

# Seminar in Computer Architecture Meeting 2: GateKeeper

Dr. Mohammed Alser

 @mealser

ETH Zurich

Fall 2021

30 September 2021

# Example Paper Presentation I



# Let's Review This Paper [Alser+, Bioinformatics 2017]

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](#)]

Bioinformatics



Article Navigation

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

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

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

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

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

# 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, 2017**

Presented by: Mohammed Alser



**Bilkent University**



TOBB  
UNIVERSITY OF  
ECONOMICS & TECHNOLOGY

**ETH** zürich **Carnegie Mellon**

# Executive Summary

- **Problem:** Genomic similarity measurement is a computational bottleneck. Examining the similarity of **highly-dissimilar genomic** sequences consumes an overwhelming majority of a modern read mapper's execution time.
- **Goal:** Develop a fast and effective *filter* that can detect highly-dissimilar genomic sequences and eliminate them *before* invoking computationally costly alignment algorithms.
- **Key observation:** If two strings differ by  $E$  edits, then every pairwise match can be aligned in at most  $2E$  shifts.
- **Key ideas:**
  - Quickly find similar sequences using *Hamming Distance*.
  - Compute “*Shifted Hamming Distance*” for the rest of sequence pairs: ANDing  $2E+1$  Hamming vectors of two strings, to identify dissimilar sequences.
  - Use only bit-parallel operations that nicely map to:
    - SIMD instructions, FPGA, Logic layer of the 3D-stacked memory, and In-memory accelerators (e.g., Ambit)
- **Key results:**
  - Provides a huge speedup of up to **130x** compared to the previous state of the art software solution.

# Background, Problem, & Goal

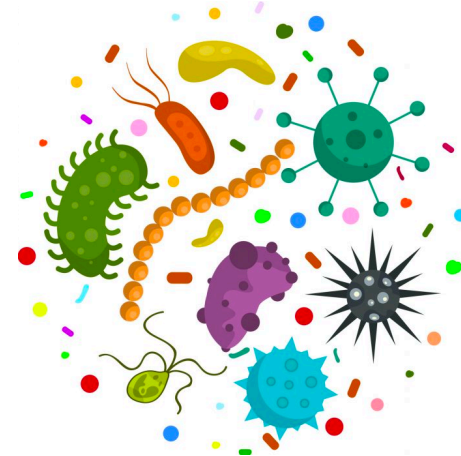


# Applications of Genome Analysis

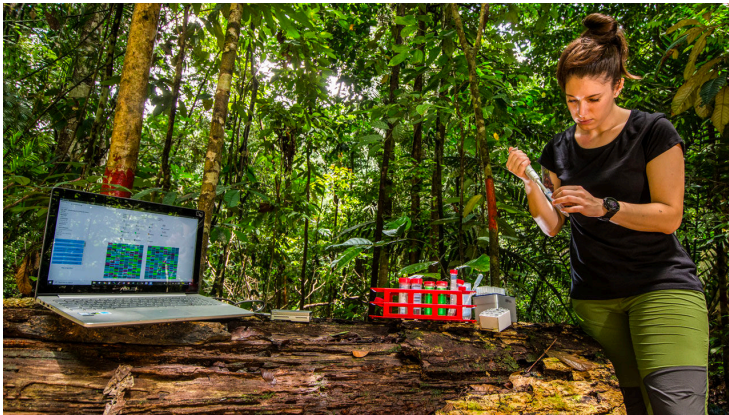
---



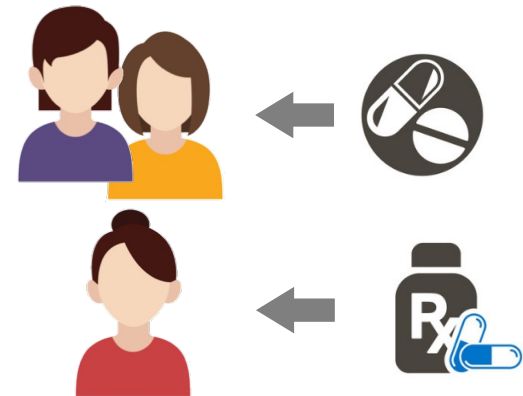
Understanding **genetic variations**



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



Rapid surveillance of **disease outbreaks**



Developing **personalized medicine**

**And many other applications ...**

# How to Analyze a Genome?

---

**NO**

machine gives the **complete sequence** of genome as output



```
>CCTCCTCAGTGCCACCCAGCCCACTGGCAGCTCCCAAACAGGCTCTTATTAAACACCCTGTTCCCTGCCCTTGGAGTGAGGTGTCAAG
GACCTAAACTAAAAAAAAAAAAAAAAAGAAAAAGAAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTCTT
CATGTCAAGGACCTAATGTGCTAAACAGCACTTTTTTGACCATTATTTTGGATCTGAAAGAAATCAAGAATAAATGAAGGACTTGATACATTG
GAAGAGGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAAAAAGAAAAAGAAAAAGAAAAAGAATTAAATTTAAGTAATTCTTTGAAAAAA
ACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCTGTGTTGCAGGTCTTCTTGCATTTCCCTGTCAAAAGAAAAAGAATTTAAATTT
AAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCAAGGCCAAGAGTTGCAAAAAAAAAAAAAAGAAAA
GAAAAGAAAAAGAATTTAAATTTAAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTAGCCAGAATGG
TTGTGGGATGGGAGCCTCTGTGGACCGACCAGGTAGCTCTCTTTCCACACTGTAGTCTCAAAGCTTCTTCATGTGGTTTTCTCTGAGTGAAA
AAAAAAAAAAGAAAAAGAAAAAGAAAAAGAATTTAAATTTAAGTAATTCTTTGAAAAAACTAATTTCTAAGCTTTTCATGTCAAGGACC
TAATGTAGCTATACTGAACGTTATCTAGGGGAAAGATTGAAGGGGAGCTCTAAGGTCAACACACCACCACTTCCCAGAAAGCTTCTTCA.....
```



# DNA Testing



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

Order now

The promotion ends today in 12 more hours!



<https://www.myheritage.ch/dna>



# DNA Testing



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



<https://www.myheritage.ch/dna>  
<https://www.23andme.com/>



Health + Ancestry  
Service

**\$199**

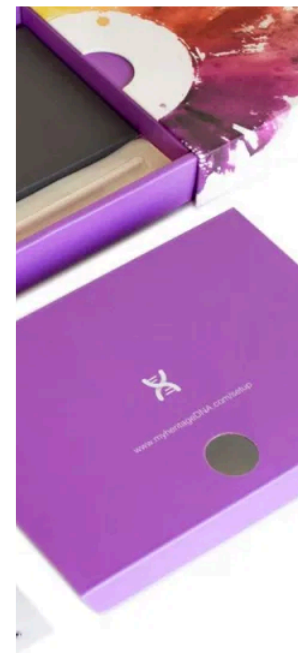
- Includes everything in Ancestry + Traits Service

*PLUS*

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

now

only in 12 more hours!



# High-Throughput Sequencers



Illumina MiSeq



Pacific  
Biosciences  
Sequel II

Oxford  
Nanopore  
PromethION



Illumina NovaSeq 6000



Pacific Biosciences RS II



Oxford Nanopore MinION



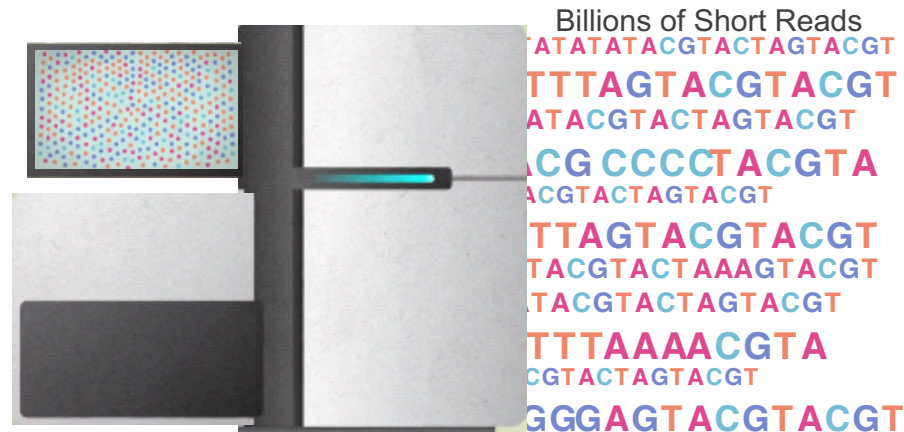
Oxford  
Nanopore  
SmidgION

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

# Genome Sequencer is a Chopper

---

Regardless the sequencing machine,  
reads still lack information about their order and location  
(which part of genome they are originated from)



# Solving the Puzzle

---

.FASTA file



Reference  
genome



.FASTQ file



Reads



# Reference Genome

---

## .FASTA file:

```
>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCTCTTTTCTTATCATTGACATTTAAACTCTGGGGCAGGTCCTCGCGTAGAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGGAACCGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACCGCCCGGGCTCCGGCCCCGGCCCCGGCTCGGGGCCCCGCGGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCGCCCCAAGTGGCCCCGGGGCTTGATTTTTTGCTTTTAAAAG
GAGGCATACAAAGATGGAAGCGAGTTACTGAGGGAGGGATAGGAAGGGGGGTGGAGGAGGGACTTGTCTT
TGCCGAGTGTGCTCTTCTGCAAAAGTAGCAAAATGTTCCACTCCTAAGAGTGGACTTCCAGTCCGGCCCT
GAGCTGGGAGTAGGGGGCGGGAGTCTGCTGCTGCTGTCTGCTAAAGCCACTCGCGACCGCGAAAAATGCA
GGAGGTGGGGACGCACTTTGCATCCAGACCTCCTCTGCATCGCAGTTCACGACATCCACGCTTGGGAAAG
TCCGTACCCGCGCCTGGAGCGCTTAAAGACACCCTGCCGCGGGTCGGGCGAGGTGCAGCAGAAGTTTCCC
GCGGTTGCAAAGTGCAGATGGCTGGACCGCAACAAAGTCTAGAGATGGGGTTCGTTTCTCAGAAAGACGC
```

# 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](https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39)

[illegible]

# Genomic Reads

---

.FASTQ file:

Identifier	@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
Sequence	TTAATTGGTAAATAAATCTCCTAATAGCTTAGATNTTACCTTNNNNNNNNNNNTAGTTTCTTGAGA
'+' sign	+
Quality scores	efcfffffcfeeffffcfffffdddf`feed]`]_Ba_^__[YBBBBBBBBBBRTT\]]][[] dddd`

Base T  
phred Quality ] = 29

# Obtaining .FASTQ Files

- <https://www.ncbi.nlm.nih.gov/sra/ERR240727>



Full ▾

Send to: ▾

**[ERX215261](#): Whole Genome Sequencing of human TSI NA20754**

1 ILLUMINA (Illumina HiSeq 2000) run: 4.1M spots, 818.7M bases, 387.2Mb downloads

**Design:** Illumina sequencing of library 6511095, constructed from sample accession SRS001721 for study accession SRP000540. This is part of an Illumina multiplexed sequencing run (9340\_1). This submission includes reads tagged with the sequence TTAGGCAT.

**Submitted by:** The Wellcome Trust Sanger Institute (SC)

**Study:** Whole genome sequencing of (TSI) Toscani in Italia HapMap population

[PRJNA33847](#) • [SRP000540](#) • [All experiments](#) • [All runs](#)

**Sample:** Coriell GM20754

[SAMN00001273](#) • SRS001721 • [All experiments](#) • [All runs](#)

*Organism:* [Homo sapiens](#)

**Library:**

*Name:* 6511095

*Instrument:* Illumina HiSeq 2000

*Strategy:* WGS

*Source:* GENOMIC

*Selection:* RANDOM

*Layout:* PAIRED

*Construction protocol:* Standard

**Runs:** 1 run, 4.1M spots, 818.7M bases, [387.2Mb](#)

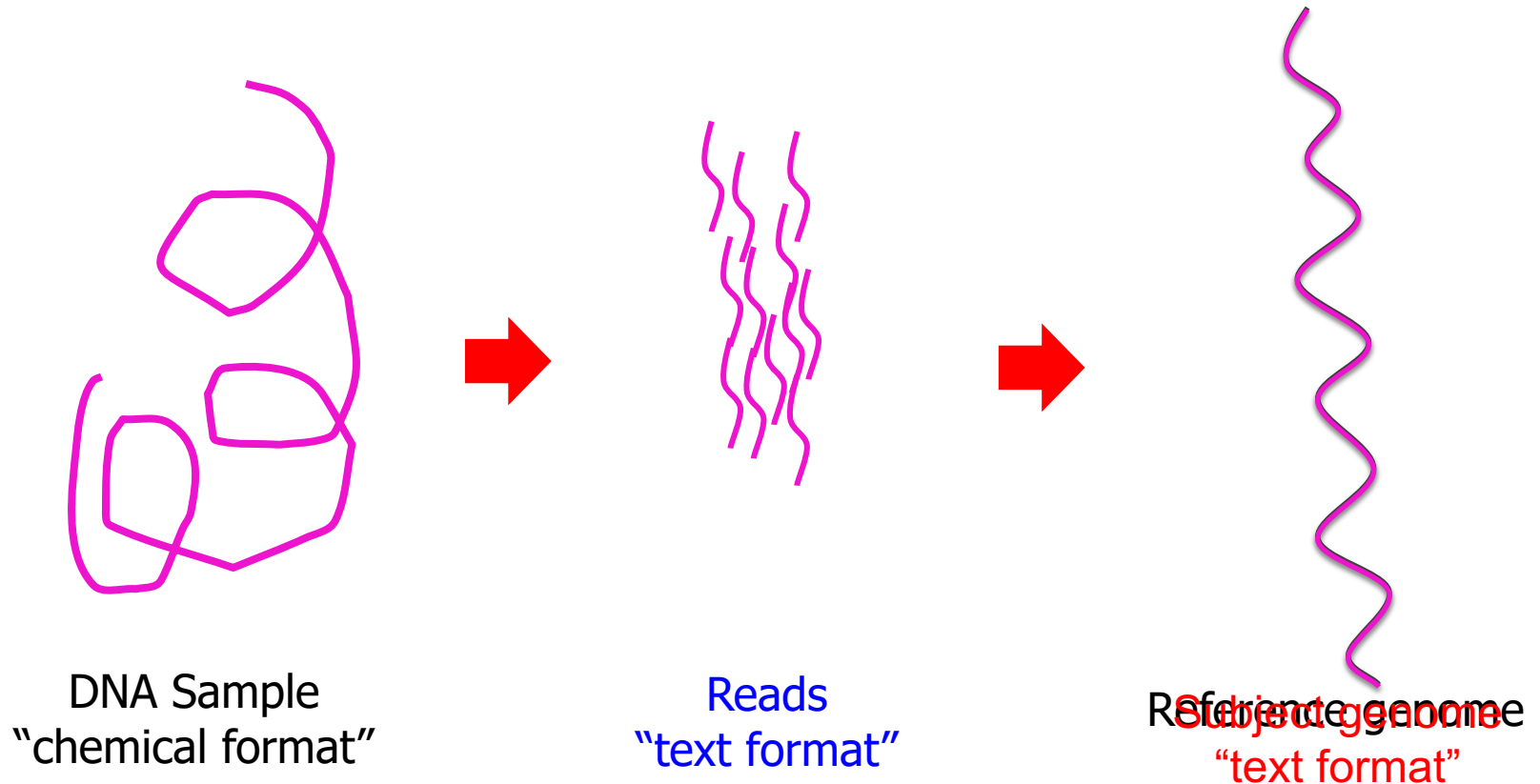
Run	# of Spots	# of Bases	Size	Published
<a href="#">ERR240727</a>	4,093,747	818.7M	387.2Mb	2013-03-22



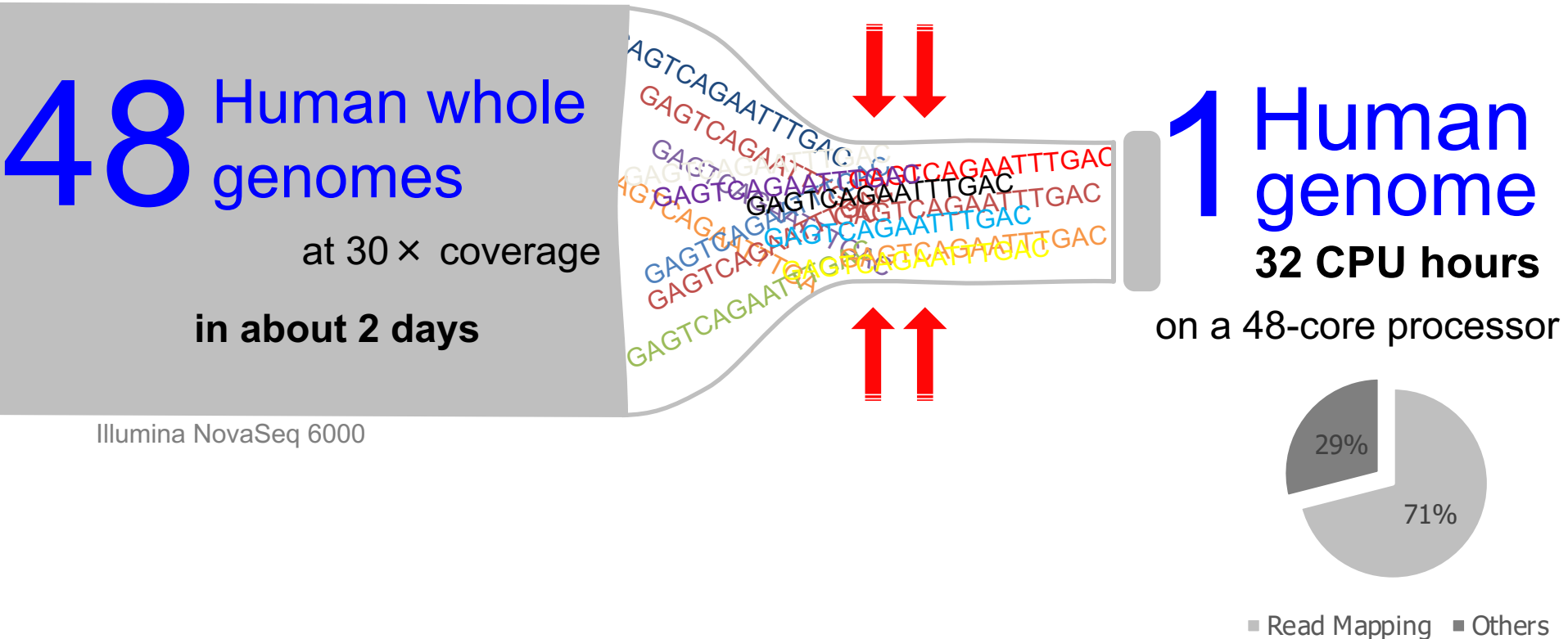
# Read Mapping

---

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



# Analysis is Bottlenecked in Read Mapping!!



---

What makes  
read mapping  
a **bottleneck**?

