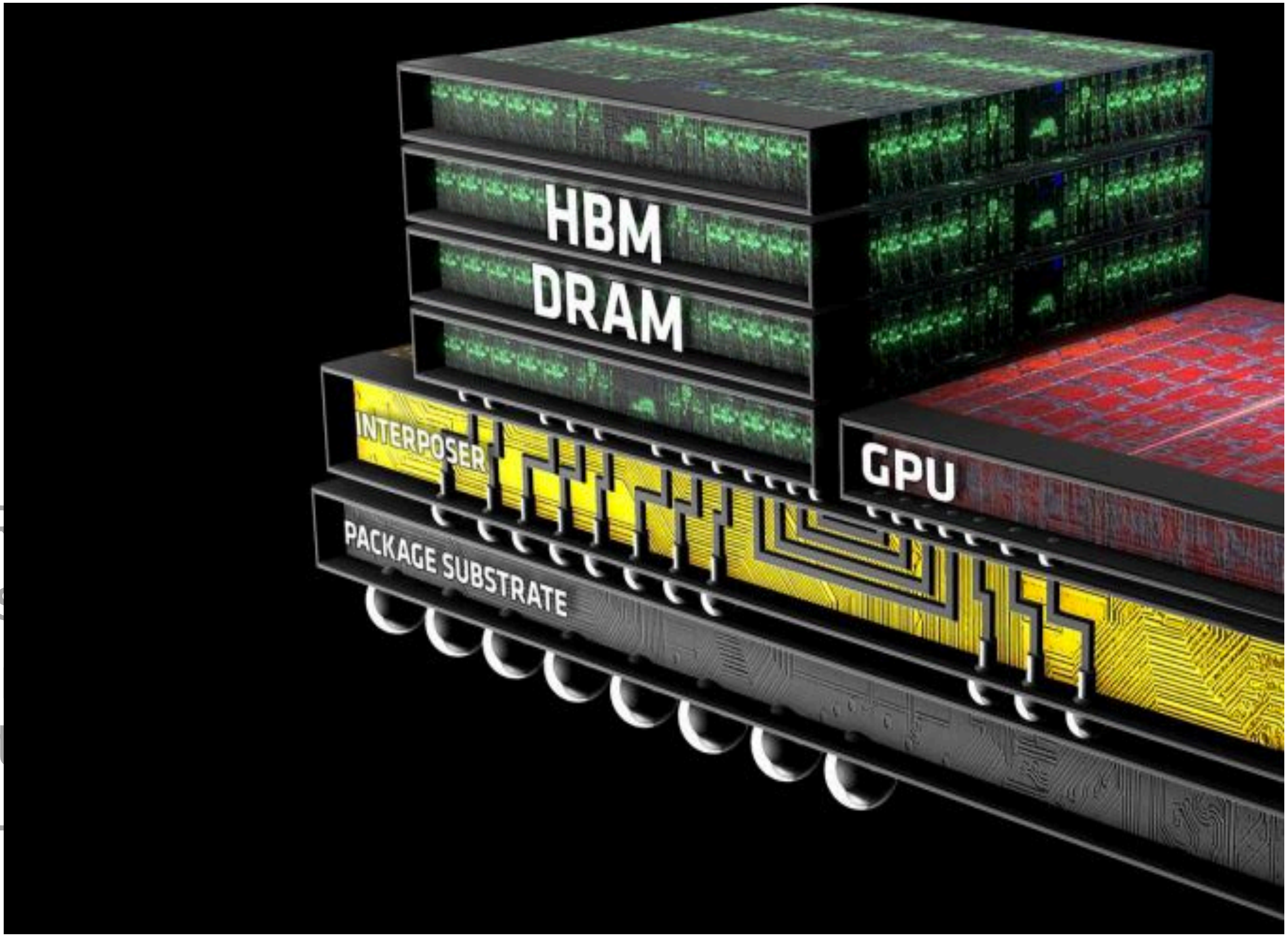
Kim+, "Ramulator: A Flexible and Extensible DRAM Simulator", IEEE CAL 2015.

3D-Stacked Memory



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

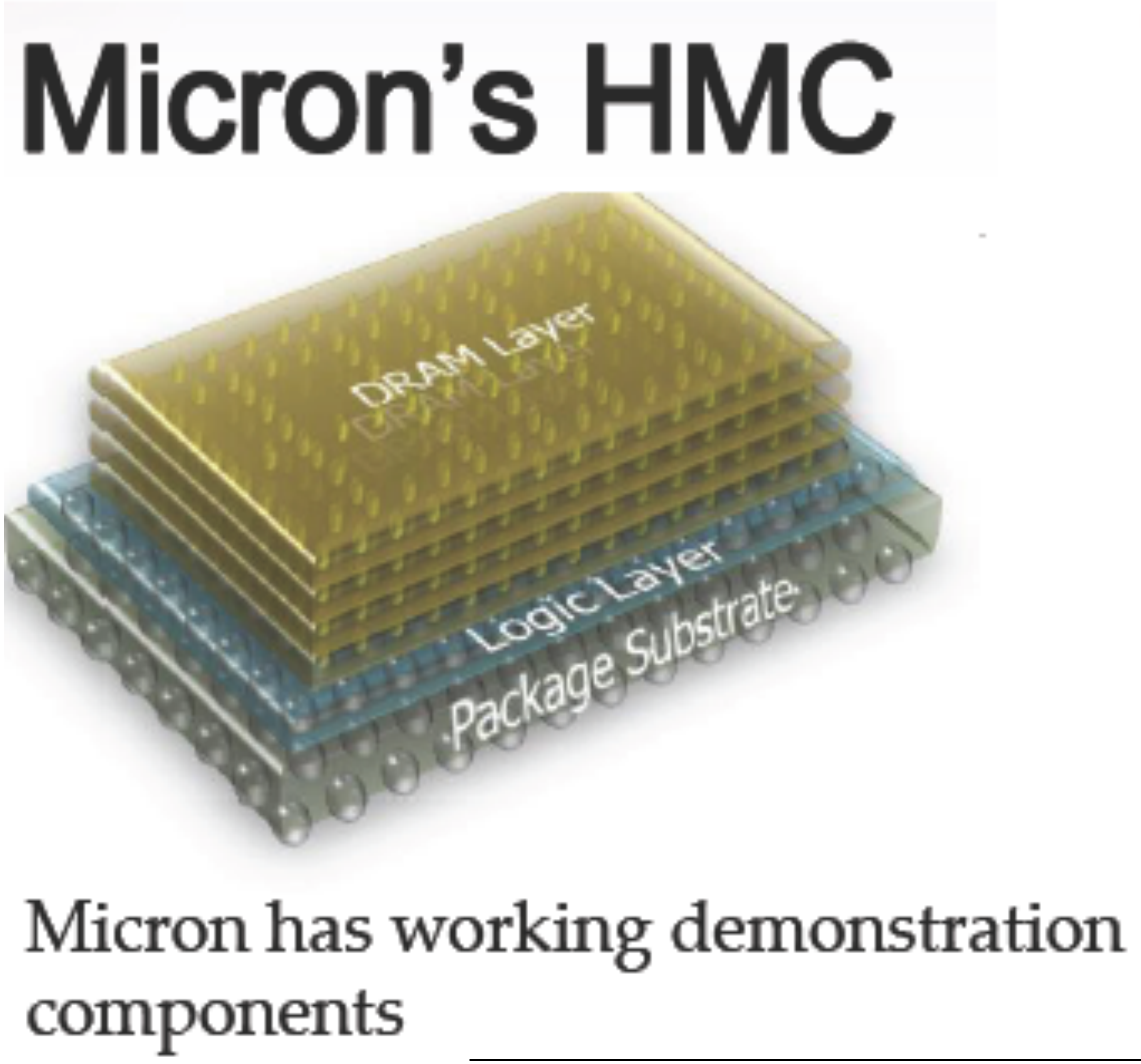
3D-Stacked Memory

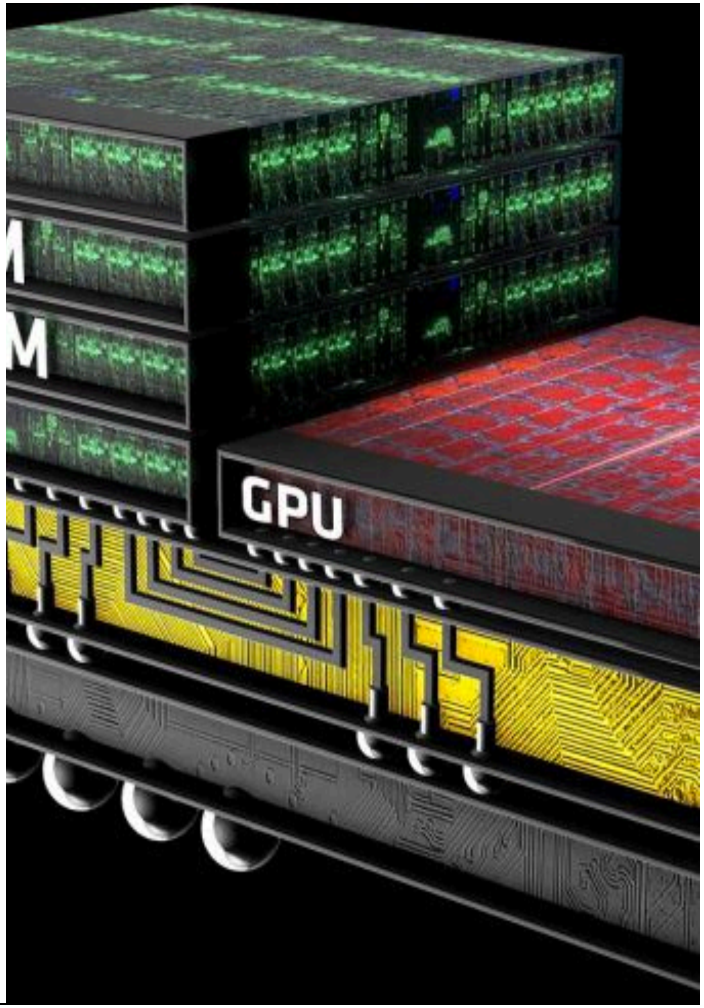


- 3D-Stacked DR **bandwidth** as
  - Logic Layer e computation t
  - Embed GRIM- appropriately

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

3D-Stacked Memory

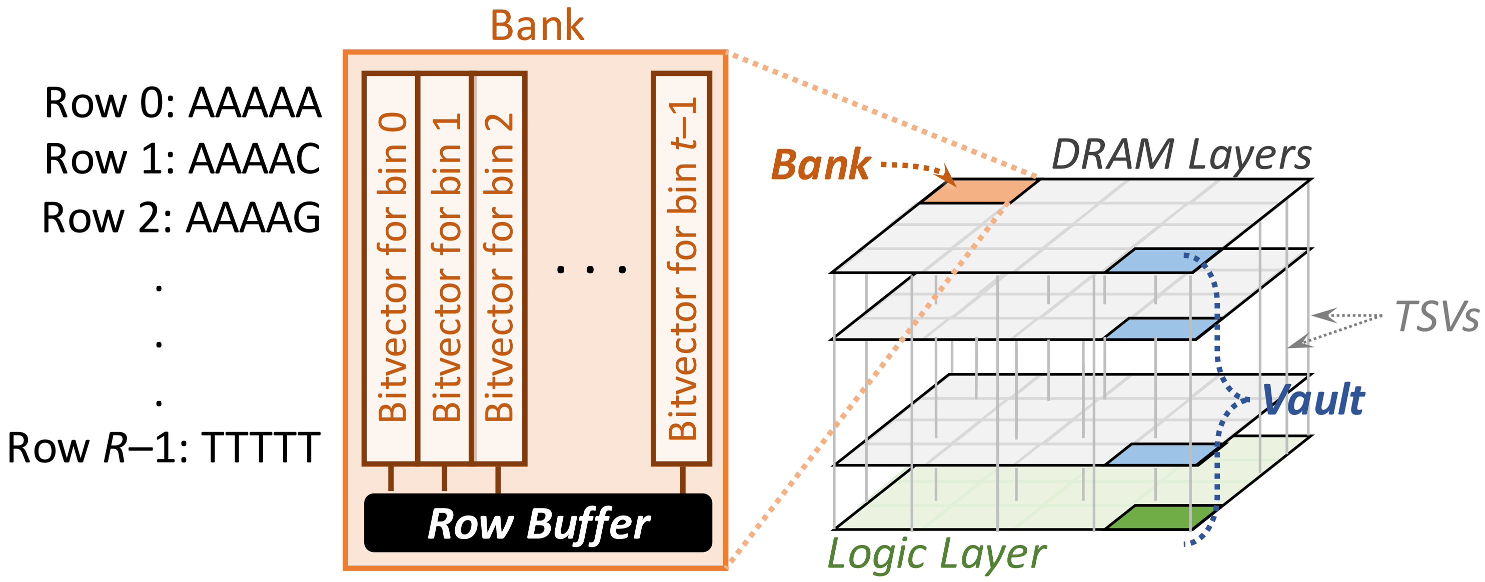




http://images.anandtech.com/doci/9266/HBMCar_678x452.jpg

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

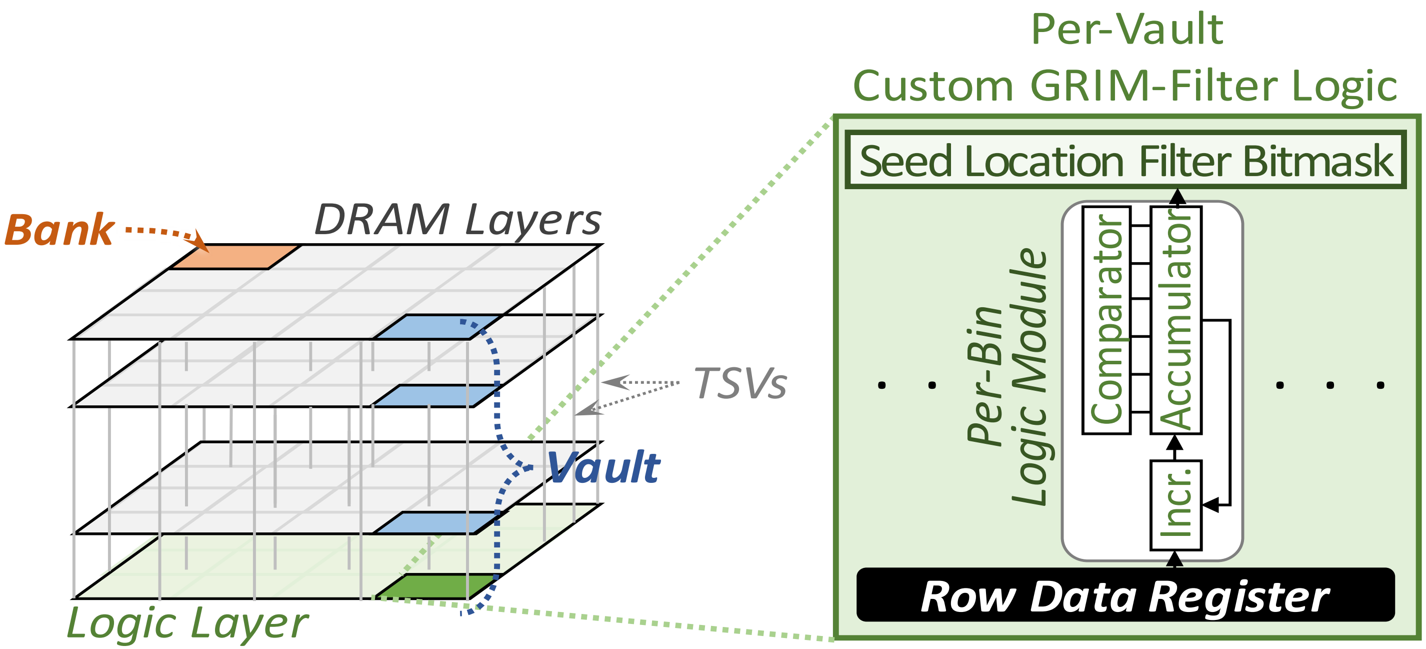
GRIM-Filter in 3D-Stacked DRAM



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

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

GRIM-Filter in 3D-Stacked DRAM



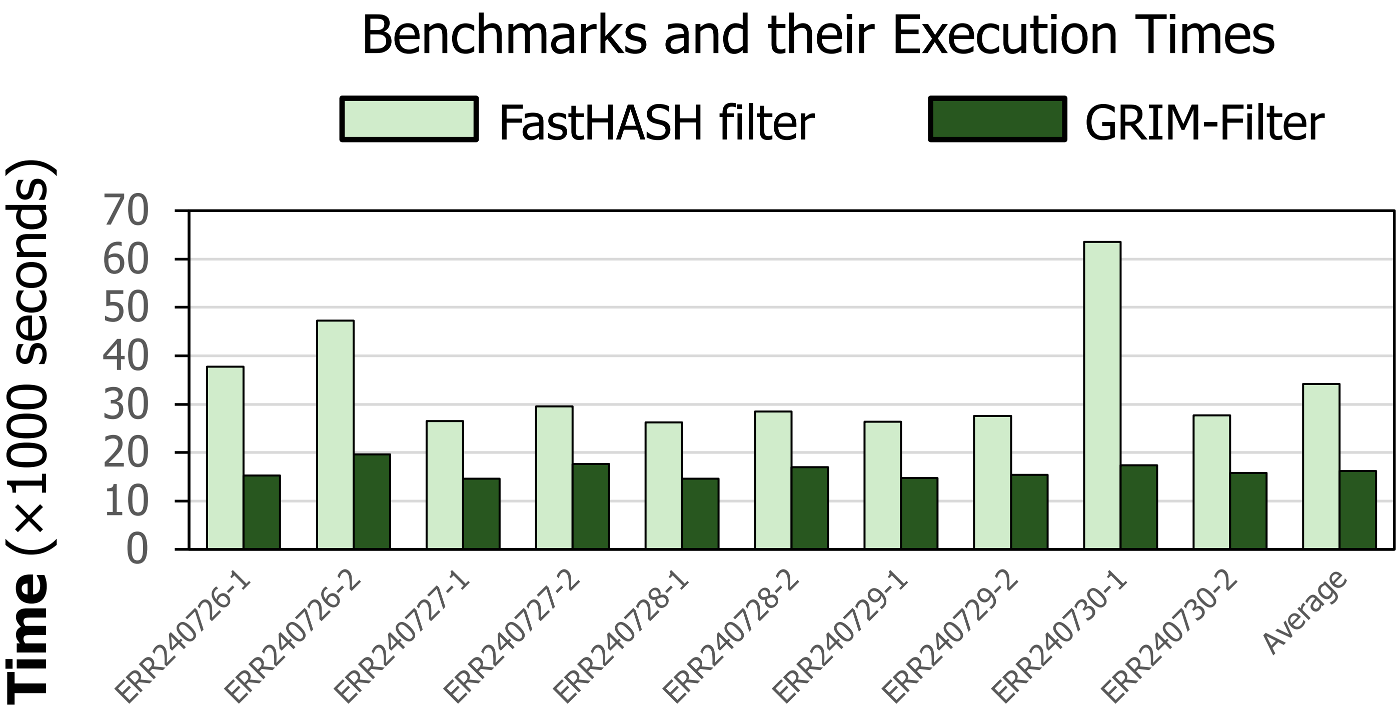
- Customized logic for accumulation and comparison per genome segment
  - Low area overhead, simple implementation
  - For HBM2, we use 4096 incrementer LUTs, 7-bit counters, and comparators in logic layer

Details are in [Kim+, BMC Genomics 2018]

Methodology

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

GRIM-Filter Performance



Sequence Alignment Error Tolerance (*e*)

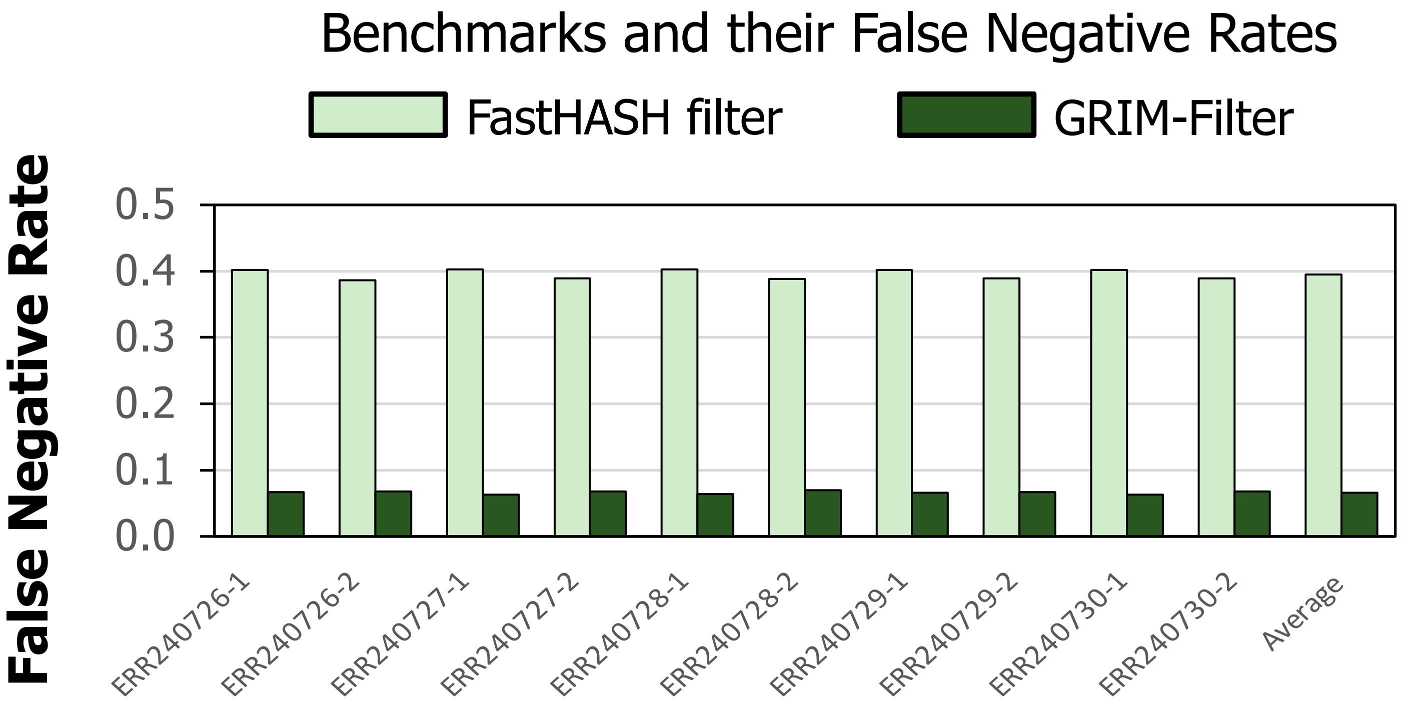
***e* = 0.05**

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



Sequence Alignment Error Tolerance (e)

e **= 0.05**

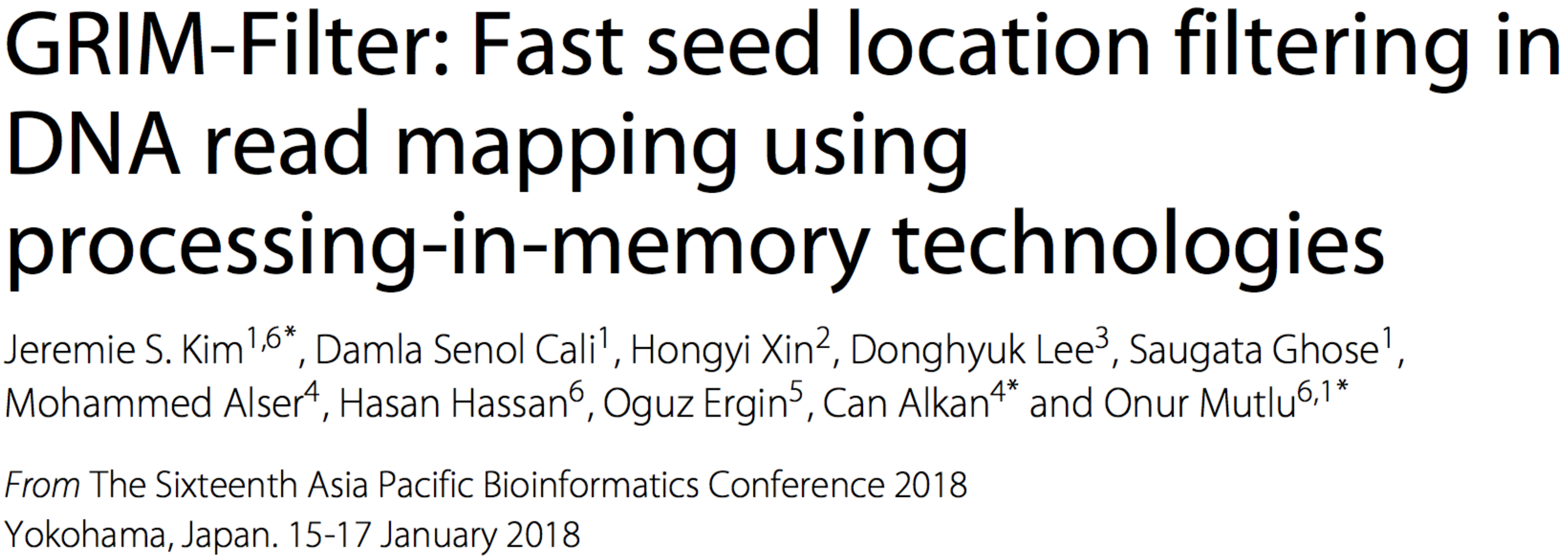
5.6x-6.4x False Negative reduction across real data sets

