Seminar in Computer Architecture

Meeting 4: GateKeeper

Dr. Mohammed Alser

ALSERM@ethz.ch

ETH Zürich

Spring 2021

18 March 2021

Mohammed Alser

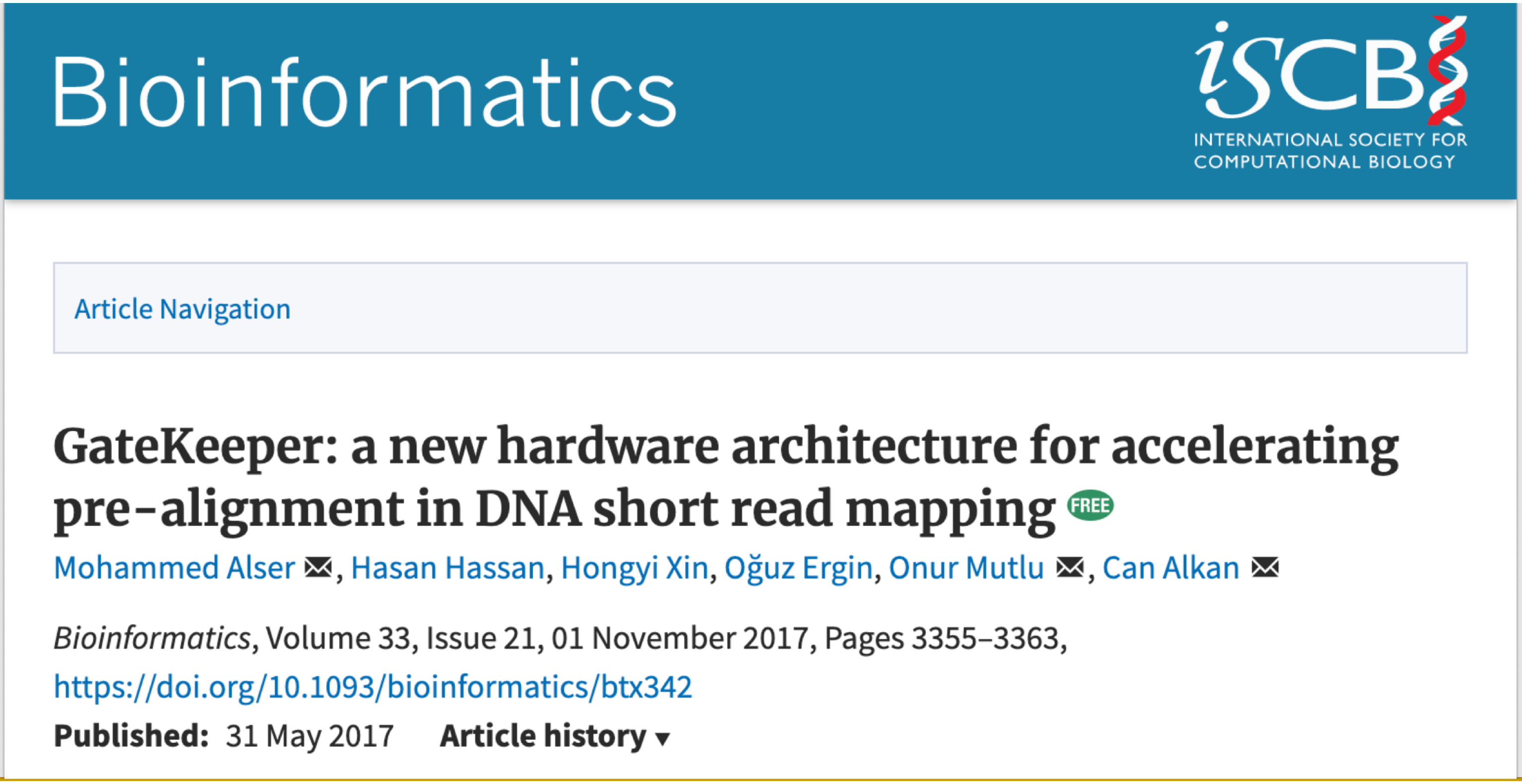
- Senior Researcher and Lecturer, SAFARI Research Group, ETH Zürich, since Sept. 2018.
- PhD from Bilkent University (Turkey) 2018, worked at UCLA, TU Dresden, and PETRONAS.
- PhD these in accelerating genome analysis, advisors: Can Alkan and Onur Mutlu, awarded:
  - IEEE Turkey Doctoral Dissertation Award
  - TÜBITAK doctoral fellowship
  - The Best Palestinian PhD Student in Turkey
  - HiPEAC Collaboration Grant
- ALSERM@ethz.ch, https://mealser.github.io/, https://twitter.com/mealser

- My main research is in bioinformatics, computational genomics, metagenomics, and computer architecture.
- I am especially excited about **building** new data structures, algorithms, and architectures that **make intelligent genome analysis a reality**.