---

Let's first **learn**  
how to **map** a read

# Matching Each Read with Reference Genome

---

## .FASTA file:

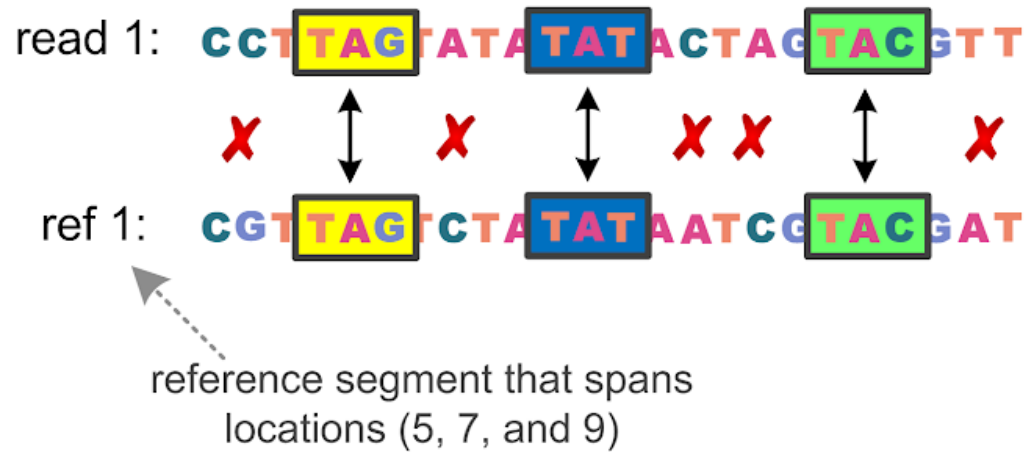
```
>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCC[redacted]TCATTGACATTTAAACTCTGGGGCAGG[redacted]GAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGGAACCGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACCGCC[redacted]CCCCGGCCCCGGCTCGGGGCCCCGCGGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCCGCCCAAGTGGCCCCGGGGCTTGATTTTTGCTTTTAAAAG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

## .FASTQ file:

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

# Base-by-Base Comparison

---



# Sequence Alignment (Verification)

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

organization x operation

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

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

Edit distance = 7

match
deletion
insertion
mismatch

organization x translation

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

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

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

Edit distance = 4

---

What makes  
read mapping  
a **bottleneck**?



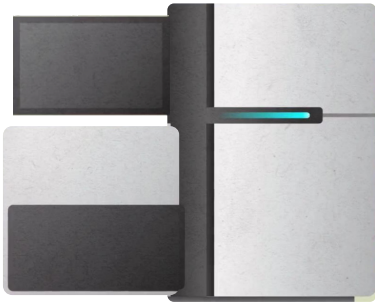
# A Tsunami of Sequencing Data

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

Source:  
[@kyrpides](#)

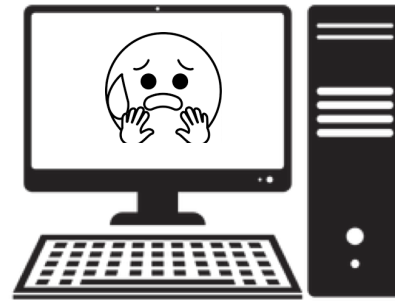
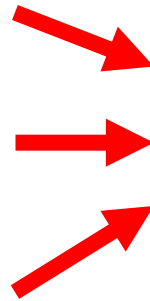
# Lack of Specialized Compute Capability

---



**Specialized** Machine  
for Sequencing

**FAST**

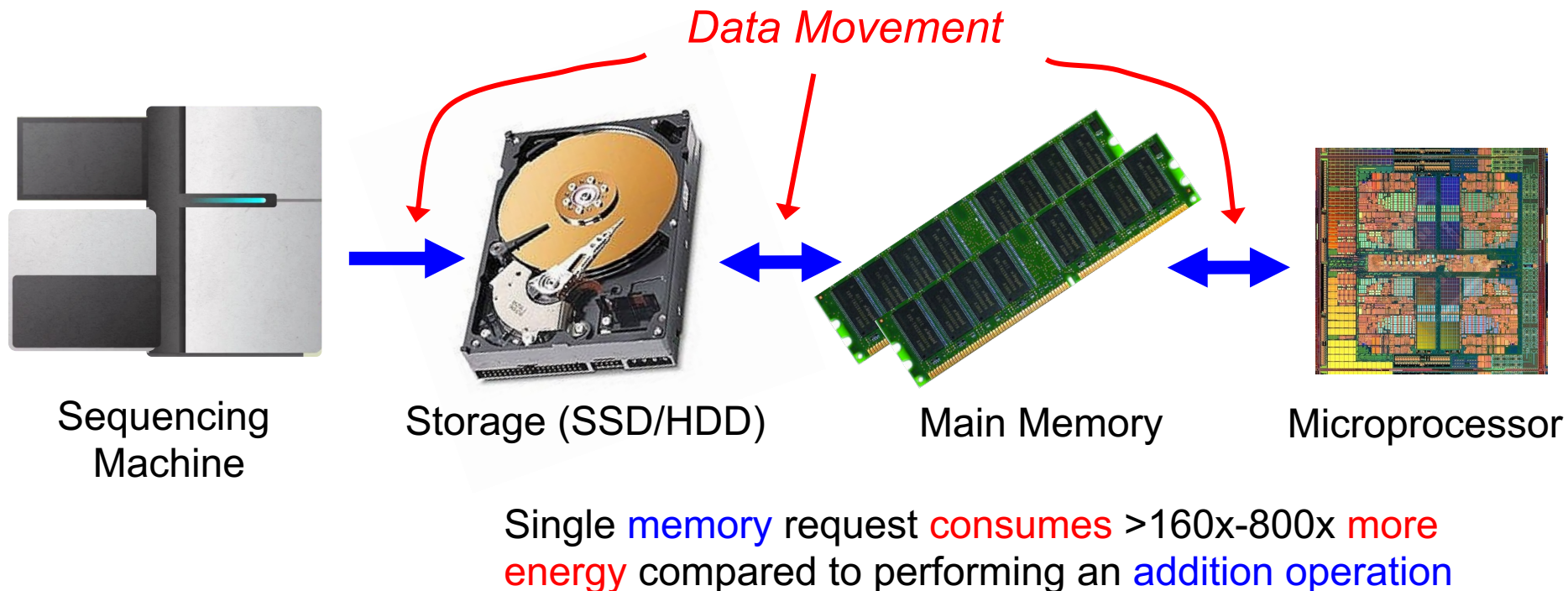


**General-Purpose** Machine  
for Analysis

**SLOW**

# Data Movement Dominates Performance

- **Data movement** dominates performance and is a **major** system **energy bottleneck** (accounting for 40%-62%)



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

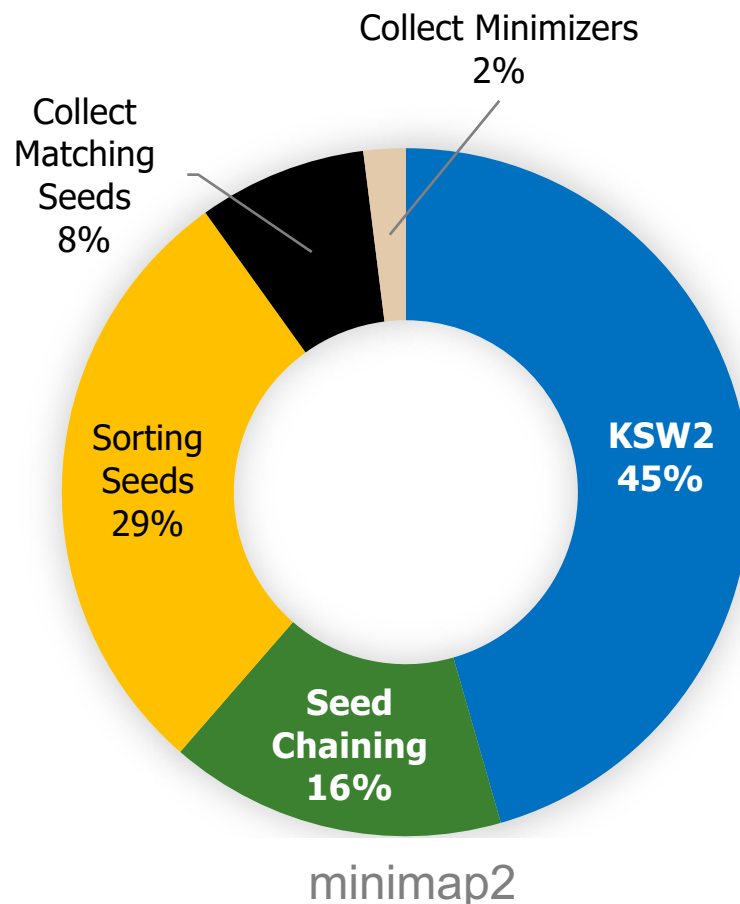
★ Kestor et al., "Quantifying the Energy Cost of Data Movement in Scientific Applications," IISWC 2013

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

# Read Mapping Execution Time

**> 60%**

**of the read mapper's  
execution time is spent in  
sequence alignment**



ONT FASTQ size: 103MB (151 reads), Mean length: 356,403 bp, std: 173,168 bp, longest length: 817,917 bp

# Why Alignment is Computationally Costly?

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

Enumerating all possible prefixes

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

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

The diagram illustrates the alignment of 'NETHERLANDS' (top row) and 'SWITZERLAND' (left column) using a dynamic programming table. The table contains numerical values representing the edit distance. Red arrows trace the optimal alignment path from the bottom-right cell (D, 11) to the top-left cell (S, 1). The path follows the sequence of matches: D (11) to N (10), N (10) to W (9), W (9) to I (8), I (8) to T (7), T (7) to Z (6), Z (6) to E (5), E (5) to R (4), R (4) to L (3), L (3) to A (2), A (2) to N (1), and N (1) to S (0). The 'etc' labels indicate the continuation of the sequence for the remaining characters in the words.

# Why Alignment is Computationally Costly?

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

Enumerating all possible prefixes

- **Data dependencies** limit the computation parallelism

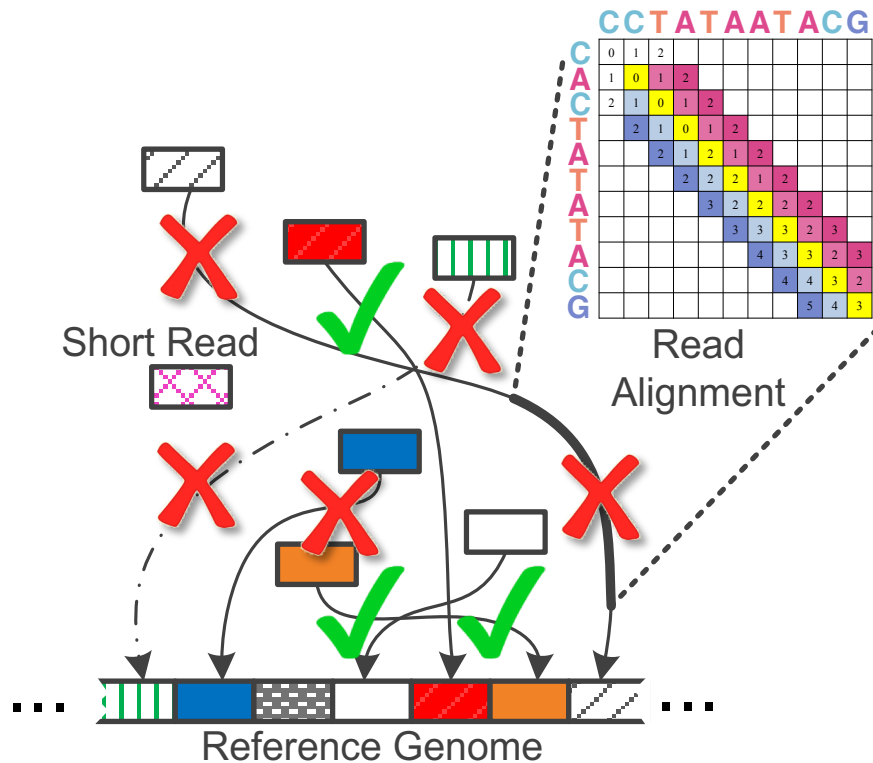
Processing row (or column) after another

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

Number of differences is computed only at the backtracking step.

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

# Large Search Space for Mapping Location



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

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

---

We need intelligent algorithms  
and intelligent architectures  
that handle data well

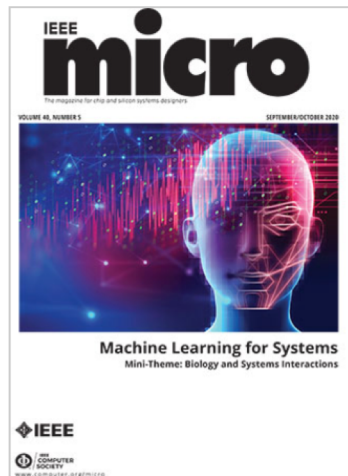


# Detailed Analysis of Tackling the Bottleneck

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

[“Accelerating Genome Analysis: A Primer on an Ongoing Journey”](#)

IEEE Micro, August 2020.



[Home](#) / [Magazines](#) / [IEEE Micro](#) / [2020.05](#)

*IEEE Micro*

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

Sept.-Oct. 2020, pp. 65-75, vol. 40

DOI Bookmark: [10.1109/MM.2020.3013728](#)

### Authors

[Mohammed Alser](#), ETH Zürich

[Zulal Bingöl](#), Bilkent University

[Damla Senol Cali](#), Carnegie Mellon University

[Jeremie Kim](#), ETH Zurich and Carnegie Mellon University

[Saugata Ghose](#), University of Illinois at Urbana-Champaign and Carnegie Mellon University

[Can Alkan](#), Bilkent University

[Onur Mutlu](#), ETH Zurich, Carnegie Mellon University, and Bilkent University

◀  
Previous

▶  
Next

☰ Table of Contents

📄 Past Issues

# Goal: Minimizing Alignment Time

---

Sequence Alignment is expensive

Our goal is to accelerate read mapping  
by reducing the need for  
dynamic programming algorithms

# Novelty, Key Approach, and Ideas

# Key Idea

---

Genomic Strings