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]



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

Jeremie S. Kim[1,6*], Damla Senol Cali[1], Hongyi Xin[2], Donghyuk Lee[3], Saugata Ghose[1], Mohammed Alser[4], Hasan Hassan[6], Oguz Ergin[5], Can Alkan[4*] and Onur Mutlu[6,1*]

From The Sixteenth Asia Pacific Bioinformatics Conference 2018
Yokohama, Japan. 15-17 January 2018

Aside: In-Memory Graph Processing

- Large graphs are everywhere (circa 2015)

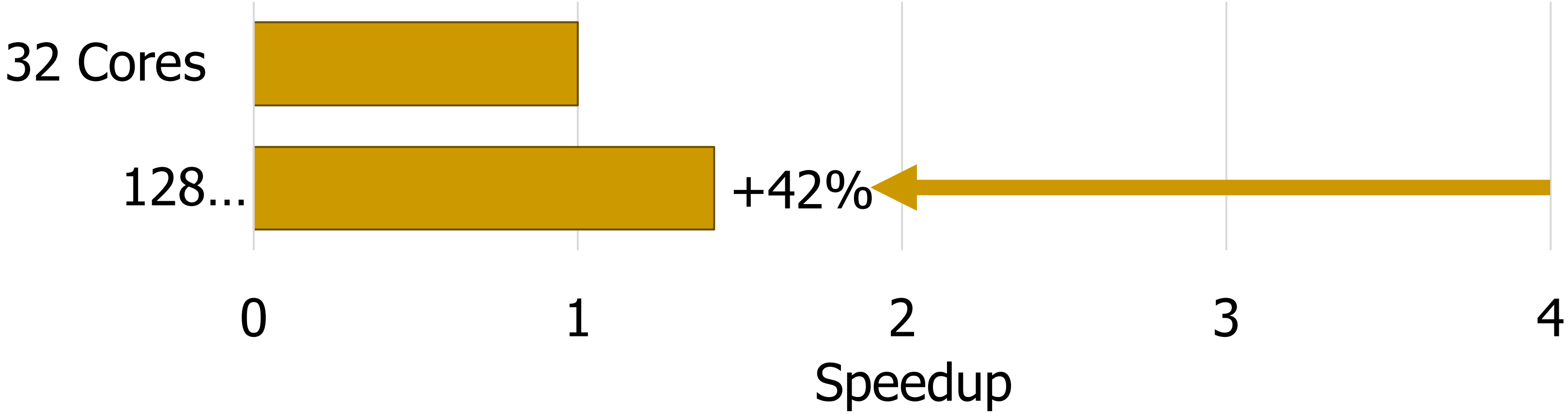
36 Million
Wikipedia Pages

1.4 Billion
Facebook Users

300 Million
Twitter Users

30 Billion
Instagram Photos

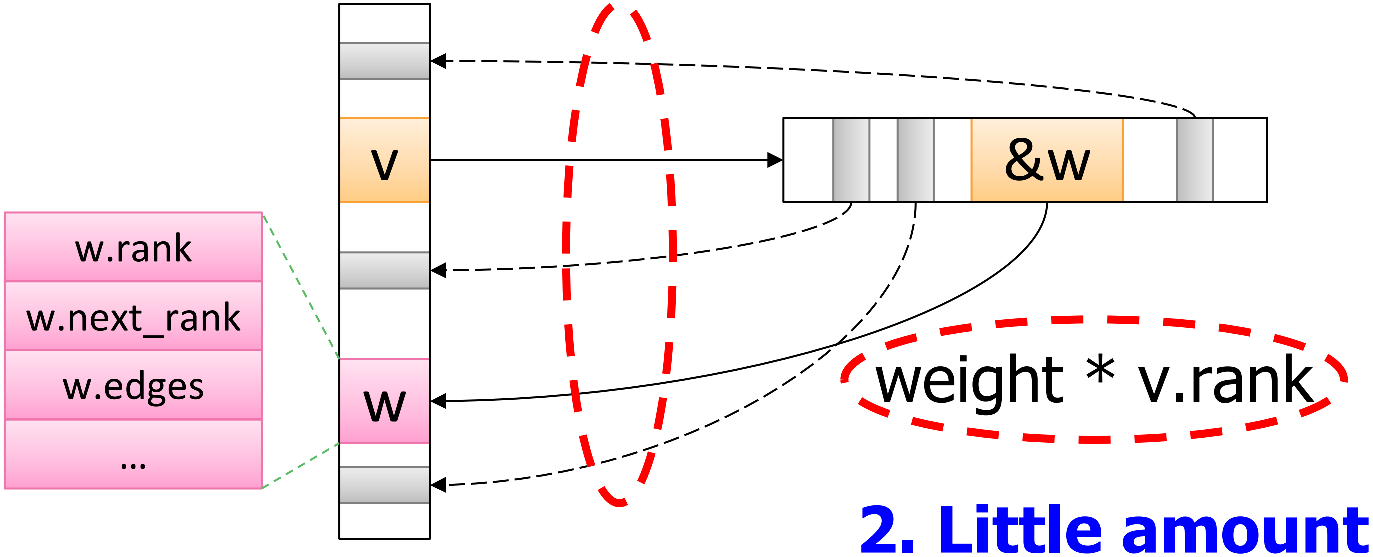
- Scalable large-scale graph processing is challenging



Key Bottlenecks in Graph Processing

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

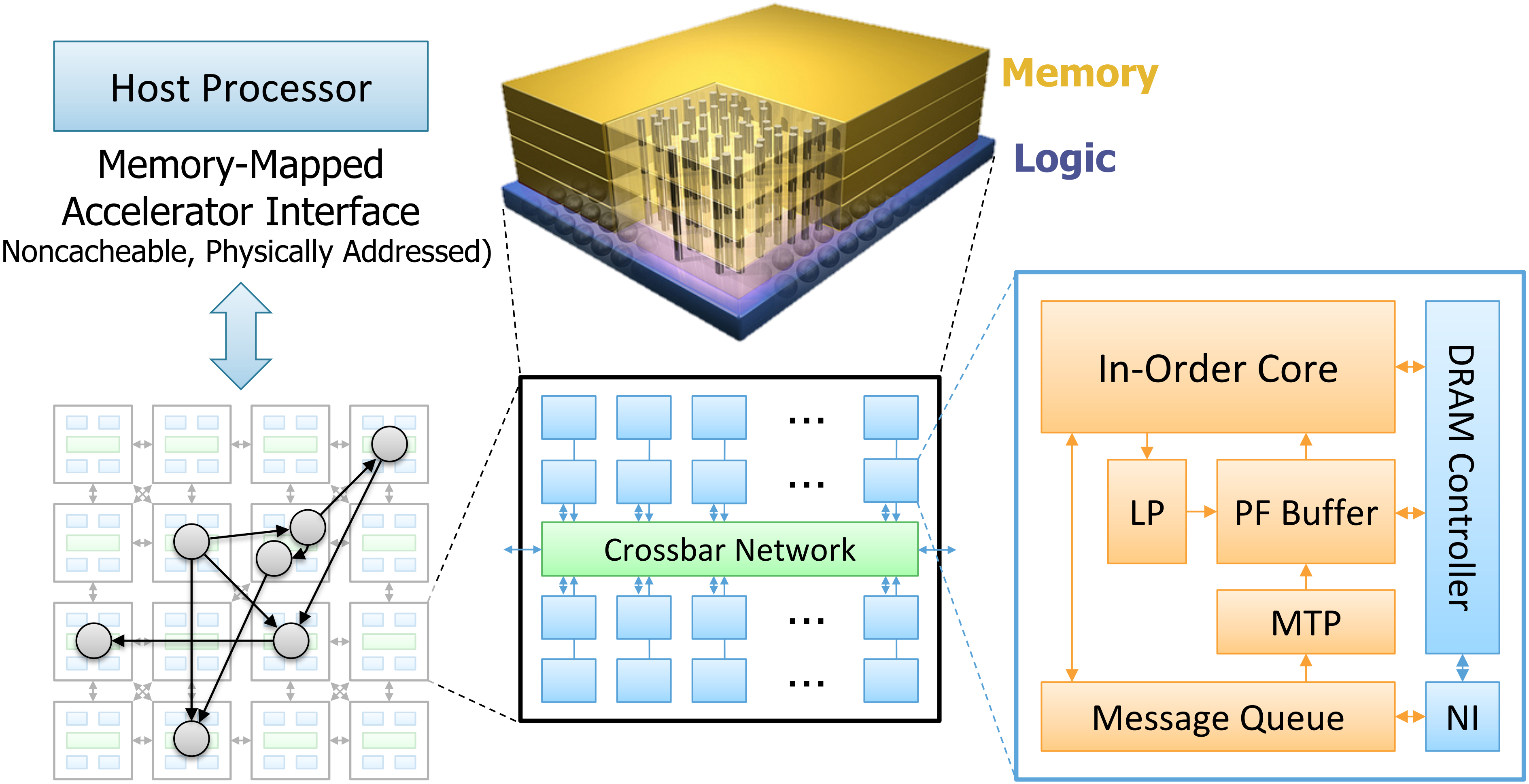
1. **Frequent random memory accesses**



2. **Little amount of computation**

Tesseract System for Graph Processing

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



Host Processor

Memory-Mapped Accelerator Interface (Noncacheable, Physically Addressed)

Memory

Logic

Crossbar Network

In-Order Core

LP

PF Buffer

MTP

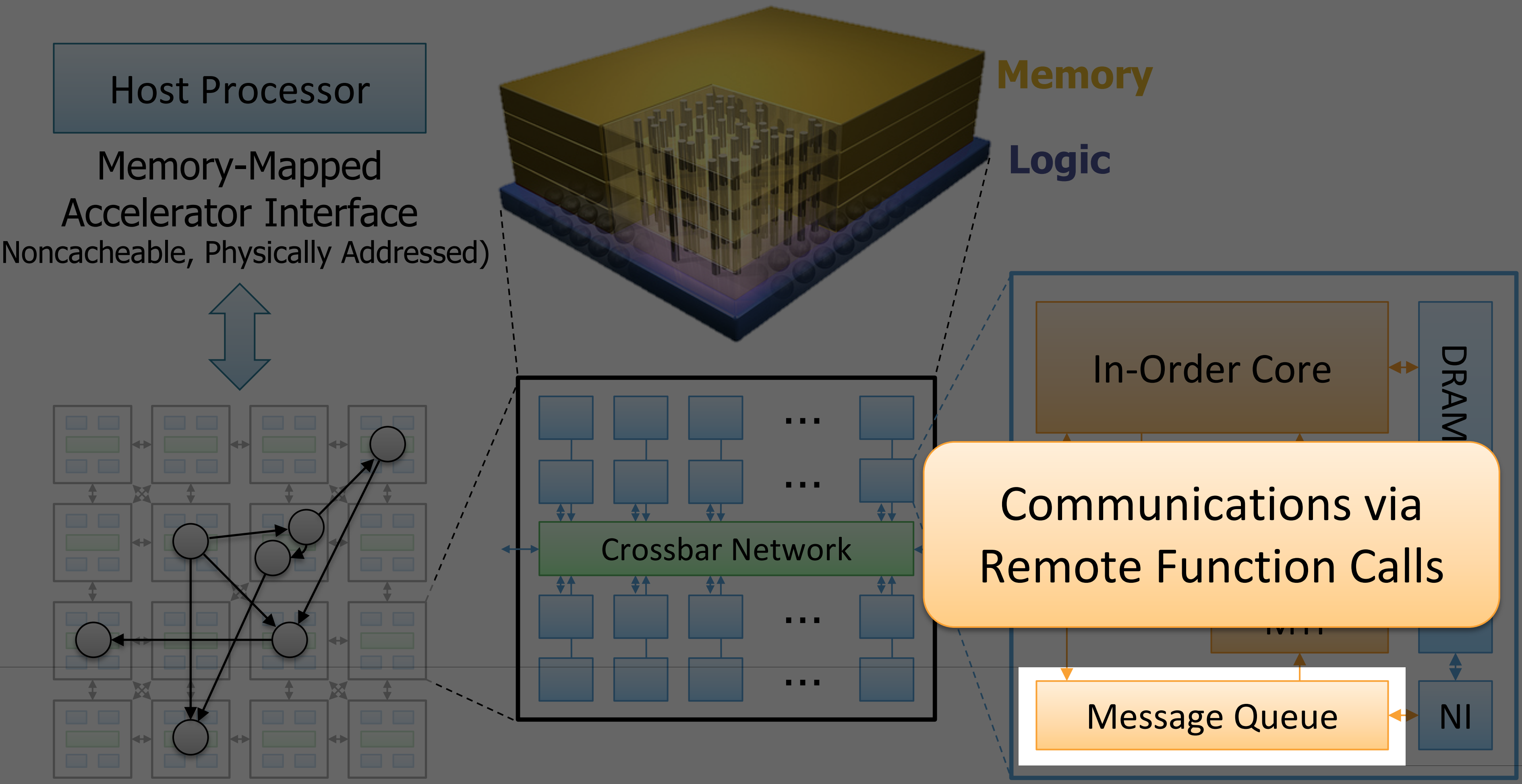
Message Queue

DRAM Controller

NI

Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

Tesseract System for Graph Processing



Host Processor

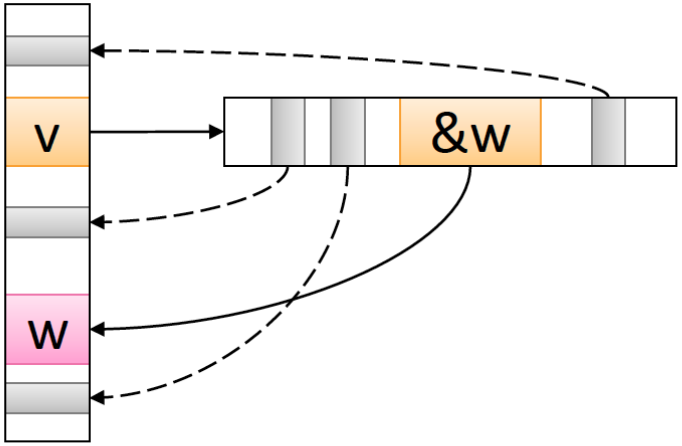
Memory-Mapped Accelerator Interface

Noncacheable, Physically Addressed)

Communications via Remote Function Calls

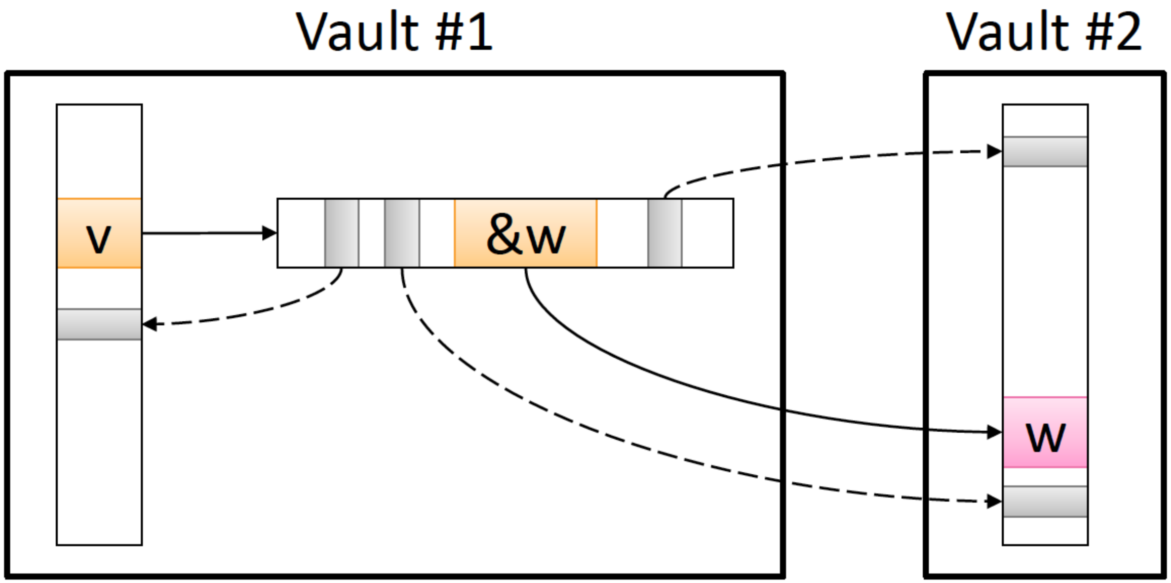
Communications In Tesseract (I)

```
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;
    }
}
```

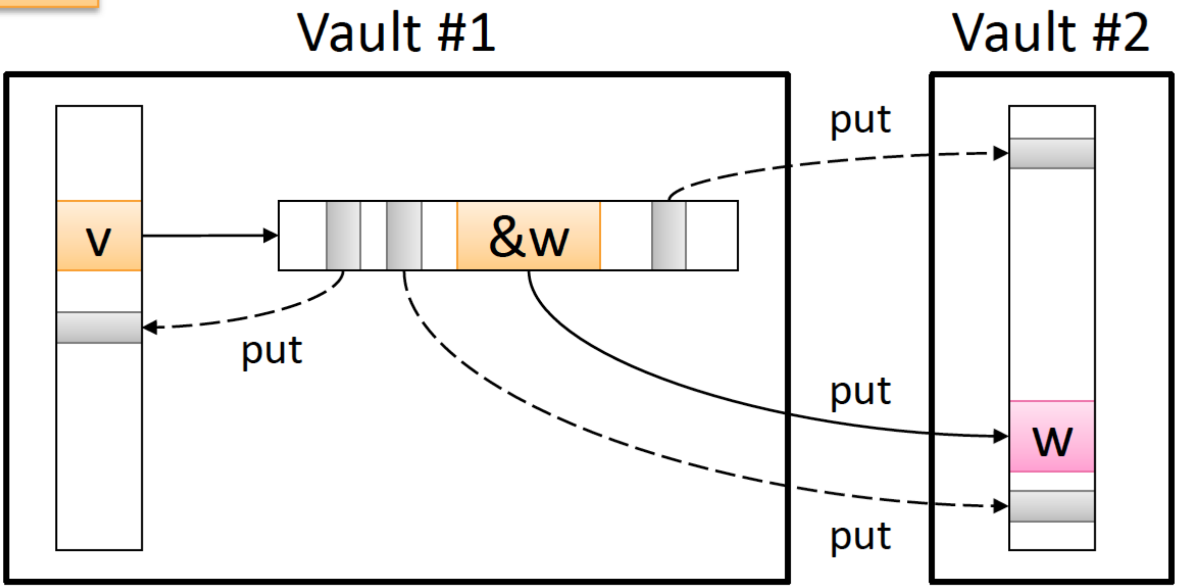


Communications In Tesseract (III)

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

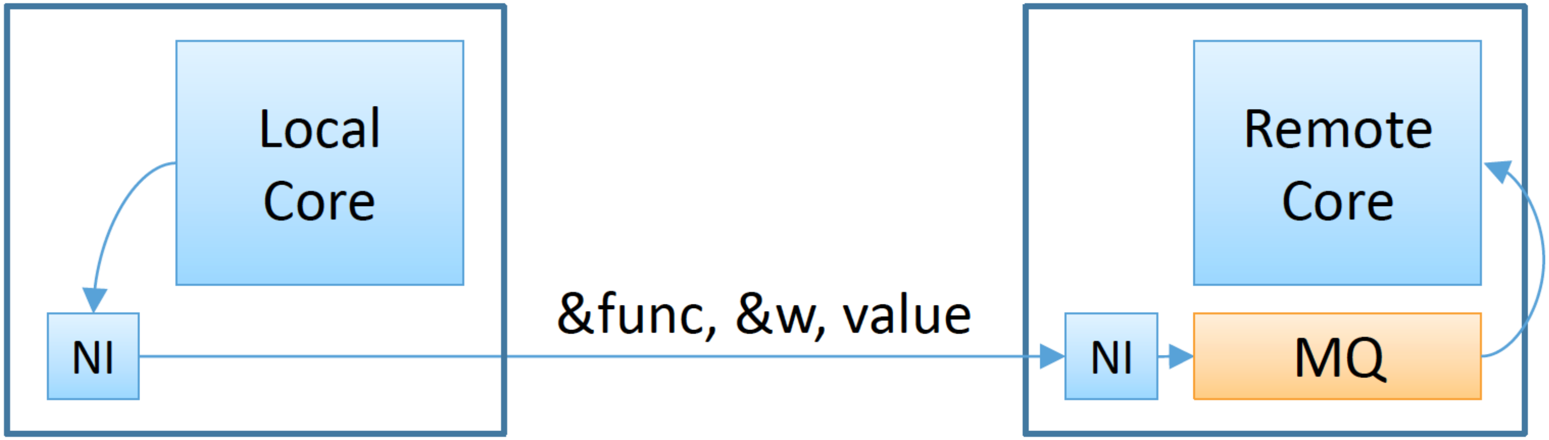
Non-blocking Remote Function Call

Can be **delayed**
until the nearest barrier



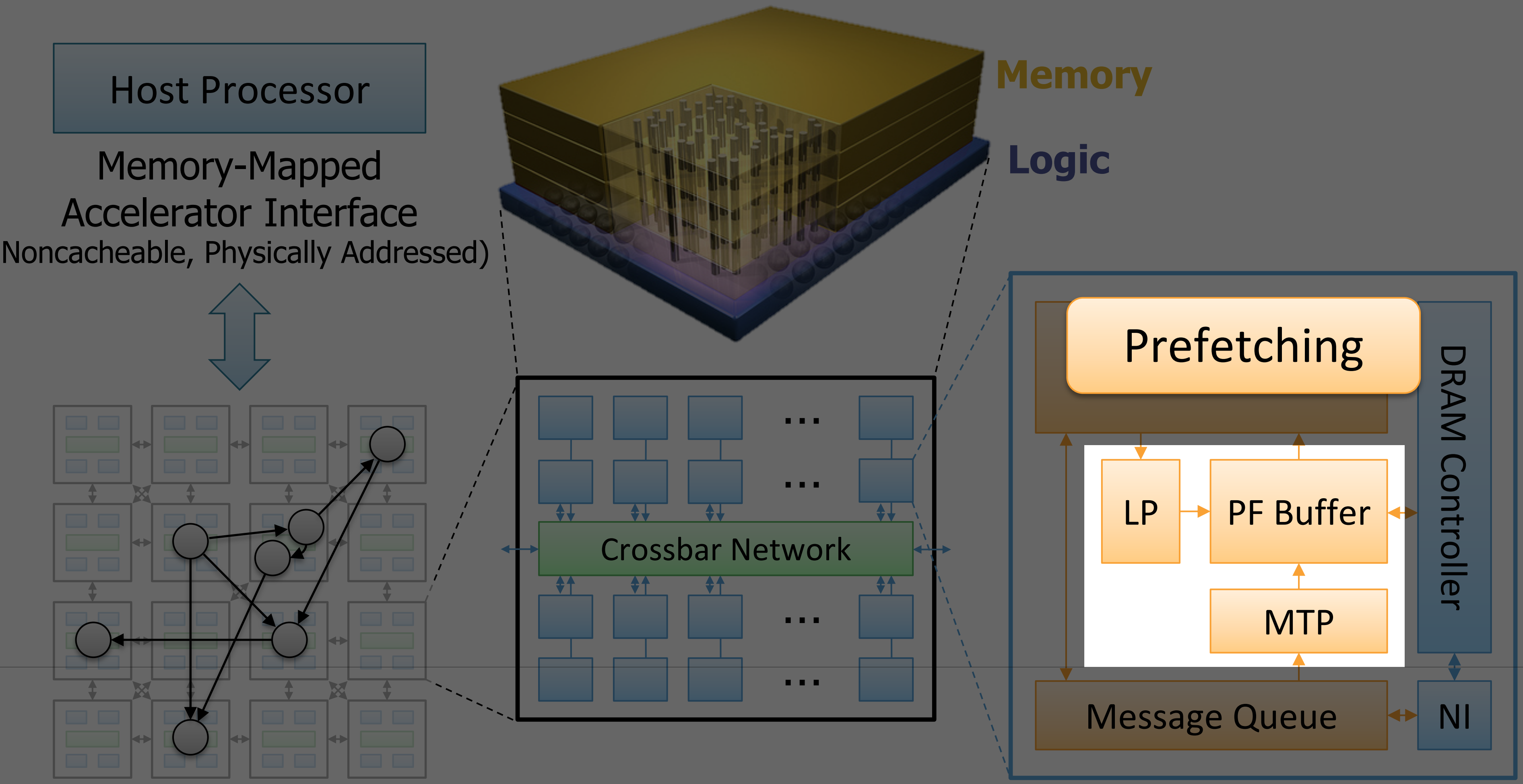
Remote Function Call (Non-Blocking)

