

# P&S Genomics

## Lecture 6b: MAGNET & Shouji

Dr. Mohammed Alser

ETH Zürich

Spring 2023

6 April 2023

---

We need intelligent algorithms  
and intelligent architectures  
that handle data well

# Goal: Minimizing Alignment Time

---

Sequence Alignment is **expensive**

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

# Key Idea

---

Genomic Strings

```
graph TD; A[Genomic Strings] --> B[Dissimilar Strings]; A --> C[Similar Strings];
```

**EXPENSIVE!**

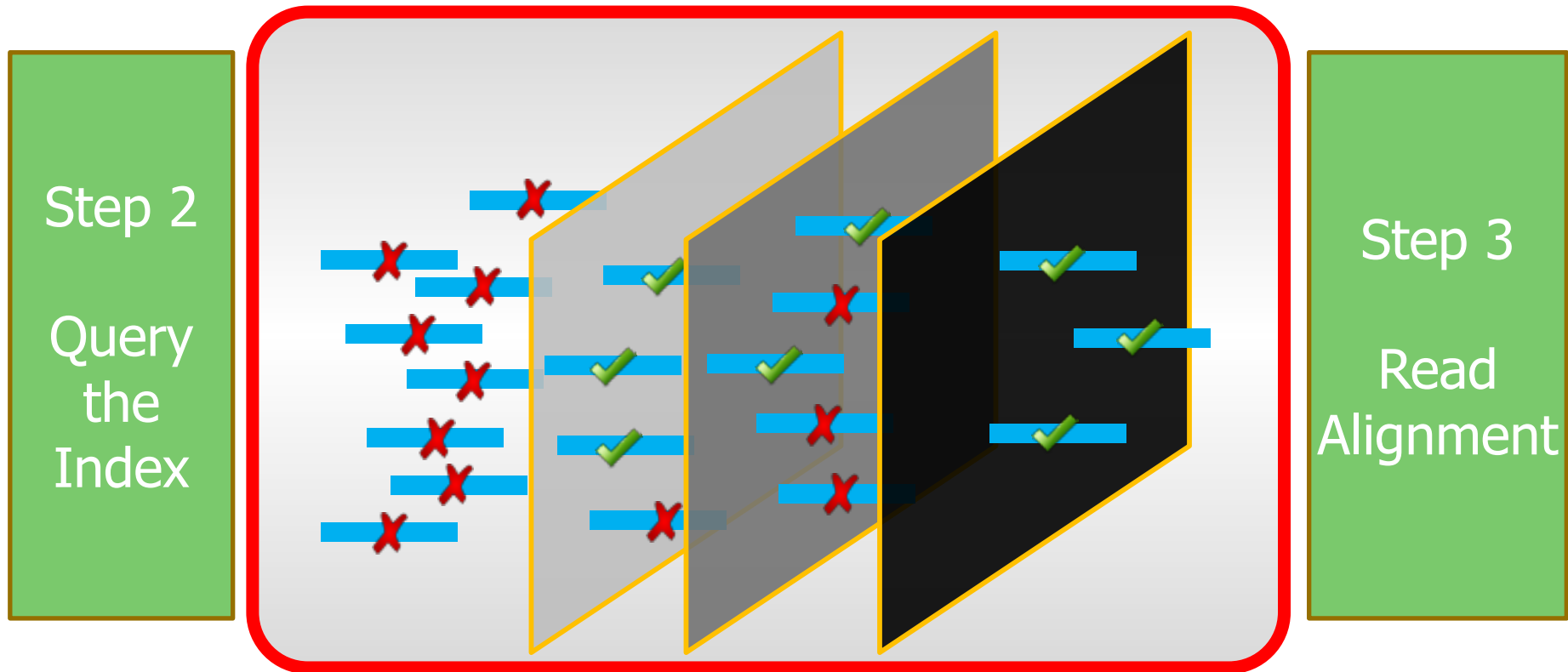
Dissimilar  
Strings

Ignore them if the number  
of differences exceeds a  
threshold.

Similar  
Strings

Find number, location, and  
type of differences?

# Ideal Filtering Algorithm



1. **Filter out** most of dissimilar sequences.
2. **Preserve** all similar sequences.
3. Do it **quickly**.

# MAGNET

---

Alser, Mohammed, Onur Mutlu, and Can Alkan.

["MAGNET: understanding and improving the accuracy of genome pre-alignment filtering"](#)

*IPSI Transaction* (2017).