```
graph TD; A[Genomic Strings] --> B[Dissimilar Strings]; A --> C[Similar Strings]; B --> D[Ignore them if the number of differences exceeds a threshold.]; C --> E[Find number and location of differences?];
```

**EXPENSIVE!**

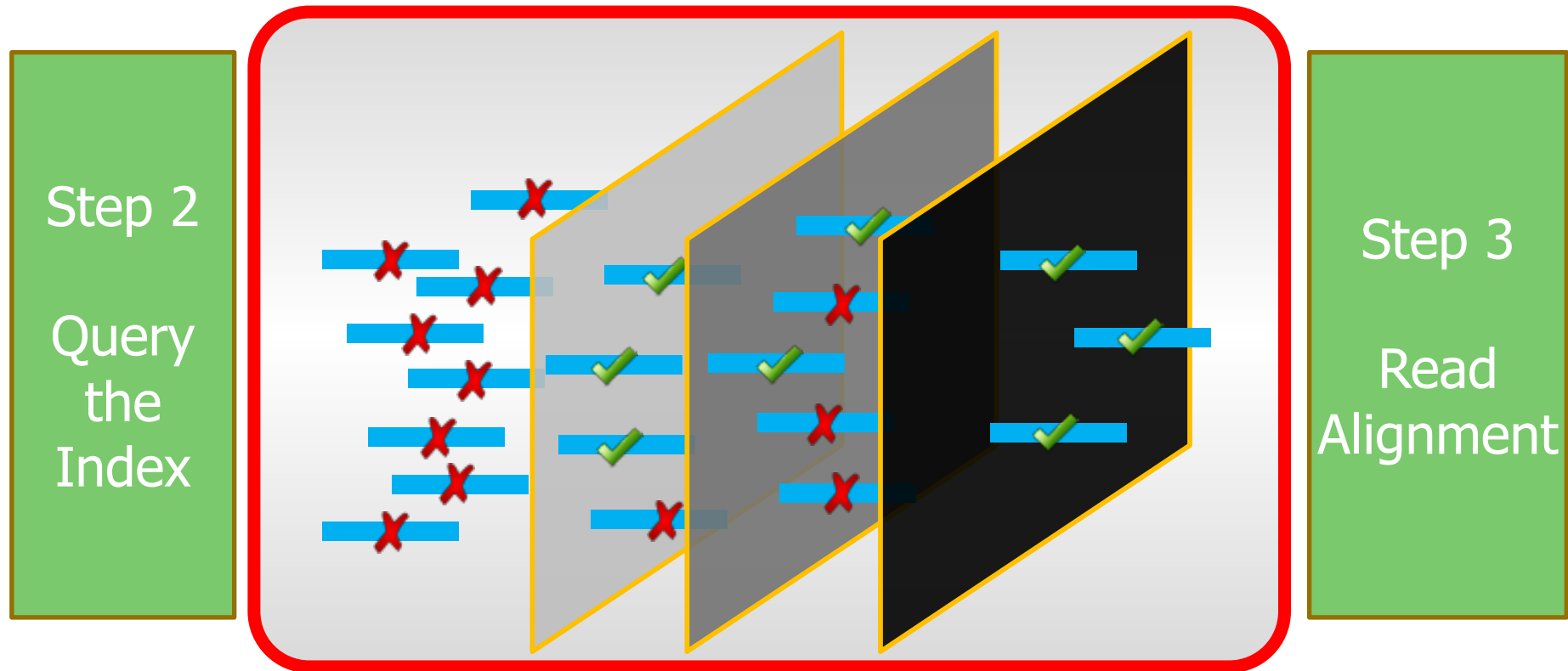
Dissimilar  
Strings

Ignore them if the number  
of differences exceeds a  
threshold.

Similar  
Strings

Find number and location  
of differences?

# Ideal Filtering Algorithm



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

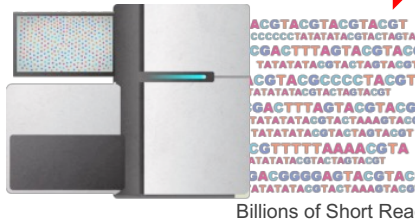
# Proposed Solution: GateKeeper



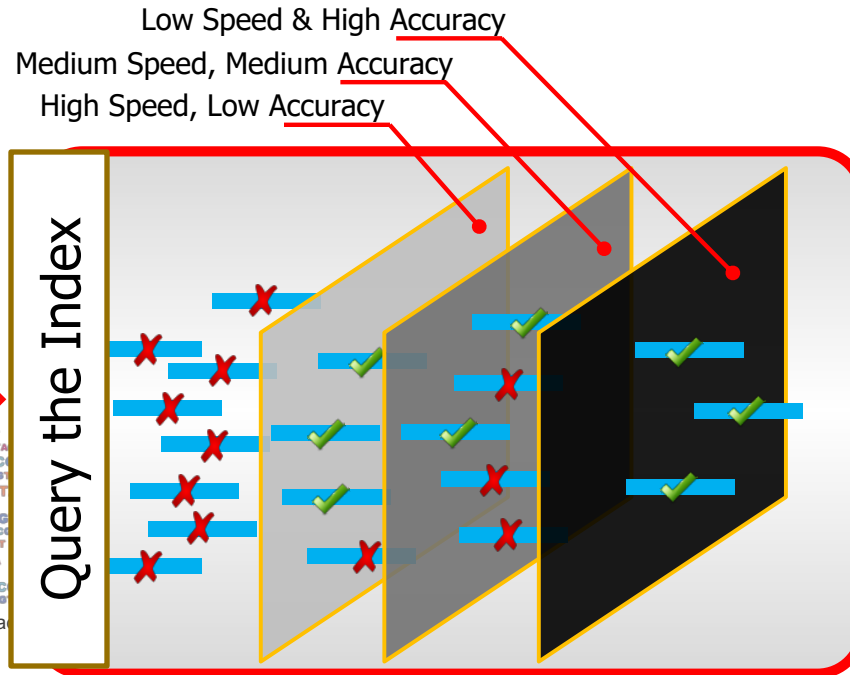
1<sup>st</sup>

FPGA-based  
Alignment Filter

$\times 10^{12}$   
mappings



Query the Index



$\times 10^3$   
mappings

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

- 1 High throughput DNA sequencing (HTS) technologies
- 2 Read Pre-Alignment Filtering  
Fast & Low False Positive Rate
- 3 Read Alignment  
Slow & Zero False Positives

# Mechanisms (in some detail)

# GateKeeper

---

## ■ Key observation:

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

## ■ Key ideas:

- Quickly find similar sequences using *Hamming Distance*.
- Compute “*Shifted Hamming Distance*”: AND of  $2E+1$  Hamming vectors of two strings, to identify invalid mappings
- Use only bit-parallel operations that nicely map to:
  - SIMD instructions
  - FPGA
  - Logic layer of the 3D-stacked memory
  - In-memory accelerators (e.g., Ambit)



# Mechanisms

---

- **Key observation:**

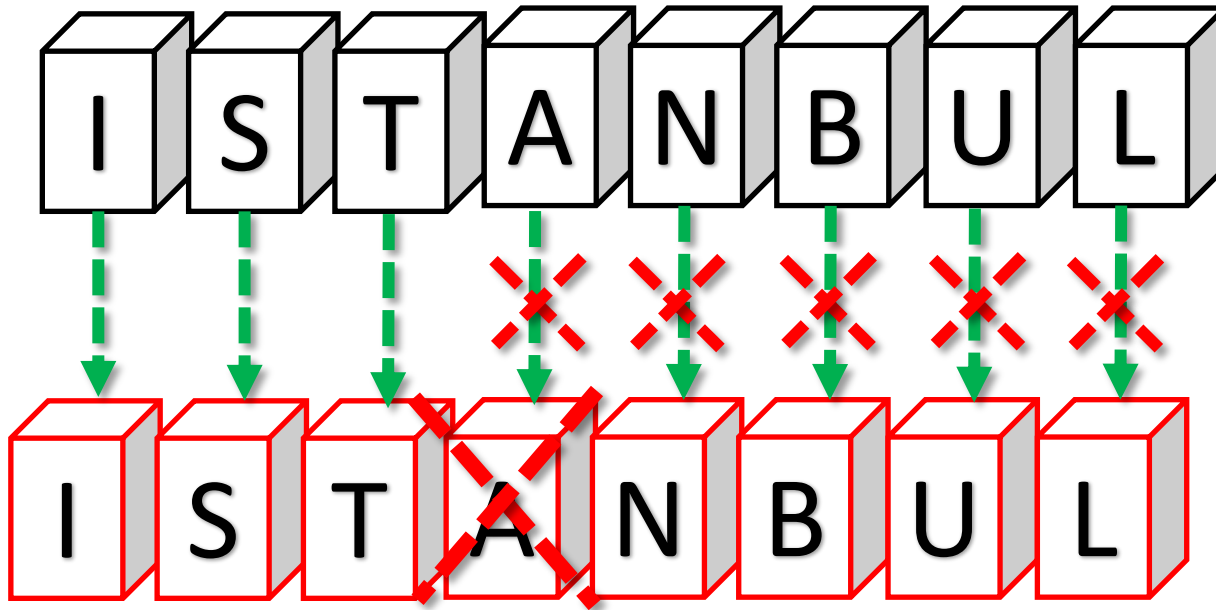
- If two strings differ by  $E$  edits, then every pairwise match can be aligned in at most  $2E$  shifts.

# Hamming Distance ( $\Sigma \oplus$ )

3 matches

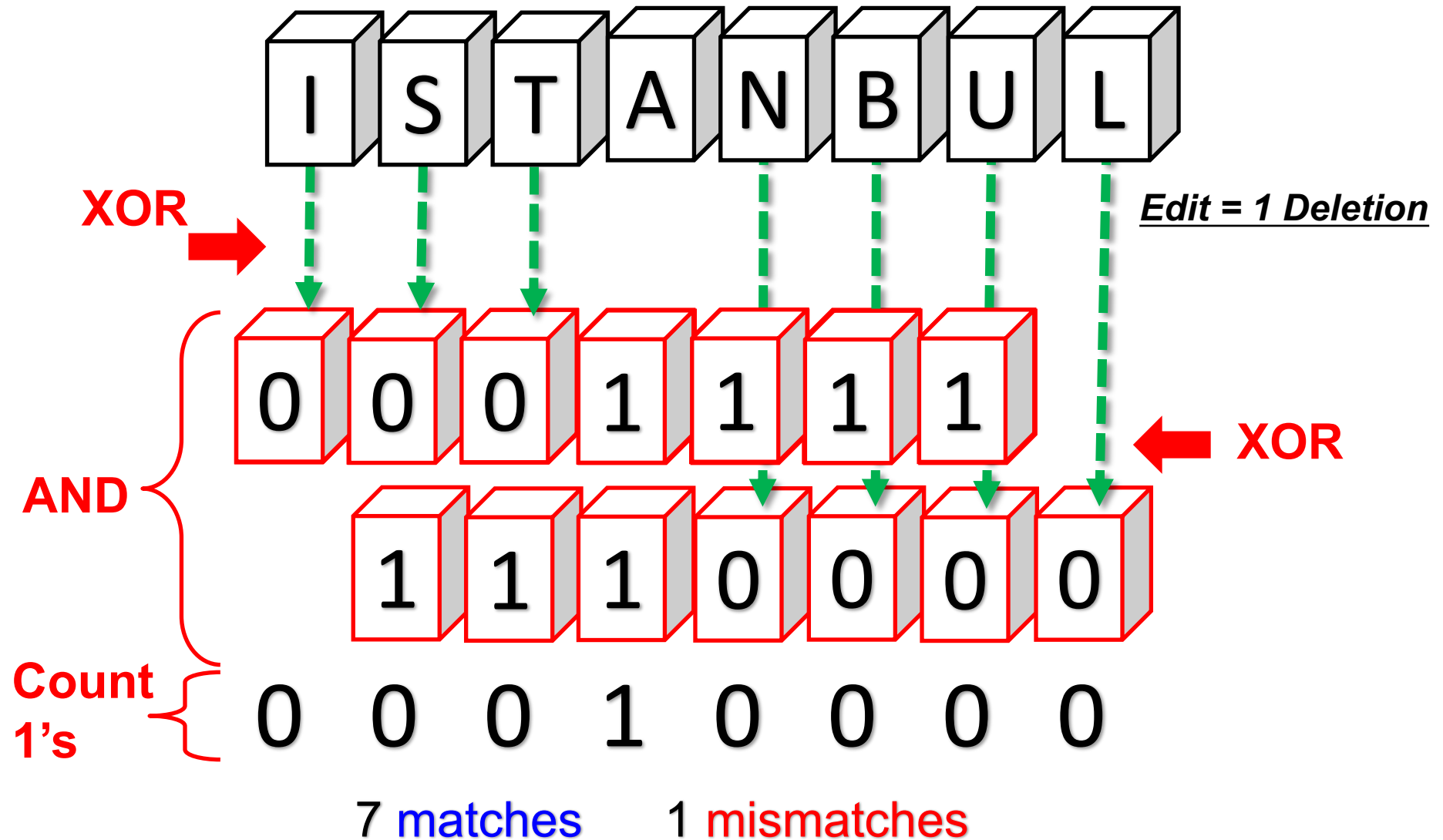
5 mismatches

Edit = 1 Deletion



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

# Shifted Hamming Distance (Xin+ 2015)



# Mechanisms

---

## ■ Key observation:

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

## ■ Key ideas:

- *Quickly* find similar sequences using *Hamming Distance*.
- Compute “Shifted Hamming Distance”: AND of  $2E+1$  Hamming vectors of two strings, to identify invalid mappings

# GateKeeper Walkthrough

Generate  $2E+1$  masks

Amend random zeros:  
101 → 111 & 1001 → 1111

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

Query :GAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGGAACATTGTTGGGCCGGA

Reference :GAGAGAGATAGTTAGTGTTCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGGCCGG

**Hamming Mask :** 0000000000010000000000001111111101111100011101101011011111111100010001011110110110010101

[illegible]

**2-Deletion Mask** : 000000001011011100111111111111111011110001110110101101111111111000100100111101101001010

**3-Deletion Mask :**111111111110111011001101110111011000100100111111111111100101100110101101110111011101111