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



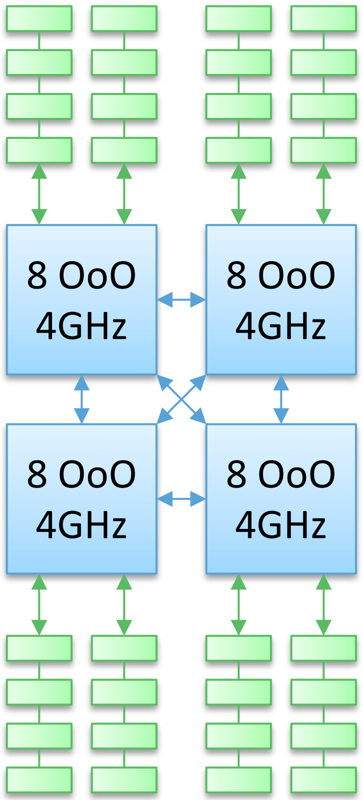
put(w.id, function() { w.next_rank += value; })

Tesseract System for Graph Processing



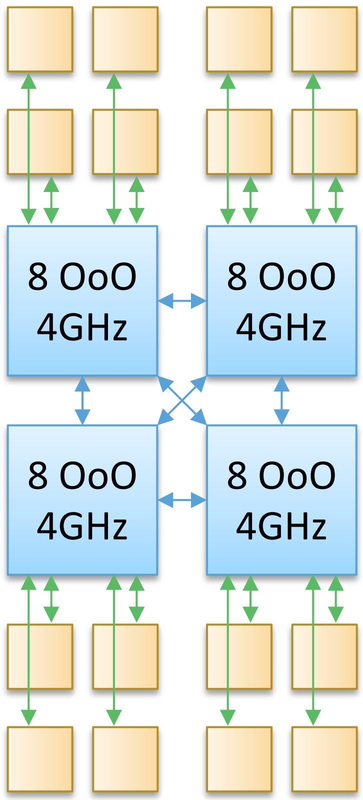
Evaluated Systems

DDR3-OoO



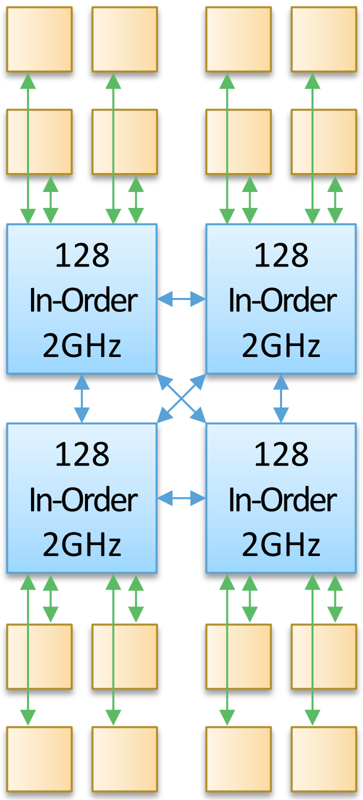
102.4GB/s

HMC-OoO



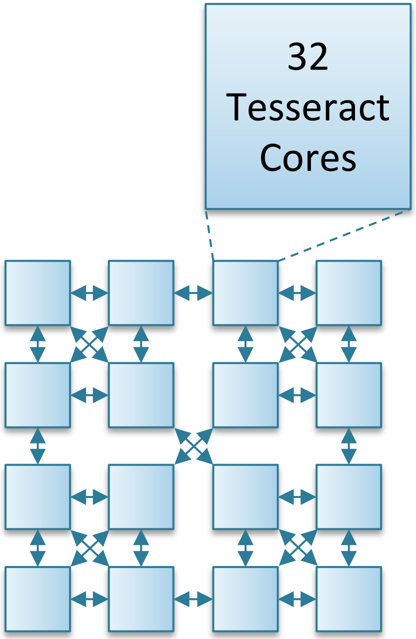
640GB/s

HMC-MC



640GB/s

Tesseract

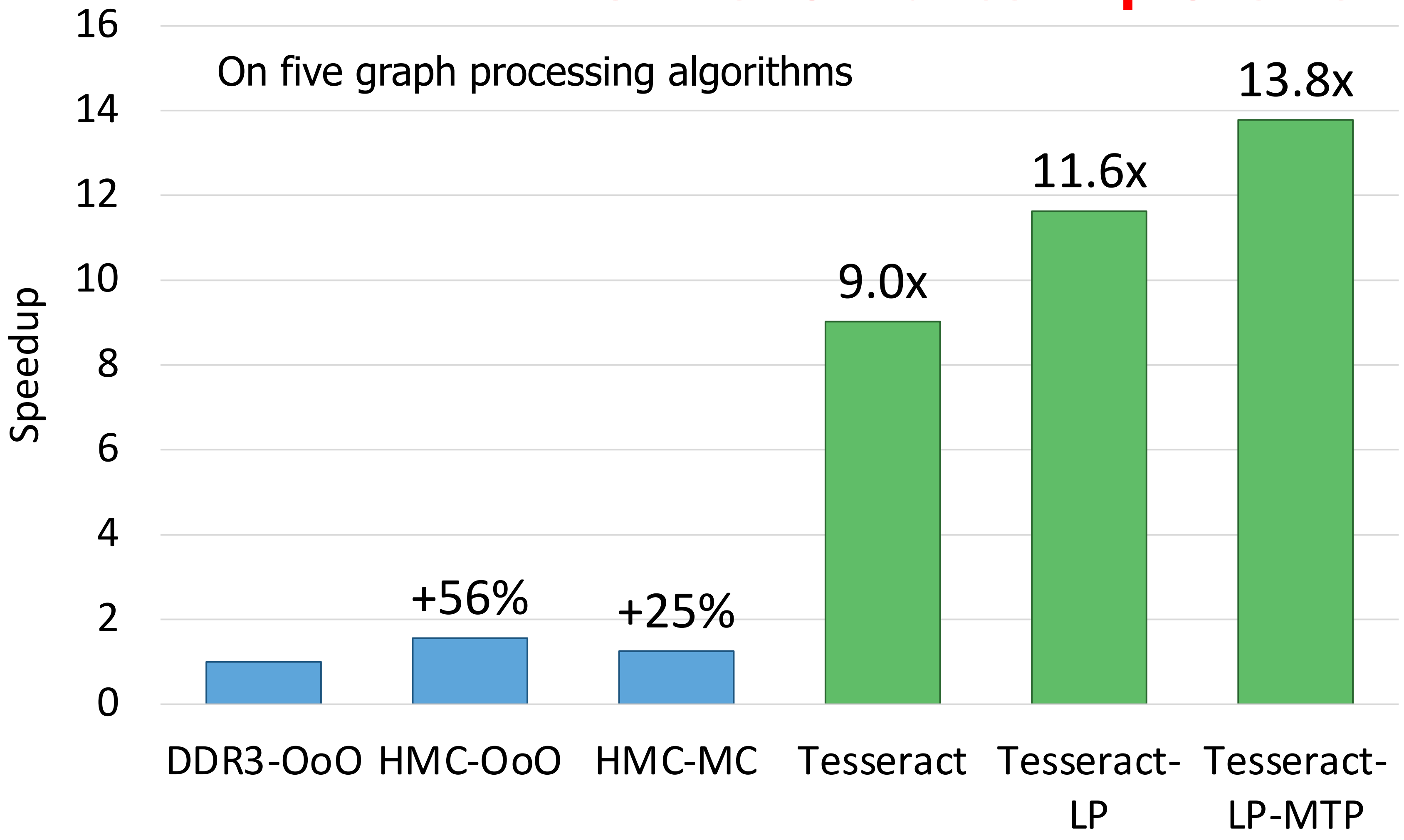


8TB/s

SAFARI Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

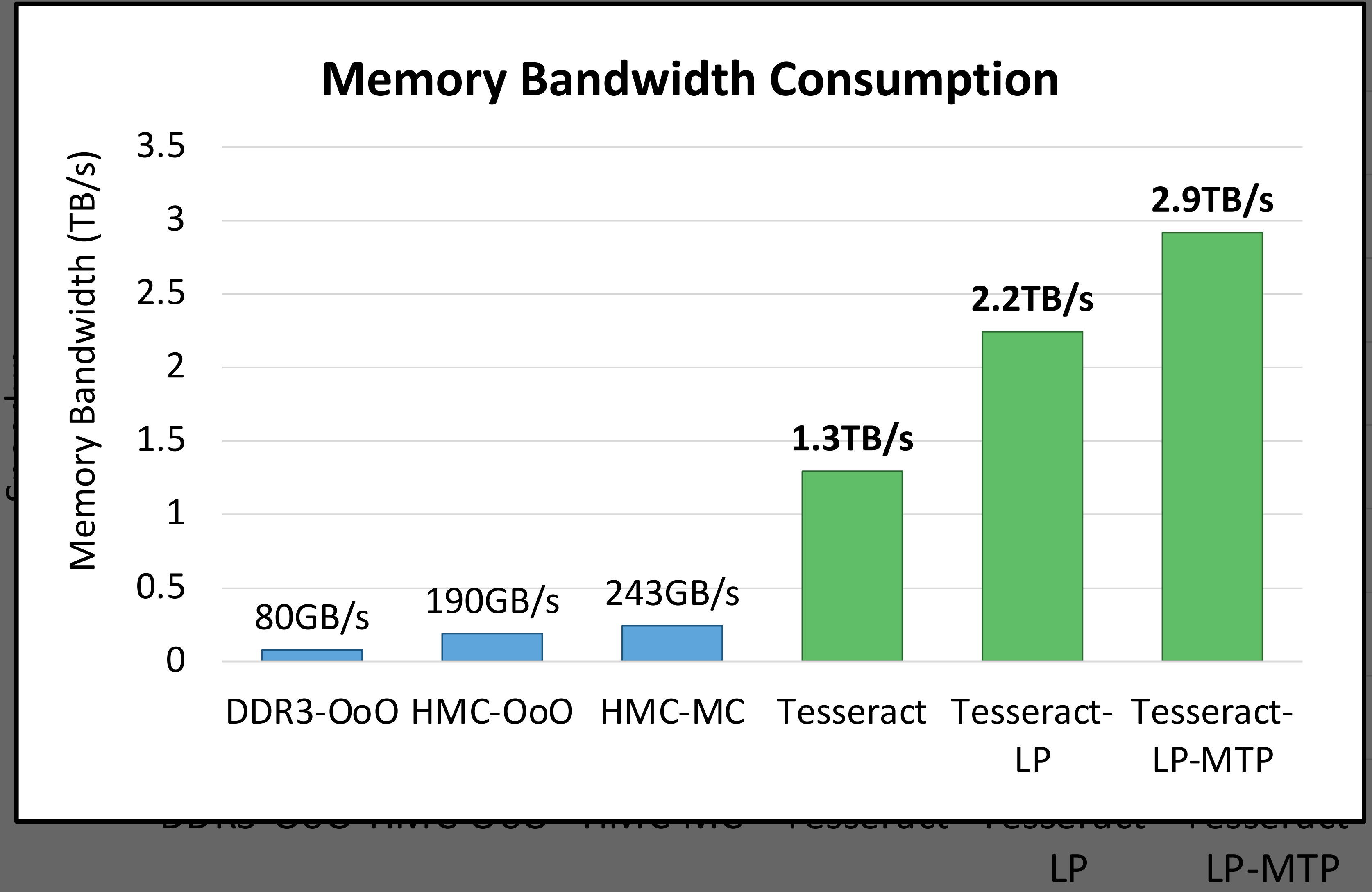
Tesseract Graph Processing Performance

>13X Performance Improvement

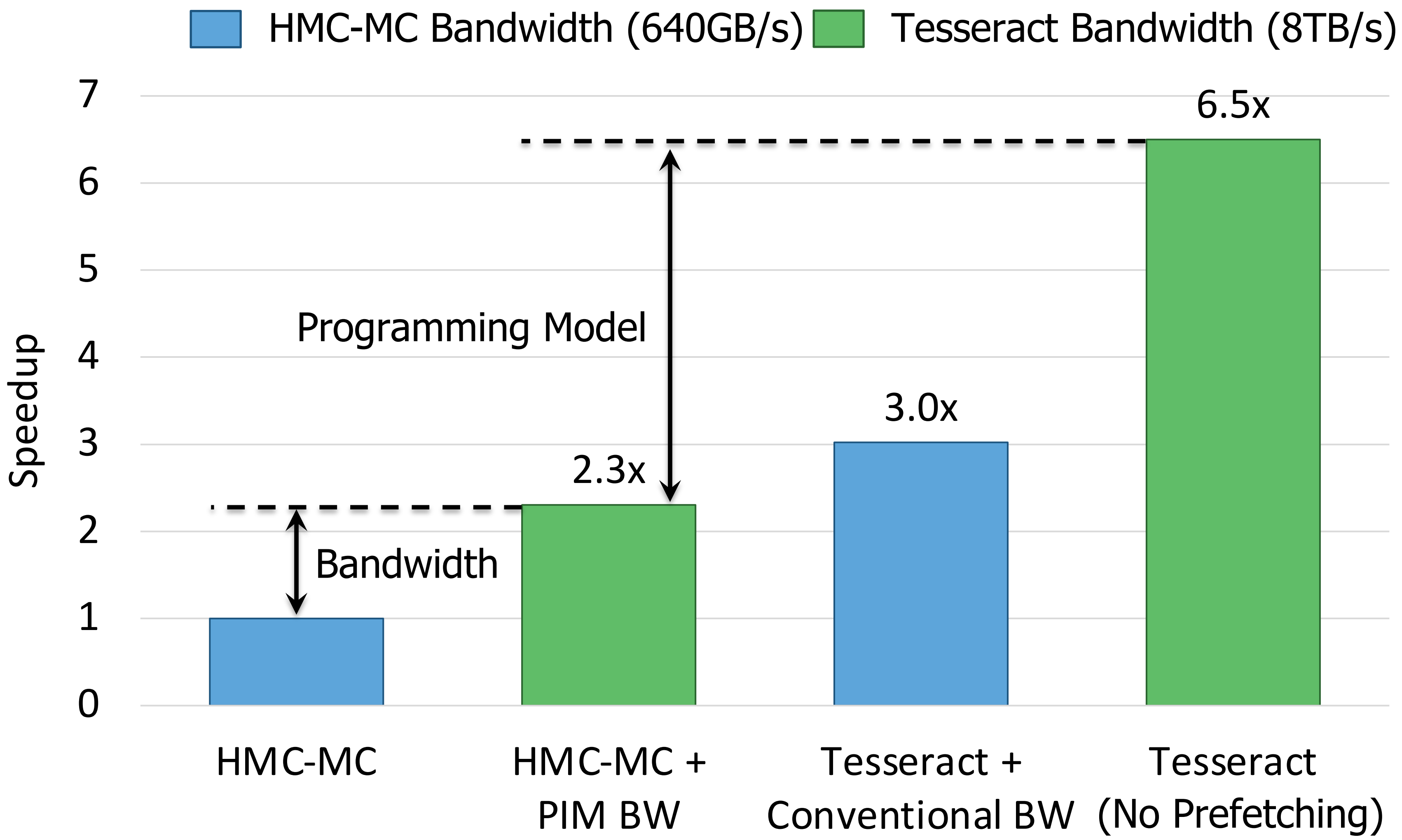


Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

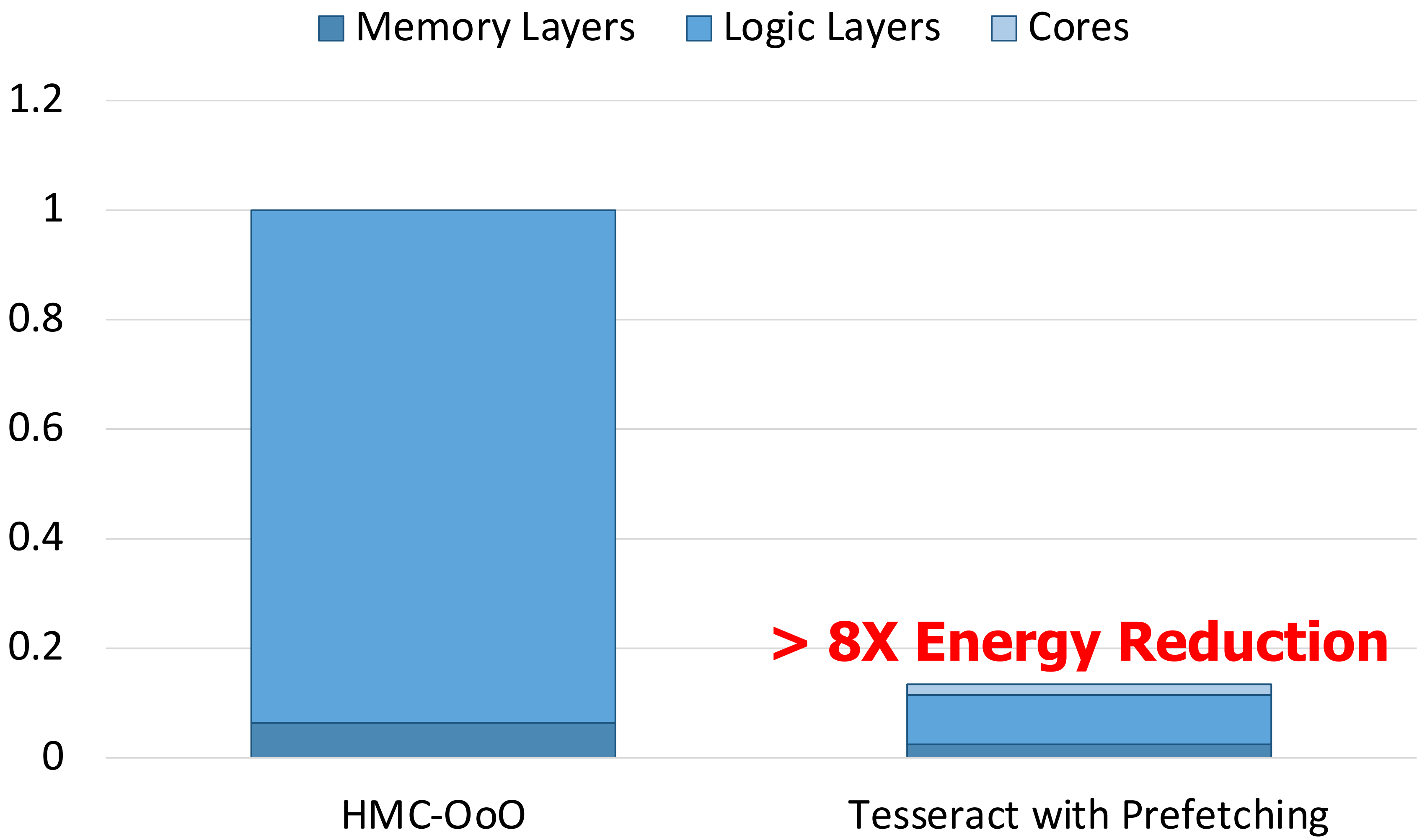
Tesseract Graph Processing Performance



Effect of Bandwidth & Programming Model



Tesseract Graph Processing System Energy

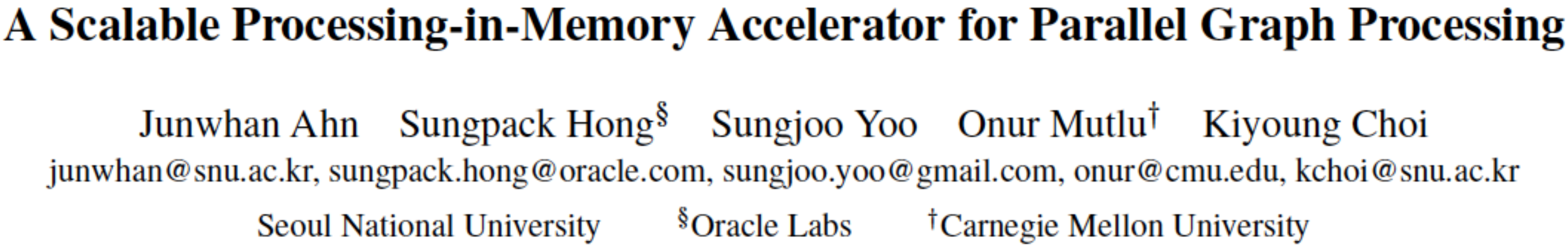


> 8X Energy Reduction

Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

More on Tesseract

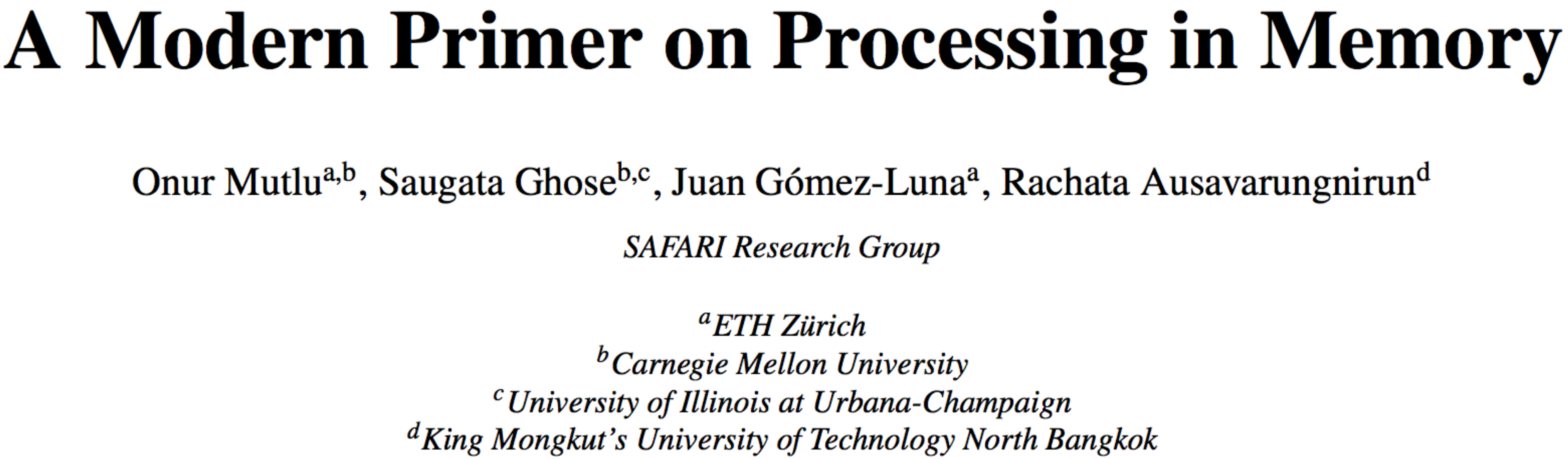
- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi, **"A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"** *Proceedings of the 42nd International Symposium on Computer Architecture* (***ISCA***), Portland, OR, June 2015. [Slides (pdf)] [Lightning Session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