[\[Source code\]](#)

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

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

# On the False Positives of GateKeeper

---

We investigate four major sources of false positives:

- Leading and trailing zeros
- Random zeros
- Conservative counting
- Lack of backtracking







# Conservative Counting

The 3-bit ones are a result of substitutions and not the amendment

```
Query : AAAAAACAAACAACCCCATCAAAAAGTGGGTGAAGGATATGAATTCACACTTCTCAAAGAAGACATTCTCAGCCAAAAAACACATGAAAAATGCTC
Reference : AAAAAACAAACAACCCCATCAAAAAGTGGGTGAAGGATATGAACAGACACTTCTCAAAGAAGACATTTACTCAGCCAAAAAACACATGAAAAATGCT
Hamming Mask : 0000000000000000000000000000000000000000001110000000000000000001111111110000011111110000011111
1-Deletion Mask : 00000001111111100011110000111111111111111111111111111111111111111100011111111111000000000000000
2-Deletion Mask : 00000001111111111111110001111111111111111111111111111111000111111111111111111111100000111111000001111
3-Deletion Mask : 000000011111000111111111111111111111111111111111111111111111111111100001111111111111100001000111100001111
1-Insertion Mask : 00000011111111000111100001111111111111111111111111111111111111111110001111111111111111111110001111100001111100
2-Insertion Mask : 0000011111111111111111000111111111111111111111111111111111111111111111111111111111111000111111110001111100
3-Insertion Mask : 00001111110001111111111111111111111111111111111111111111111111111111111111111111111111111111111111000
Final bit-vector : 000000000000000000000000000000000000000000000000000000001110000000000000000000100000000000000000000000000000000000
Needleman-Wunsch AAAAAACAAACAACCCCATCAAAAAGTGGGTGAAGGATATGAATTCACACTTCTCAAAGAAGACATTT-CTCAGCCAAAAAACACATGAAAAATGCT
Alignment:          ||| : ||| |
AAAAAACAAACAACCCCATCAAAAAGTGGGTGAAGGATATGAACAGACACTTCTCAAAGAAGACATTTACTCAGCCAAAAAACACATGAAAAATGCT
```

Misinterpreted as a single edit

Fig 5: An example of an incorrect mapping that passes the SHD filter due to conservative counting of the short streak of '1's in the final bit-vector.



---

Can we improve the **accuracy**?

# MAGNET (AACBB 2018, TIR 2017)

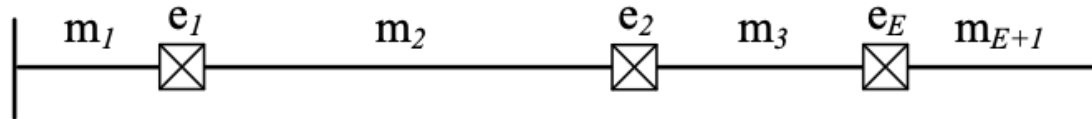
---

- **Key observation:**
  - Correct alignment always includes **non-overlapping long** identical subsequences.
- **Key idea:**
  - Count the **consecutive zeros** in each mask and select the longest in a divide-and-conquer approach.



# Number of Iterations in MAGNET

---



**Fig. 1: Random edit distribution in a read sequence. The edits ( $e_1, e_2, \dots, e_E$ ) act as dividers resulting in several identical subsequences ( $m_1, m_2, \dots, m_{E+1}$ ) between the read and the reference.**