```
1-Insertion Mask :11111111111101111110111111011110110001001001111111111111110010110011000 01011110111011111110
```

2-Insertion Mask :0000001001111110011111111110010001101010100110101111111111111011100111111000111101100

3-Insertion Mask : 111111111011110110011000111111111110101101111111001100101111011111111101110111010111001000

**AND Mask :**000000000010000000000001000

111

1-1	0000
-----	------

2-1110


3-1 Our goal is to track the diagonally consecutive matches in the 111

1-1r Car year to track the sequentially consecutive materials in the 110

2- Ir neighborhood map 100

3-Ir

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

Needleman-Wunsch Alignment : 

# GateKeeper

---

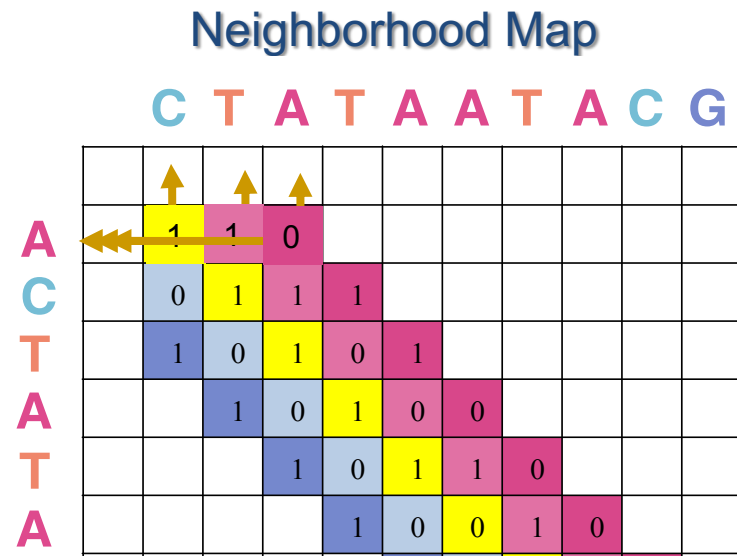
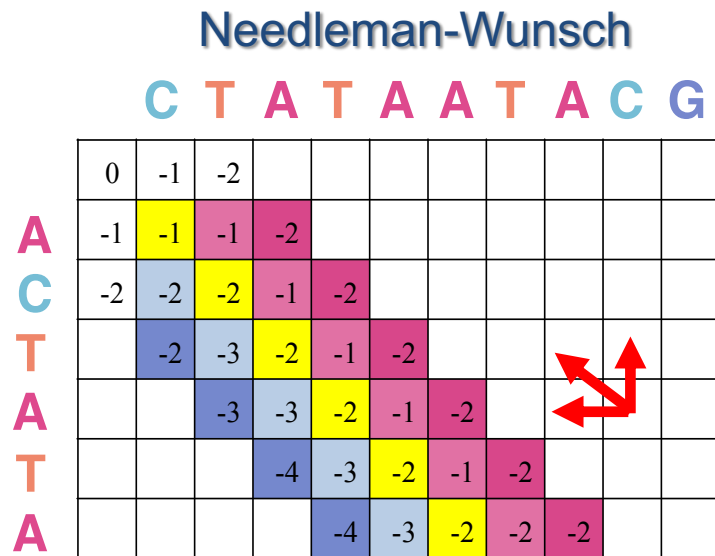
## ■ Key observation:

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

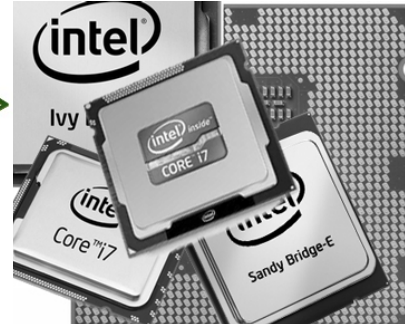
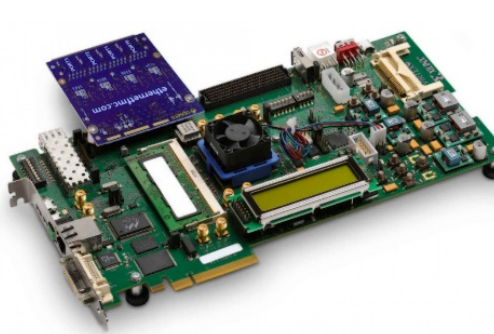
## ■ Key ideas:

- Quickly find similar sequences using *Hamming Distance*.
- Compute “*Shifted Hamming Distance*”: AND of  $2E+1$  Hamming vectors of two strings, to identify invalid mappings
- Use only bit-parallel operations that nicely map to:
  - SIMD instructions
  - FPGA
  - Logic layer of the 3D-stacked memory
  - In-memory accelerators (e.g., Ambit)

# Alignment Matrix vs. Neighborhood Map



Independent vectors can be processed in parallel using hardware technologies



# Hardware Architecture



# GateKeeper Walkthrough (cont'd)

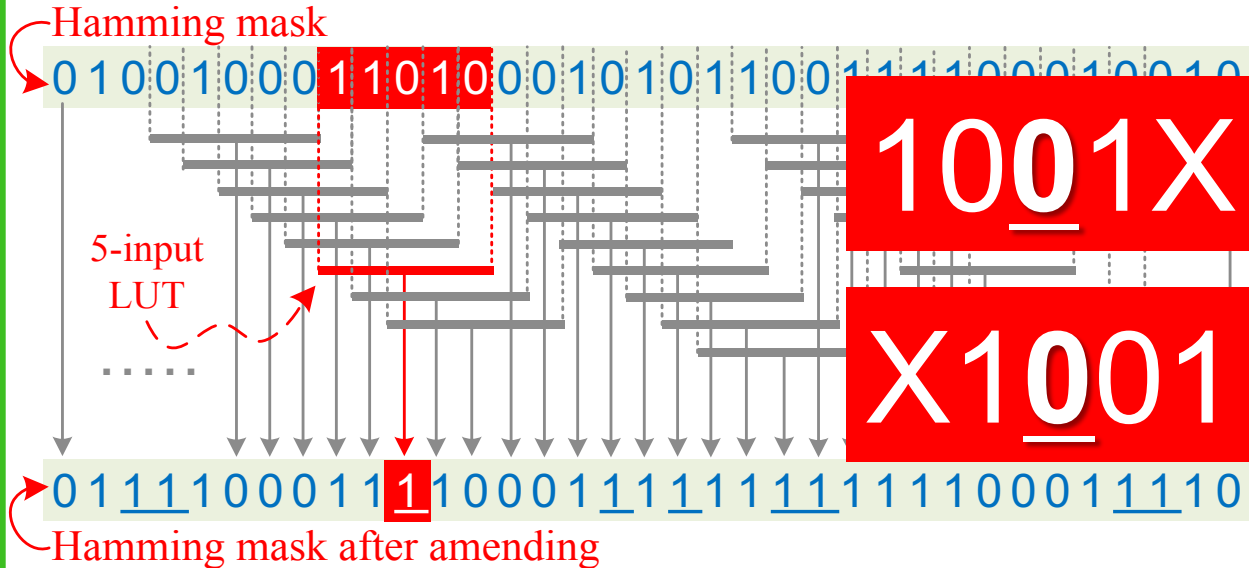
Generate  $2E+1$  masks

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

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

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

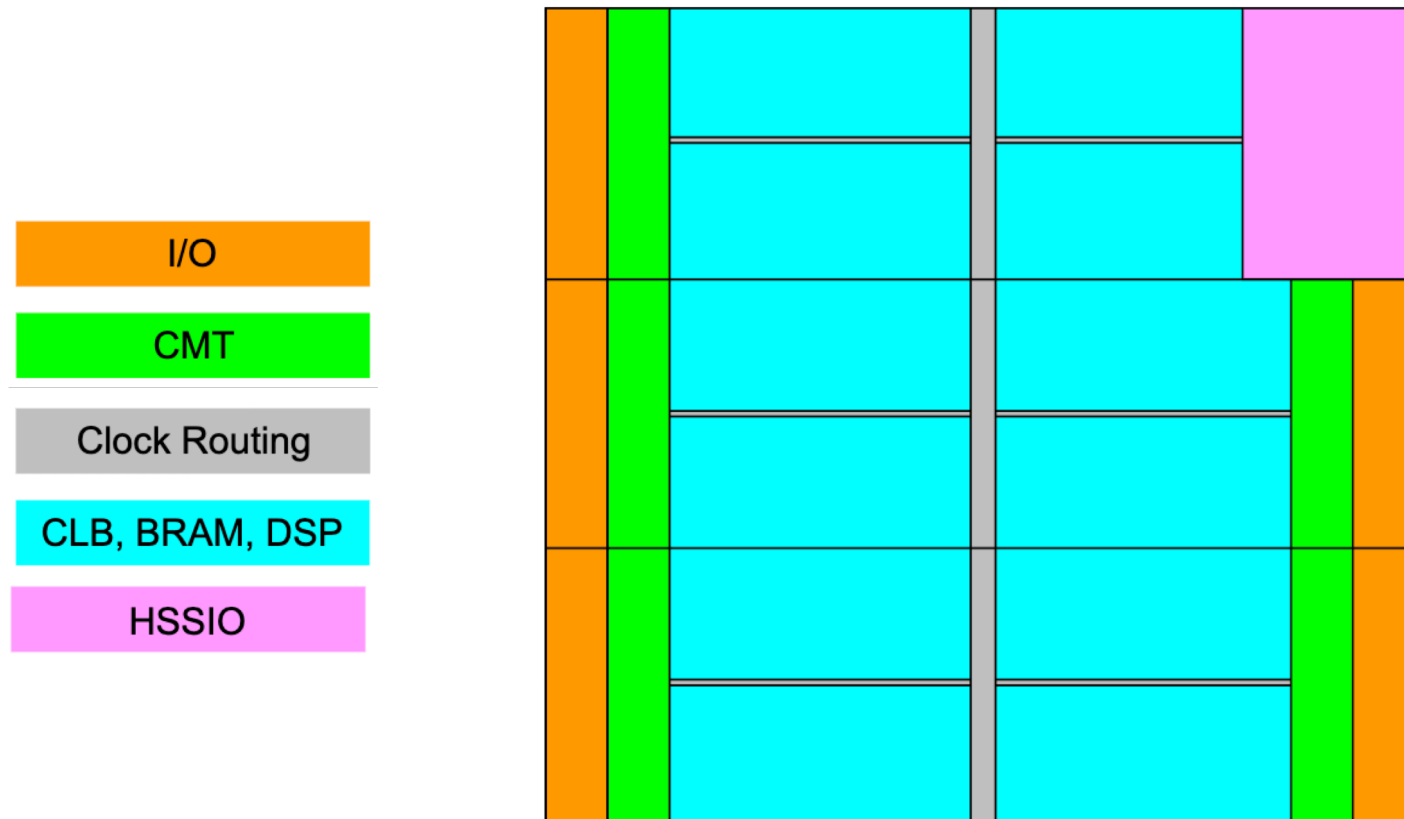
- $(2E) * (\text{ReadLength})$  2-AND operations.
- $(\text{ReadLength}/4)$  5-input LUT.
- $\log_2 \text{ReadLength}$ -bit counter.



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

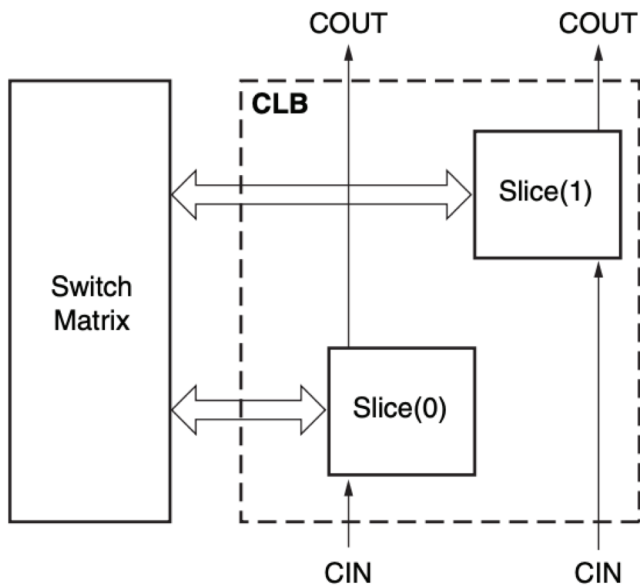
# Virtex-7 FPGA Layout

---



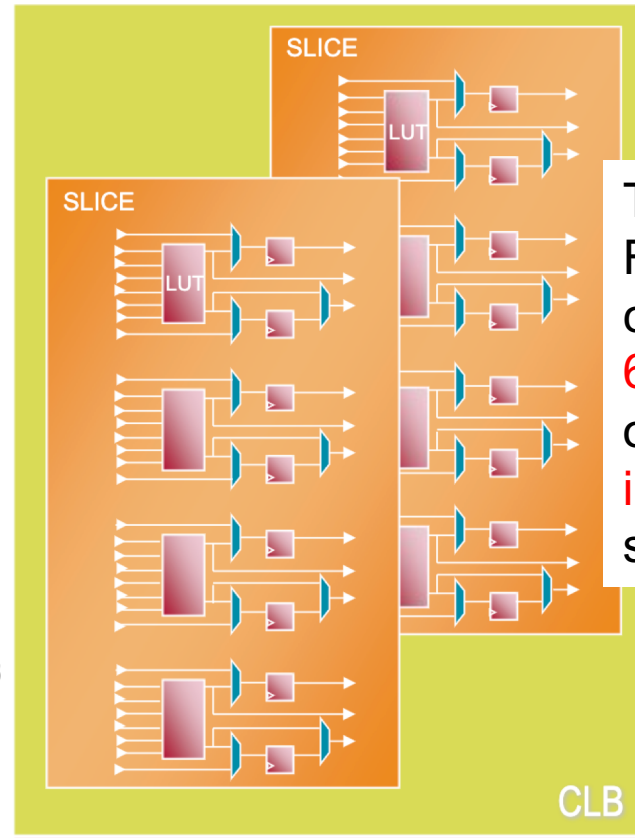
Configurable logic blocks (CLBs) are the main logic resources for implementing sequential as well as combinatorial circuits

# Virtex-7 FPGA Layout



UG474\_c1\_01\_071910

Figure 1-1: Arrangement of Slices within the CLB



The LUTs in 7 series FPGAs can be configured as either a 6-input LUT with one output, or as two 5-input LUTs with separate outputs

Table 2-1: Logic Resources in One CLB

Slices	LUTs	Flip-Flops	Arithmetic and Carry Chains	Distributed RAM <sup>(1)</sup>	Shift Registers <sup>(1)</sup>
2	8	16	2	256 bits	128 bits

# Key Results:

## Methodology and Evaluation

# Methodology

---

- System setup:
  - 3.6 GHz Intel i7-3820 (supports only PCIe 2.0)
  - Xilinx VC709 (~\$5000)
    - Architecture implementation using Vivado 2014.4 in Verilog
    - RIFFA 2.2 to perform Host-FPGA PCIe communication



- Evaluated dataset:
  - Real sequencing read set (ERR240727\_1.fastq)
  - Five simulated read sets of 100 bp and 300 bp long Illumina-like reads with different type and number of edits.

# Prior Work on Pre-Alignment Filtering

---

- Adjacency Filter (*BMC Genomics, 2013*)
  - ❑ Slow
  - ❑ Accepts a large number of dissimilar sequences.
- Shifted Hamming Distance (SHD) (*Bioinformatics, 2015*)
  - ❑ It requires the same execution time as the Adjacency Filter
  - ❑ It accepts 4X fewer dissimilar sequences compared to the Adjacency Filter.
  - ❑ It suffers from a limited sequence length ( $\leq 128$  bp)

# VC709 Resource Utilization

---

## Theoretically:

- Up to 140 GateKeeper Processing cores on a single FPGA (E=5, 100bp)
- BUT bottlenecked by PCIe bandwidth
- Small area allows integration into FPGAs already inside of sequencers

**Table 2.** FPGA resource utilization for a single GateKeeper core

Resource utilization %	Resource utilization %				
	100 bp		300 bp		
	2	5	2	5	15
Read length					
Edit distance					
Slice LUT <sup>a</sup>	0.39%	0.71%	1.27%	2.2%	4.82%
Slice Register <sup>b</sup>	0.01%	0.01%	0.01%	0.01%	0.01%

<sup>a</sup>LUT: look-up tables.

<sup>b</sup>Flip-flop.

# VC709 Resource Utilization

## Experimentally:

- **GateKeeper** aligns each read against up to 8 and 16 different reference segments in parallel, without violating the timing constraints for a sequence lengths of 300 and 100 bp, respectively.

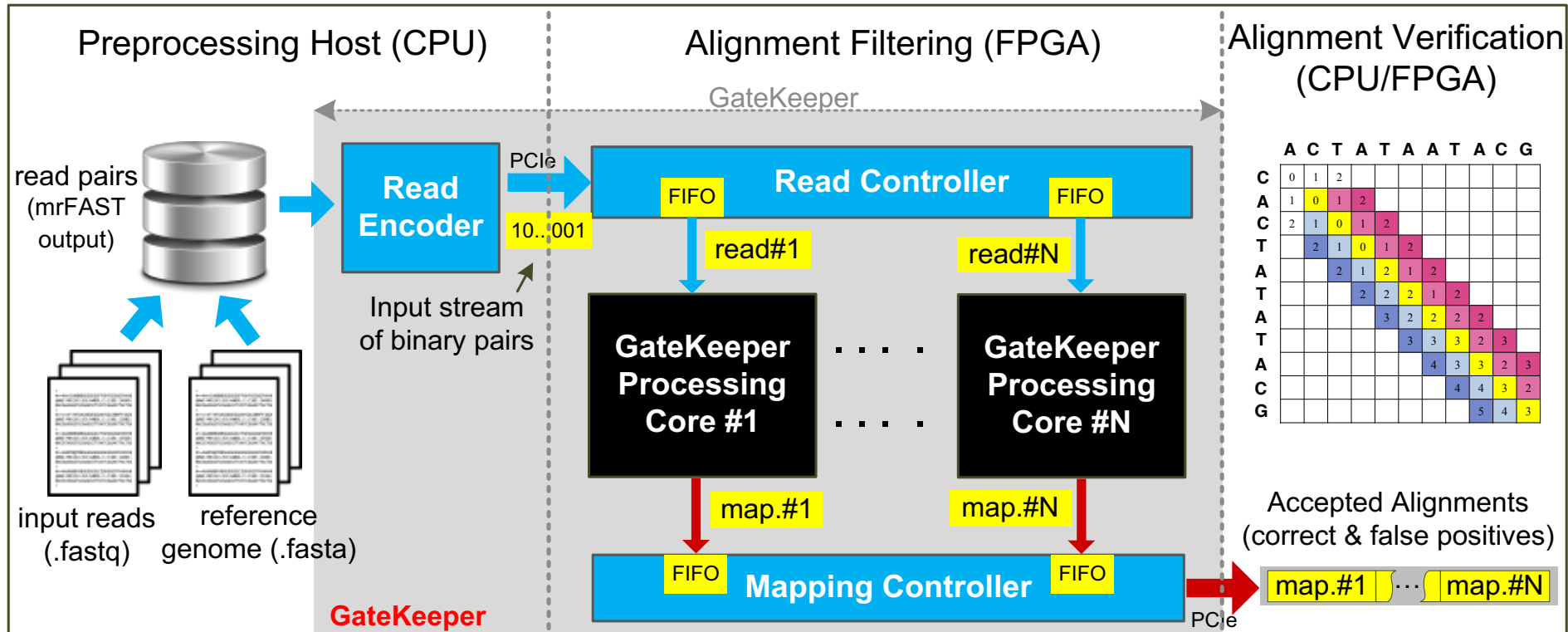
**Table 3.** Overall system resource utilization under different read lengths and edit distance thresholds

Read length	Resource utilization %			
	100 bp 16 GateKeeper cores		300 bp 8 GateKeeper cores	
Edit distance	2	5	2	15
Slice LUT	32%	45%	50%	69%
Slice register	2%	2%	17%	91%
Block memory	2%	2%	2%	2%

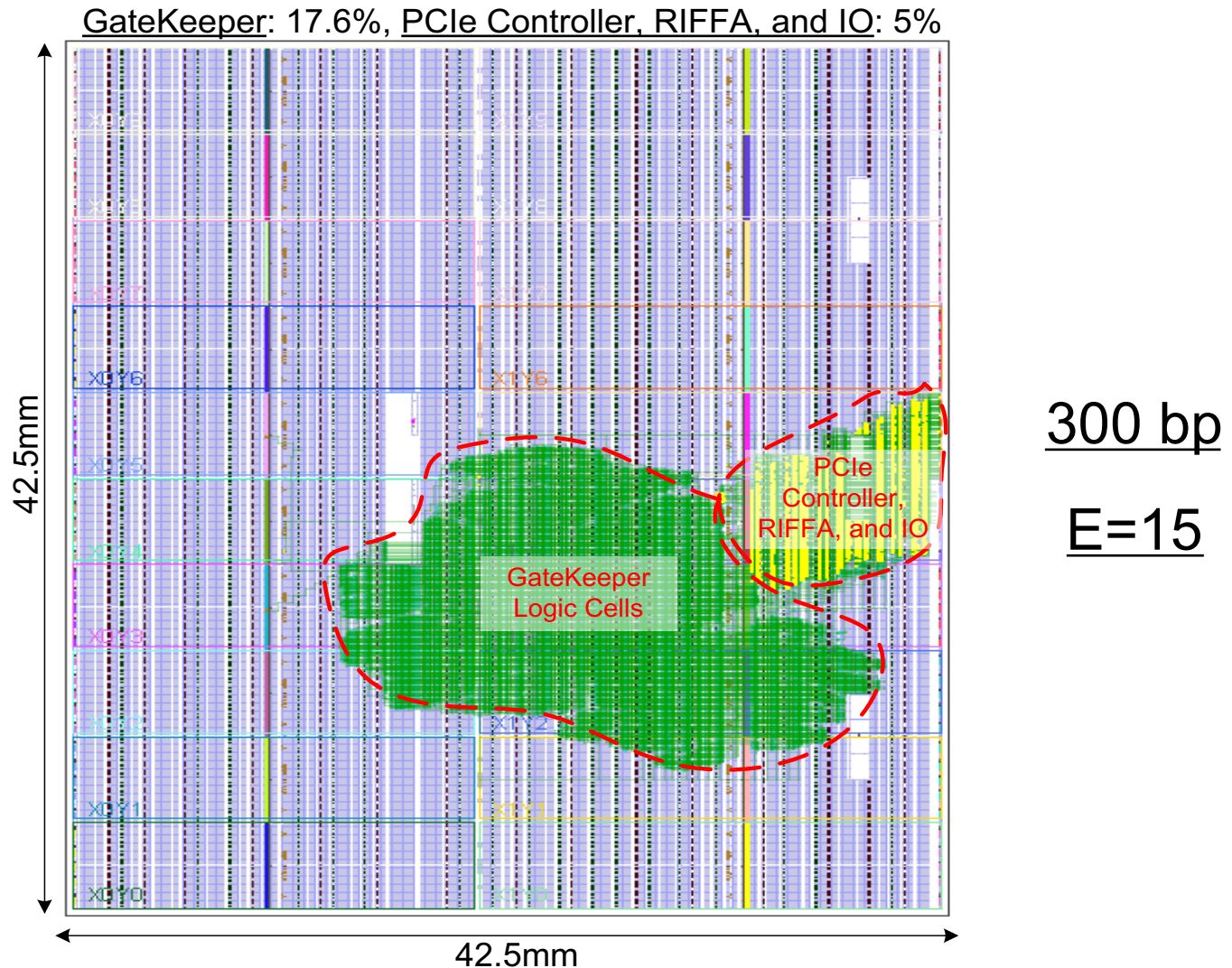


# GateKeeper Accelerator Architecture

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



# FPGA Chip Layout



# Speed & Accuracy Results

---

**90x-130x faster**

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

**Accepts 4x fewer dissimilar strings**

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

**10x speedup**

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

**Freely available online**

[github.com/BilkentCompGen/GateKeeper](https://github.com/BilkentCompGen/GateKeeper)

# Summary

# GateKeeper Conclusions

---

- There is a significant performance gap between high-throughput DNA sequencers and read mapper
- Sequence alignment is computationally expensive and unavoidable
- **GateKeeper** is the first hardware accelerator architecture (as a pre-alignment filter) for quickly rejecting dissimilar sequences
- It provides a huge speedup of up to 130x compared to the previous state of the art software solution.

# GateKeeper Conclusions

---

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

# More on SHD (SIMD Implementation)

---

- Download and test for yourself
- <https://github.com/CMU-SAFARI/Shifted-Hamming-Distance>

*Bioinformatics*, 31(10), 2015, 1553–1560

doi: 10.1093/bioinformatics/btu856

Advance Access Publication Date: 10 January 2015

Original Paper

OXFORD

---

Sequence analysis

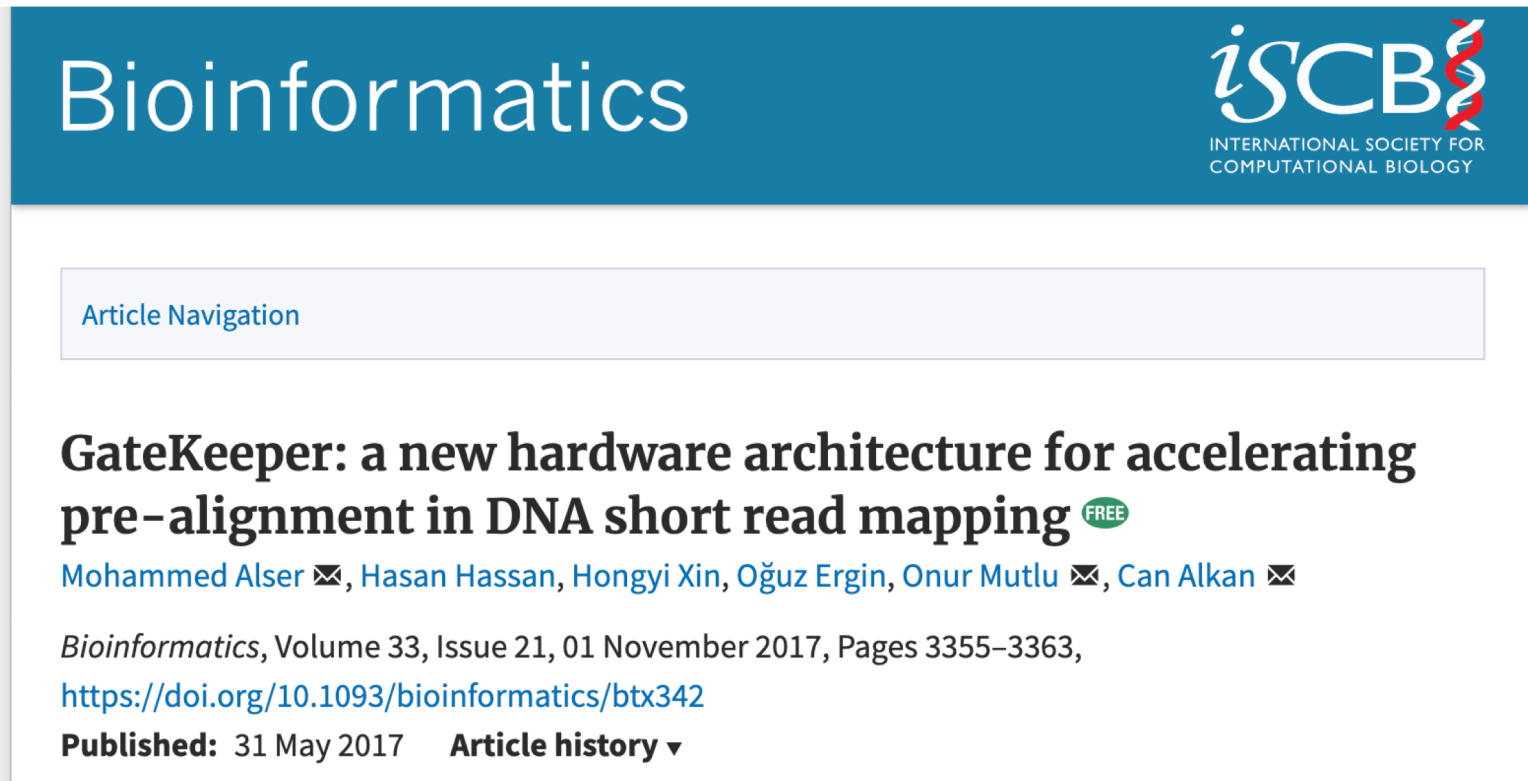
## **Shifted Hamming distance: a fast and accurate SIMD-friendly filter to accelerate alignment verification in read mapping**

**Hongyi Xin<sup>1,\*</sup>, John Greth<sup>2</sup>, John Emmons<sup>2</sup>, Gennady Pekhimenko<sup>1</sup>,  
Carl Kingsford<sup>3</sup>, Can Alkan<sup>4,\*</sup> and Onur Mutlu<sup>2,\*</sup>**

# More on GateKeeper

- Download and test for yourself

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



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

Bioinformatics

iSCB  
INTERNATIONAL SOCIETY FOR  
COMPUTATIONAL BIOLOGY

Article Navigation

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

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

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

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

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



# Strengths

# Strengths

---

- New and simple solution to a critical problem. New algorithm and hardware architecture.
- GateKeeper does not sacrifice any of the aligner capabilities, as it does not modify or replace the alignment step.
- Design is scalable; could add more processing cores in the future.
- Some sequencers use FPGAs as well, so GateKeeper could be integrated into them.

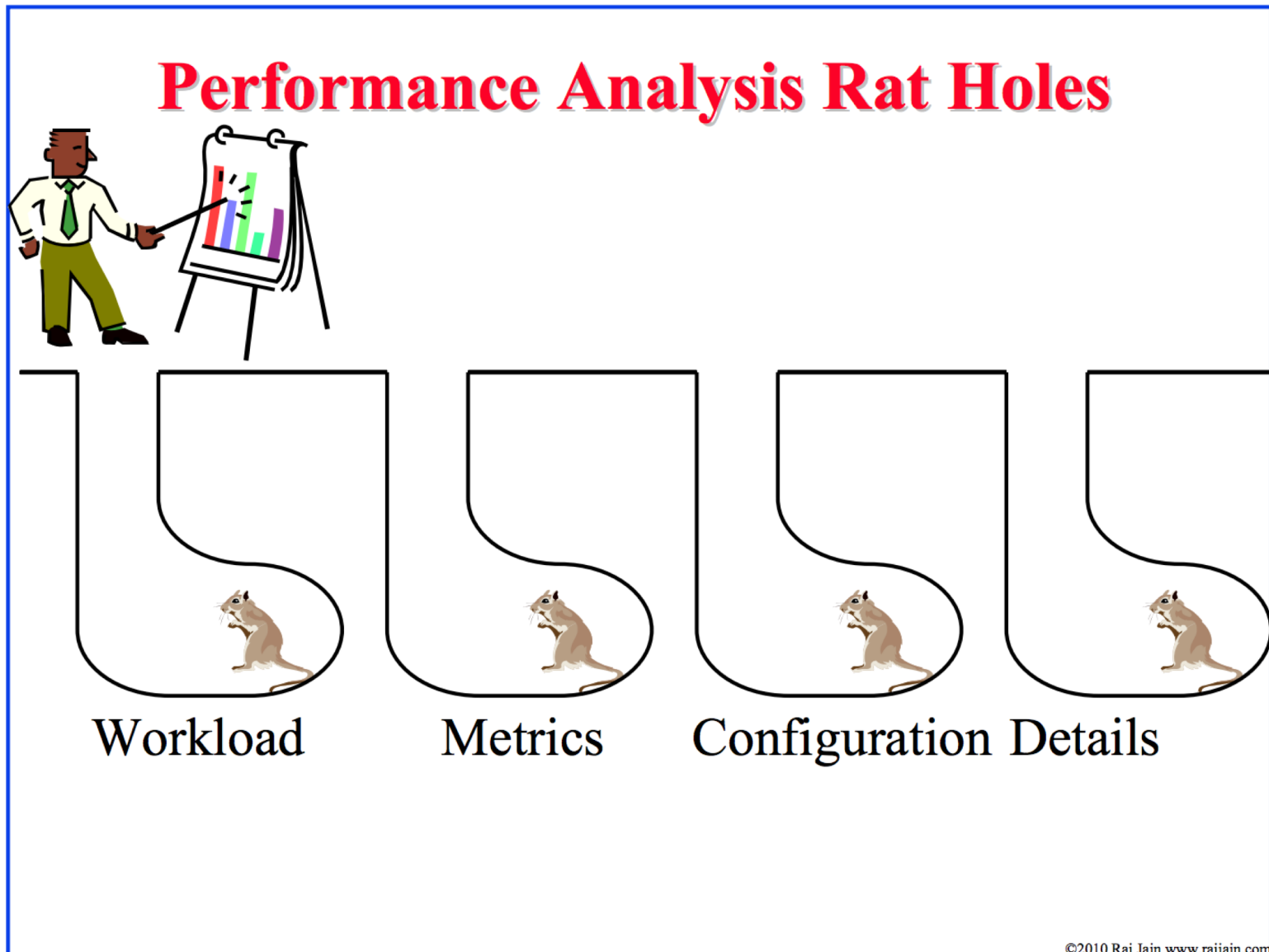
## Strengths (cont'd)

---

- Authors understand and highlight limitations of GateKeeper
- Greatly improves filtering speed and accuracy