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

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

PIM Review and Open Problems



A Modern Primer on Processing in Memory

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

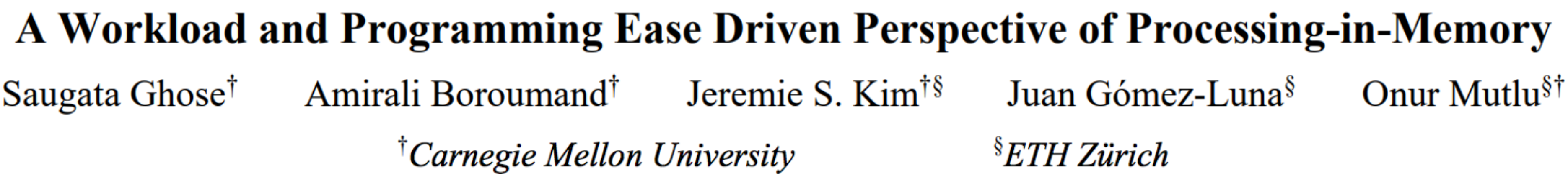
SAFARI Research Group

[a] *ETH Zürich*
[b] *Carnegie Mellon University*
[c] *University of Illinois at Urbana-Champaign*
[d] *King Mongkut's University of Technology North Bangkok*

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

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

PIM Review and Open Problems (II)



A Workload and Programming Ease Driven Perspective of Processing-in-Memory

Saugata Ghose[†] Amirali Boroumand[†] Jeremie S. Kim[†§] Juan Gómez-Luna[§] Onur Mutlu[§†]

[†]*Carnegie Mellon University* [§]*ETH Zürich*

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

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

More on Processing-in-Memory

- Onur Mutlu,
  "Memory-Centric Computing Systems"
  Invited Tutorial at *66th International Electron Devices Meeting (**IEDM**)*, Virtual, 12 December 2020.
  [Slides (pptx) (pdf)]
  [Executive Summary Slides (pptx) (pdf)]
  [Tutorial Video (1 hour 51 minutes)]
  [Executive Summary Video (2 minutes)]
  [Abstract and Bio]
  [Related Keynote Paper from VLSI-DAT 2020]
  [Related Review Paper on Processing in Memory]

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

Agenda

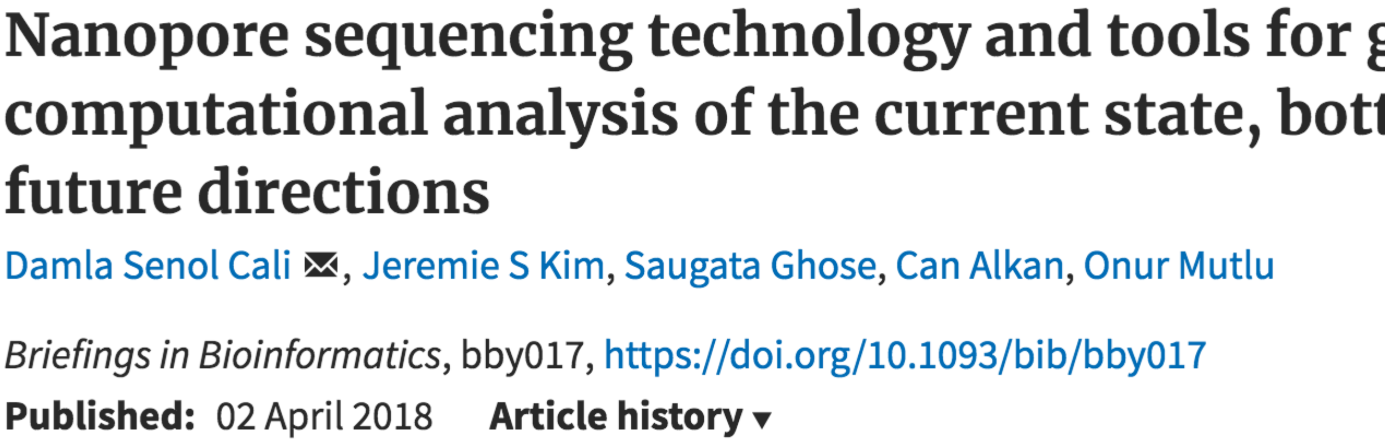
- The Problem: DNA Read Mapping
  - State-of-the-art Read Mapper Design

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

New 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.
[Preliminary arxiv.org version]

Recall: High-Throughput Sequencing

- Massively parallel sequencing technology
  - Illumina, Roche 454, Ion Torrent, SOLID…

- Small DNA fragments are first amplified and then sequenced in parallel, leading to
  - High throughput
  - High speed
  - Low cost
  - Short reads
    - Amplification step limits the read length since too short or too long fragments are not amplified well.

- Sequencing is done by either reading optical signals as each base is added, or by detecting hydrogen ions instead of light, leading to:
  - Low error rates (relatively)
  - Reads lack information about their order and which part of genome they are originated from

Nanopore Sequencing Technology

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

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

Nanopore Sequencing Technology

an emerging and a promising
ncing technology

read length → Longer read length

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

Oxford Nanopore Sequencers

MinION Mk1B

MinION Mk1C

GridION Mk1

PromethION 24/48

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

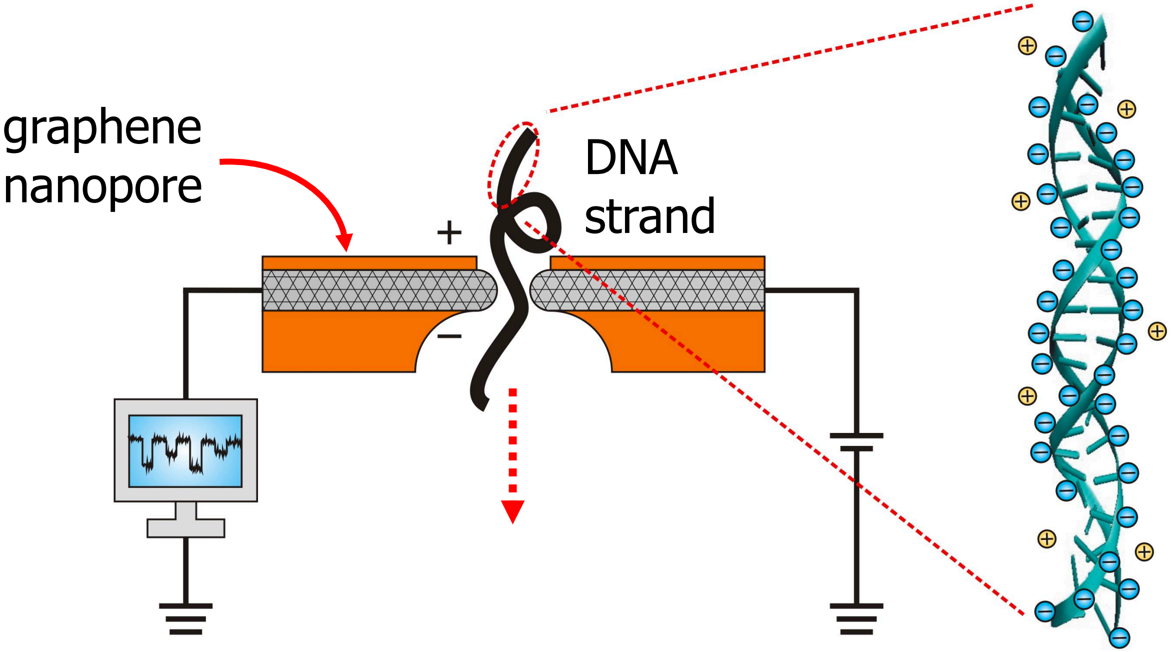
https://nanoporetech.com/products/comparison

Illumina Sequencers

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

https://www.illumina.com/systems/sequencing-platforms.html

How Does Nanopore Sequencing Work?



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

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

Advantages of Nanopore Sequencing

Nanopores:

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

Challenges of Nanopore Sequencing

- One major drawback: high error rates

- Nanopore sequence analysis tools have a critical role to:
  - overcome high error rates
  - take better advantage of the technology

- Faster tools are critically needed to:
  - Take better advantage of the real-time data production capability of nanopore sequencing
  - Enable fast, real-time data analysis

Nanopore Genome Assembly Pipeline

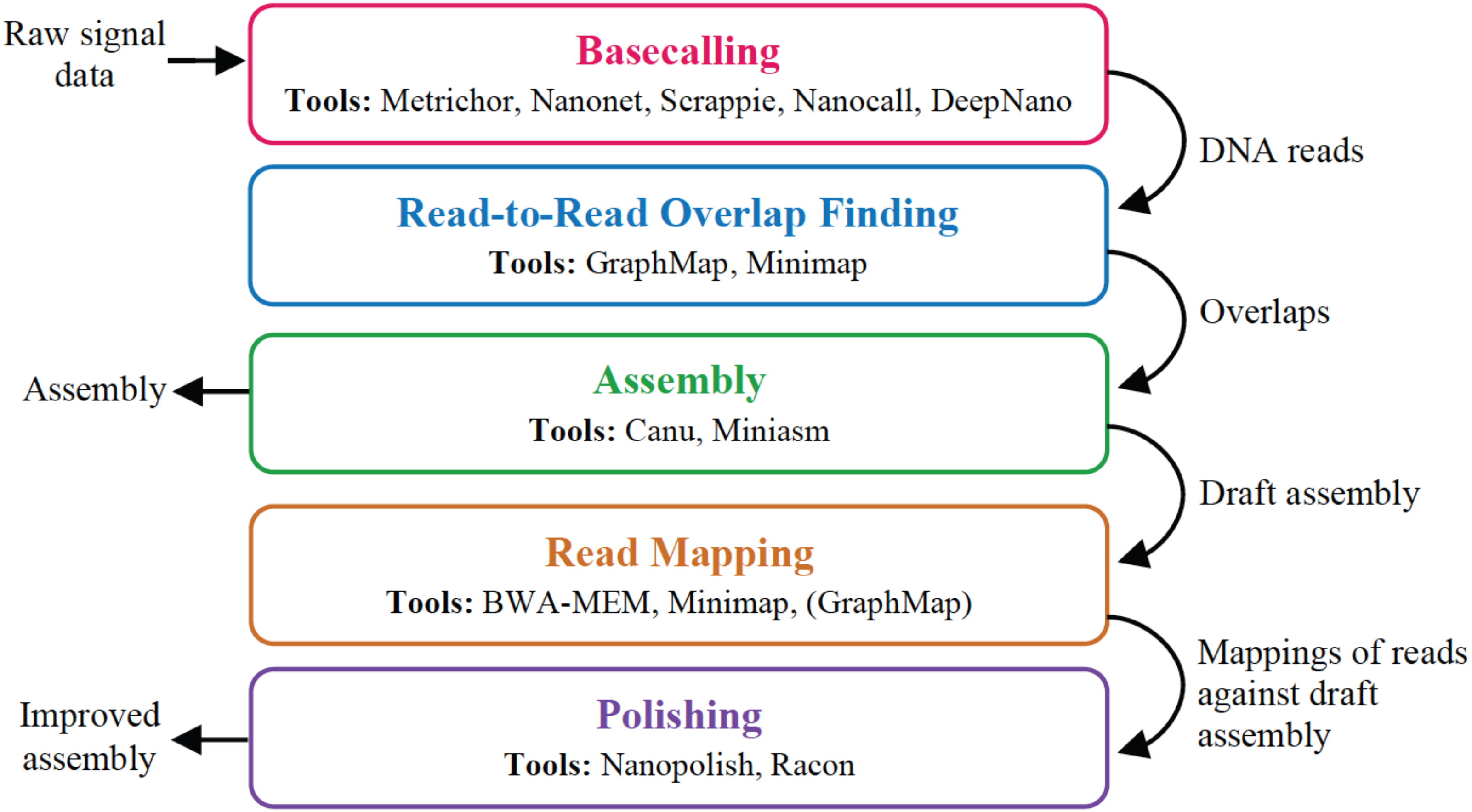


Figure 1. The analyzed genome assembly pipeline using nanopore sequence data, with its five steps and the associated tools for each step.

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

Nanopore Genome Assembly Tools (I)

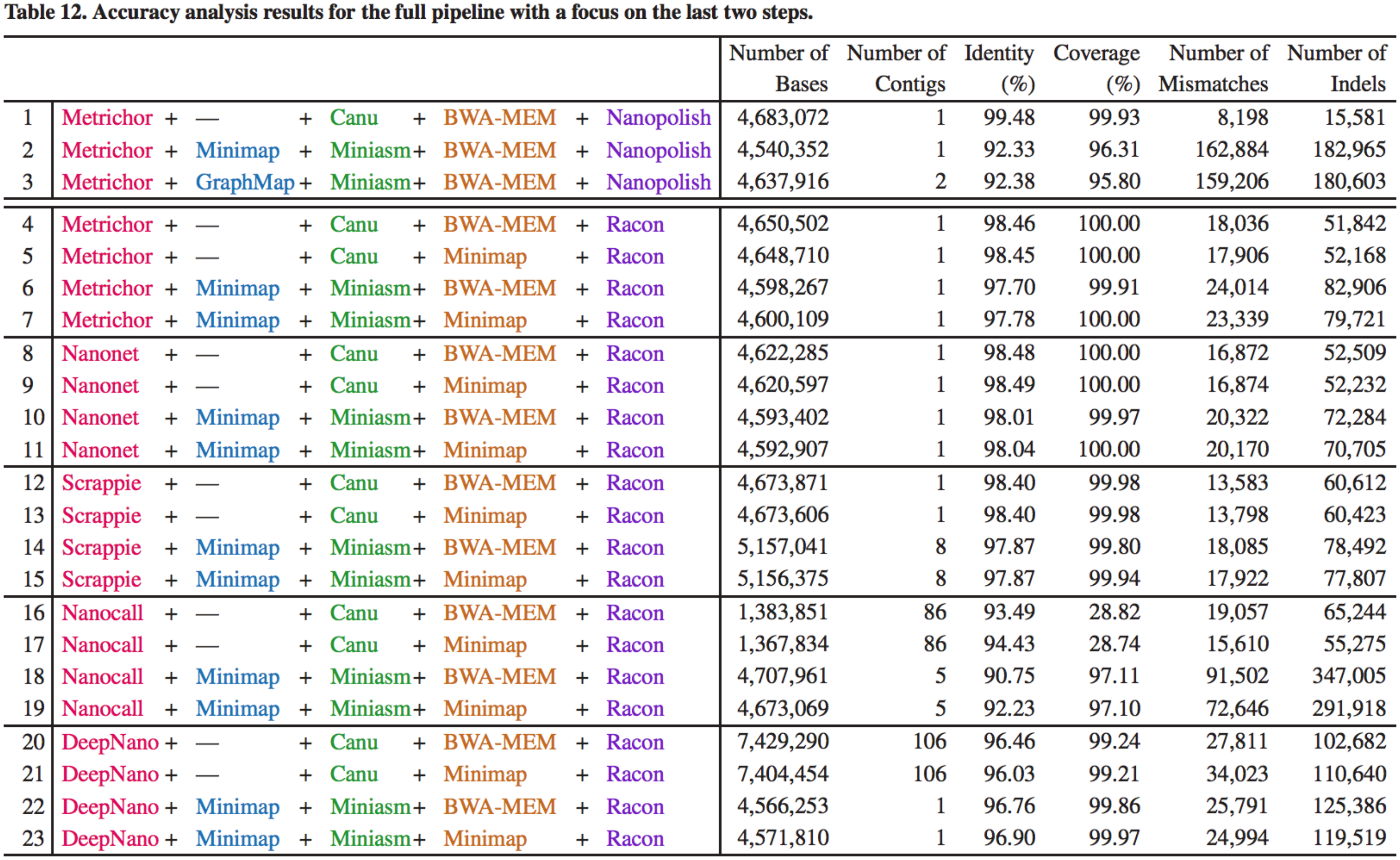


Table 12. Accuracy analysis results for the full pipeline with a focus on the last two steps.

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

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

Nanopore Genome Assembly Tools (II)

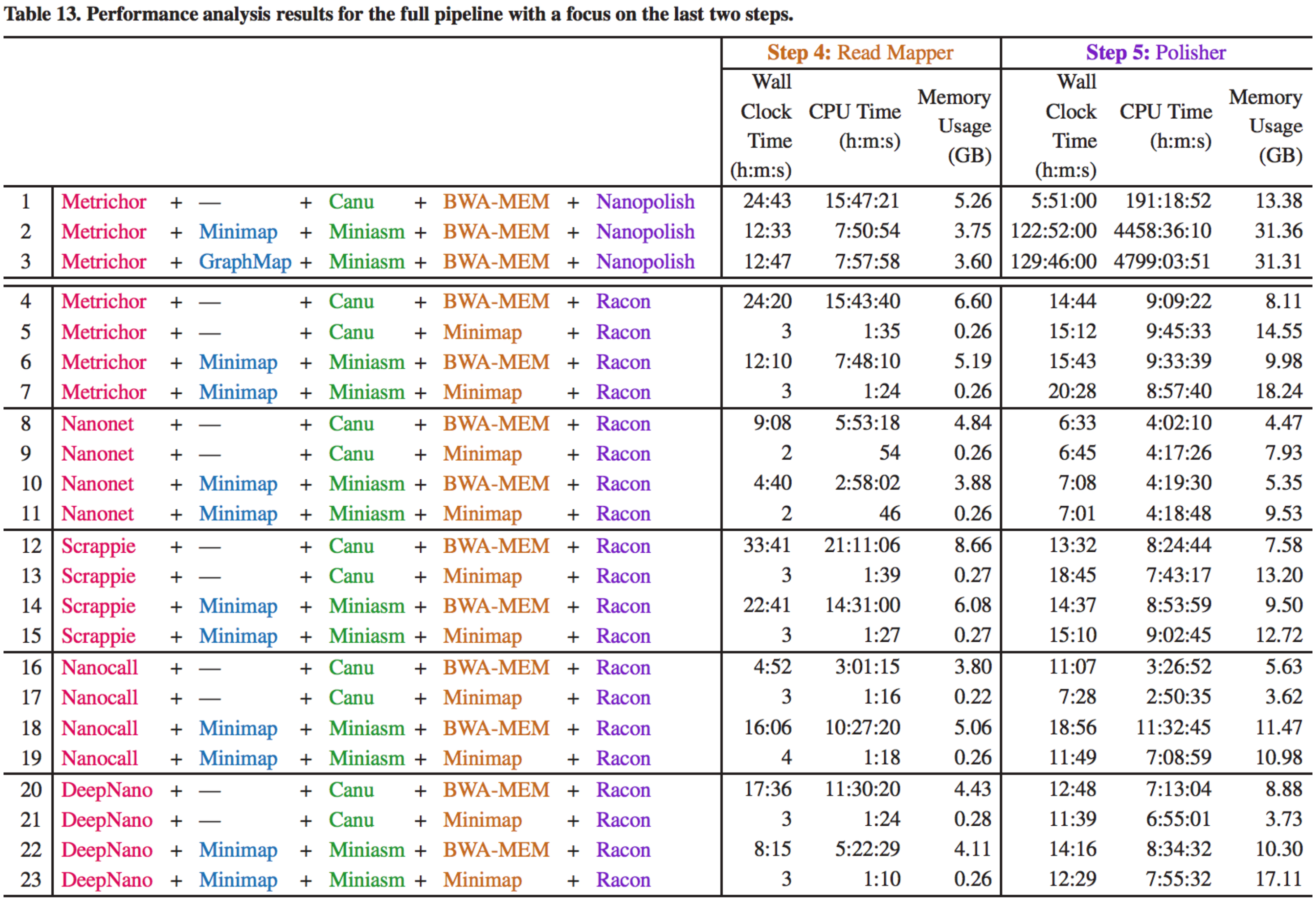
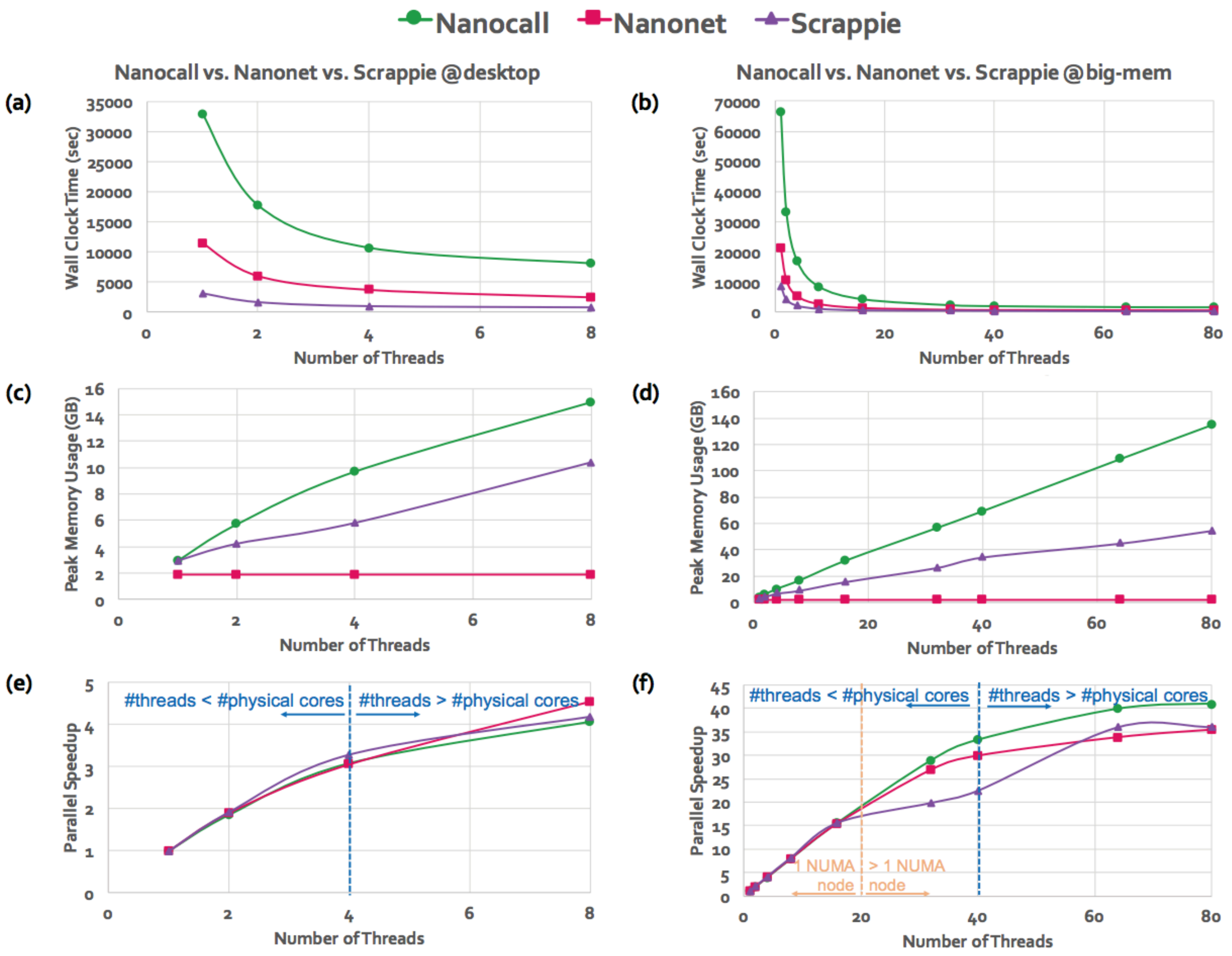


Table 13. Performance analysis results for the full pipeline with a focus on the last two steps.