# MAGNET Walkthrough

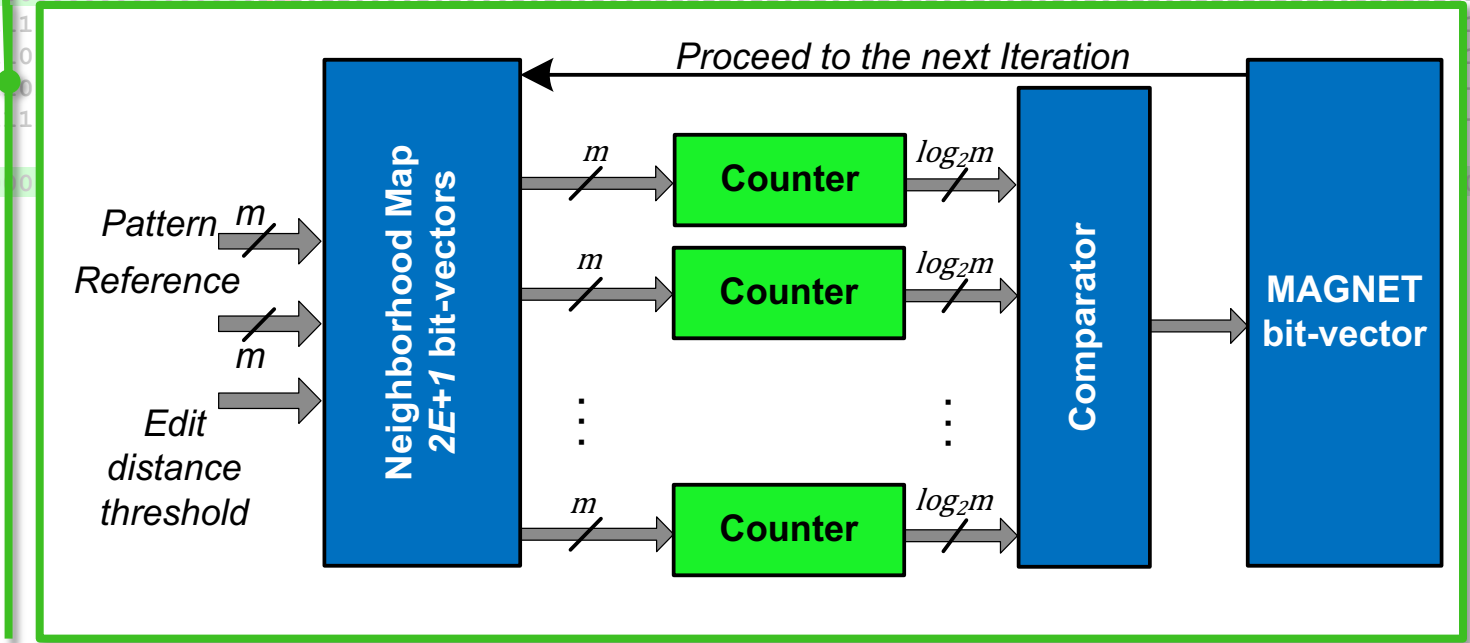
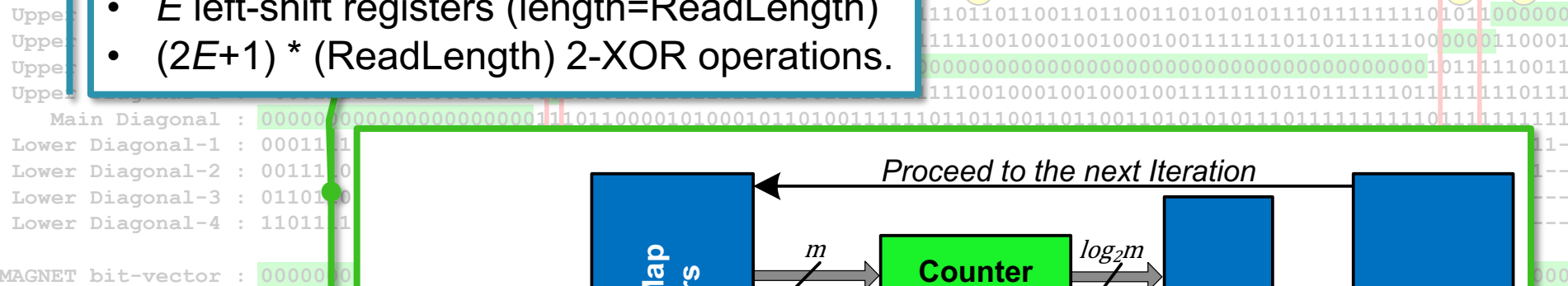
Build Neighborhood Map