- Spurred quite a few papers that build on GateKeeper
- Well-written, interesting and easy to understand paper

# Weaknesses

# Recall: Try to Avoid Rat Holes



# Weaknesses

---

- The benefits of such a mechanism require an FPGA and advanced knowledge with computers, this may be **problematic for some biologists/genomicists/geneticists**
- The amendment of the random zeros is a simple “**hack**” to reduce the number of false positives, but there is **no explanation** why GateKeeper only flips the patterns 101 and 1001, what about 10001? And  $10^n1$ ?
- The paper can be **confusing at times** due to the use of a “supplementary material” document that is constantly referred to (but understandable as there was a page limit set by the publication journal).

# Weaknesses (cont'd)

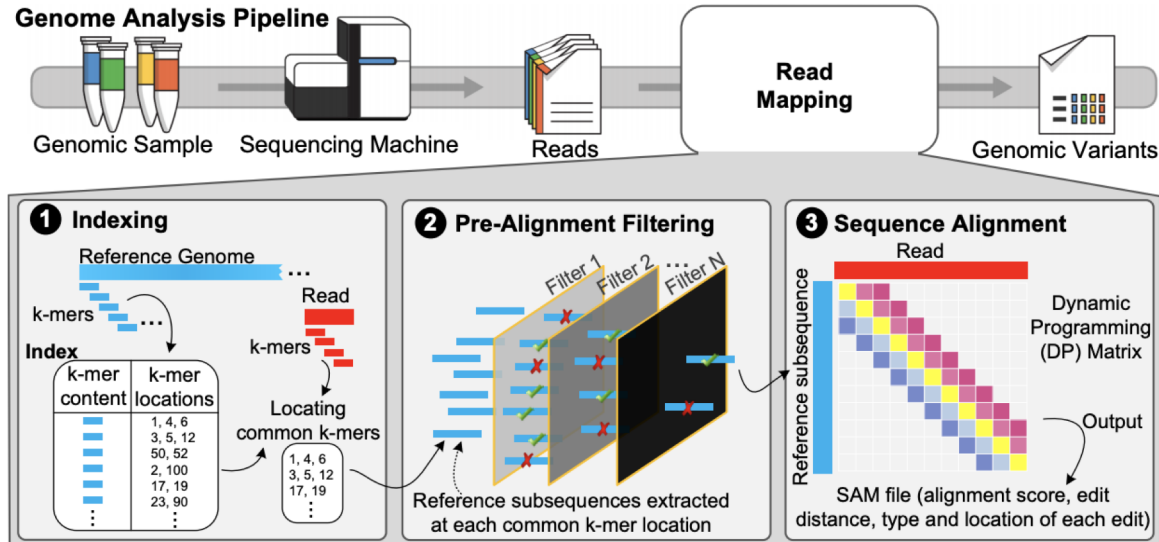
---

- GateKeeper's **accuracy degrades** exponentially for  $E > 2\%$ , and becomes ineffective for  $E > 8\%$ .
- GateKeeper is tested using short reads
  - 3<sup>rd</sup> generation sequencing machines produce much **longer reads**

# Thoughts and Ideas



# Accelerating Read Mapping



## Accelerating Indexing

Reducing the number of seeds

Reducing data movement during indexing

## Accelerating Pre-Alignment Filtering

q-gram filtering

Pigeonhole principle

Base counting

Sparse DP

## Accelerating Alignment

Accurate alignment accelerators

Heuristic-based alignment accelerators

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

# Our Ongoing Journey

## Near-memory/In-memory Pre-alignment Filtering

**GRIM-Filter** [BMC Genomics'18]

**SneakySnake** [IEEE Micro'21]

**GenASM** [MICRO 2020]

## Near-memory Sequence Alignment

**GenASM** [MICRO 2020]

## Specialized Pre-alignment Filtering Accelerators (GPU, FPGA)

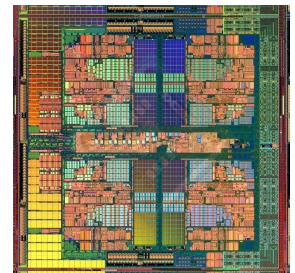
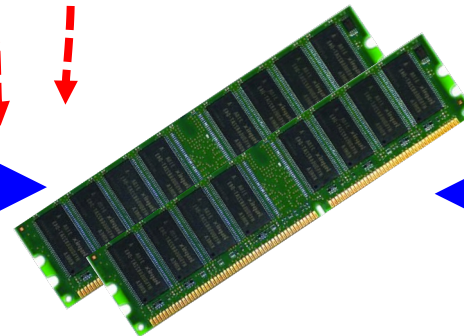
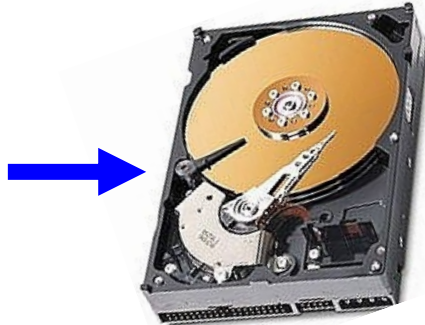
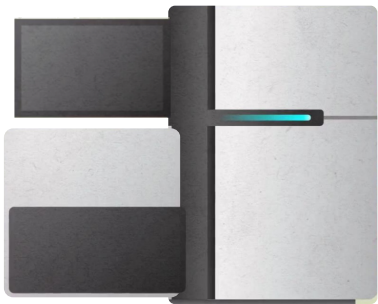
**GateKeeper** [Bioinformatics'17]

**MAGNET** [AACBB'18]

**Shouji** [Bioinformatics'19]

**GateKeeper-GPU** [arXiv'21]

**SneakySnake** [Bioinformatics'20]



Sequencing Machine

Storage (SSD/HDD)

Main Memory

Microprocessor

# Extensions

---

- Can we improve the filtering accuracy
  - Don't amend, count the number of matches accurately.
    - Yes, **see** [MAGNET paper \[Alser et al. \*arXiv preprint\* 2017\]](#). But this requires large number of LUTs.

# MAGNET [Alser+, arXiv 2017]

---

- Mohammed Alser, Onur Mutlu, and Can Alkan,  
**"MAGNET: Understanding and Improving the Accuracy of  
Genome Pre-Alignment Filtering"**  
*IPSI Transactions on Internet Research*, July 2017.  
[arXiv.org version](#), July 2017.  
[\[Source Code\]](#)

## MAGNET: Understanding and Improving the Accuracy of Genome Pre-Alignment Filtering

Alser, Mohammed; Mutlu, Onur; and Alkan, Can

# MAGNET Walkthrough

## Build Neighborhood Map

## Track the Diagonally Consecutive Matches

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

[illegible]

## Find the longest segment of consecutive zeros

## Exclude the errors from the search space

Divide the problem into two subproblems and repeat

Total number of edits = number of 1's in MAGNET bit-vector

# Extensions

---

- Can we improve the filtering accuracy
  - Don't amend, count the number of matches accurately.
    - Yes, **see** [MAGNET paper \[Alser et al. \*arXiv preprint\* 2017\]](#). But this requires large number of LUTs.
- Can we improve the filtering accuracy and scalability
  - Yes, **see** [Shouji paper \[Alser et al. \*Bioinformatics\* 2019\]](#).

# 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<sup>1,2,3,\*</sup>, Hasan Hassan<sup>1</sup>, Akash Kumar<sup>2</sup>, Onur Mutlu<sup>1,3,\*</sup>  
and Can Alkan<sup>3,\*</sup>**

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

\*To whom correspondence should be addressed.

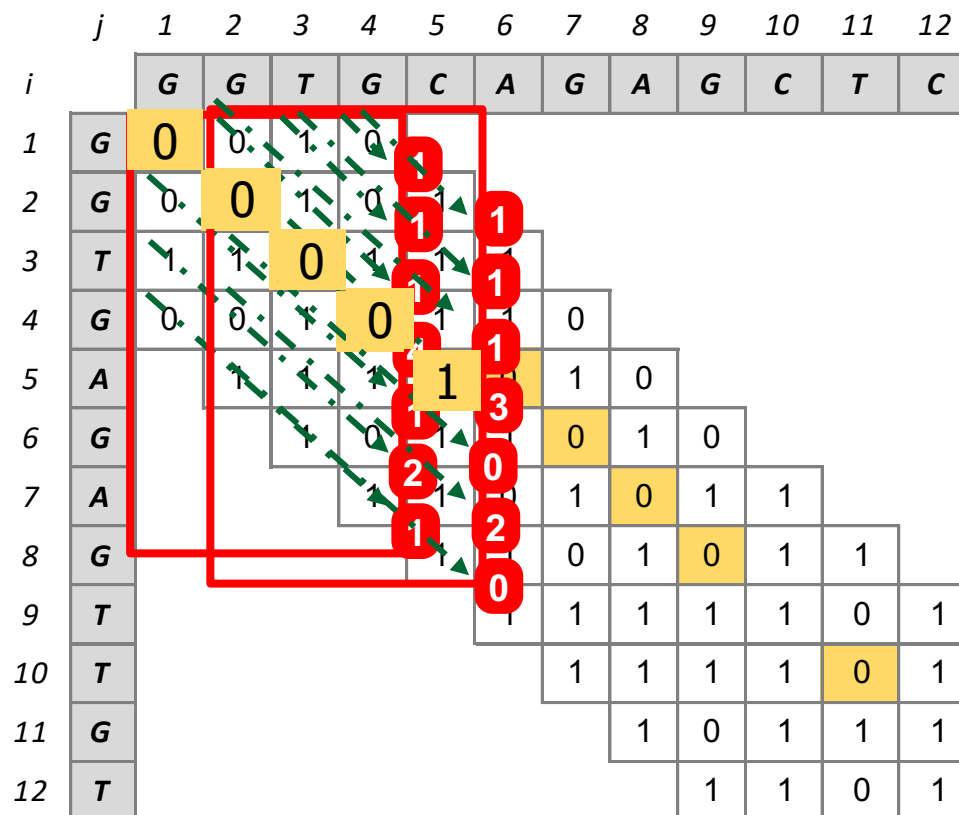
Associate Editor: Inanc Birol

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

# Shouji Walkthrough

Building the  
Neighborhood Map

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



Storing it @ Shouji Bit-vector

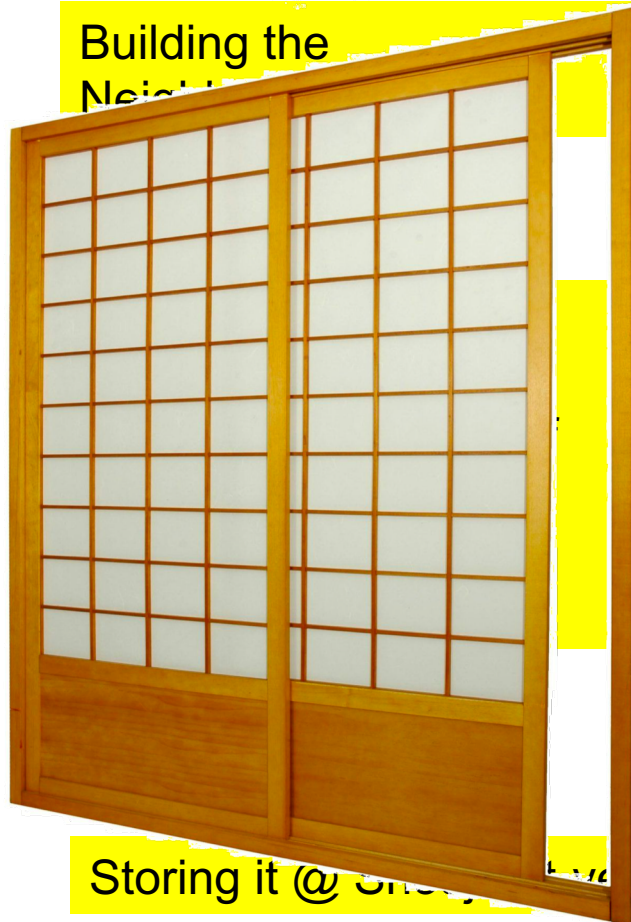
0 0 0 0 1 0 0 0 0 1 0 1

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

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



# Shouji Walkthrough



Building the  
Neighbor

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

Storing it @ Shouji Vector

0	0	0	0	1	0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---

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

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

# Extensions

---

- Can we improve the filtering accuracy
  - Don't amend, count the number of matches accurately.
    - Yes, **see** [MAGNET paper \[Alser et al. \*arXiv preprint\* 2017\]](#). But this requires large number of LUTs.
- Can we improve the filtering accuracy and scalability
  - Yes, **see** [Shouji paper \[Alser et al. \*Bioinformatics\* 2019\]](#).
- Can we solve the FPGA-CPU communication bottleneck?
  - **Where it makes sense**: Processing-in-memory, Processing-near-storage, Processing-while-sequencing?
  - Yes, **see** [GRIM-Filter \[Kim et al. \*BMC Genomics\* 2018\]](#).

# GRIM-Filter [Kim+, BMC Genomics 2018]

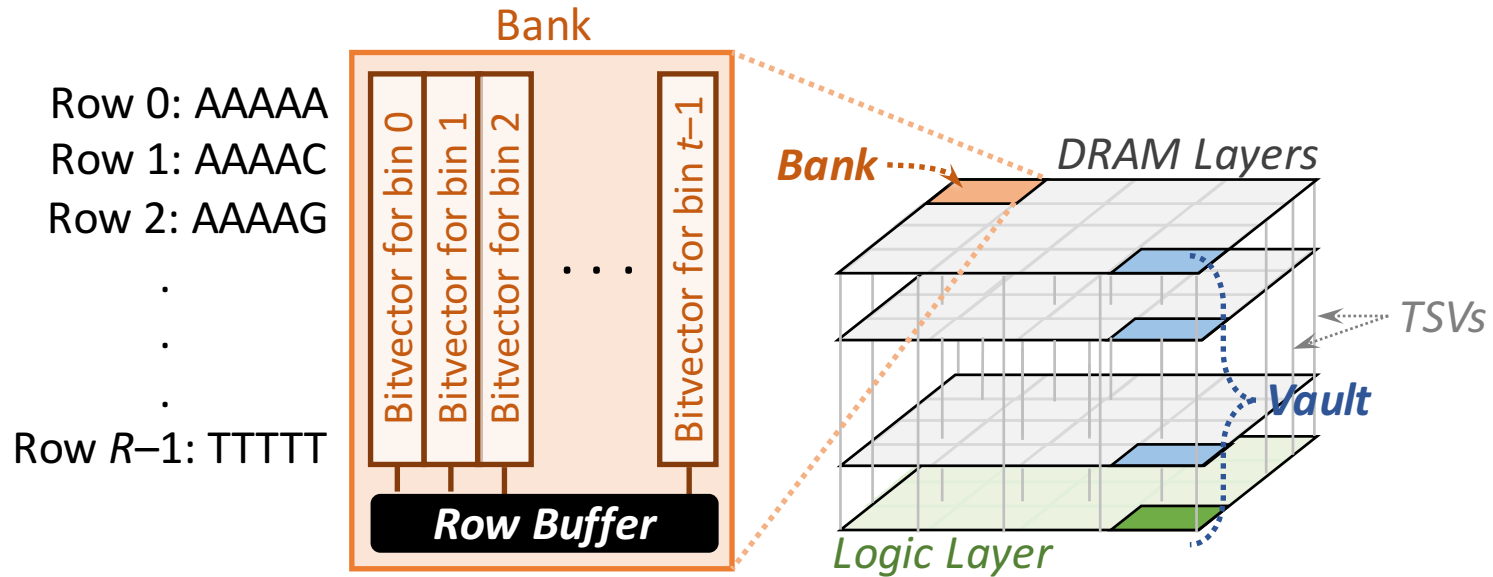
---

- Jeremie S. Kim, Damla Senol Cali, Hongyi Xin, Donghyuk Lee, Saugata Ghose, Mohammed Alser, Hasan Hassan, Oguz Ergin, Can Alkan, and Onur Mutlu, **"GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies"**  
to appear in [\*BMC Genomics\*](#), 2018.  
*Proceedings of the 16th Asia Pacific Bioinformatics Conference (APBC)*,  
Yokohama, Japan, January 2018.  
[arxiv.org Version \(pdf\)](#)

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

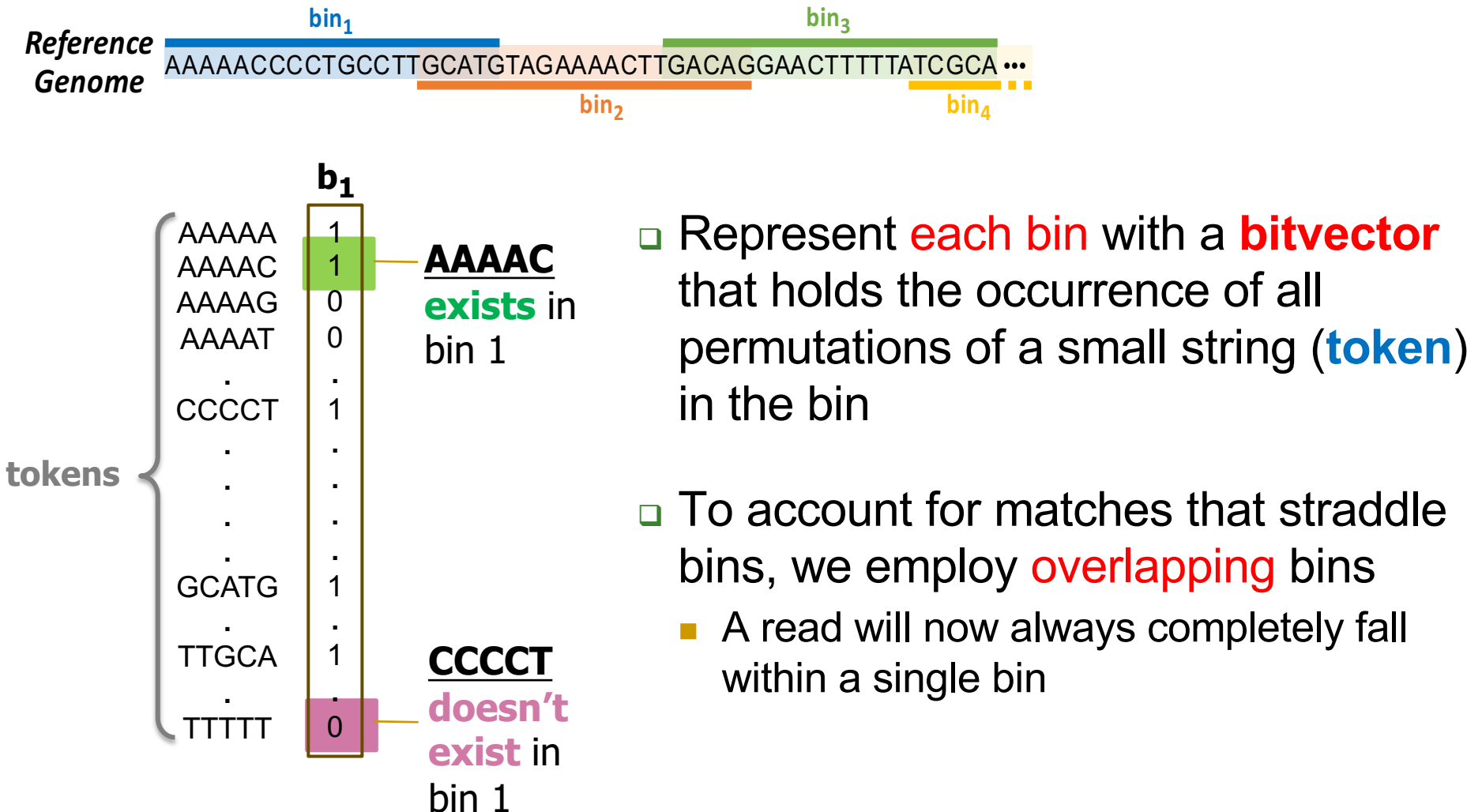
Jeremie S. Kim<sup>1,6\*</sup>, Damla Senol Cali<sup>1</sup>, Hongyi Xin<sup>2</sup>, Donghyuk Lee<sup>3</sup>, Saugata Ghose<sup>1</sup>,  
Mohammed Alser<sup>4</sup>, Hasan Hassan<sup>6</sup>, Oguz Ergin<sup>5</sup>, Can Alkan<sup>\*4</sup>, and Onur Mutlu<sup>\*6,1</sup>

# GRIM-Filter in 3D-Stacked DRAM

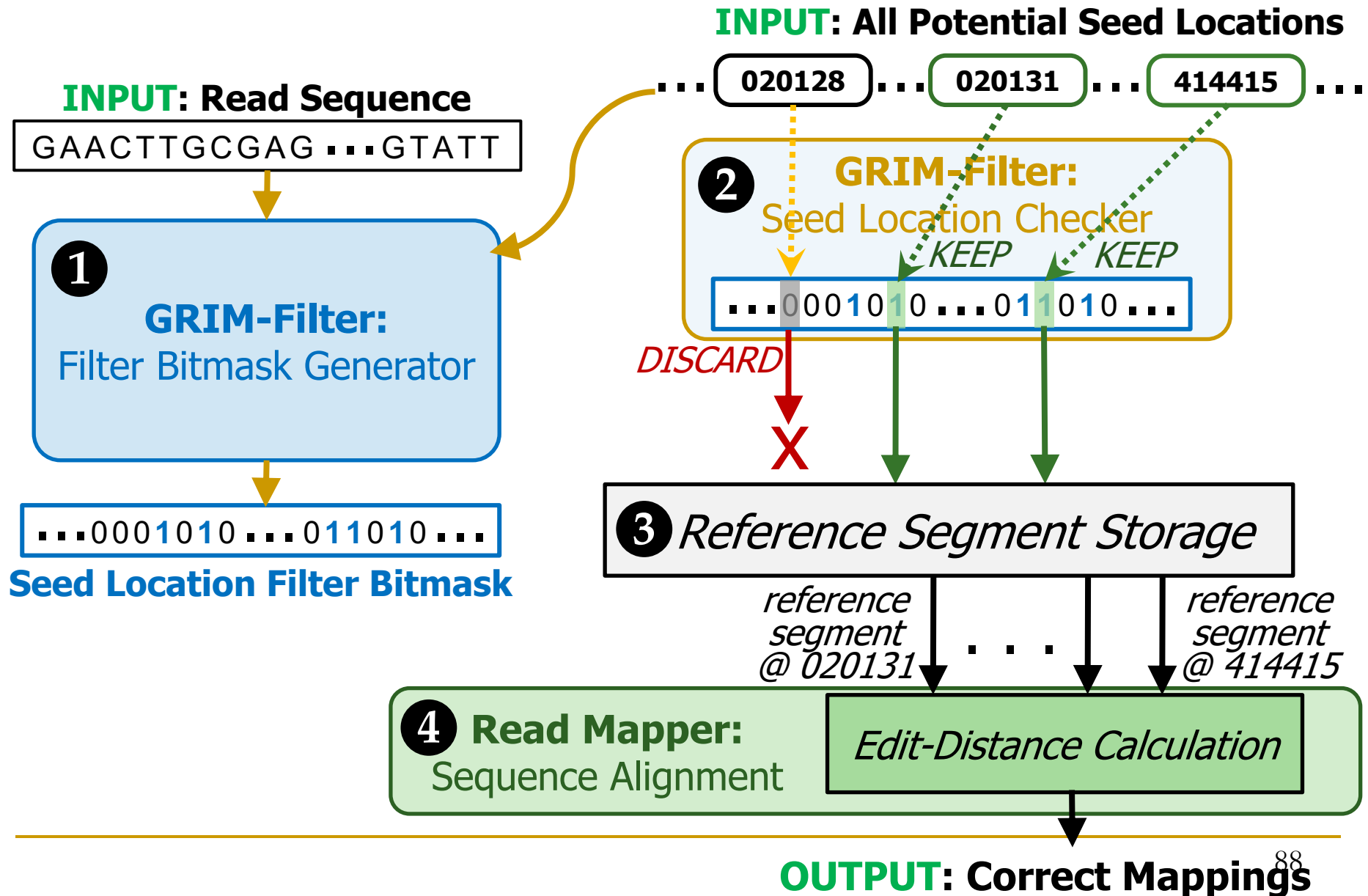


- Each DRAM layer is organized as an array of **banks**
  - A **bank** is an array of cells with a row buffer to transfer data
- The layout of bitvectors in a bank enables filtering many bins in parallel

# GRIM-Filter: Bitvectors



# Integrating GRIM-Filter into a Read Mapper



# Can We Do Better?

---

Faster, More Accurate,  
More Scalable

Pre-Alignment Filtering

# Specialized Hardware for Pre-alignment Filtering

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**, 2020.

[\[Source Code\]](#)

[\[Online link at Bioinformatics Journal\]](#)

## Bioinformatics



## SneakySnake: a fast and accurate universal genome pre-alignment filter for CPUs, GPUs and FPGAs

Mohammed Alser ✉, Taha Shahroodi, Juan Gómez-Luna, Can Alkan ✉, Onur Mutlu ✉

*Bioinformatics*, btaa1015, <https://doi.org/10.1093/bioinformatics/btaa1015>

**Published:** 26 December 2020    **Article history** ▼



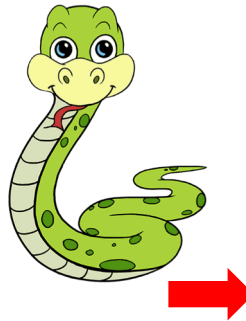
# SneakySnake

## ■ Key observation:

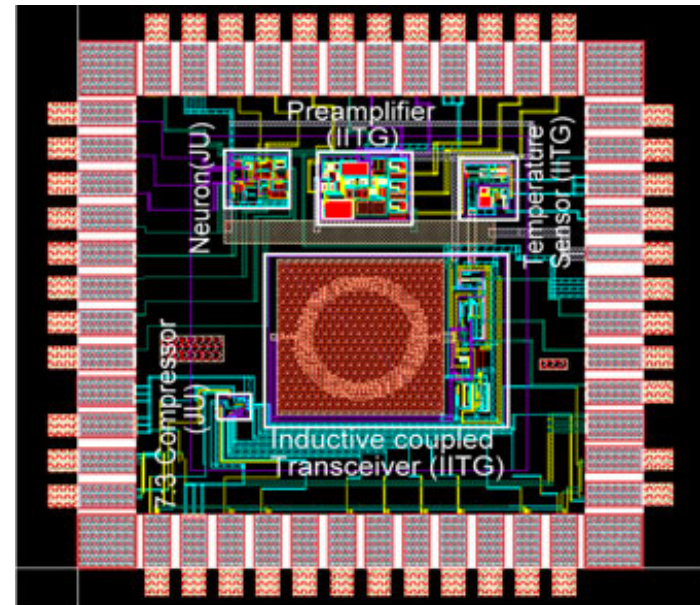
- ❑ Correct alignment is a sequence of non-overlapping long matches

## ■ Key idea:

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



VLSI chip layout



# SneakySnake Walkthrough

Building Neighborhood Map

Finding the Optimal Routing Path

Examining the Snake Survival