Example Paper Presentation III

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

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 ▾**

GateKeeper: Accelerating Pre-Alignment in DNA Read Mapping

Mohammed Alser[1], Hasan Hassan[2,3], Hongyi Xin[4], Oğuz Ergin[2], Onur Mutlu[1,3,4], Can Alkan[1]

Bioinformatics, 2017

[1] Bilkent University

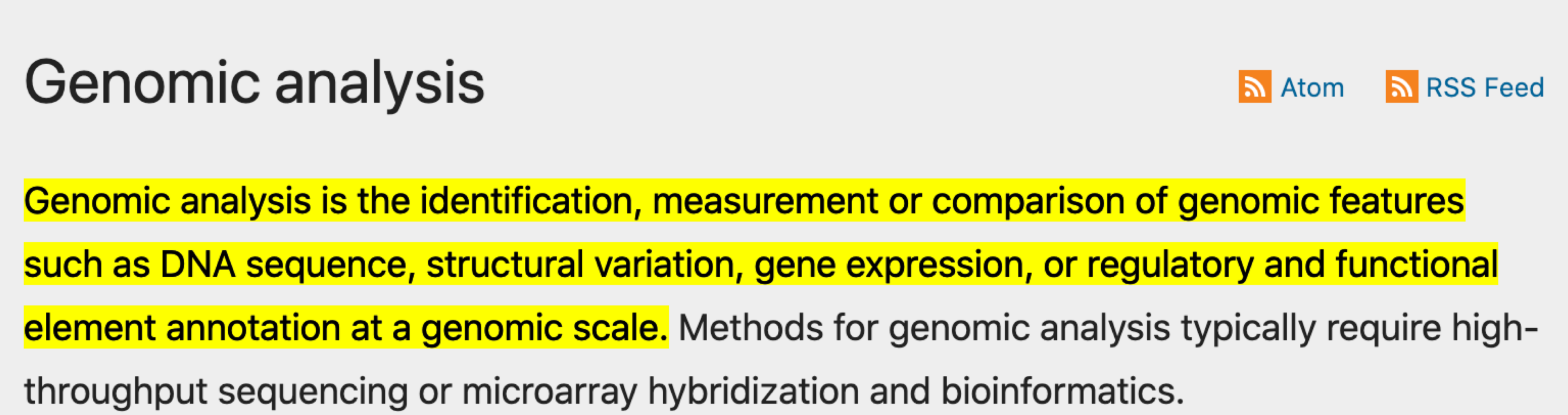
[2] TOBB University of Economics & Technology

[3] ETH zürich

[4] Carnegie Mellon

Background, Problem, & Goal

What is Genome Analysis?



Genomic analysis

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

https://onlinelearning.hms.harvard.edu/hmx/courses/genetic-testing/
https://www.nature.com/subjects/genomic-analysis

Genome Analysis

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

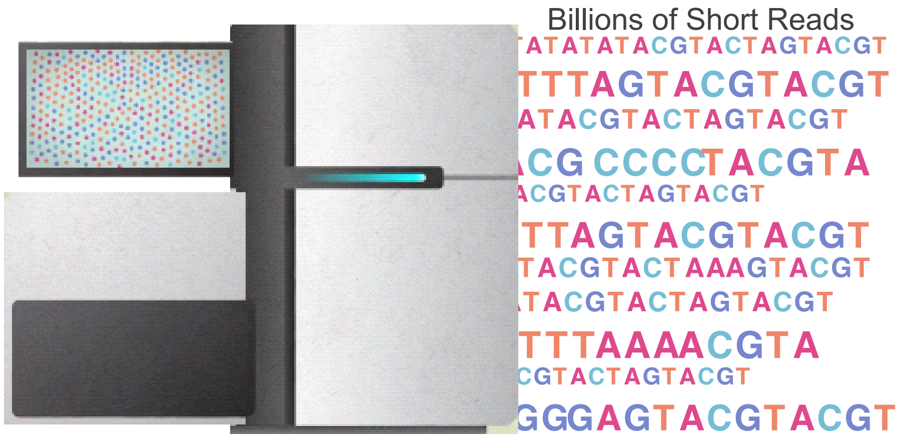
```
>CCTCCTCAGTGCCACCCAGCCCACTGGCAGCTCCCAAACAGGCTCTTATTAAAACACCCTGTTCCCTGCCCCTTGGAGTGAGGTGTCAAG
GACCTAAACTAAAAAAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTT
CATGTCAAGGACCTAATGTGCTAAACAGCACTTTTTTGACCATTATTTTGGATCTGAAAGAAATCAAGAATAAATGAAGGACTTGATACATTG
GAAGAGGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAA
ACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCTGTGTTGCAGGTCTTCTTGCATTTCCCTGTCAAAAGAAAAAGAATTTAAAATTT
AAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTCAGGCCAAGAGTTGCAAAAAAAAAAAAAAGAAAAA
GAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTAGCCAGAATGG
TTGTGGGATGGGAGCCTCTGTGGACCGACCAGGTAGCTCTCTTTTCCACACTGTAGTCTCAAAGCTTCTTCATGTGGTTTCTCTGAGTGAAA
AAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTTTTCATGTCAAGGACC
TAATGTAGCTATACTGAACGTTATCTAGGGGAAAGATTGAAGGGGAGCTCTAAGGTCAACACACCACCACTTCCCAGAAAGCTTCTTCA……
```