Identifying  $E+1$  non-overlapping subsequences

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

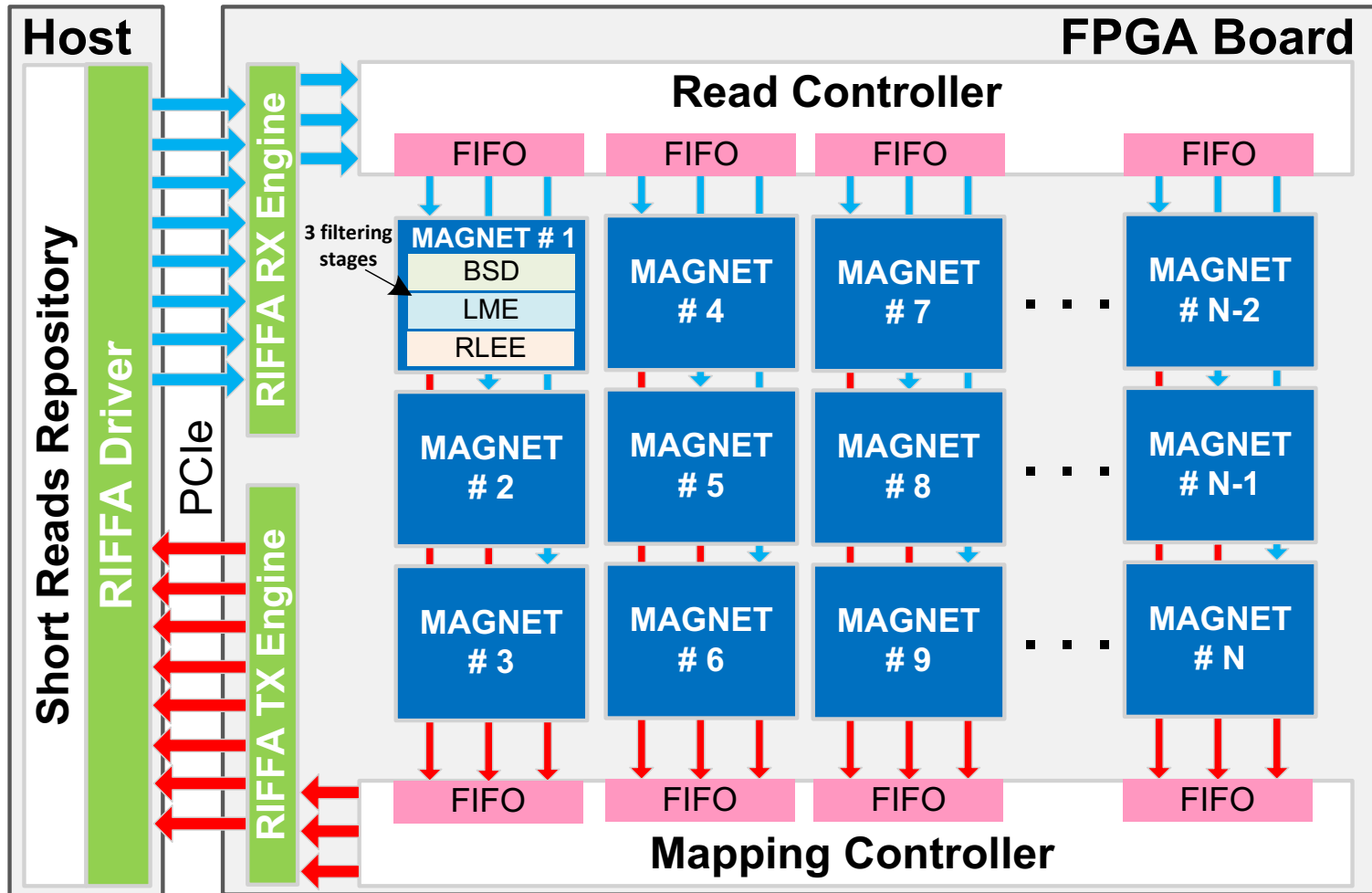
- $E$  right-shift registers (length=ReadLength)
- $E$  left-shift registers (length=ReadLength)
- $(2E+1) * (\text{ReadLength})$  2-XOR operations.

- $\log_2 \text{ReadLength}$ -bit counter.





# MAGNET Accelerator



# MAGNET (AACBB 2018, TIR 2017)

---

- **Key observation:**
    - Correct alignment always includes **non-overlapping long** identical subsequences.
  - **Key idea:**
    - Count the **consecutive zeros** in each mask and select the longest in a divide-and-conquer approach.
  - **Key result:**
    - MAGNET is 74x - 460x **faster** than its CPU implementation.
    - Contains up to **2 or 8 filtering units**, each of which has **10 folds the footprint** of that of GateKeeper on the FPGA.
    - MAGNET is 3.5x to 25552x (as GateKeeper stop filtering after  $E=4\%$ [250bp] or  $8\%$ [100bp]) **more accurate** than GateKeeper and SHD.
  - **Weaknesses:** Challenging to be implemented on FPGA due to random search.
-

# More on MAGNET

---

- Download and test for yourself

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

Alser, Mohammed, Onur Mutlu, and Can Alkan. "MAGNET: understanding and improving the accuracy of genome pre-alignment filtering." *IPSI Transaction* (2017).

---

Can we do **better?** Scalability?

---

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

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

# Shouji

---

- **Key observation:**
  - Correct alignment always includes **long identical subsequences**.
  - Processing the entire mapping at once (as in GateKeeper) is ineffective for hardware design.

# Dot Plot [Lipman and Pearson, 1985]

---

	T	G	T	G	C	A	G	G	G
T	*		*						
G		*		*			*	*	*
G		*		*			*	*	*
C					*				
A						*			
G		*		*			*	*	*
G		*		*			*	*	*

"dot plot" or "dot matrix" is a visual representation of the similarities between two closely similar genomic sequences that is used in FASTA/FASTP (Lipman and Pearson, 1985).

# Shouji

---

- **Key observation:**

- Correct alignment always includes **long identical subsequences**.
- Processing the entire mapping at once (as in GateKeeper) is ineffective for hardware design.