of value '0') in its corresponding HRT. Given two genomic sequences, a reference sequence  $R[1 \dots m]$  and a query sequence  $Q[1 \dots m]$ , and an edit distance threshold  $E$ , we calculate the entry  $Z[i, j]$  of the chip maze, where  $1 \leq i \leq (2E + 1)$  and  $1 \leq j \leq m$ , as follows:

$$E = 3$$

$$Z[i, j] = \begin{cases} 0, & \text{if } i = E + 1, Q[j] = R[j], \\ 0, & \text{if } 1 \leq i \leq E, Q[j - i] = R[j], \\ 0, & \text{if } i > E + 1, Q[j + i - E - 1] = R[j], \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

	column	1	2	3	4	5	6	7	8	9	10	11	12
<i>3<sup>rd</sup> Upper Diagonal</i>		1	1	1	0	1	1	0	0	0	1	1	1
<i>2<sup>nd</sup> Upper Diagonal</i>		1	1	1	0	1	1	1	1	1	1	0	1
<i>1<sup>st</sup> Upper Diagonal</i>		1	0	1	1	1	0	0	0	0	1	0	1
<i>Main Diagonal</i>		0	0	0	0	1	1	1	1	1	1	1	1
<i>1<sup>st</sup> Lower Diagonal</i>		0	1	1	1	1	0	0	1	1	1	0	1
<i>2<sup>nd</sup> Lower Diagonal</i>		1	0	1	0	1	1	1	1	0	1	1	1
<i>3<sup>rd</sup> Lower Diagonal</i>		0	1	1	1	1	1	1	1	1	1	1	1

# SneakySnake Walkthrough

Building Neighborhood Map

Finding the Optimal Routing Path

Examining the Snake Survival

$$E = 3$$

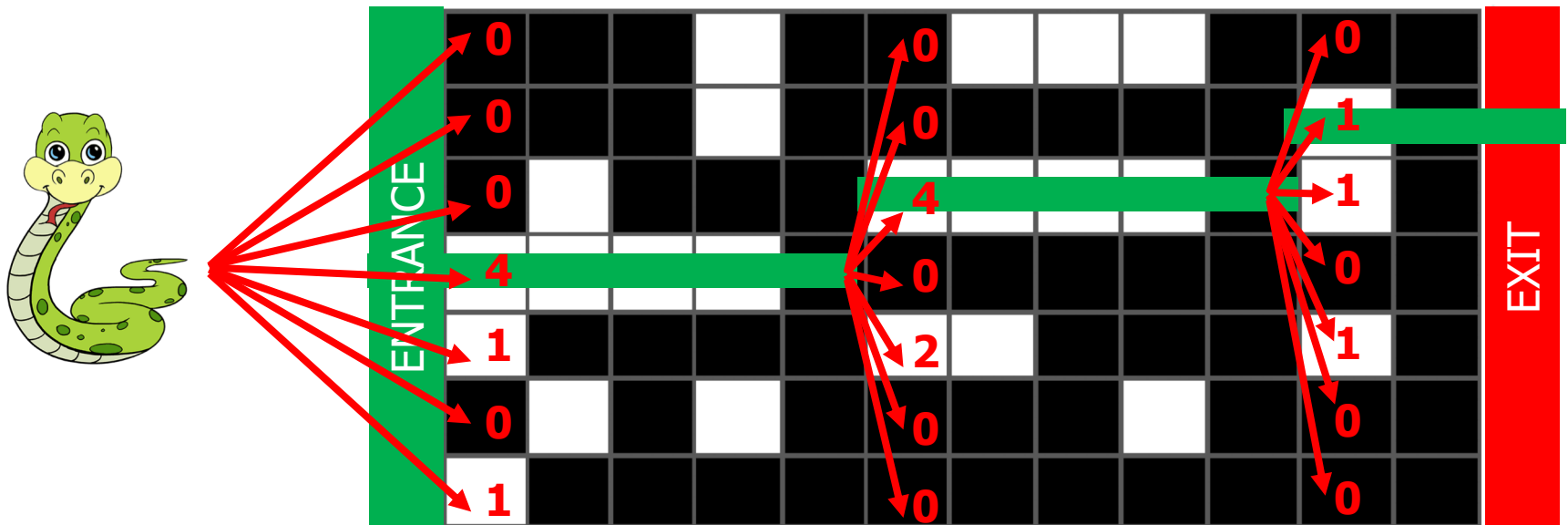
	column	1	2	3	4	5	6	7	8	9	10	11	12	
<i>3<sup>rd</sup> Upper Diagonal</i>	ENTRANCE	■	■	■	□	■	■	□	□	□	■	■	■	EXIT
<i>2<sup>nd</sup> Upper Diagonal</i>		■	■	■	□	■	■	■	■	■	■	□	■	
<i>1<sup>st</sup> Upper Diagonal</i>		■	□	■	■	■	□	□	□	□	■	□	■	
<i>Main Diagonal</i>		□	□	□	□	■	■	■	■	■	■	■	■	
<i>1<sup>st</sup> Lower Diagonal</i>		□	■	■	■	■	□	□	■	■	■	□	■	
<i>2<sup>nd</sup> Lower Diagonal</i>		■	□	■	□	■	■	■	■	□	■	■	■	
<i>3<sup>rd</sup> Lower Diagonal</i>		□	■	■	■	■	■	■	■	■	■	■	■	

# SneakySnake Walkthrough

Building Neighborhood Map

Finding the Optimal Routing Path

Examining the Snake Survival



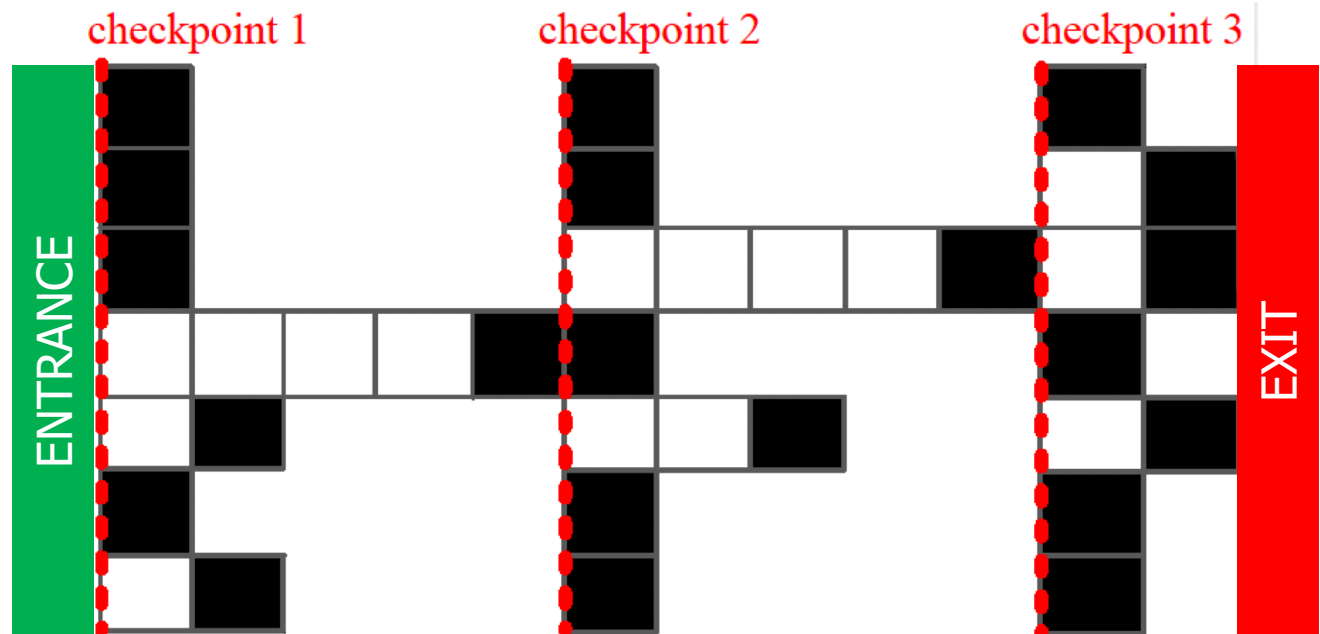
# SneakySnake Walkthrough

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!**



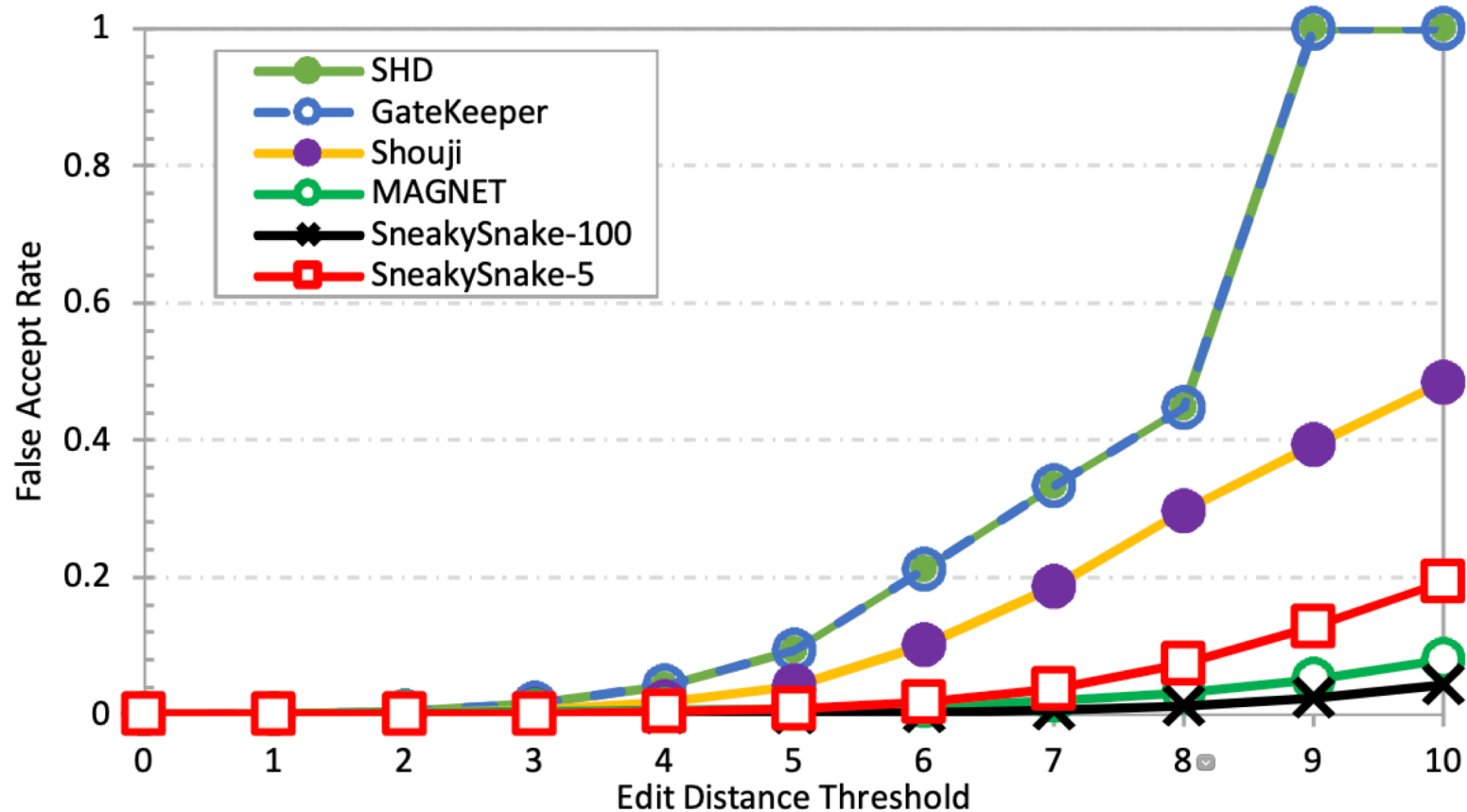
# FPGA Resource Analysis

---

- 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
<b>GateKeeper</b>	2	0.39%	0.01%	16
	5	0.71%	0.01%	16
<b>Shouji</b>	2	0.69%	0.08%	16
	5	1.72%	0.16%	16
<b>Snake-on-Chip</b>	2	0.68%	0.16%	16
	5	1.42%	0.34%	16

# Filtering Accuracy



Alser, "[Accelerating the Understanding of Life's Code Through Better Algorithms and Hardware Design](#)", *arXiv preprint arXiv:1910.03936*, 2019.

# Key Results of SneakySnake

---

- ❑ SneakySnake is up to **four orders of magnitude more accurate** than **Shouji** (Bioinformatics'19) and **GateKeeper** (Bioinformatics'17)
- ❑ Using short reads, SneakySnake **accelerates Edlib** (Bioinformatics'17) and **Parasail** (BMC Bioinformatics'16) by
  - up to **37.7 × and 43.9 ×** ( $>12 \times$  on average), on CPUs
  - up to **413 × and 689 ×** ( $>400 \times$  on average) with **FPGA/GPU acceleration**
- ❑ Using long reads, SneakySnake **accelerates Parasail** and **KSW2** by **140.1 × and 17.1 ×** on average, respectively, on CPUs



# Takeaways

# Key Takeaways

---

- A **novel** method to **accelerate Sequence Alignment** in genome analysis.
- Simple and effective
- Hardware/software cooperative
- Good potential for work **building on it** to extend it
  - To make things more efficient and effective
  - Multiple works have already built on the paper (see MAGNET, Shouji, GRIM-Filter, SneakySnake, GenCache)
- Easy to read and understand paper

# Open Discussion

# Discussion Starters (I)

---

- Thoughts on the previous ideas?
- Rethinking Alignment and Pre-alignment?
  - Re-use the results of the pre-alignment filter?
  - Improve the accuracy of pre-alignment filtering to achieve an optimal alignment?
- Extend the solution to longer reads, higher edit distance thresholds?
- Is this solution clearly advantageous in some cases?

# Discussion Starters (II)

---

- Data movement is still a bottleneck. How could we try to reduce it?
  - ❑ Placing the accelerator **closer to memory**
  - ❑ Using newer and faster **I/O**
  - ❑ Closely integrate the accelerator into sequencers for **real-time pre-alignment filtering**
  - ❑ Offer **cloud computing** with access to advanced FPGA chips

Can We Do Better?

---

Alleviating  
Data Movement  
Bottlenecks

# Near-memory Pre-alignment Filtering

Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gomez-Luna, Henk Corporaal, Onur Mutlu,

## [“FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications”](#)

IEEE Micro, 2021.

[\[Source Code\]](#)



[Home](#) / [Magazines](#) / [IEEE Micro](#) / [2021.04](#)

*IEEE Micro*

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

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

DOI Bookmark: [10.1109/MM.2021.3088396](#)

### Authors

[Gagandeep Singh](#), ETH Zürich, Zürich, Switzerland

[Mohammed Alser](#), ETH Zürich, Zürich, Switzerland

[Damla Senol Cali](#), Carnegie Mellon University, Pittsburgh, PA, USA

[Dionysios Diamantopoulos](#), Zürich Lab, IBM Research Europe, Rüschlikon, Switzerland

[Juan Gomez-Luna](#), ETH Zürich, Zürich, Switzerland

[Henk Corporaal](#), Eindhoven University of Technology, Eindhoven, The Netherlands

[Onur Mutlu](#), ETH Zürich, Zürich, Switzerland

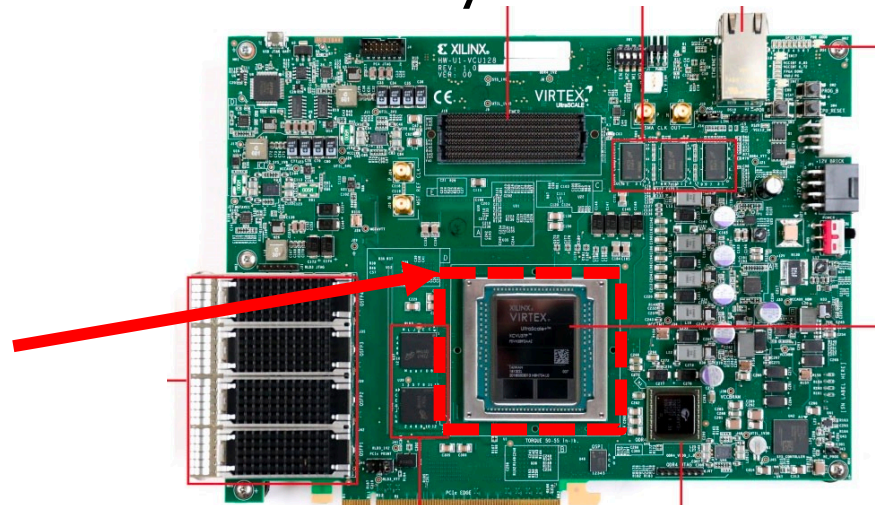
<a href="#">◀</a>	<a href="#">▶</a>
<a href="#">Previous</a>	<a href="#">Next</a>
<a href="#">☰</a>	<a href="#">Table of Contents</a>
<a href="#">📖</a>	<a href="#">Past Issues</a>

# Near-memory SneakySnake

---

- Problem: Read Mapping is heavily bottlenecked by data movement from main memory
- Solution: Perform read mapping near where data resides (i.e., near-memory)
- We carefully redesigned the accelerator logic of SneakySnake to exploit near-memory computation capability on modern FPGA boards with high-bandwidth memory

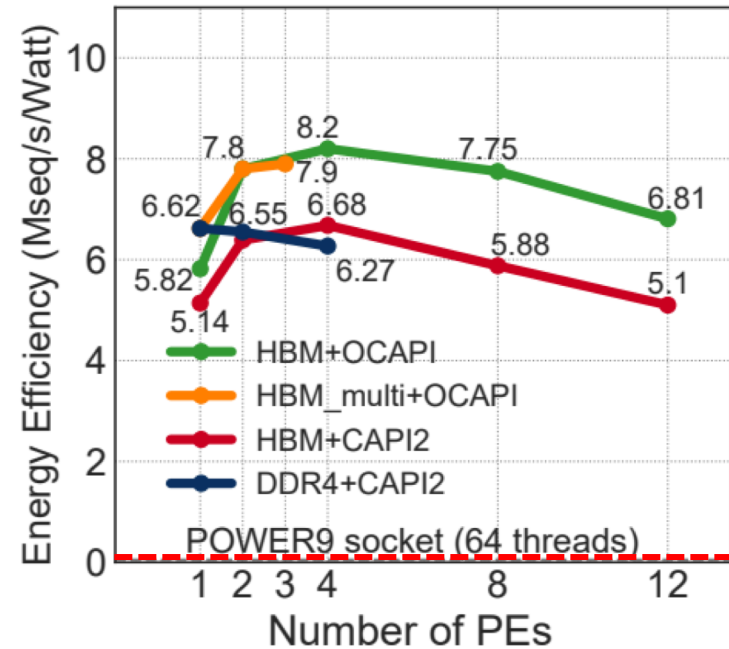
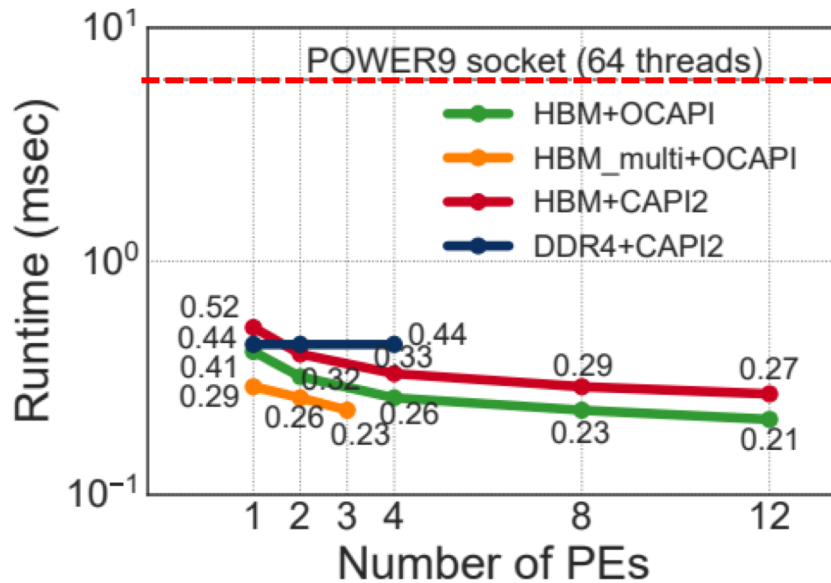
FPGA + high-bandwidth memory  
on the same package substrate



Xilinx Virtex UltraScale+ HBM VCU128 FPGA



# Key Results of Near-memory SneakySnake



**Near-memory** pre-alignment filtering improves **performance** and **energy efficiency** by  $27.4 \times$  and  $133 \times$ , respectively, over a 16-core (64 hardware threads) IBM POWER9 CPU

# 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.*  
[[Lightning 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<sup>†⋈</sup> Gurpreet S. Kalsi<sup>⋈</sup> Zülal Bingöl<sup>▽</sup> Can Firtina<sup>◇</sup> Lavanya Subramanian<sup>‡</sup> Jeremie S. Kim<sup>◇†</sup>  
Rachata Ausavarungnirun<sup>⊙</sup> Mohammed Alser<sup>◇</sup> Juan Gomez-Luna<sup>◇</sup> Amirali Boroumand<sup>†</sup> Anant Nori<sup>⋈</sup>  
Allison Scibisz<sup>†</sup> Sreenivas Subramoney<sup>⋈</sup> Can Alkan<sup>▽</sup> Saugata Ghose<sup>\*†</sup> Onur Mutlu<sup>◇†▽</sup>  
<sup>†</sup>Carnegie Mellon University   <sup>⋈</sup>Processor Architecture Research Lab, Intel Labs   <sup>▽</sup>Bilkent University   <sup>◇</sup>ETH Zürich  
<sup>‡</sup>Facebook   <sup>⊙</sup>King Mongkut's University of Technology North Bangkok   <sup>\*</sup>University of Illinois at Urbana-Champaign

# Near-memory GenASM Framework

---

- **Our goal:** Accelerate approximate string matching (ASM) by designing a fast and flexible framework, which can accelerate multiple steps of genome sequence analysis.
- **Key ideas:** Exploit the high memory bandwidth and the logic layer of 3D-stacked memory to perform highly-parallel ASM in the DRAM chip itself.
- Modify and extend Bitap<sup>1,2</sup>, ASM algorithm with fast and simple bitwise operations, such that it now:
  - Supports long reads
  - Supports traceback
  - Is highly parallelizable