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

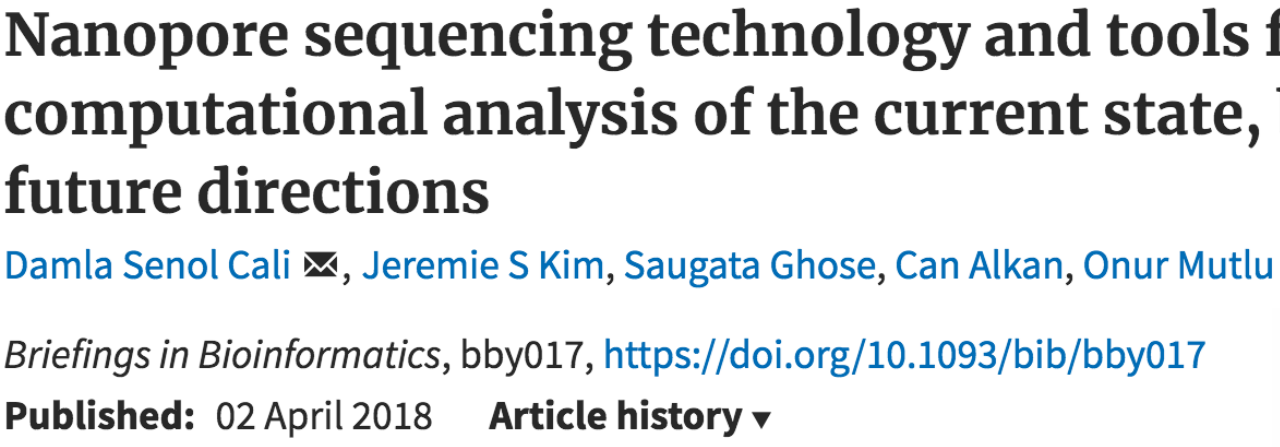
Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly**" Briefings in Bioinformatics, 2018.

Nanopore Genome Assembly Tools (III)



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

More on Nanopore Sequencing & Tools



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

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

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

Published: 02 April 2018 **Article history** ▾

BiB arXiv

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

Recall Our Dream (from 2007)

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

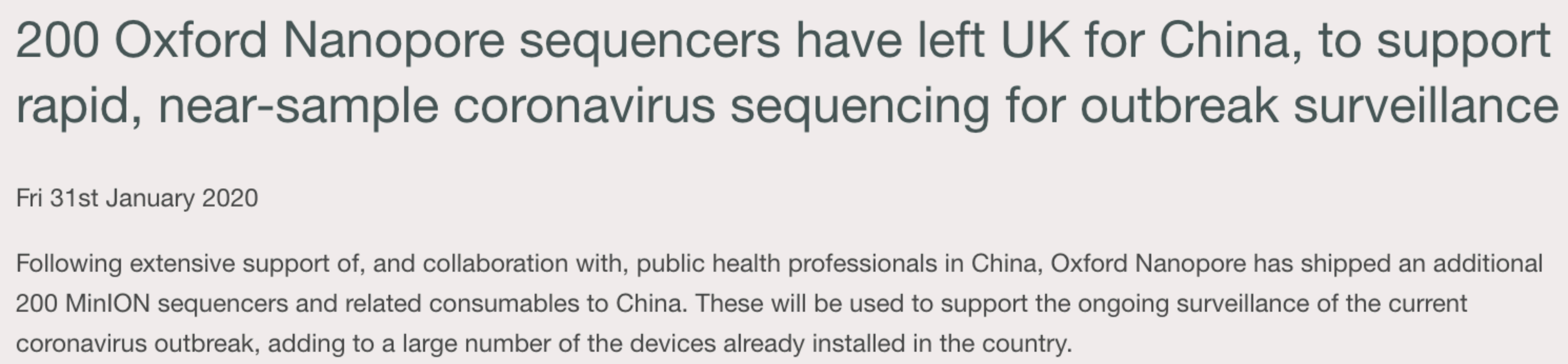
- Still a long ways to go
  - Energy efficiency
  - Performance (latency)
  - Security
  - **Huge memory bottleneck**

Future of Genome Sequencing & Analysis

MinION from ONT

SmidgION from ONT

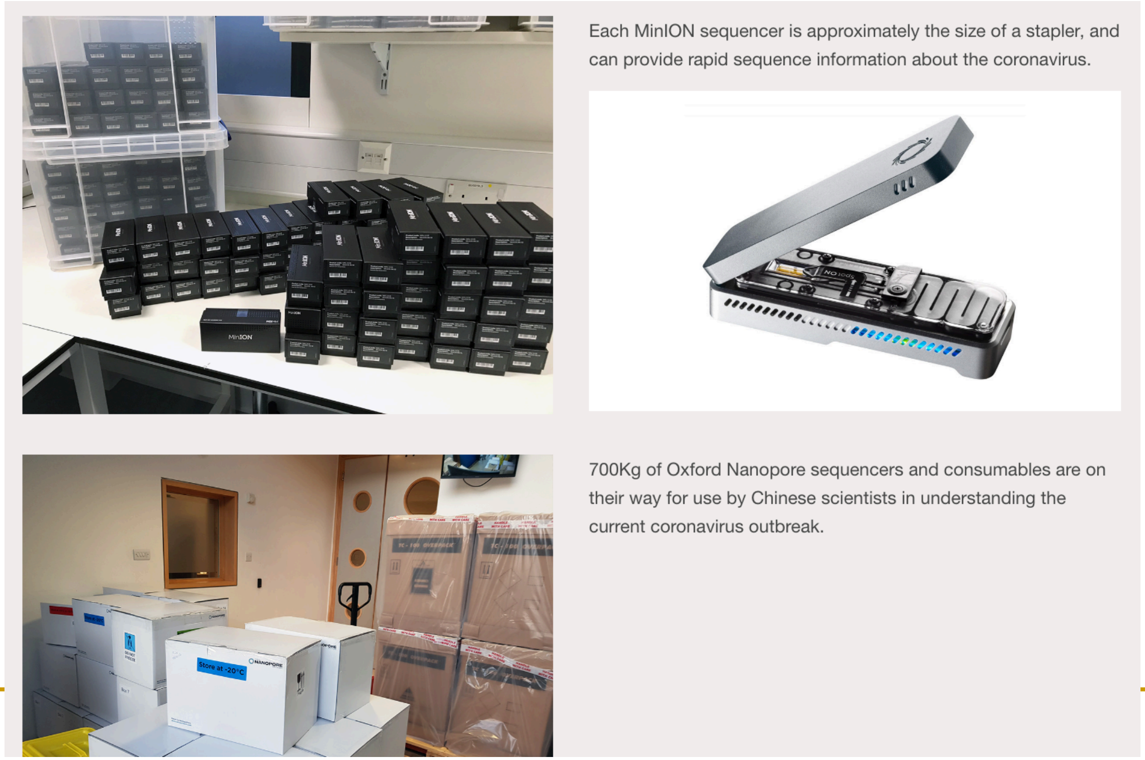
Why Do We Care? An Example from 2020



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

Fri 31st January 2020

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



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

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

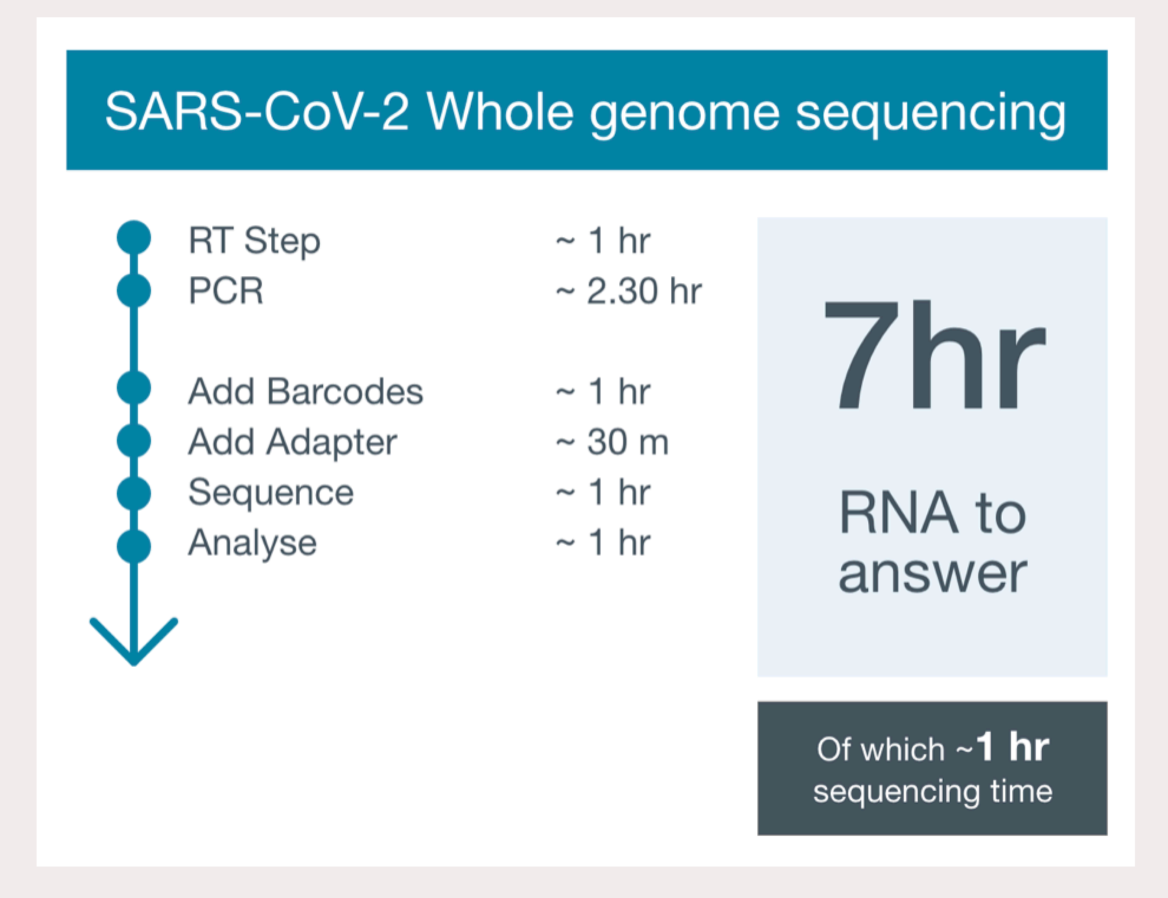
Source: https://nanoporetech.com/about-us/news/200-oxford-nanopore-sequencers-have-left-uk-china-support-rapid-near-sample

Sequencing of COVID-19

- **Whole genome sequencing (WGS) and sequence data analysis are important**
  - To detect the virus from a human sample such as saliva, Bronchoalveolar fluid etc.
  - To understand the sources and modes of transmission of the virus
  - To discover the genomic characteristics of the virus, and compare with better-known viruses (e.g., 02-03 SARS epidemic)
  - To design and evaluate the diagnostic tests and deep-dive studies

- **Two key areas of COVID-19 genomic research**
  - To sequence the genome of the virus itself, COVID-19, in order to track the mutations in the virus.
  - To explore the genes of infected patients. This analysis can be used to understand why some people get more severe symptoms than others, as well as, help with the development of new treatments in the future.

COVID-19 Nanopore Sequencing (I)



SARS-CoV-2 Whole genome sequencing

RT Step	~ 1 hr
PCR	~ 2.30 hr
Add Barcodes	~ 1 hr
Add Adapter	~ 30 m
Sequence	~ 1 hr
Analyse	~ 1 hr

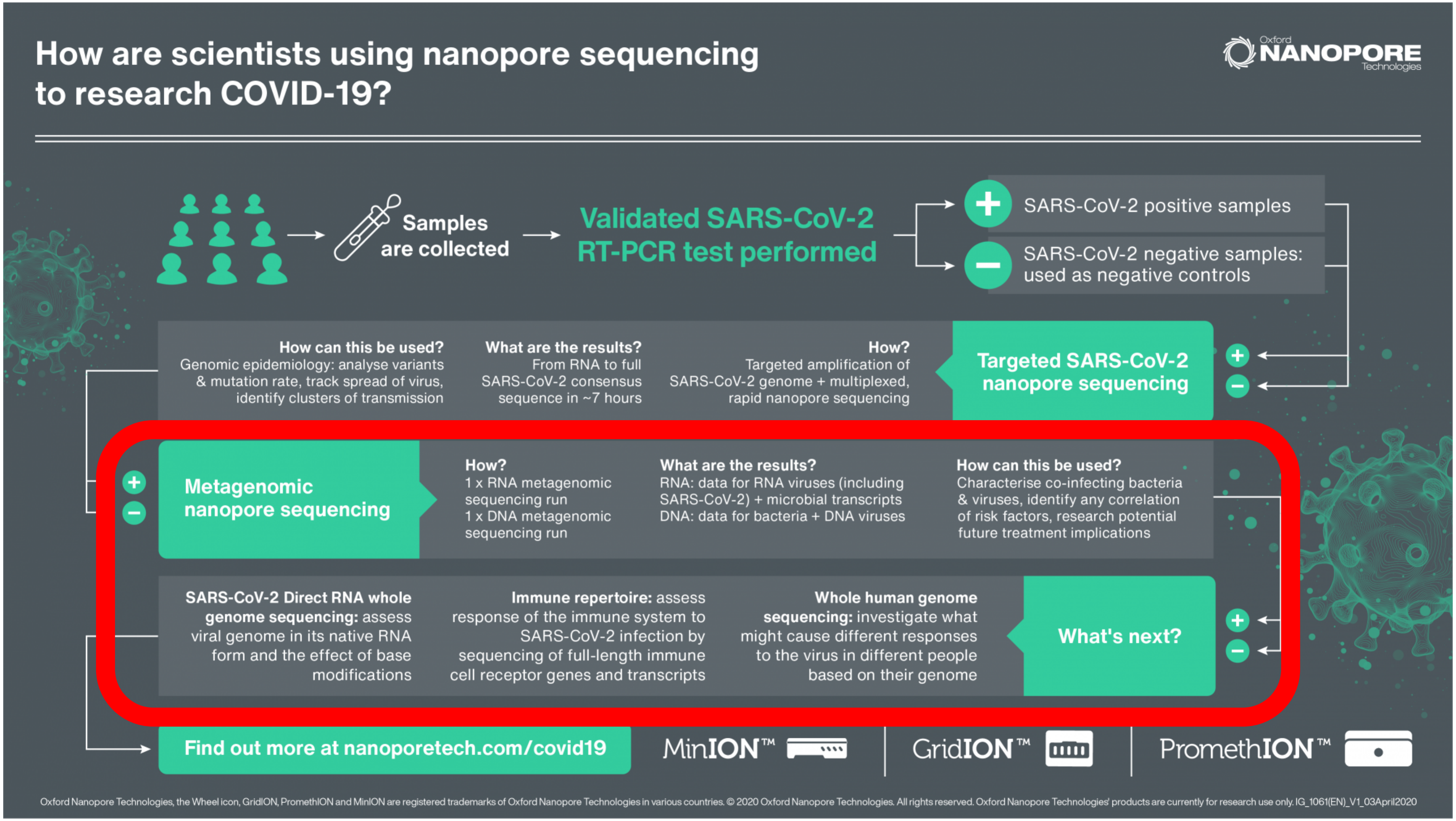
7hr

RNA to answer

Of which ~**1 hr** sequencing time

- 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

Targeted SARS-CoV-2 nanopore sequencing

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

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

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

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

What's next?

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

Find out more at nanoporetech.com/covid19

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

Future of Genome Sequencing & Analysis

MinION from ONT

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

- Future Opportunities: New Sequencing Technologies

Conclusion

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

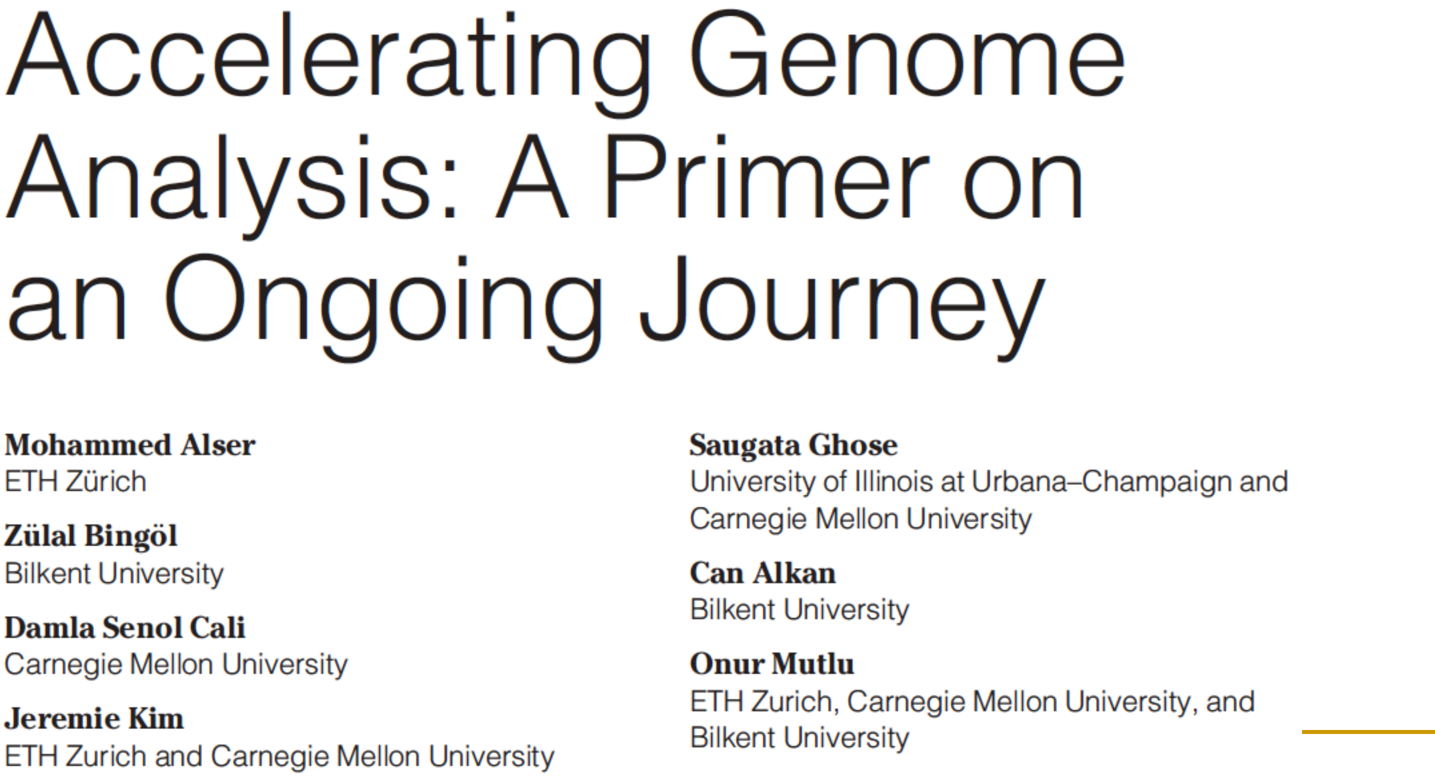
- This talk is about accelerating a key step in bioinformatics: genome sequence analysis
  - In particular, read mapping

- 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**

Accelerating Genome Analysis: Overview

- Mohammed Alser, Zulal Bingol, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
"Accelerating Genome Analysis: A Primer on an Ongoing Journey"
IEEE Micro (***IEEE MICRO***), Vol. 40, No. 5, pages 65-75, September/October 2020.
[Slides (pptx)(pdf)]
[Talk Video (1 hour 2 minutes)]



Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser
ETH Zürich

Zülal Bingöl
Bilkent University

Damla Senol Cali
Carnegie Mellon University

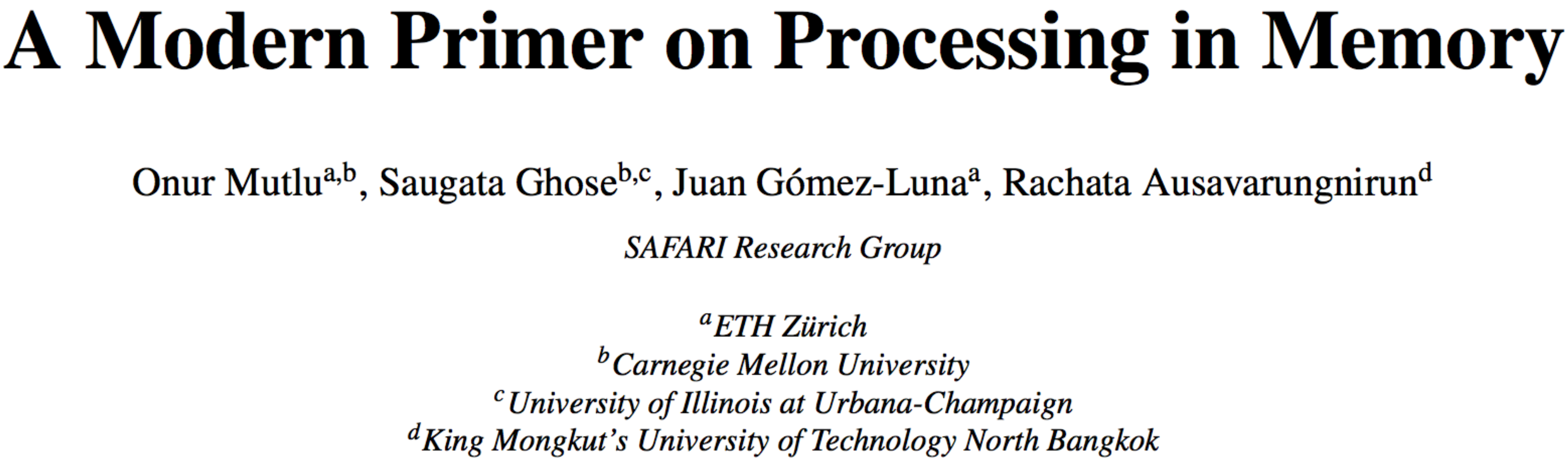
Jeremie Kim
ETH Zurich and Carnegie Mellon University

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

Can Alkan
Bilkent University

Onur Mutlu
ETH Zurich, Carnegie Mellon University, and Bilkent University

PIM Review and Open Problems



A Modern Primer on Processing in Memory

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

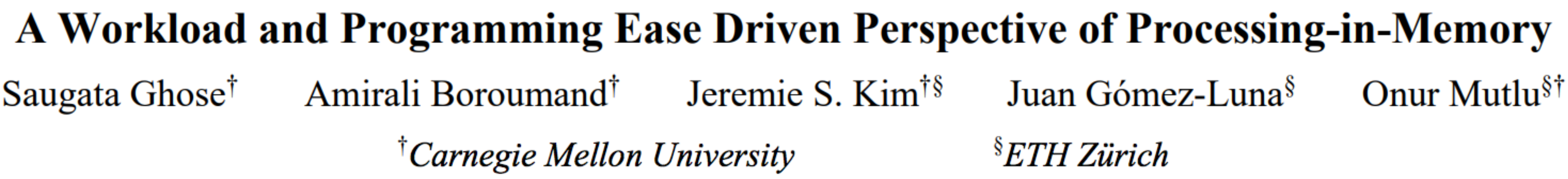
SAFARI Research Group

[a] *ETH Zürich*
[b] *Carnegie Mellon University*
[c] *University of Illinois at Urbana-Champaign*
[d] *King Mongkut's University of Technology North Bangkok*

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

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

PIM Review and Open Problems (II)



A Workload and Programming Ease Driven Perspective of Processing-in-Memory

Saugata Ghose[†] Amirali Boroumand[†] Jeremie S. Kim[†§] Juan Gómez-Luna[§] Onur Mutlu[§†]

[†]*Carnegie Mellon University* [§]*ETH Zürich*

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

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

More on 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

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

https://www.youtube.com/onurmutlulectures

Acknowledgments

- Can Alkan, Bilkent University

- Many students at ETH, CMU, Bilkent
  - Mohammed Alser, Damla Senol Cali, Jeremie Kim, Hasan Hassan, 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