- **Key idea:**

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



# Building the Neighborhood Map

---

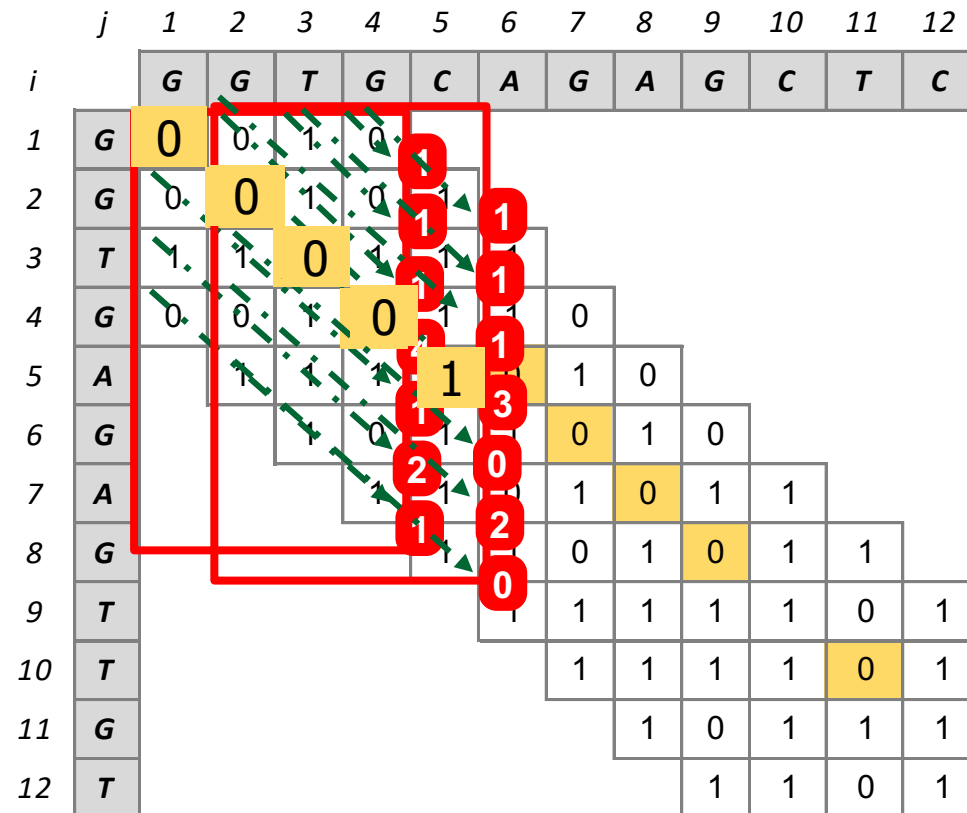
- Given a text sequence  $T[1\dots m]$ , a pattern sequence  $P[1\dots m]$ , and an edit distance threshold  $E$ , the neighborhood map represents the comparison result of the  $i^{\text{th}}$  character of  $P$  with the  $j^{\text{th}}$  character of  $T$ , where  $i$  and  $j$  satisfy  $1 \leq i \leq m$  and  $i-E \leq j \leq i+E$ . The entry  $N[i, j]$  of the neighborhood map can be calculated as follows:

$$N[i, j] = \begin{cases} 0, & \text{if } P[i] = T[j] \\ 1, & \text{if } P[i] \neq T[j] \end{cases} \quad (1)$$