- Co-design of our modified scalable and memory-efficient algorithms with low-power and area-efficient hardware accelerators

[1] R. A. Baeza-Yates and G. H. Gonnet. "A New Approach to Text Searching." *CACM*, 1992.

[2] S. Wu and U. Manber. "Fast Text Searching: Allowing Errors." *CACM*, 1992.

# 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 of the GenASM Framework

---

## (1) Read Alignment

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

## (2) Pre-Alignment Filtering

- $3.7\times$  speedup,  $1.7\times$  less power than **Shouji** (state-of-the-art **HW**)

## (3) Edit Distance Calculation

- $22\text{--}12501\times$  speedup,  $548\text{--}582\times$  less power than **Edlib** (state-of-the-art **SW**)
- $9.3\text{--}400\times$  speedup,  $67\times$  less power than **ASAP** (state-of-the-art **HW**)

# Discussion Starters (III)

---

- Can you think of fields that could be similarly in need of string alignment as in read mapping in bioinformatics?
- Natural language processing
  - OCR error correction
  - Autocorrection in text-based editors or apps
  - Reconstruction of languages using the comparative method
  - Social sciences

Combining dynamic programming with filtering to solve a four-stage two-dimensional guillotine-cut bounded knapsack problem

François Clautiaux<sup>a,b,\*</sup>, Ruslan Sadykov<sup>b,a</sup>, François Vanderbeck<sup>a,b</sup>, Quentin Viaud<sup>a,b</sup>

<sup>a</sup>IMB, Université de Bordeaux, 351 cours de la Libération, 33405 Talence, France

<sup>b</sup>INRIA Bordeaux - Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence, France

Clautiaux+, "[Combining dynamic programming with filtering to solve a four-stage two-dimensional guillotine-cut bounded knapsack problem](#)", *Discrete Optimization*, 2018.

---

# Adoption of hardware accelerators in genome analysis

# Bioinformatics: Reviewer #6 (Dec. 2016)

---

**I have a major concern with the work that is actually not a problem with the manuscript at all.** Specifically, I have the concern that there has been little to no adoption of previous specialized hardware solutions related to improving the speed of alignment. While there has been considerable work in this area (which the authors do an admirable job of citing), it does not seem that these hardware-based solutions have gained any type of real traction in the community, as the vast majority of alignment is still performed on “regular” CPUs, where the extent of hardware acceleration is the adoption of specific SIMD or vectorized instructions. While I don’t think that this practical concern should preclude publication of the current work, it is something worth considering (e.g. what, if any, of the proposed improvements to the SHD filter could be “back-ported” to a software-only solution).



# Our Response

---

We see the reviewer's point, but we do not believe this should be held against the research in the area of FPGA-based acceleration of read mapping in particular or genomics in general. It always takes time to adopt a "new" or "different" hardware technology since it requires investment into the hardware infrastructure. The main challenges/barriers that limit the popularity of FPGAs in the genomics field are the high cost, design effort, and development time. Due to the fact that the deliverable of such projects is normally a hardware product, researchers tend to commercialize their research with startup companies and engage themselves with industrial collaborators, as we describe below. Today, the cost structure of FPGAs is changing because major cloud infrastructures (e.g., by Microsoft Azure and Amazon AWS) offer FPGAs as core engines of the infrastructure. Therefore, we believe the benefits of FPGA-based acceleration has become available to many more folks in the community, especially with the open-source release of such FPGA-accelerated solutions. To increase adoption, we have decided to release our source code for GateKeeper. It is available on <https://github.com/BilkentCompGen/GateKeeper>.

Some examples of the research groups that commercialize their research and promote FPGA-based or even cloud-based products for genomics are as follows:

<http://www.timelogic.com/catalog/775>

<http://www.gidel.com/HPC-RC/HPC-Applications.asp>

[http://www.edicogenome.com/dragen\\_bioit\\_platform/the-dragen-engine-2/](http://www.edicogenome.com/dragen_bioit_platform/the-dragen-engine-2/)

<http://www.bcgsc.ca/platform/bioinfo/software/XpressAlign/releases/1.0>

<https://www.sevenbridges.com/amazon/>

<http://www.falcon-computing.com/index.php/solutions/falcon-genomics-solutions/>

# Our Response (cont'd)

---

It is also important to emphasize that the necessity of designing a mapper on hardware is currently steering the field towards more personalized medicine. Hardware-accelerated mappers (using various platforms such as SIMD, GPUs, and FPGAs) are becoming increasingly popular as they can be potentially directly integrated into sequencing machines (the Illumina sequencer, for example, includes an FPGA chip inside it

[https://support.illumina.com/content/dam/illumina-support/documents/downloads/software/hiseq/hcs\\_2-0-12/installnotes\\_hcs2-0-12.pdf](https://support.illumina.com/content/dam/illumina-support/documents/downloads/software/hiseq/hcs_2-0-12/installnotes_hcs2-0-12.pdf) ), such that we have a single machine that can perform both sequencing and mapping (Lindner, et al., Bioinformatics 2016). This approach has two benefits. First, it can hide the complexity and details of the underlying hardware from users who are not necessarily aware about FPGAs (e.g., biologists and mathematicians). Second, it allows a significant reduction in total genome analysis time by starting read mapping while still sequencing. Hence, an end user or researcher in genomics might not directly deal with the “pre-alignment on FPGA” or “mapper on FPGA”, but they might purchase a sequencer that performs pre-alignment and alignment using FPGAs inside. As such, one potential target of our research is to influence the design of more intelligent sequencing machines by integrating GateKeeper inside them.

In fact, we believe GateKeeper is very suitable to be used as part of a sequencer as it provides a complete pre-alignment system that includes many processing cores, where all processing cores work in parallel to provide extremely fast filtering. We believe such a fast approach can make sequencers more intelligent and attractive.

# Remember What We Said in the First Lecture

---

Dream  
and, they will come

- Computing landscape is very different from 10-20 years ago.
- As applications push boundaries, computing platforms will become increasingly strained.

# Illumina DRAGEN Bio-IT Platform (2018)

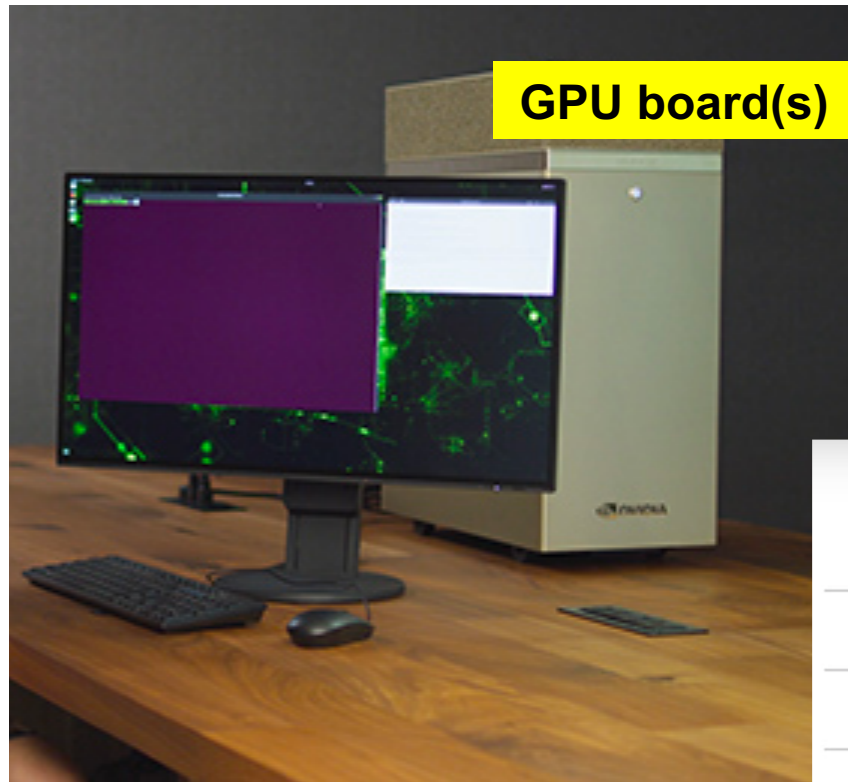
- Processes whole genome at 30x coverage in ~25 minutes with hardware support for data compression



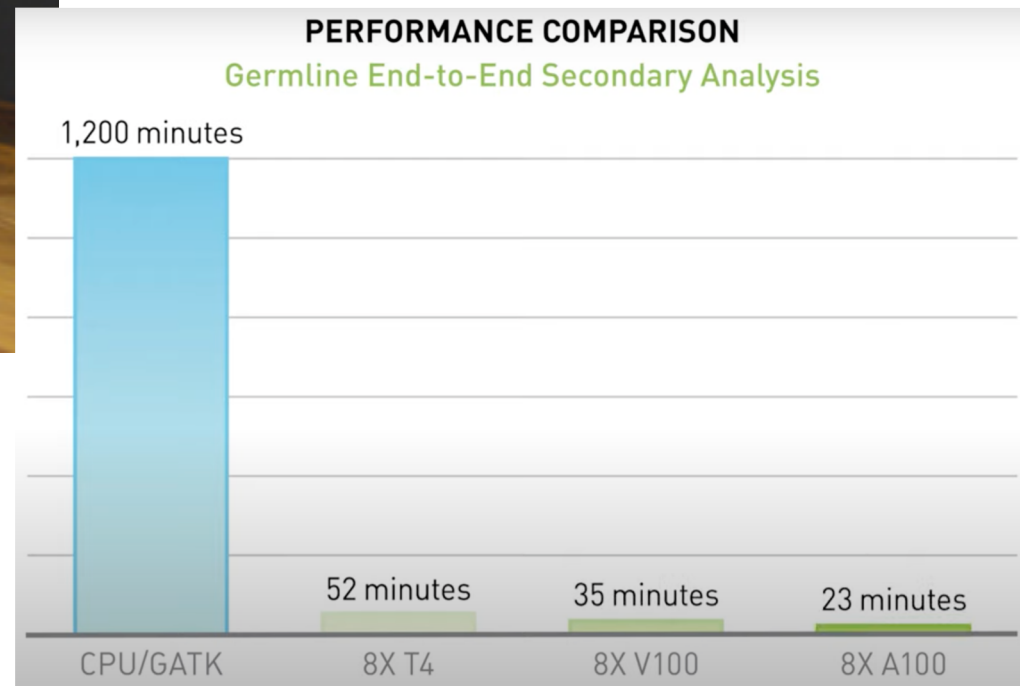
[emea.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html](https://emea.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html)  
[emea.illumina.com/company/news-center/press-releases/2018/2349147.html](https://emea.illumina.com/company/news-center/press-releases/2018/2349147.html)



# NVIDIA Clara Parabricks (2020)



**A University of Michigan's startup in 2018 and joined NVIDIA in 2020**



---

# Computing is Still Bottlenecked by Data Movement

# Adoption Challenges of Hardware Accelerators

---

- Accelerate the **entire read mapping** process rather than its **individual** steps (**Amdahl's law**)