Funding Acknowledgments

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

Acknowledgments

SAFARI

SAFARI Research Group

safari.ethz.ch

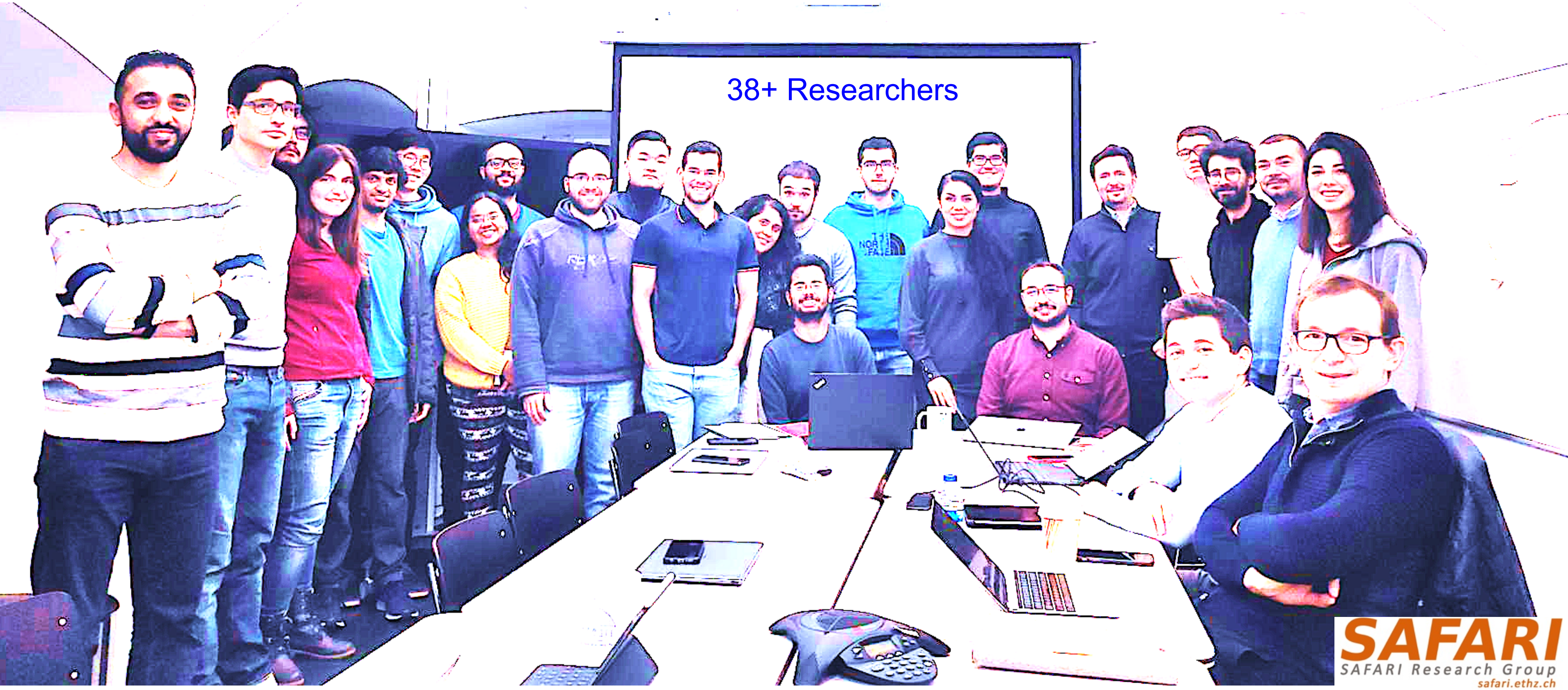
Think BIG, Aim HIGH!

https://safari.ethz.ch

Onur Mutlu's SAFARI Research Group

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

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



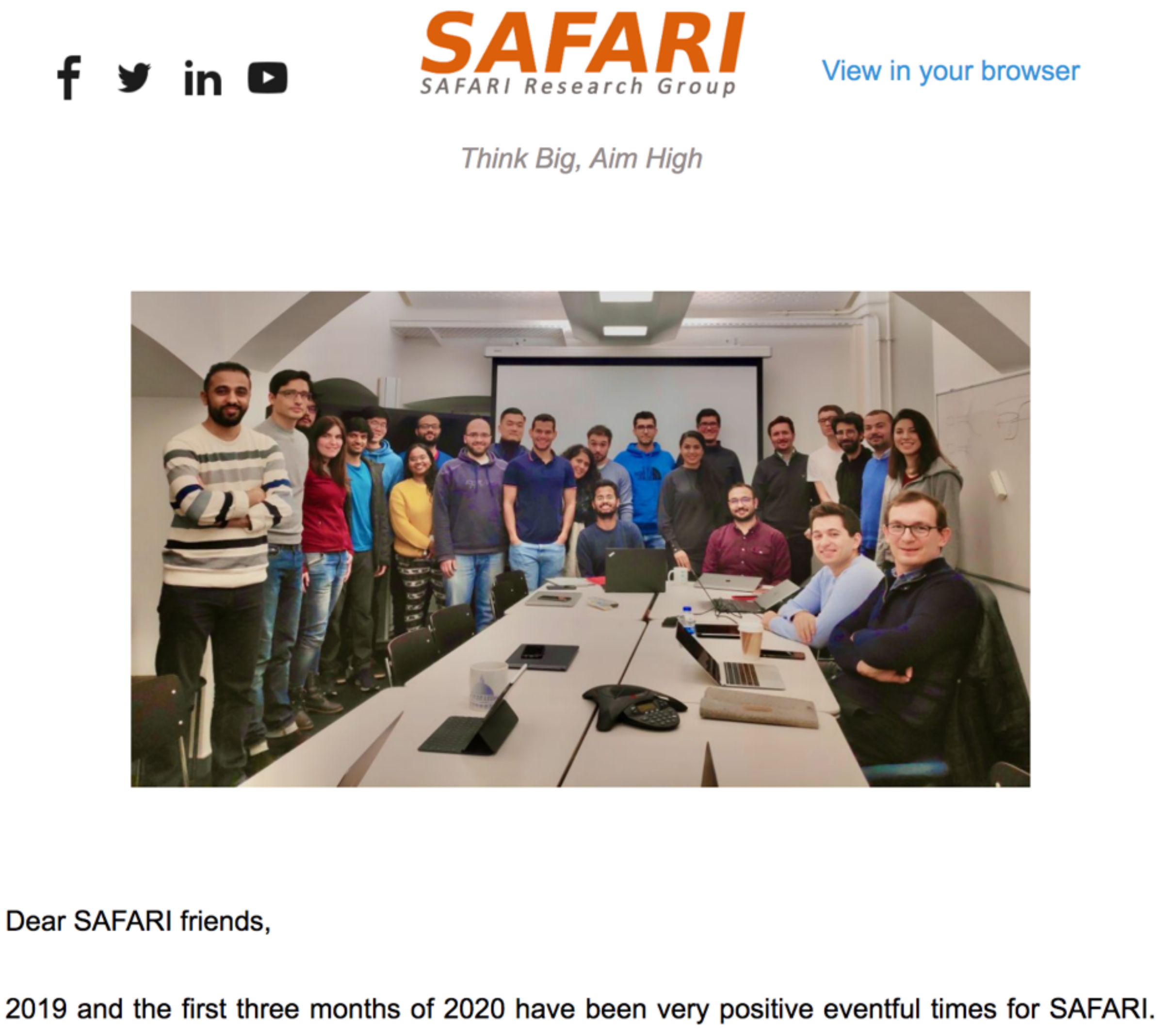
Think BIG, Aim HIGH!

SAFARI

https://safari.ethz.ch

SAFARI Newsletter April 2020 Edition

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



SAFARI
SAFARI Research Group

View in your browser

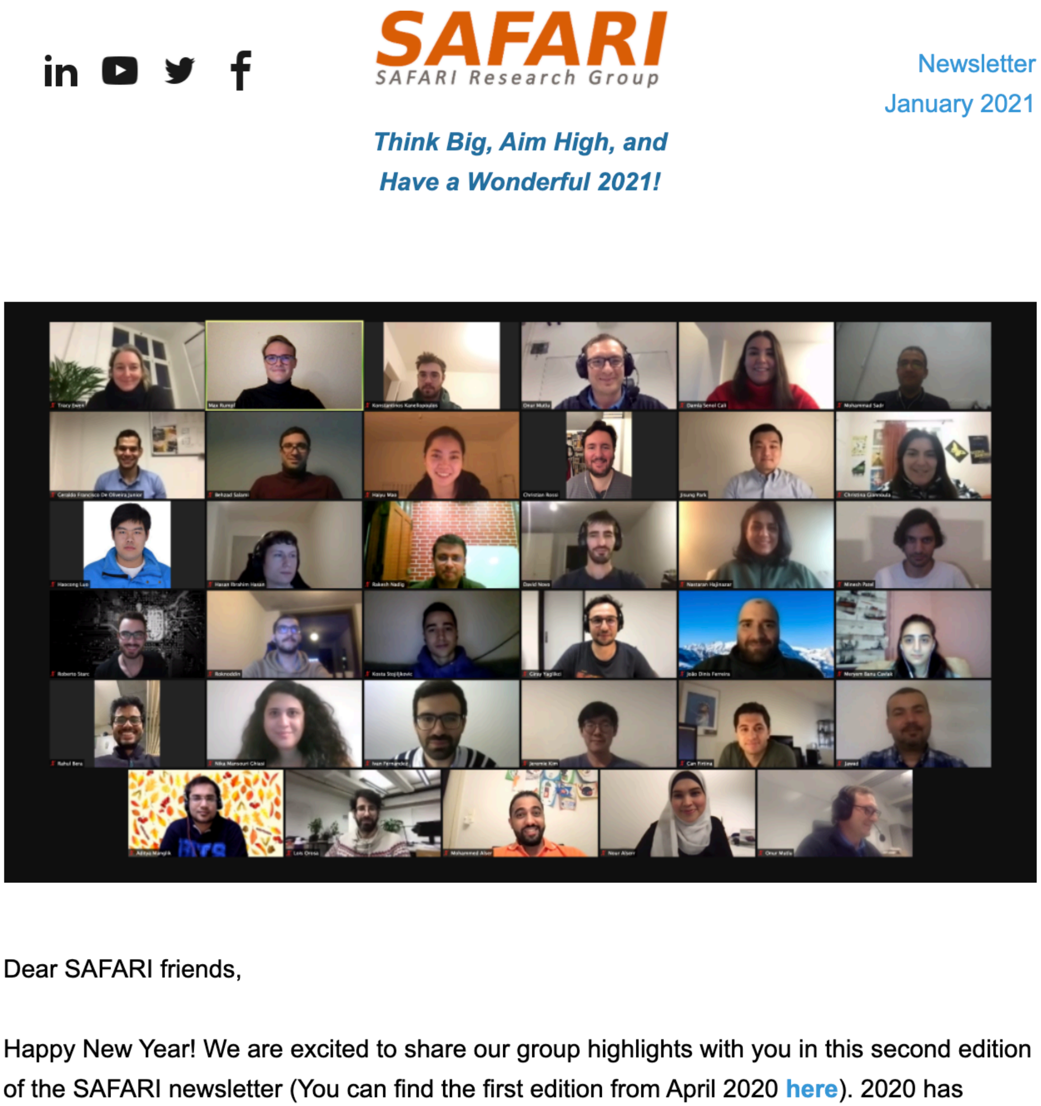
Think Big, Aim High

Dear SAFARI friends,

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

SAFARI Newsletter January 2021 Edition

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



SAFARI
SAFARI Research Group

Newsletter
January 2021

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

Dear SAFARI friends,

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

Accelerating Genome Analysis

A Primer on an Ongoing Journey

Onur Mutlu

omutlu@gmail.com

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

26 January 2021

Technion Invited Lecture

Backup Slides for Further Info

Referenced Papers and Talks

- All are available at

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

  http://scholar.google.com/citations?user=7XyGUGkAAAAJ&hl=en

  https://www.youtube.com/onurmutlulectures

Research & Teaching: Some Overview Talks

https://www.youtube.com/onurmutlulectures

- Future Computing Architectures
  - https://www.youtube.com/watch?v=kgiZlSOcGFM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=1
- Enabling In-Memory Computation
  - https://www.youtube.com/watch?v=njX_14584Jw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=16
- Accelerating Genome Analysis
  - https://www.youtube.com/watch?v=hPnSmfwu2-A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=9
- Rethinking Memory System Design
  - https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=3
- Intelligent Architectures for Intelligent Machines
  - https://www.youtube.com/watch?v=n8Aj_A0WSg8&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=22
- Revisiting RowHammer
  - https://www.youtube.com/watch?v=B58YT9hZM4g&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=25

An Interview on Research and Education

- Computing Research and Education (@ ISCA 2019)
  - https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi_4oP9LdL3cc8G6NIjD2Ydz

- Maurice Wilkes Award Speech (10 minutes)
  - https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=15

More Thoughts and Suggestions

- Onur Mutlu,
  "Some Reflections (on DRAM)"
  *Award Speech for ACM SIGARCH Maurice Wilkes Award, at the **ISCA** Awards Ceremony*, Phoenix, AZ, USA, 25 June 2019.
  [Slides (pptx) (pdf)]
  [Video of Award Acceptance Speech (Youtube; 10 minutes) (Youku; 13 minutes)]
  [Video of Interview after Award Acceptance (Youtube; 1 hour 6 minutes) (Youku; 1 hour 6 minutes)]
  [News Article on "ACM SIGARCH Maurice Wilkes Award goes to Prof. Onur Mutlu"]

- Onur Mutlu,
  "How to Build an Impactful Research Group"
  *57th Design Automation Conference Early Career Workshop (**DAC**)*, Virtual, 19 July 2020.
  [Slides (pptx) (pdf)]

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

https://www.youtube.com/onurmutlulectures

Detailed Lectures on PIM (II)

- Computer Architecture, Fall 2020, Lecture 15
  - **Emerging Memory Technologies** (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=AlE1rD9G_YU&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=28

- Computer Architecture, Fall 2020, Lecture 16a
  - **Opportunities & Challenges of Emerging Memory Technologies** (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=29

- Computer Architecture, Fall 2020, Guest Lecture
  - **In-Memory Computing: Memory Devices & Applications** (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=wNmqQHiEZNk&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=41

Genome Analysis

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

>CCTCCTCAGTGCCACCCAGCCCACTGGCAGCTCCCAAACAGGCTCTTATTAAAACACCCTGTTCCCTGCCCCTTGGAGTGAGGTGTCAAG
GACCTAAACTAAAAAAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTT
CATGTCAAGGACCTAATGTGCTAAACAGCACTTTTTTGACCATTATTTTGGATCTGAAAGAAATCAAGAATAAATGAAGGACTTGATACATTG
GAAGAGGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAAAGAAAAAGAAAAGAAAAAGA**A**TTTAAAATTTAAGTAATTCTTTGAAAAAA
ACTAATTTCTAAGCTTCTT**C**ATGTCAAGGACCTAATGTCTGTGTTGCAGGTCTTCTTGCATTTCCCTGTCAAAAGAAAAAGAATTTAAAATTT
AAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCAGGCCAAGAGTTGCAAAAAAAAAAAAAAGAAAAA
GAAAAGAAAAAGAATTTAAAATTTA**A**GTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTAGCCAGAATGG
TTGTGGGATGGGAGCCTCTGTGGACCGACCAGGTAGCTCTCTTTTCCACACTGTAGTCTCAAAGCTTCTTCATGTGGTTTCTCTGAGTGAAA
AAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTT**T**TTCATGTCAAGGACC
TAATGTAGCTATACTGAACGTTATCTAGGGGAAAGATTGAAGGGGAGCTCTAAGGTCAACACACCACCACTTCCCAGAAAGCTTCTTCA……

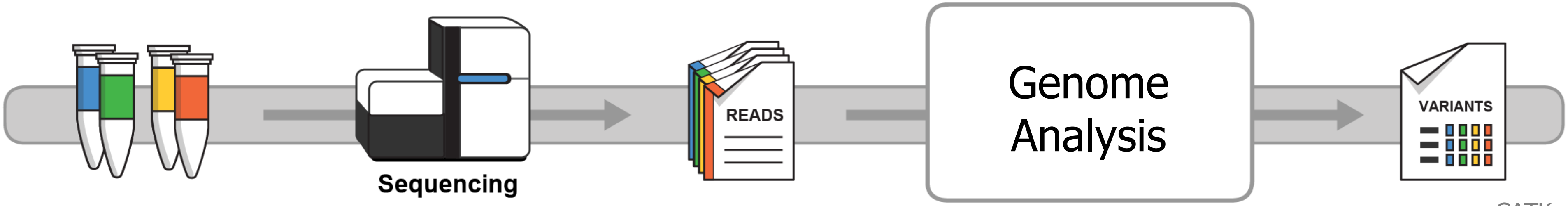
Genome Analysis

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

Why?!

>CCT CAAG
GACC TTCTT
CATGT CATTG
GAAG AAAA
ACTA ATTT
AAGT AAAA
GAAA ATGG
TTGTC GAAA
AAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTTTTTCATGTCAAGGACC
TAATGTAGCTATACTGAACGTTATCTAGGGGAAAGATTGAAGGGGAGCTCTAAGGTCAACACACCACCACTTCCCAGAAAGCTTCTTCA......

Genome Sequencer is a Chopper



GATK

CCCCCCTATATATACGTACTAGTACGT

ACGACTTTAGTACGTACGT

TATATATACGTACTAGTACGT

ACGTACG CCCCTACGTA

TATATATACGTACTAGTACGT

ACGACTTTAGTACGTACGT

TATATATACGTACTAAAGTACGT

TATATATACGTACTAGTACGT

ACG TTTTTAAAACGTA

TATATATACGTACTAGTACGT

ACGAC GGGGAGTACGTACGT

1×10^{12} bases*

44 hours*

<1000 $

* NovaSeq 6000

High-Throughput Sequencers

Illumina MiSeq

Pacific Biosciences Sequel II

Oxford Nanopore PromethION

Oxford Nanopore MinION

Illumina NovaSeq 6000

Pacific Biosciences RS II

Oxford Nanopore SmidgION

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

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

https://nanoporetech.com/products/comparison

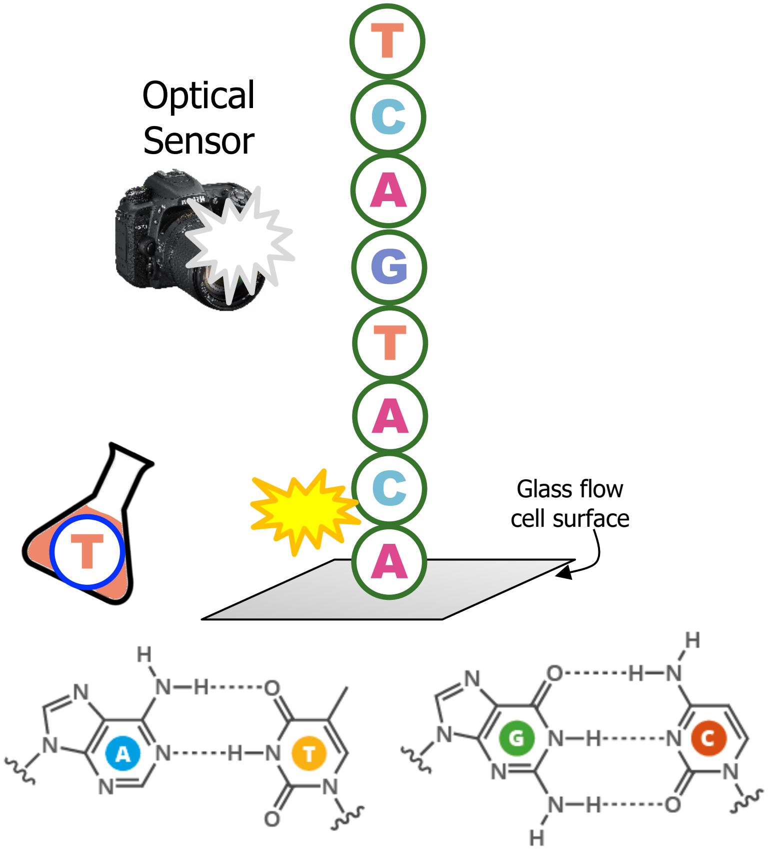
Illumina Sequencers

illumina®

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

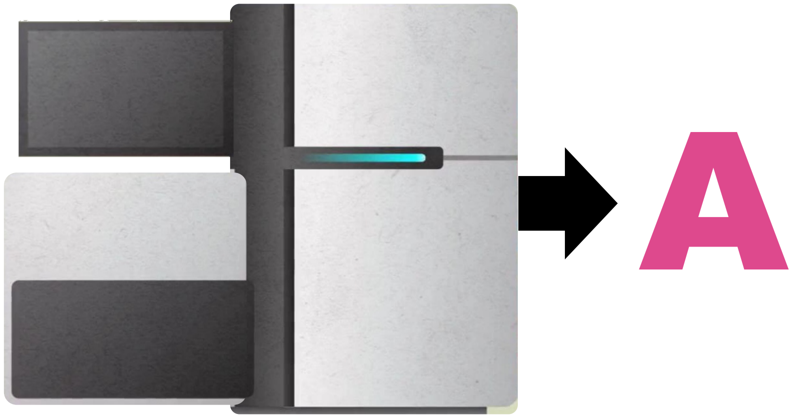
https://www.illumina.com/systems/sequencing-platforms.html

How Does Illumina Machine Work?

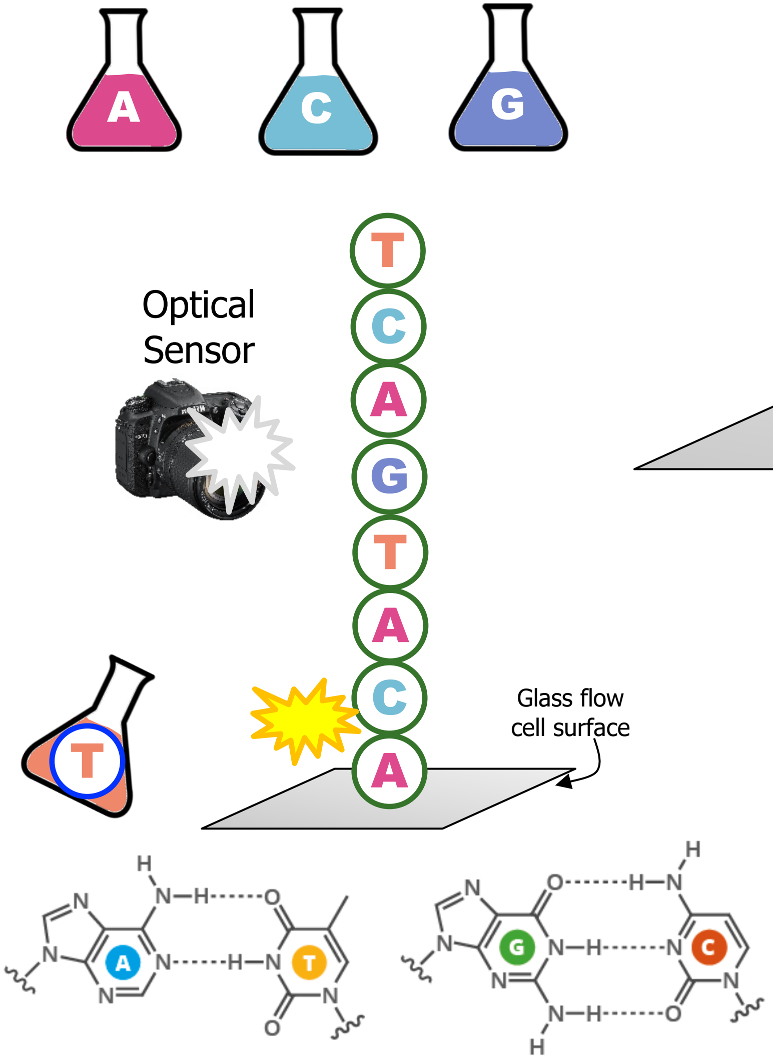


Optical Sensor

Glass flow cell surface

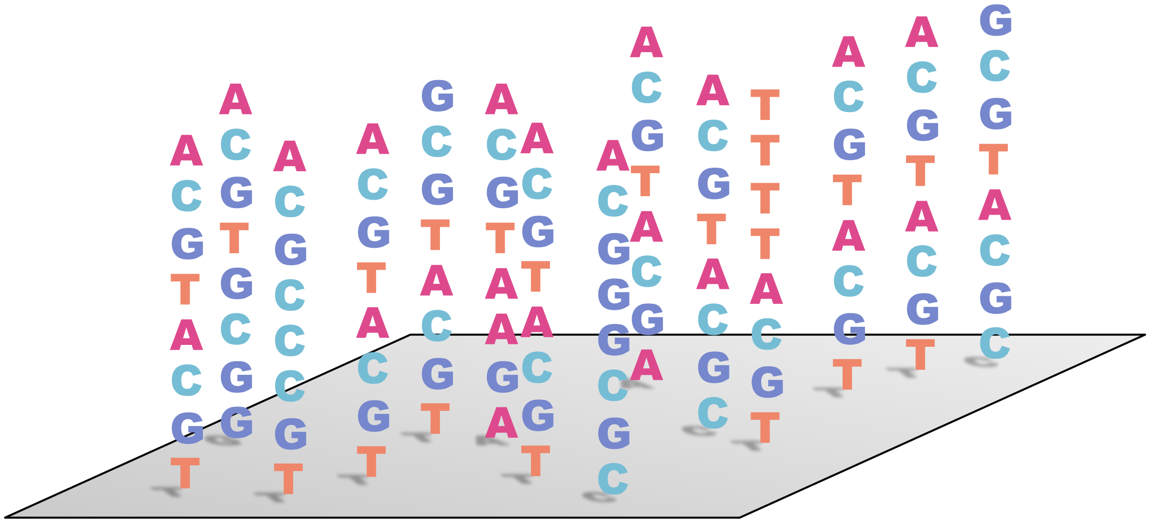


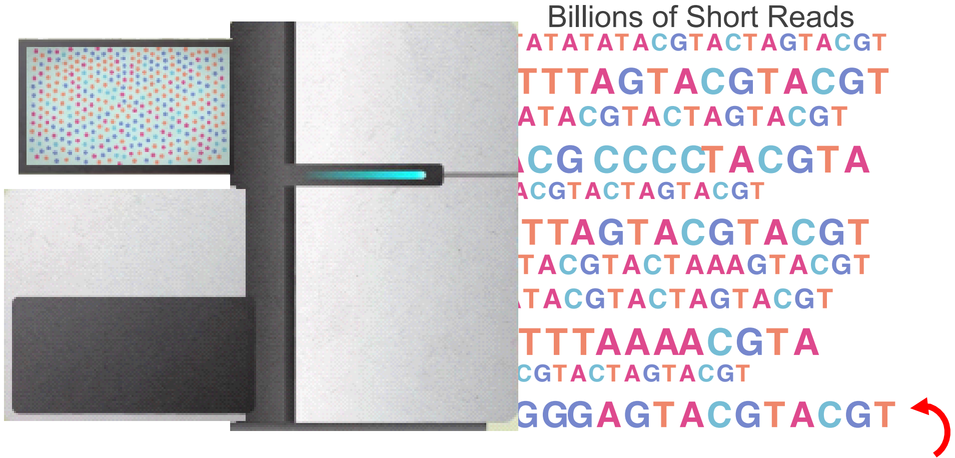
How Does Illumina Machine Work?