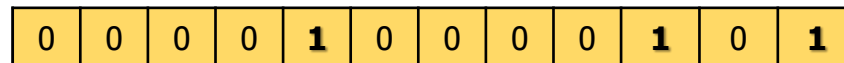
# Shouji Walkthrough

Building the Neighborhood Map

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



Storing it @ Shouji Bit-vector



ACCEPT iff number of '1' ≤ Threshold

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

# Shouji Walkthrough

Building the Neighborhood Map

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

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

Storing it @ Shouji Bit-vector

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

ACCEPT iff number of '1' ≤ Threshold

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

# What Does Shouji Mean?

---

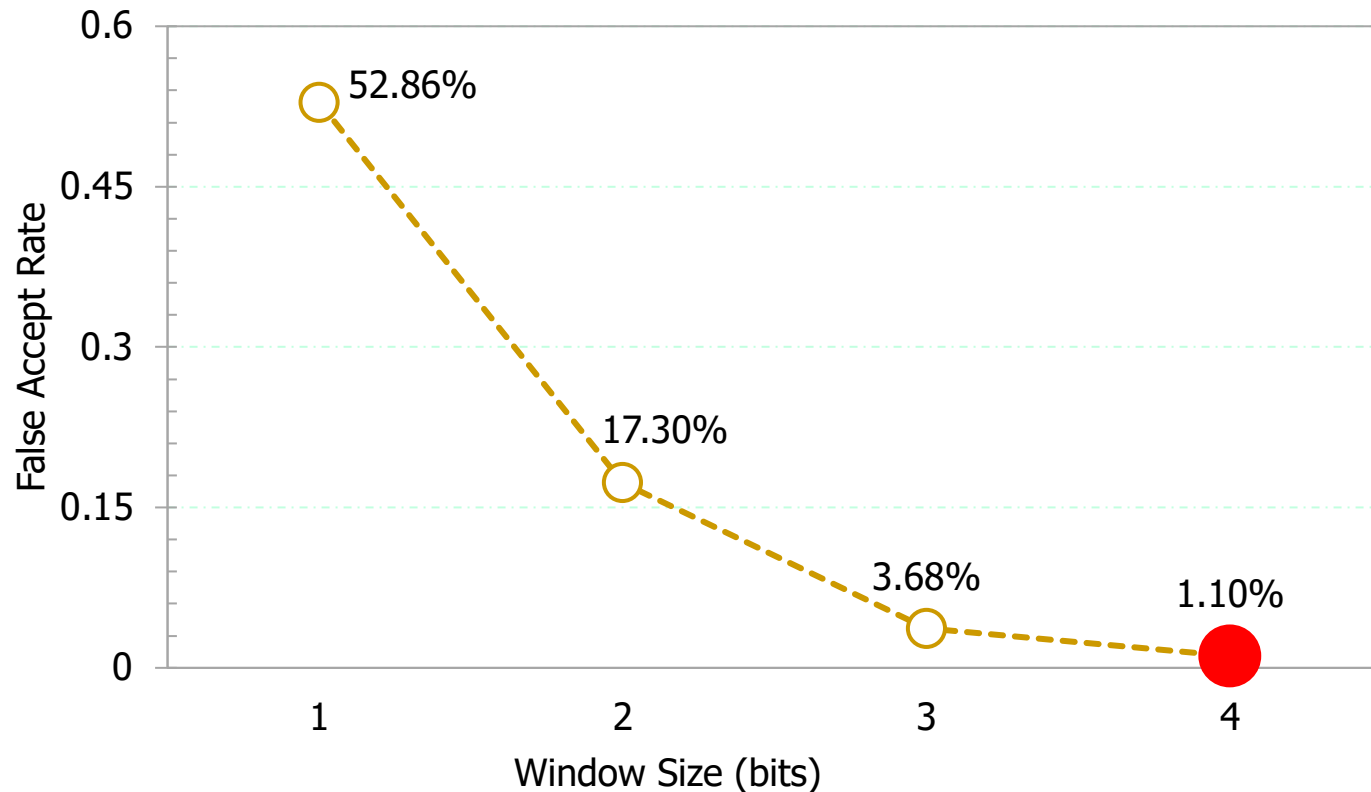


Named after a traditional Japanese door that is designed to slide open

<http://www.aisf.or.jp/~jaanus/deta/s/shouji.htm>.