- Reduce the high amount of **data movement**
  - Working directly on **compressed** data
  - Filter out **unlikely-reused data** at the very first component of the compute system
- Develop **flexible** hardware architectures that do NOT conservatively **limit the range** of supported **parameter values** at design time
- Adapt existing genomic **data formats** for hardware accelerators or develop more **efficient file formats**

# Adoption Challenges of Hardware Accelerators

---

- Maintaining the same (or better) **accuracy/sensitivity** of the output results of the **software** version
  - Using **heuristic** algorithms to gain speedup!
- High hardware **cost**
- Long **development life-cycle** for FPGA platforms



---

What **else** can be **done**?

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

.FASTA file



Reference  
genome

.FASTQ file



Reads

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

# AirLift [Kim+, arXiv 2021]

---

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

["AirLift: A Fast and Comprehensive Technique for Translating Alignments between Reference Genomes"](#), arXiv, 2021

[\[Source Code\]](#)

[\[Online link at arXiv\]](#)

## RESEARCH

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

Jeremie S. Kim<sup>1</sup>, Can Firtina<sup>1</sup>, Meryem Banu Cavlak<sup>2</sup>, Damla Senol Cali<sup>3</sup>, Nastaran Hajinazar<sup>1,4</sup>,  
Mohammed Alser<sup>1</sup>, Can Alkan<sup>2</sup> and Onur Mutlu<sup>1,2,3\*</sup>

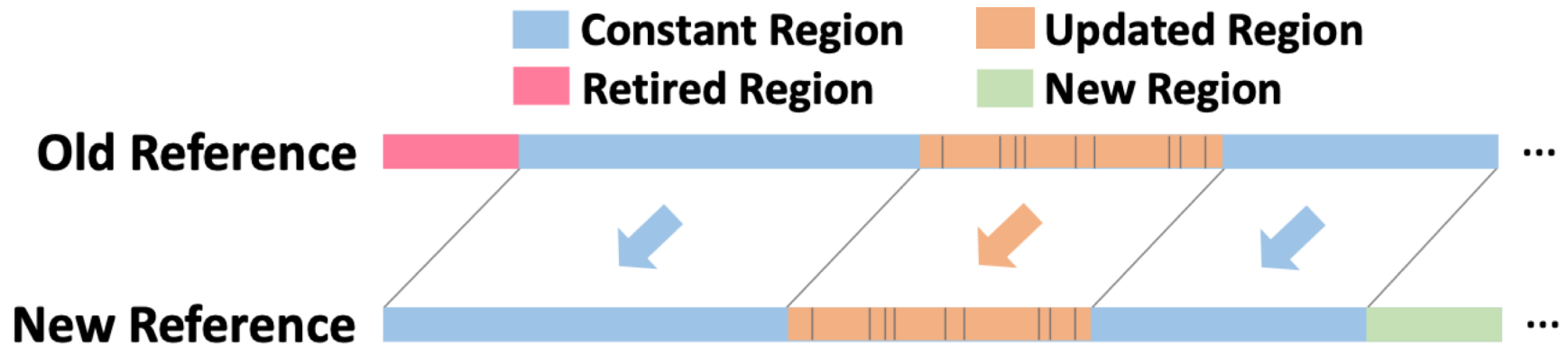
# AirLift

---

- **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 99.99%
  - ❑ reduces overall runtime to re-map reads by 6.7x, 6.6x, and 2.8x for large (human), medium (C. elegans), and small (yeast) reference genomes

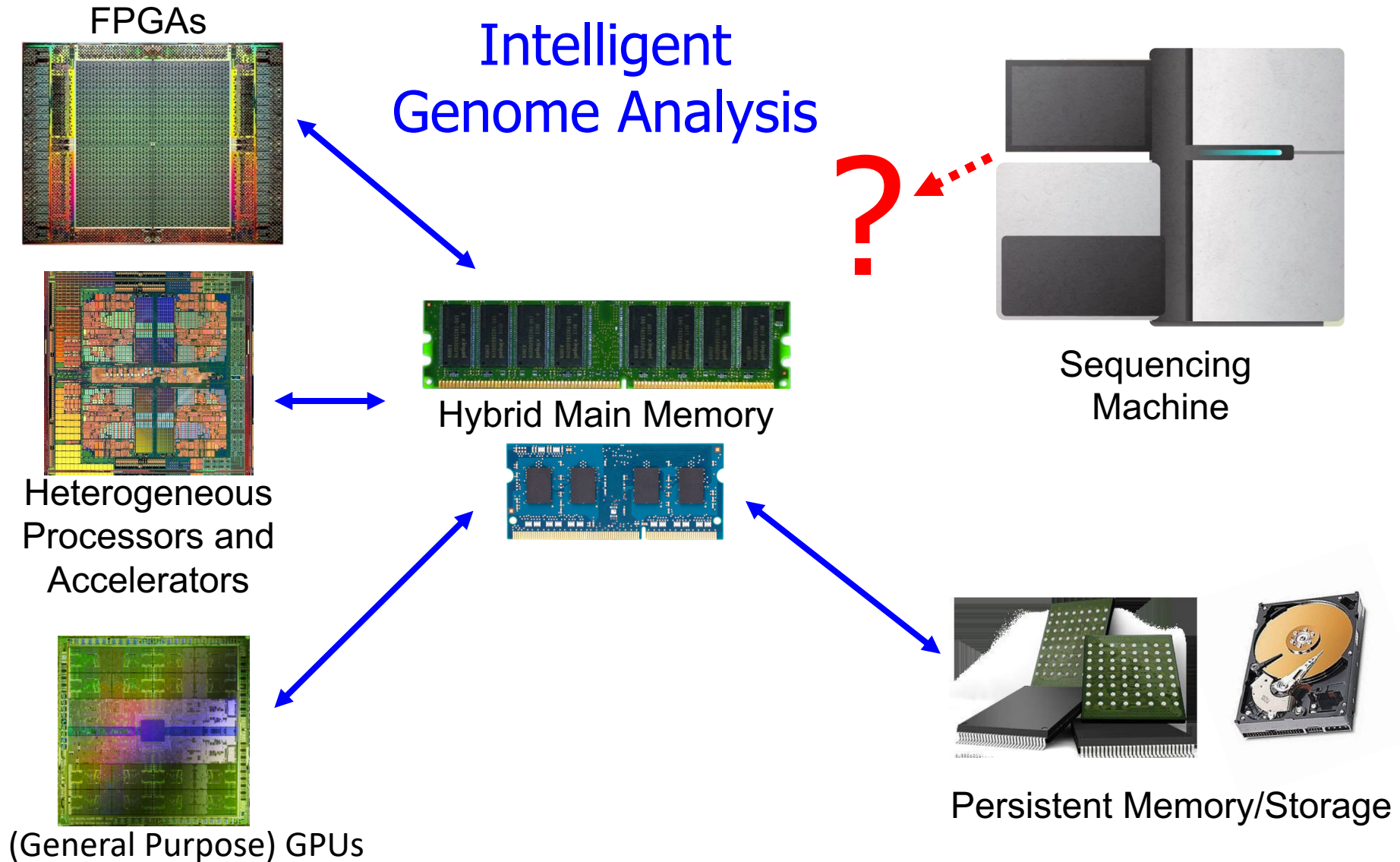
# Clustering the Reference Genome Regions

---



**Fig. 2.** Reference Genome Regions.

# Processing Genomic Data Where it Makes Sense



# What is Intelligent Genome Analysis?

---

- Fast genome analysis

- *Real-time analysis*

Bandwidth

- Using intelligent architectures

- *Specialized HW with less data movement*

Energy-efficiency &  
Latency

- DNA is a valuable asset

- *Controlled-access analysis*

Privacy

- Population-scale genome analysis

- *Sequence anywhere at large scale!*

Scalability

- Avoiding erroneous analysis

- *E.g., your father is not your father*

Accuracy

# Achieving Intelligent Genome Analysis?

---

How and where to enable

fast, accurate, cheap,

privacy-preserving, and exabyte scale

analysis of genomic data?



---

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

# Read Mapping in 111 pages!

In-depth analysis of 107 read mappers (1988-2020)

**Mohammed Alser**, Jeremy Rotman, Dhrithi Deshpande, Kodi Taraszka, Huwenbo Shi, Pelin Icer Baykal, Harry Taegyun Yang, Victor Xue, Sergey Knyazev, Benjamin D. Singer, Brunilda Balliu, David Koslicki, Pavel Skums, Alex Zelikovsky, Can Alkan, Onur Mutlu, Serghei Mangul

["Technology dictates algorithms: Recent developments in read alignment"](#)

Genome Biology, 2021

[\[Source code\]](#)

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


Genome Biology

REVIEW

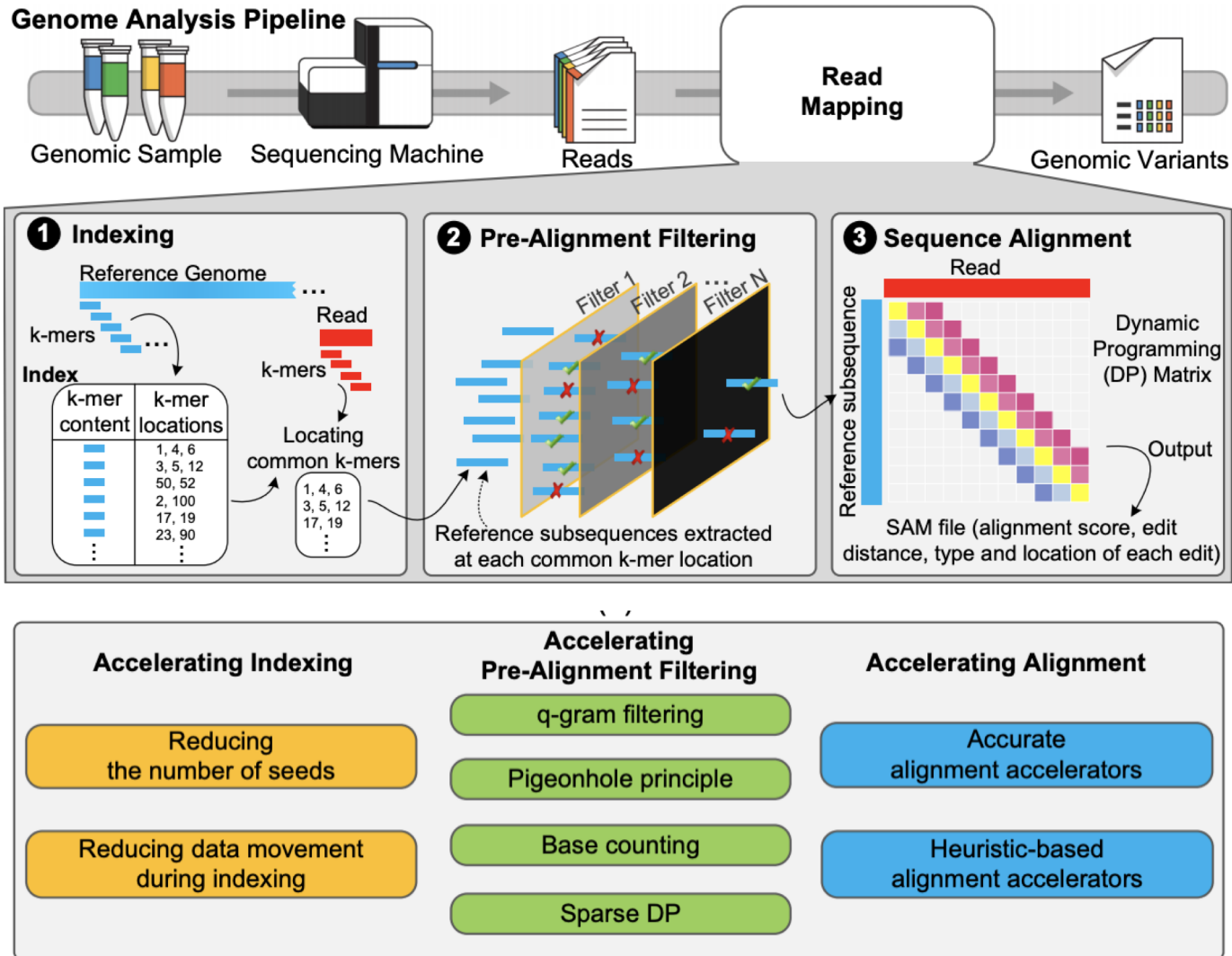
Open Access

## Technology dictates algorithms: recent developments in read alignment



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

# Accelerating Read Mapping



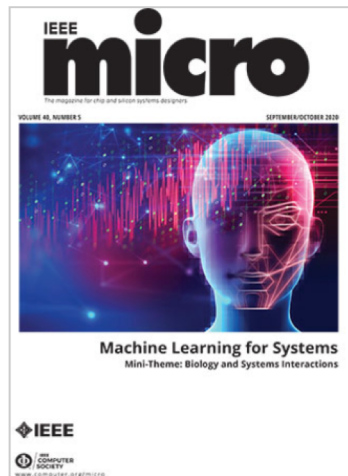
Alser+, “[Accelerating Genome Analysis: A Primer on an Ongoing Journey](#)”, IEEE Micro, 2020.

# Detailed Analysis of Tackling the Bottleneck

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

[“Accelerating Genome Analysis: A Primer on an Ongoing Journey”](#)

IEEE Micro, August 2020.



[Home](#) / [Magazines](#) / [IEEE Micro](#) / [2020.05](#)

*IEEE Micro*

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

Sept.-Oct. 2020, pp. 65-75, vol. 40

DOI Bookmark: [10.1109/MM.2020.3013728](#)

### Authors

[Mohammed Alser](#), ETH Zürich

[Zulal Bingöl](#), Bilkent University

[Damla Senol Cali](#), Carnegie Mellon University

[Jeremie Kim](#), ETH Zurich and Carnegie Mellon University

[Saugata Ghose](#), University of Illinois at Urbana-Champaign and Carnegie Mellon University

[Can Alkan](#), Bilkent University

[Onur Mutlu](#), ETH Zurich, Carnegie Mellon University, and Bilkent University



Previous



Next



Table of Contents



Past Issues

# More on Fast Genome Analysis ...

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

Insight: Shifting a String Helps Similarity Search

7 matches 1 mismatch

I S T A N B U L

I S T N B U L

I S T N B U L

81

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

566 views · Premiered Feb 6, 2021

Onur Mutlu Lectures  
13.9K subscribers

ANALYTICS EDIT VIDEO

# More on Intelligent Genome Analysis ...

Our Solution: GateKeeper

The diagram illustrates the GateKeeper solution. It starts with 'High throughput DNA sequencing (HTS) technologies' (labeled 1) producing 'Billions of Short Reads'. These are processed by 'Read Pre-Alignment Filtering' (labeled 2), which is described as 'Fast & Low False Positive Rate'. This step reduces the volume from  $\times 10^{12}$  mappings to  $\times 10^3$  mappings. The filtered data then undergoes 'Read Alignment' (labeled 3), described as 'Slow & Zero False Positives', resulting in a final set of mappings. A video player interface is overlaid on the diagram, showing a progress bar at 2:08:58 / 2:54:18 and the title 'GateKeeper >'. A small video feed of the presenter is visible in the top right corner of the player.

1 High throughput DNA sequencing (HTS) technologies

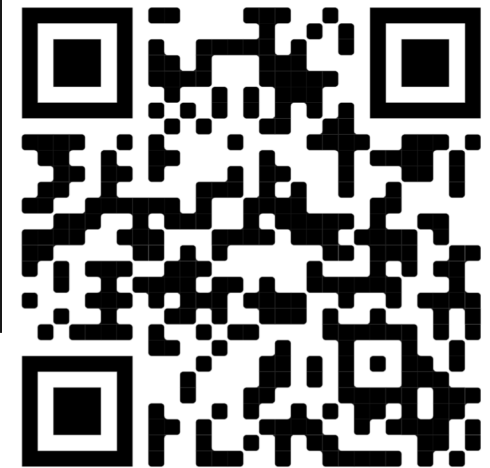
2 Read Pre-Alignment Filtering  
Fast & Low False Positive Rate

3 Read Alignment  
Slow & Zero False Positives

108

ETH ZENTRUM

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



<https://www.youtube.com/watch?v=ygmQpdDTL7o>

# 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>

# Prior Research on Genome Analysis (1/2)

---

- Alser + ["SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs."](#), *Bioinformatics*, 2020.
- Senol Cali+, ["GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"](#), *MICRO* 2020.
- Alser+, ["Technology dictates algorithms: Recent developments in read alignment"](#), *arXiv*, 2020.
- Kim+, ["AirLift: A Fast and Comprehensive Technique for Translating Alignments between Reference Genomes"](#), *arXiv*, 2020
- Alser+, ["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#), *IEEE Micro*, 2020.



# Prior Research on Genome Analysis (2/2)

---

- Firtina+, "[Apollo: a sequencing-technology-independent, scalable and accurate assembly polishing algorithm](#)", *Bioinformatics*, 2019.
- Alser+, "[Shouji: a fast and efficient pre-alignment filter for sequence alignment](#)", *Bioinformatics* 2019.
- Kim+, "[GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies](#)", *BMC Genomics*, 2018.
- Alser+, "[GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping](#)", *Bioinformatics*, 2017.
- Alser+, "[MAGNET: understanding and improving the accuracy of genome pre-alignment filtering](#)", *IPSI Transaction*, 2017.

# 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, 2017**

Presented by: Mohammed Alser



**Bilkent University**



TOBB  
UNIVERSITY OF  
ECONOMICS & TECHNOLOGY

**ETH** zürich **Carnegie Mellon**

Thank you. Questions?

# Seminar in Computer Architecture Meeting 2: GateKeeper

Dr. Mohammed Alser

 @mealser

ETH Zurich

Fall 2021

30 September 2021

**SAFARI**

**ETH** zürich