Optical Sensor

Glass flow cell surface





Billions of Short Reads

ATATATACGTACTAGTACGT

TTTAGTACGTACGT

ATACGTACTAGTACGT

CG CCCCTACGTA

ACGTACTAGTACGT

TTAGTACGTACGT

TACGTACTAAAGTACGT

TACGTACTAGTACGT

TTTAAAACGTA

CGTACTAGTACGT

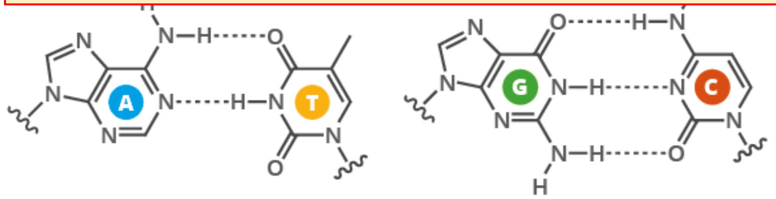
GGGAGTACGTACGT

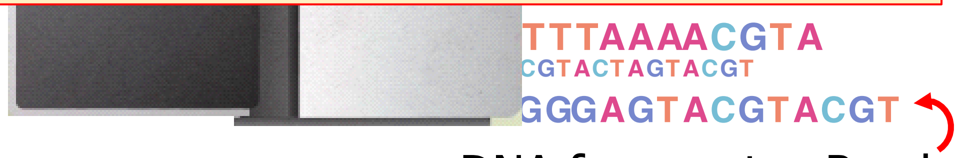
DNA fragment = Read

How Does Illumina Machine Work?

Check Illumina virtual tour:

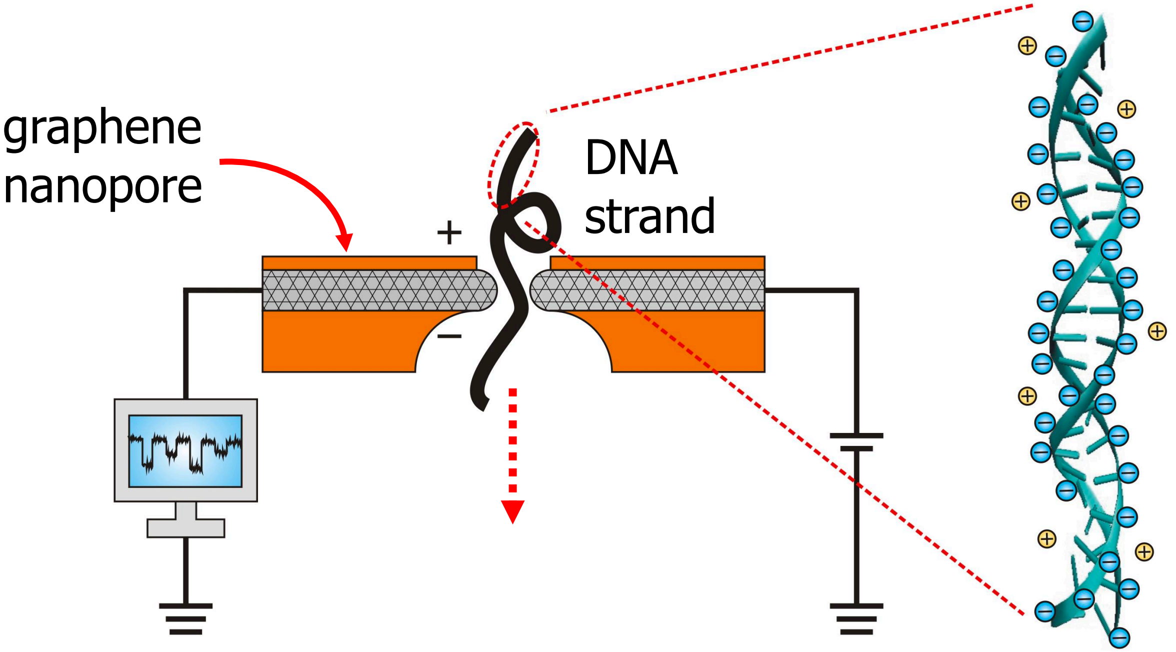
https://emea.illumina.com/systems/sequencing-platforms/iseq/tour.html





DNA fragment = Read

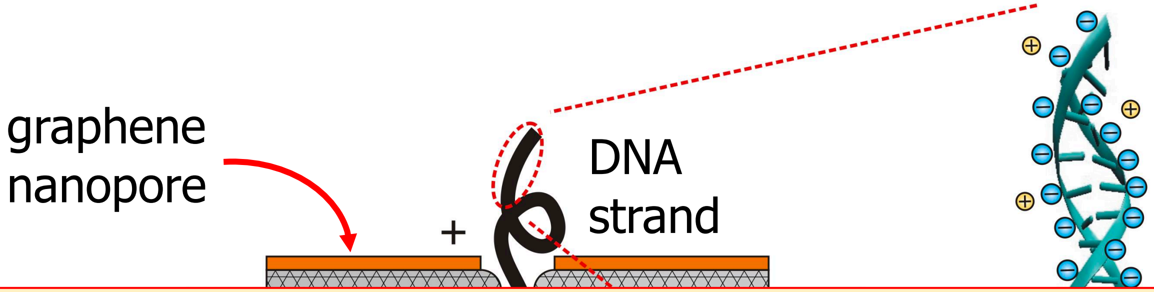
How Does Nanopore Machine Work?



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

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

How Does Nanopore Machine Work?



Check Nanopore virtual tour:

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

measures the the **change in current**

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

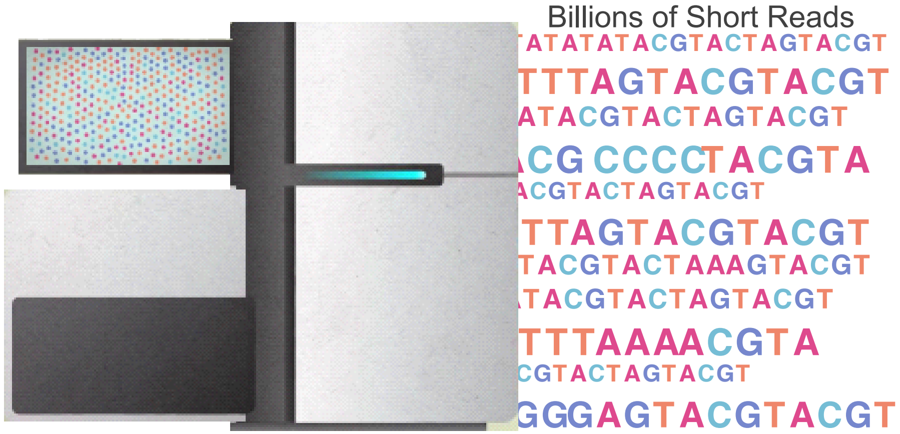
Figure is adapted from: https://phys.org/news/2013-12-gene-sequencing-future.html

Common Disadvantages!

Regardless the sequencing machine,

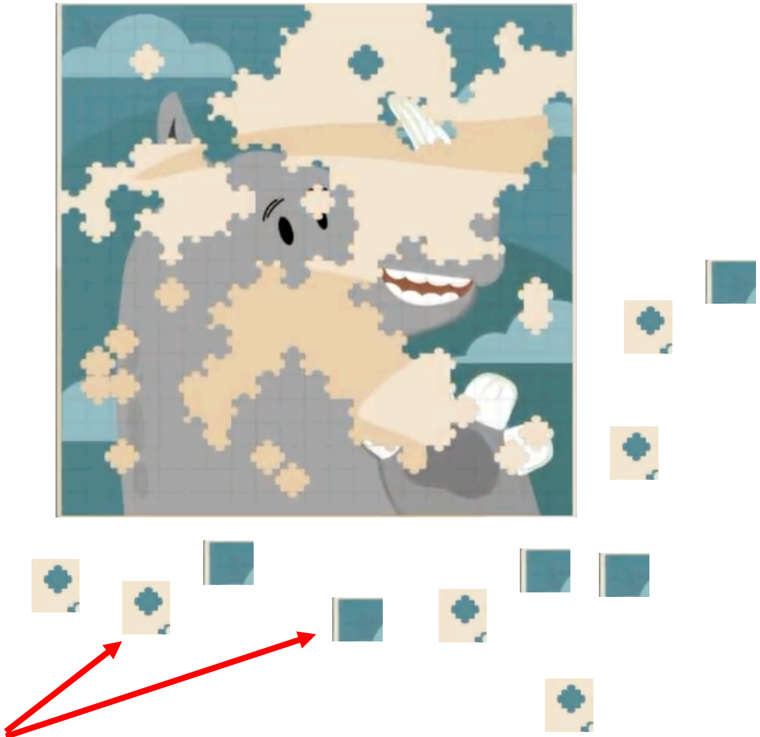
reads still lack information about their order and location

(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)

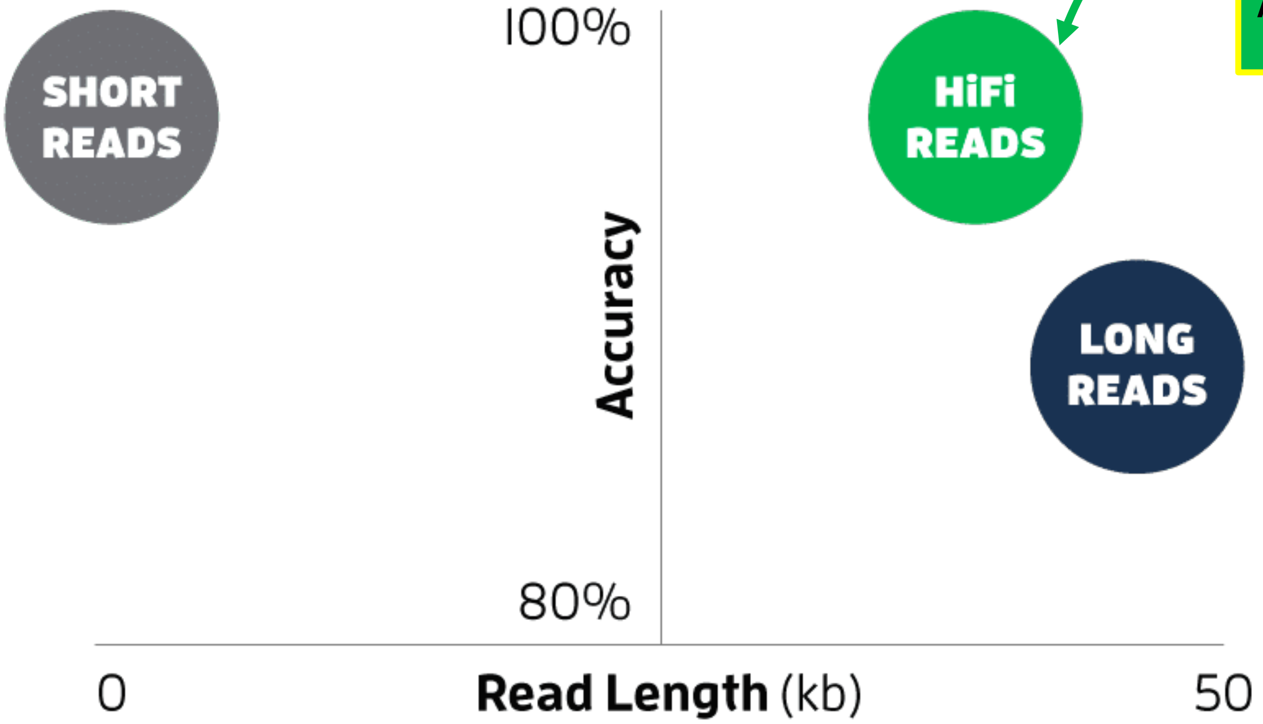
Which sequencing technology is the best?

- ❑ 100-300 bp
- ❑ low error rate (~0.1%)

- ❑ 500-2M bp
- ❑ high error rate (~15%)

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

HiFi Reads (PacBio)



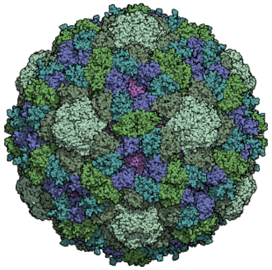
Long: 10-20 kb
Accurate: 99.8%

But still very expensive!

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

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

How Long is DNA?



Phi X174 virus
5.386 Killo bp

E. coli O157:H7
5.44 Million bp

Homo Sapiens
3.2 Billion bp

Onion, Allium Cepa
16 Billion bp

Paris Japonica
149 Billion bp

Cracking the 1st Human Genome Sequence

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

ws Print"

The New York Times

National Edition

Arizona and New Mexico: M cloudy in New Mexico, thunderst in the mountains. Partly sunny where. Highs 80 mountains, ove deserts. Weather map is on Page

No. 51,432 Copyright © 2000 The New York Times TUESDAY, JUNE 27, 2000 Printed in Arizona ONE DOLL

tic Code of Human Life Is Cracked by Scientist

A SHARED SUCC

2 Rivals' Announcem Marks New Medica Era, Risks and All

By NICHOLAS WADE

WASHINGTON, June 26 — achievement that represents a nacle of human self-knowledge rival groups of scientists said that they had deciphered the he tary script, the set of instruc that defines the human organis

The Book of Life
The 3 billion base pairs ...
... of the intertwining double helix of DNA ...
... that make up the set of chromosomes in our cells, have been sequenced.
BASE PAIRS
Rungs between the strands of the double helix
BASES
A adenine
C cytosine
G guanine
T thymine
By ordering the base units, scientists hope to locate the genes and determine their functions.
The New York Times
nature
the human genome
Science
THE HUMAN GENOME
TIME
EXCLUSIVE Q&A DR. LAURA ON THE OFFENSIVE
Cracking The Code!
CELERA

A C G T 3.2 x10^9 bases

13 years

>3x10^9 $

SAFARI

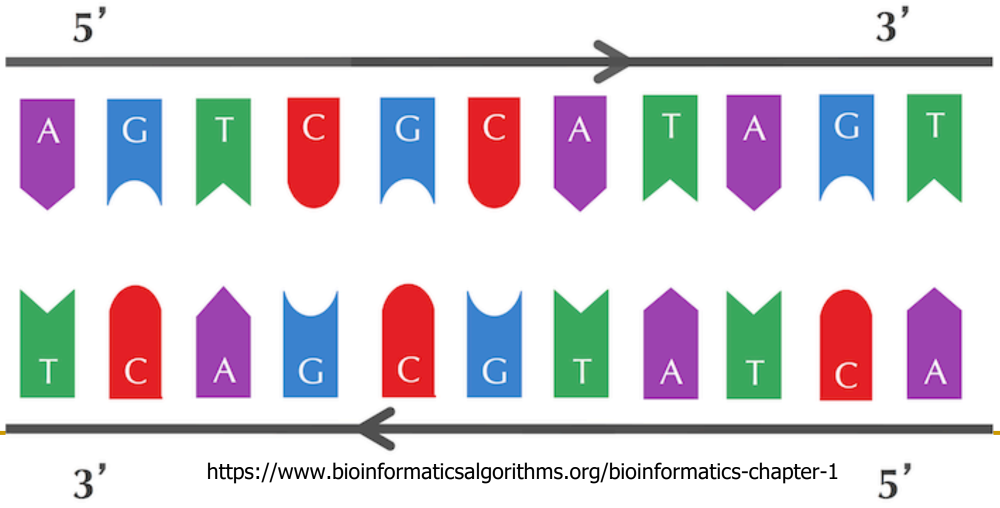
Obtaining the Human Reference Genome

- **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
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

Challenges in Read Mapping

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

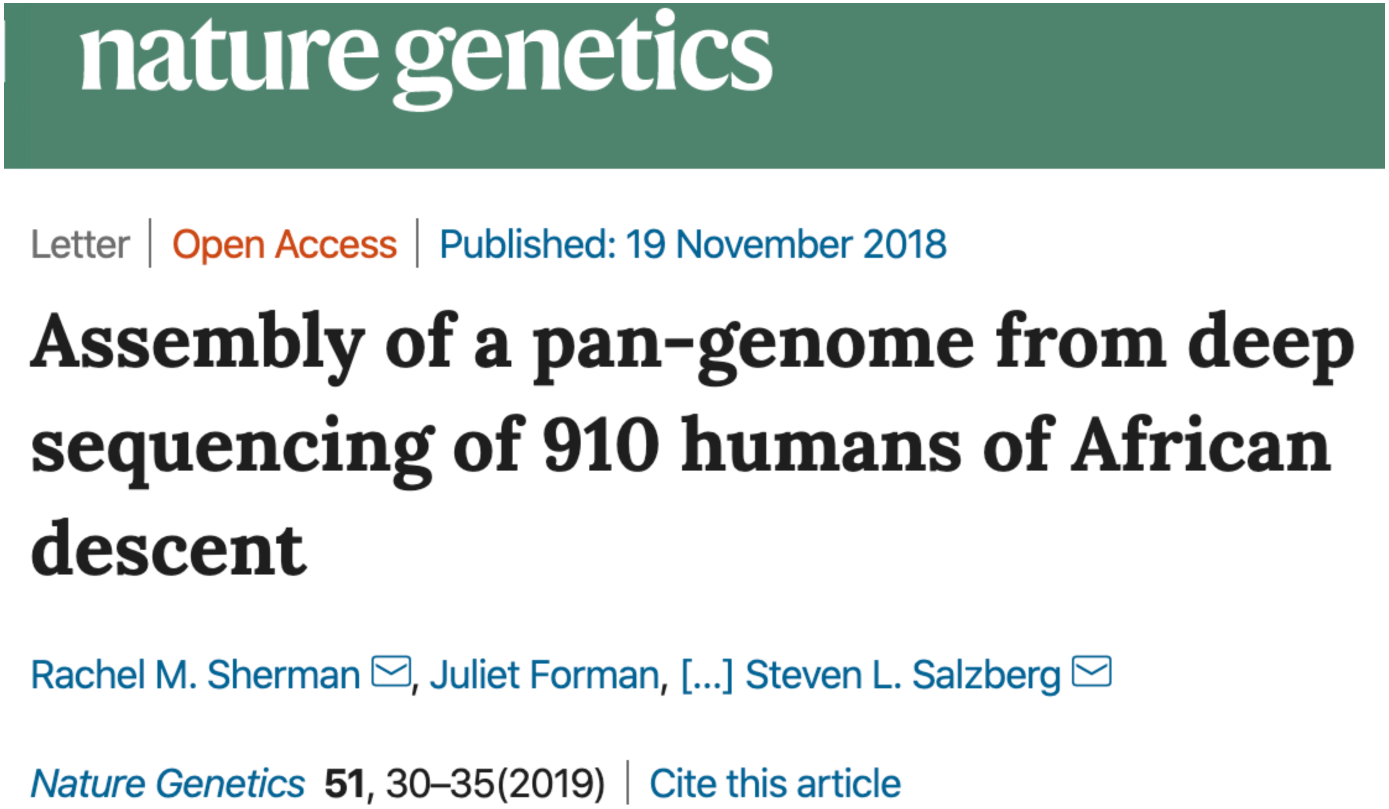


https://www.bioinformaticsalgorithms.org/bioinformatics-chapter-1

Revisiting the Puzzle

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

Reference Genome Bias



nature genetics

Letter | Open Access | Published: 19 November 2018

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

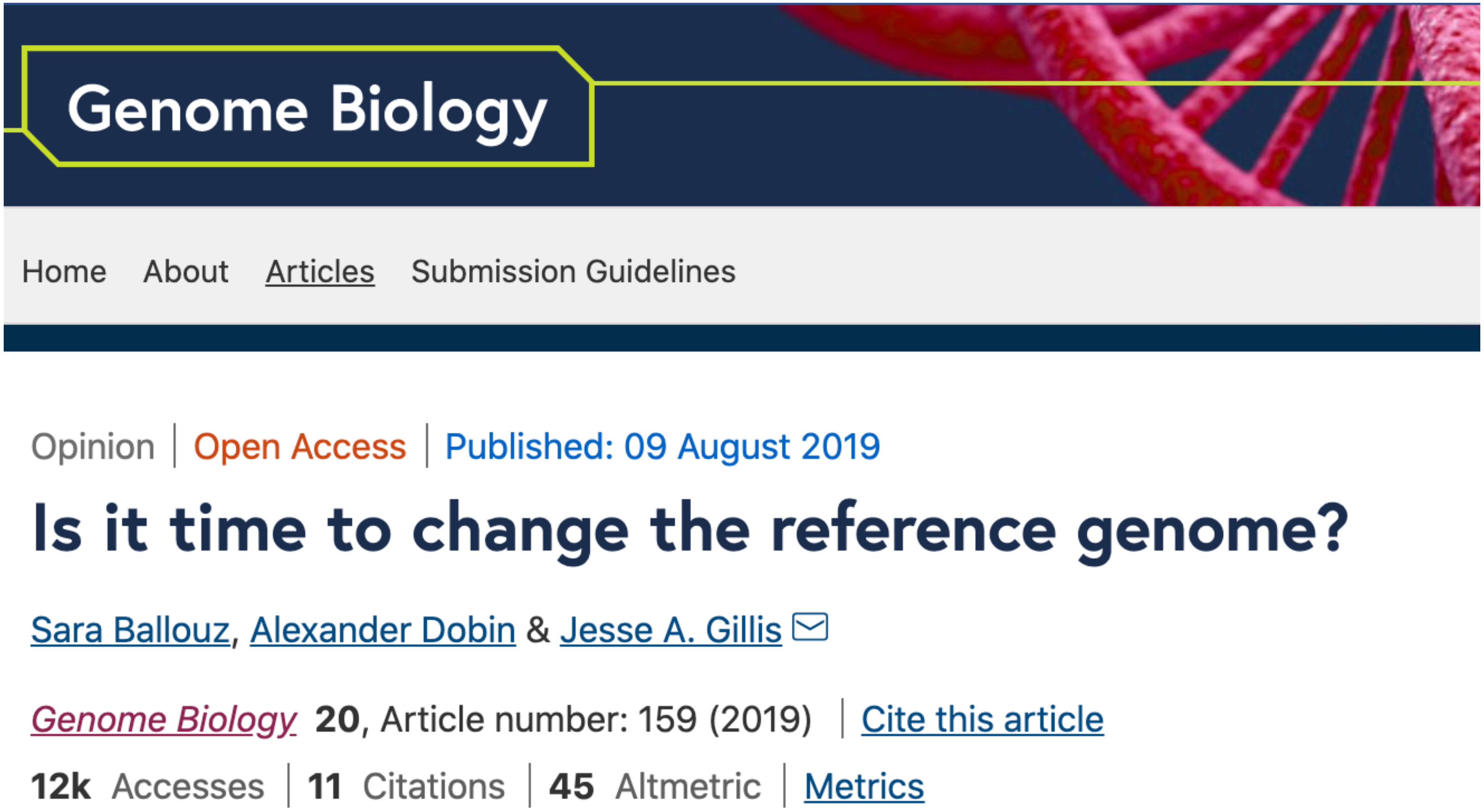
Rachel M. Sherman ✉, Juliet Forman, […] Steven L. Salzberg ✉

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

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

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

Time to Change the Reference Genome



Opinion | Open Access | Published: 09 August 2019

Is it time to change the reference genome?