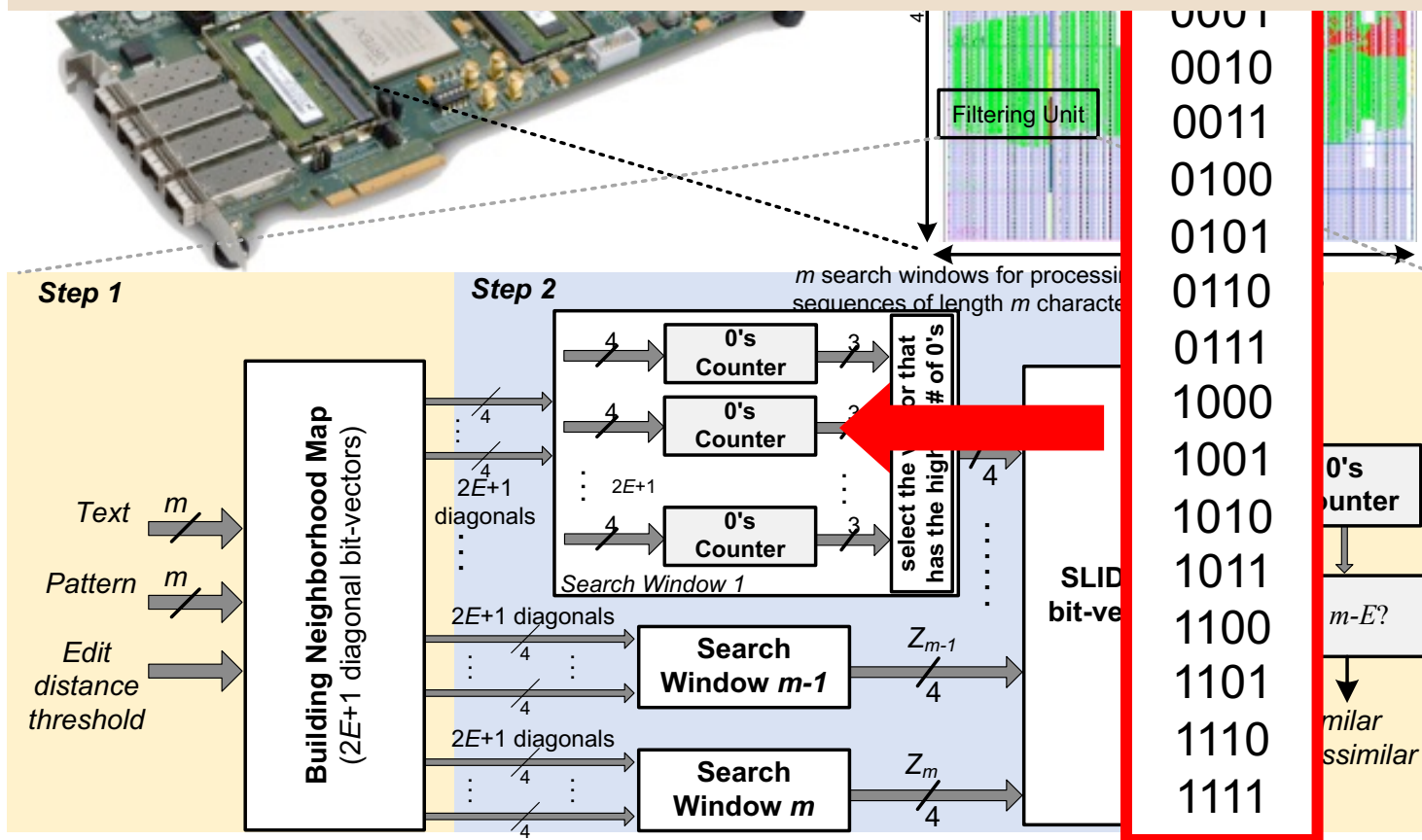
# Sliding Window Size

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



# Hardware Implementation

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



# Shouji

---

- **Key observation:**

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

- **Key idea:**

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

- **Key result:**

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

# More on Shouji

Download and test for yourself

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

*Bioinformatics*, 2019, 1–9

doi: 10.1093/bioinformatics/btz234

Advance Access Publication Date: 28 March 2019

Original Paper

OXFORD

---

Sequence alignment

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

**Mohammed Alser<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

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

<https://doi.org/10.1093/bioinformatics/btz234>



# Assignment #2

---

- Which one out of the three pre-alignment filters, GateKeeper, MAGNET, and Shouji is more efficient? why?
  - Hint:

	<b>GateKeeper</b>	<b>MAGNET</b>	<b>Shouji</b>	<b>Justification</b>
<b>Time complexity</b>				
<b>Space complexity</b>				
<b>FPGA resource</b>				
<b>Accuracy</b>				
<b>Speed</b>				

---

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

# More on GateKeeper [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** ▼

# 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

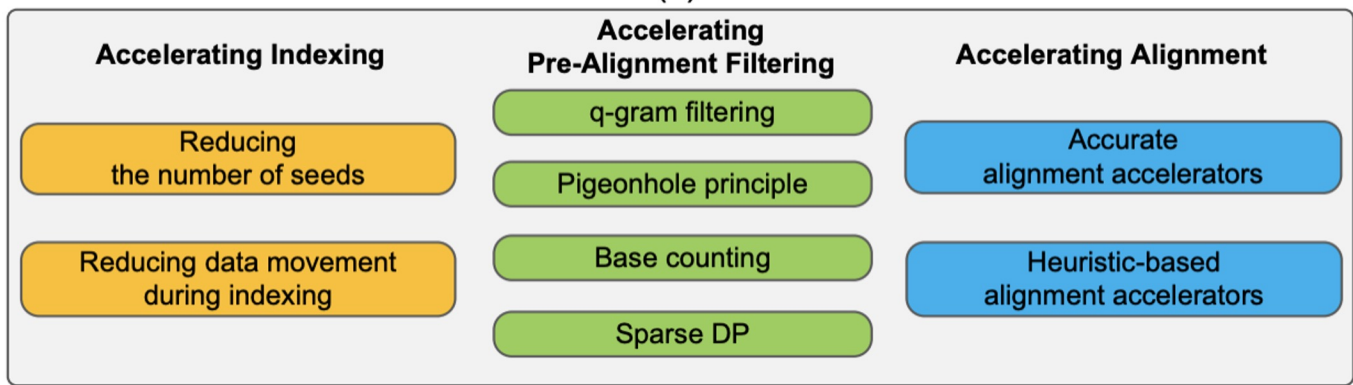
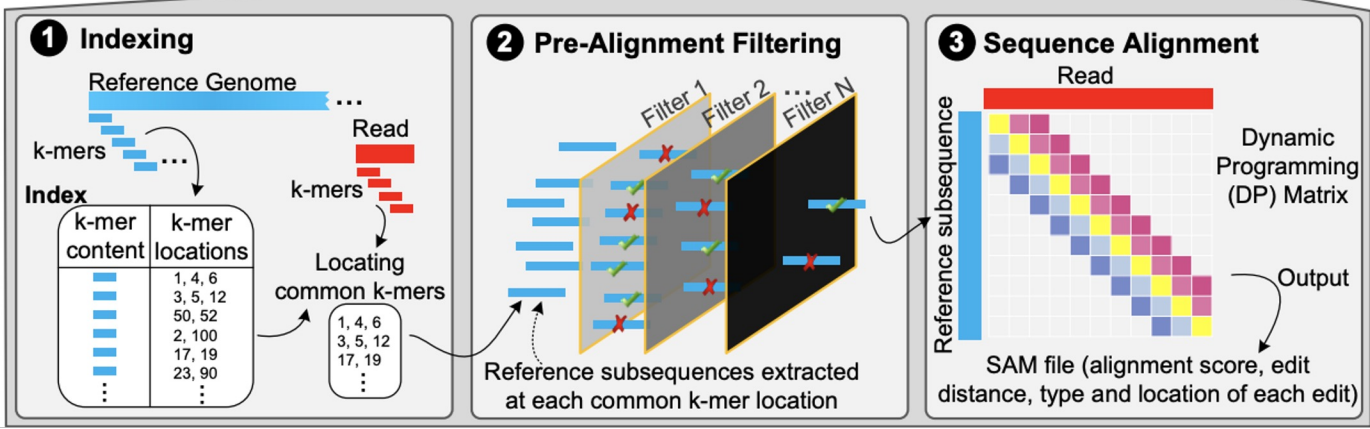
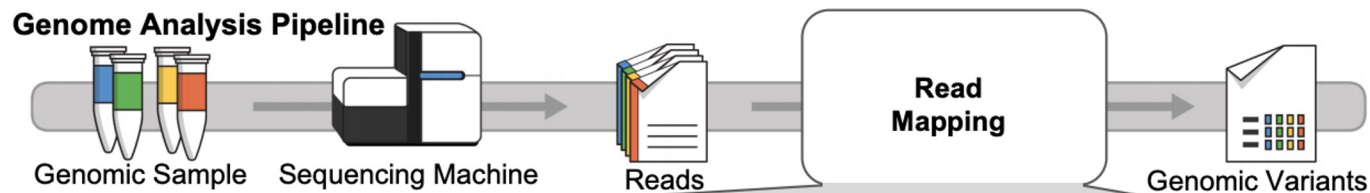
Open Access