Sara Ballouz, Alexander Dobin & Jesse A. Gillis

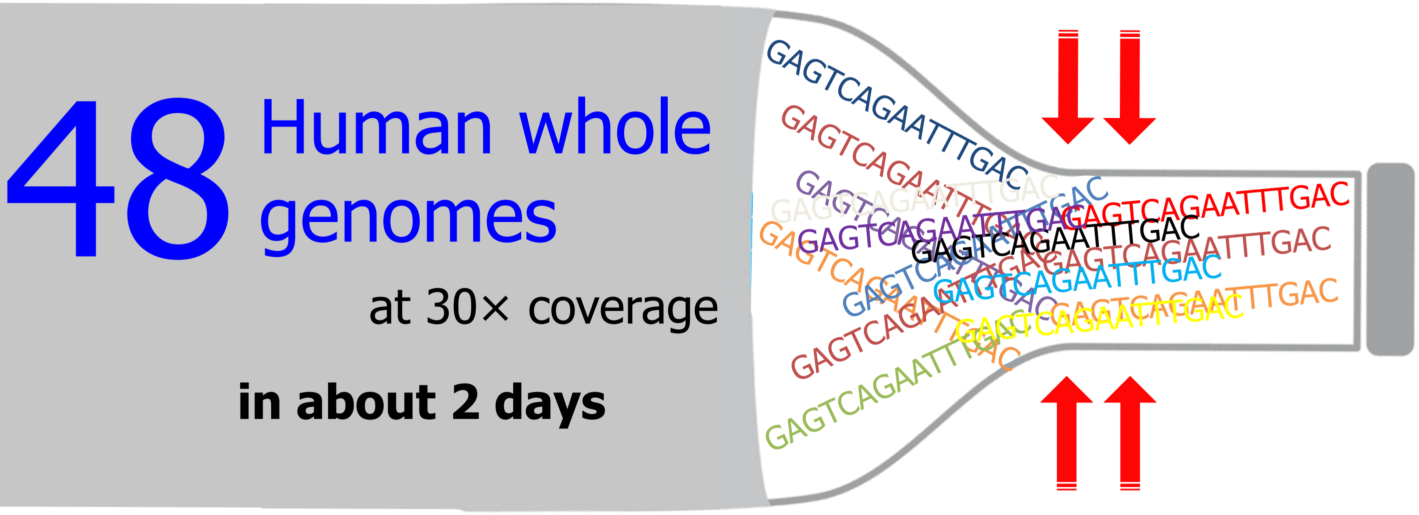
Genome Biology **20**, Article number: 159 (2019) | Cite this article

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

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

Ballouz+, "Is it time to change the reference genome?", Genome Biology, 2019

Bottlenecked in Read Mapping!!



48 Human whole genomes

at 30× coverage

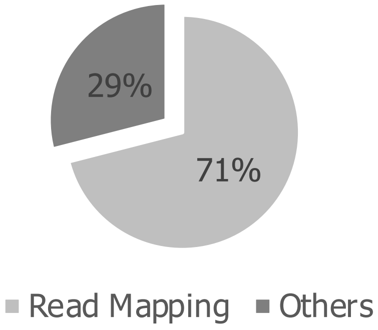
in about 2 days

Illumina NovaSeq 6000

1 Human genome

32 CPU hours

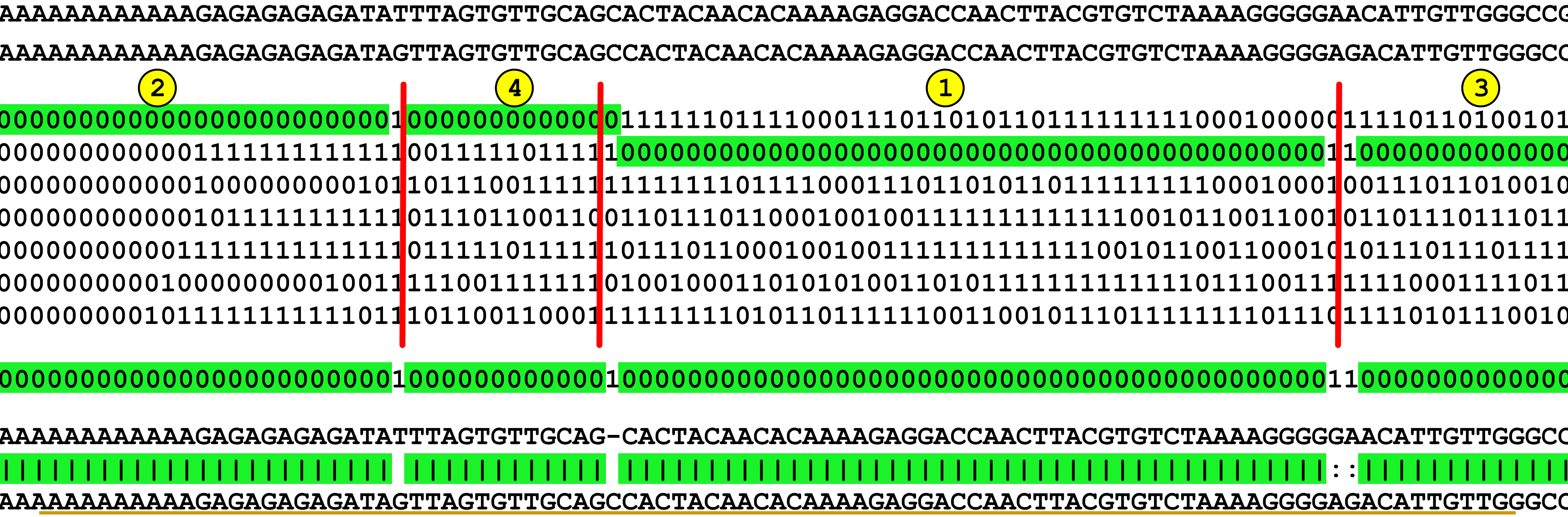
on a 48-core processor



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

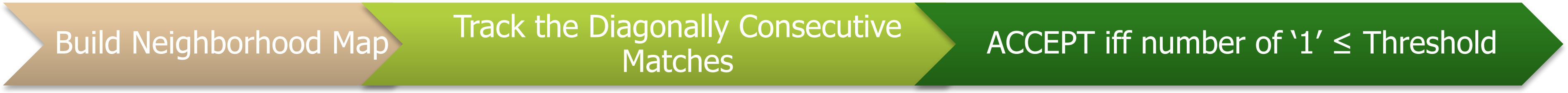
MAGNET (AACBB 2018, TIR 2017)

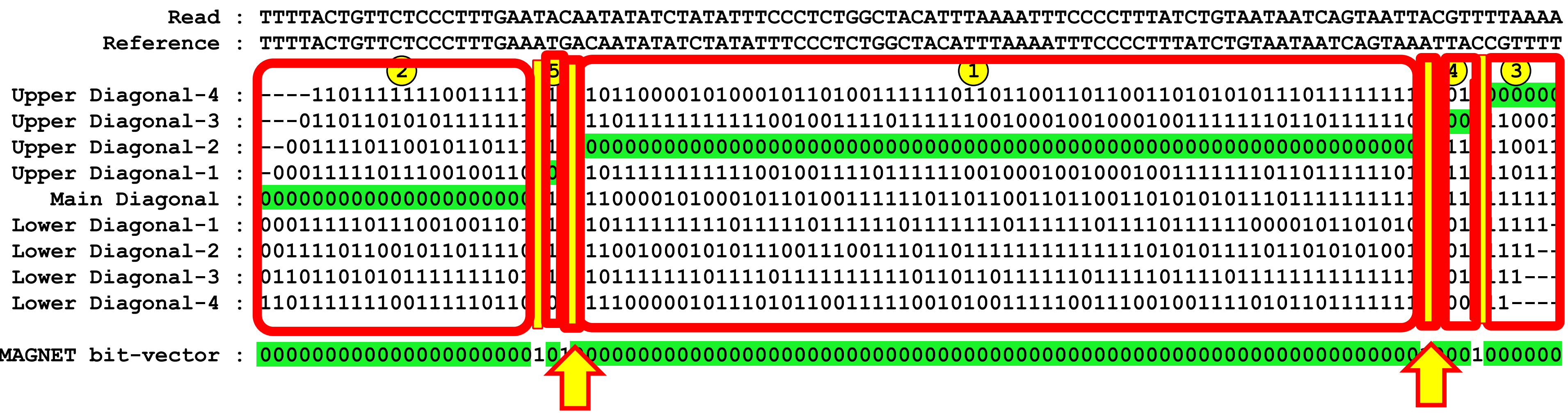
- Key observation: the use of **AND operation** to check if a zero (match) exists in a column introduces filtering inaccuracy.
- Key Idea: count the **consecutive zeros** in each mask and select the longest in a divide-and-conquer approach.
- **MAGNET** is **17x to 105x more accurate** than GateKeeper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



```
AAAAAAAAAAAAGAGAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAACATTGTTGGCCG
AAAAAAAAAAAAGAGAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGAGACATTGTTGGCC

00000000000000000000000010000000000000011111101111100011101101011011111111100010000011110110100101
00000000000011111111111110011111011111100000000000000000000000000000000000000000000110000000000000
0000000000001000000000101101110011111111111110111110001110110101101111111110001000100111011010010
00000000000010111111111110111011001100011011101100010010011111111111111001011001100101101110111011
00000000000111111111111110111110111111101110110001001001111111111111100101100110001010111011101111
00000000001000000000100111110011111111010010001101010100110101111111111110111001111110001111011
00000000010111111111110111011001100011111111101011011111111001100101110111111110111011110101110010

00000000000000000000000010000000000001000000000000000000000000000000000000000000000110000000000000

AAAAAAAAAAAAGAGAGAGAGATATTTAGTGTTGCAG-CACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAACATTGTTGGCC
|||||||||||||||||||||||| |||||||||||| |||||||||||||||||||||||||||||||||||||||||||||::|||||||||||||
AAAAAAAAAAAAGAGAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGAGACATTGTTGGCC
```

MAGNET Walkthrough


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



```
             Read : TTTTACTGTTCTCCCTTTGAATACAATATATCTATATTTCCCTCTGGCTACATTTAAAATTTCCCCTTTATCTGTAATAATCAGTAATTACGTTTTAAAA
        Reference : TTTTACTGTTCTCCCTTTGAAATGACAATATATCTATATTTCCCTCTGGCTACATTTAAAATTTCCCCTTTATCTGTAATAATCAGTAAATTACCGTTTT

 Upper Diagonal-4 : ----110111111110011111 1  1011000010100010110100111111011011001101100110101010111011111111  01 000000
 Upper Diagonal-3 : ---0110110101011111111 1  1101111111111100100111101111111001000100100010011111110110111110  00 10001
 Upper Diagonal-2 : --0011110110010110111 1  0000000000000000000000000000000000000000000000000000000000000000  11 10011
 Upper Diagonal-1 : -0001111101110010011 0  1011111111111001001111011111110010001001000100111111101101111110  11 10111
    Main Diagonal : 0000000000000000000 1  1100001010001011010011111101101100110110011010101011101111111111  11 11111
 Lower Diagonal-1 : 0001111101110010011 1  1011111111101111101111110111111011111101111011111100001011010 10  01 1111-
 Lower Diagonal-2 : 0011110110010110111 1  1100100010101110011100111011011111111111111010101110110101010 01  01 111--
 Lower Diagonal-3 : 0110110101011111111 1  1011111110111101111111111011011011111101111101110111111111111 11  01 11---
 Lower Diagonal-4 : 1101111111001111101 0  1110000010111010110011111001010011111001110010011110101101111 11  00 1----

MAGNET bit-vector : 0000000000000000000000101000000000000000000000000000000000000000000000000000000000000000001000000
```

Find the longest segment of consecutive zeros

Exclude the errors from the search space

Divide the problem into two subproblems and repeat

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

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

AirLift

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

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

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

Clustering the Reference Genome Regions

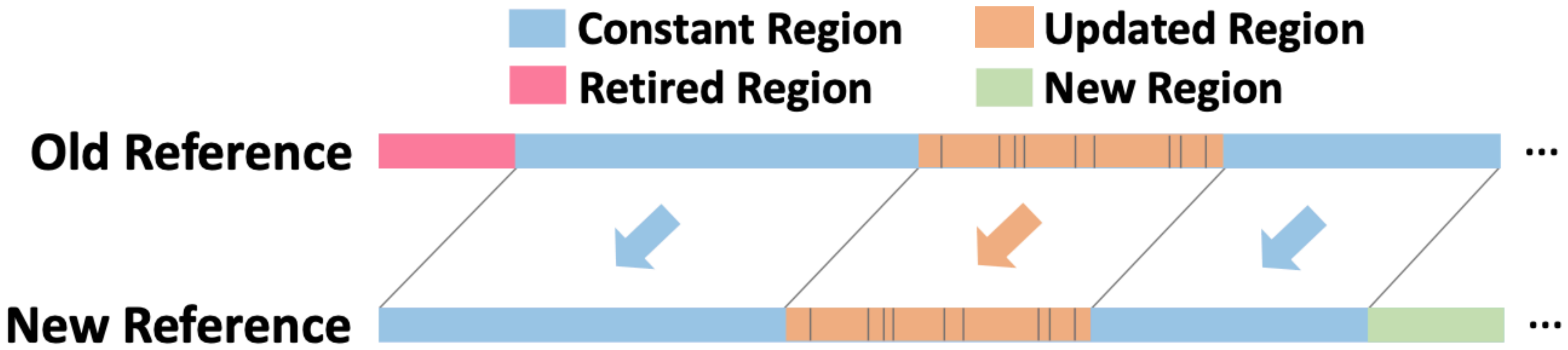
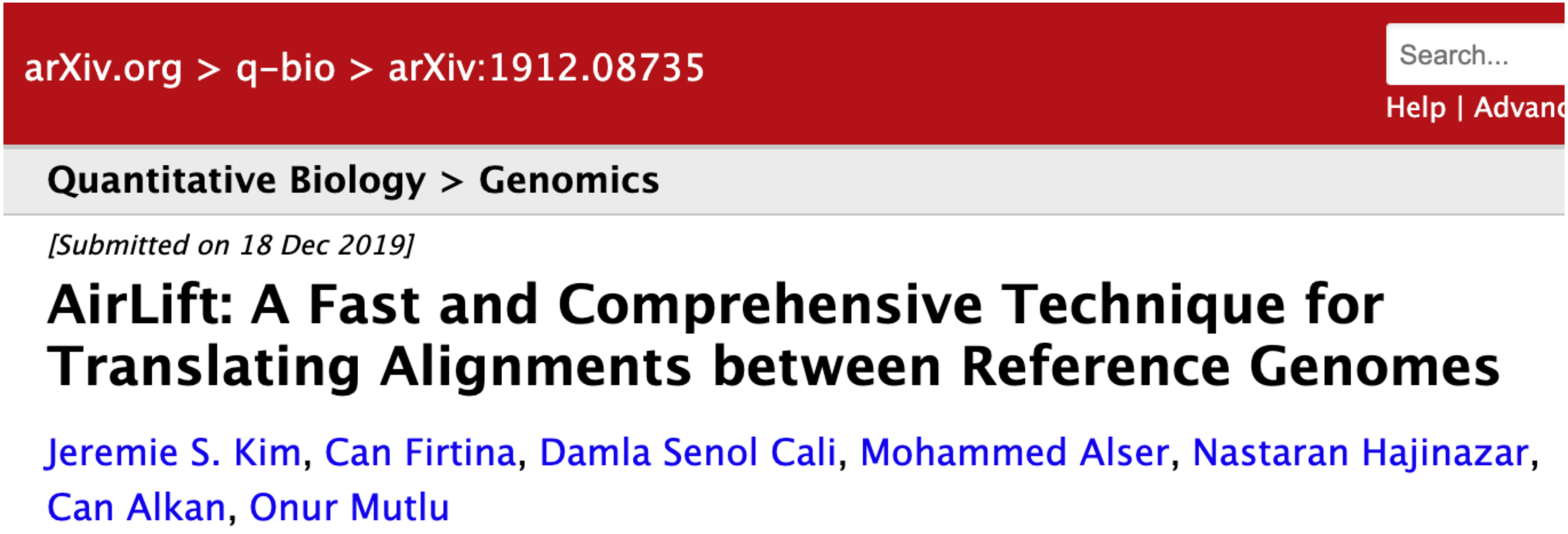


Fig. 2. Reference Genome Regions.

More Details on AirLift



arXiv.org > q-bio > arXiv:1912.08735

Quantitative Biology > Genomics

[Submitted on 18 Dec 2019]

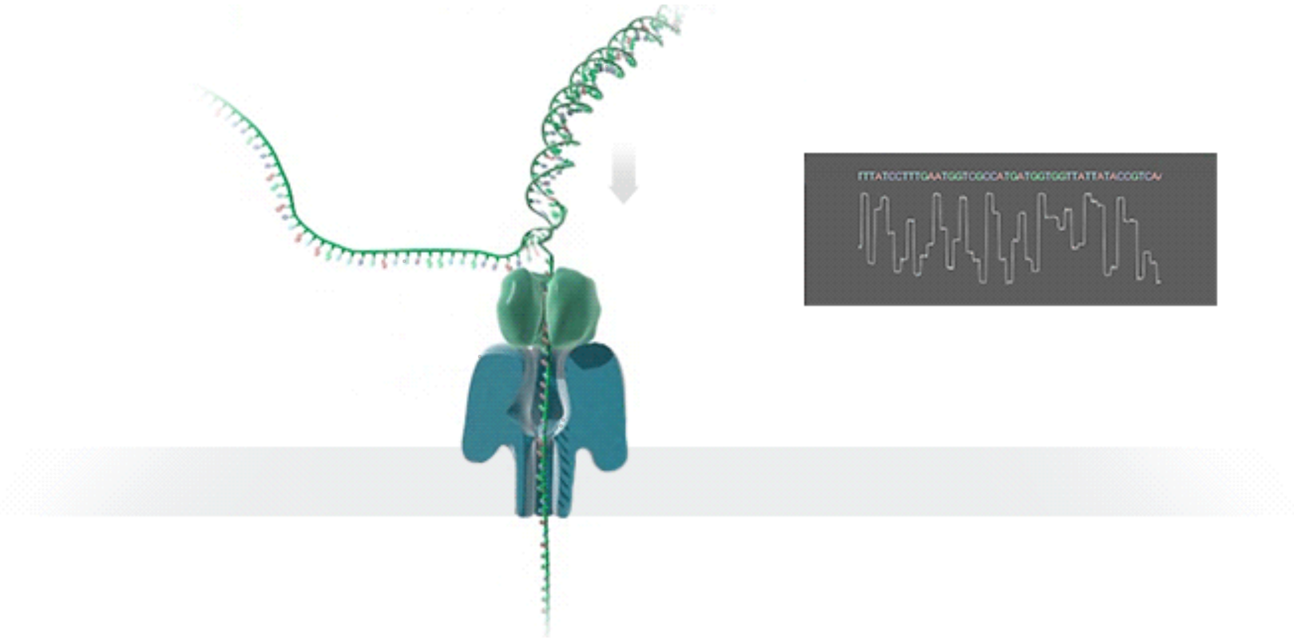
AirLift: A Fast and Comprehensive Technique for Translating Alignments between Reference Genomes

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

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

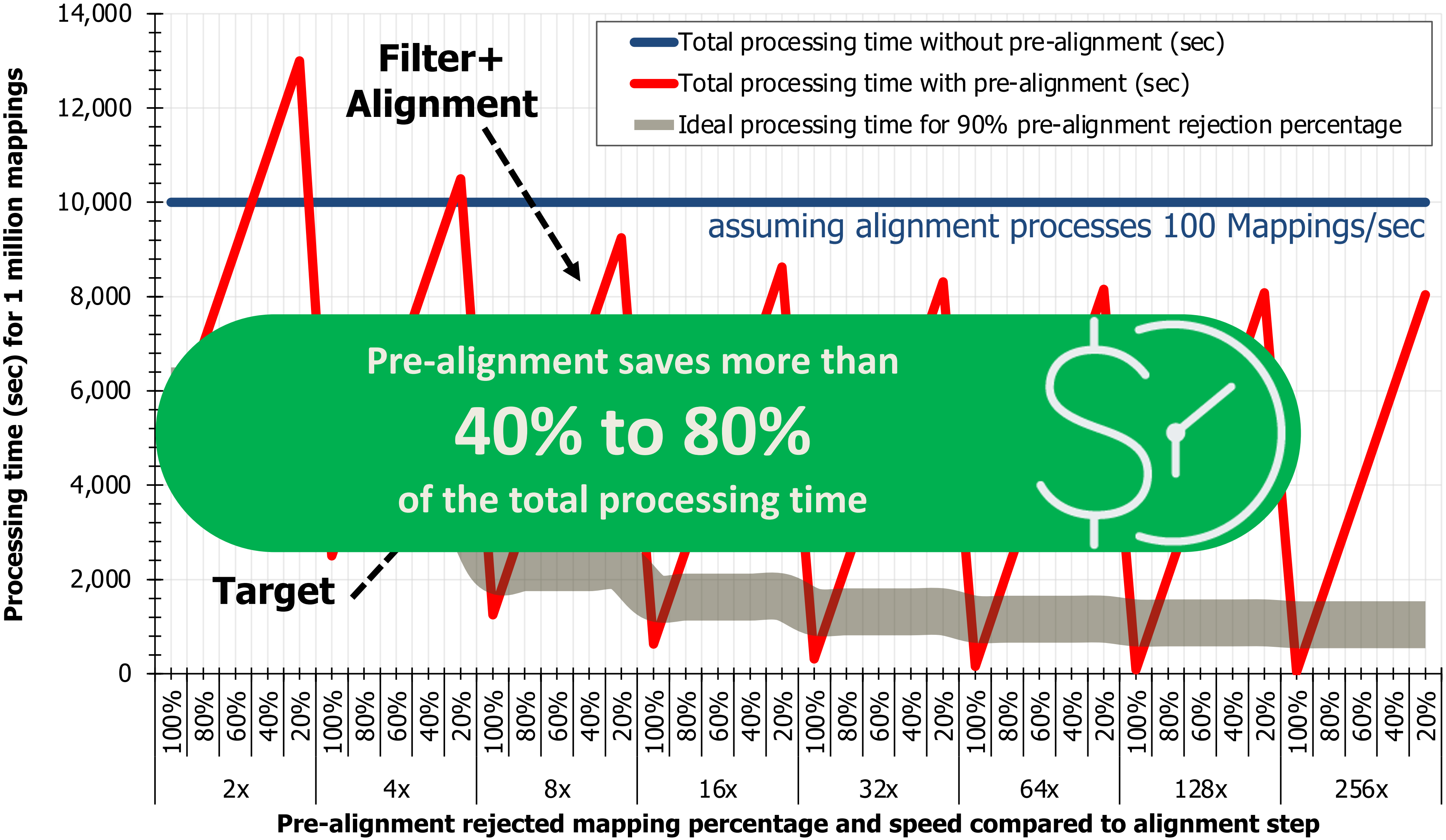
Kim+, "AirLift: A Fast and Comprehensive Technique for Translating Alignments between Reference Genomes", arXiv, 2020

Nanopore Sequencing



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

The Effect of Pre-Alignment (Theoretically)



Pre-alignment saves more than

40% to 80%

of the total processing time