## 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



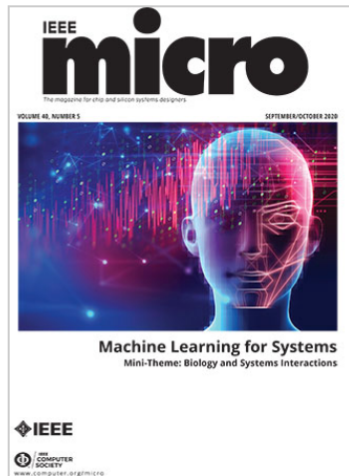
Alser+, "Accelerating Genome Analysis: A Primer on an Ongoing Journey", IEEE Micro, 2020.

# 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](https://doi.org/10.1109/MM.2020.3013728)

### Authors

[Mohammed Alser](#), ETH Zürich

[Zulal Bingol](#), Bilkent University

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

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

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

[Can Alkan](#), Bilkent University

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

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

# 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

46:08 / 1:37:37

Onur Mutlu

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

566 views · Premiered Feb 6, 2021

31 0 SHARE SAVE ...

 Onur Mutlu Lectures  
13.9K subscribers

ANALYTICS EDIT VIDEO

# 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) which produce 'Billions of Short Reads' (x10<sup>12</sup> mappings). These reads pass through 'Read Pre-Alignment Filtering' (labeled 2), which is 'Fast & Low False Positive Rate'. This step is shown as a 3D stack of filters with arrows indicating the flow of reads. The filters are categorized by speed and accuracy: 'Low Speed & High Accuracy', 'Medium Speed, Medium Accuracy', and 'High Speed, Low Accuracy'. The filtered reads then go to 'Read Alignment' (labeled 3), which is 'Slow & Zero False Positives', resulting in 'x10<sup>3</sup> mappings'. A small video inset in the top right shows a man speaking.

1 High throughput DNA sequencing (HTS) technologies

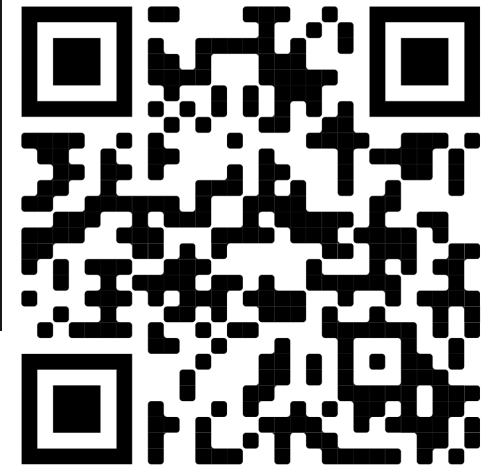
2 Read Pre-Alignment Filtering  
Fast & Low False Positive Rate

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.

# P&S Genomics

## Lecture 6b: MAGNET & Shouji

Dr. Mohammed Alser

ETH Zürich

Spring 2023

6 April 2023