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)



Reference Genome

.FASTA file:

```
>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCTCTTTTCTTATCATTGACATTTAAACTCTGGGGCAGGTCCTCGCGTAGAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGGAACCGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACCGCCCGGGCTCCGGCCCCGGCCCGGCTCGGGGCCCGCGGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCGCCCCAAGTGGCCCCGGGGCTTGATTTTTGCTTTTAAAAG
GAGGCATACAAAGATGGAAGCGAGTTACTGAGGGAGGGATAGGAAGGGGGTGGAGGAGGGACTTGTCTT
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

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

Genomic Reads

.FASTQ file:

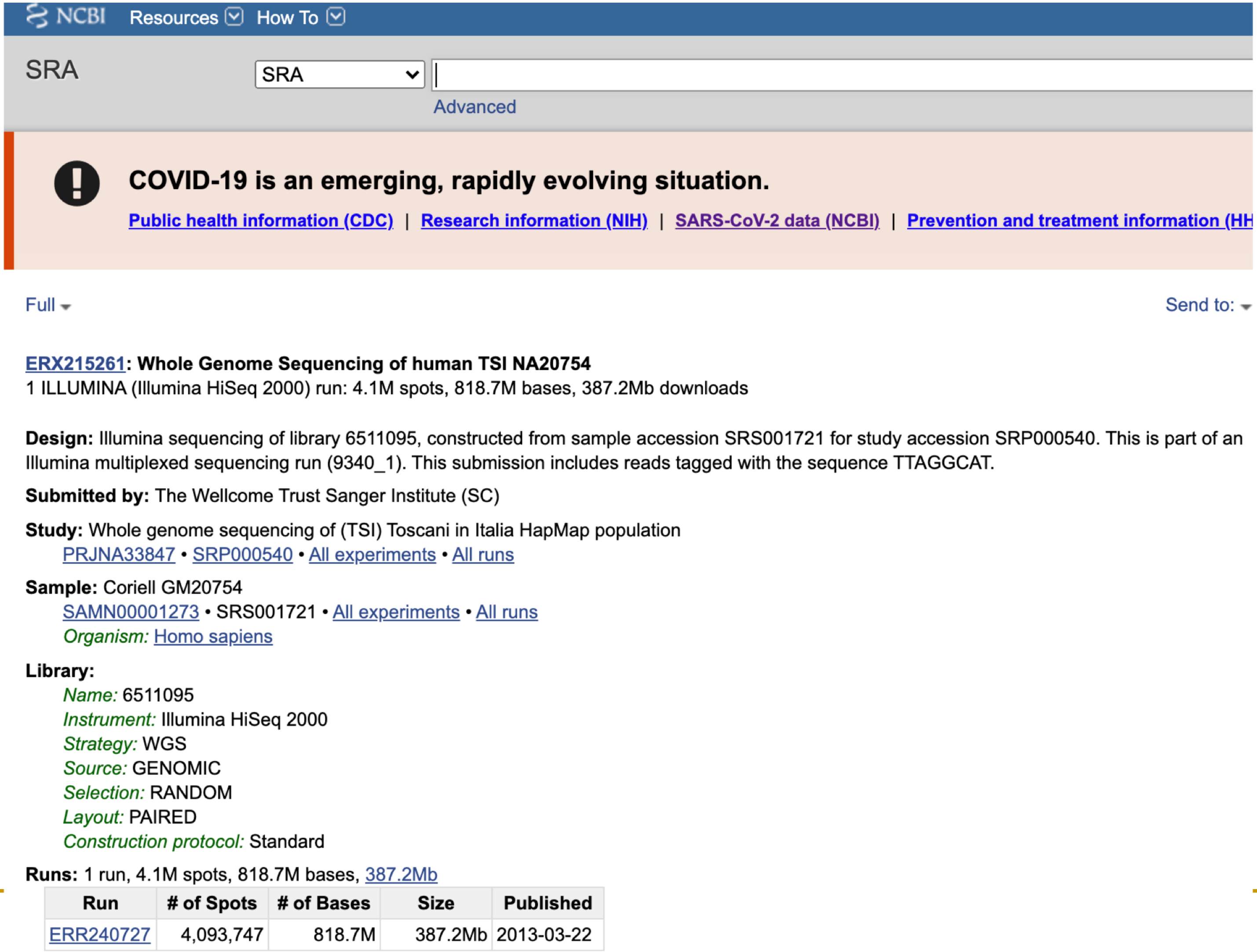
Identifier
Sequence
+ sign & identifier
Quality scores

```
@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
TTAATTGGTAAATAAATCTCCTAATAGCTTAGATNTTACCTTNNNNNNNNNNTAGTTTCTTGAGA
+HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
efcfffffcfeefffcffffffddf`feed]`]_Ba_^__[YBBBBBBBBBBRTT\]][]dddd`
```

Base T
phred Quality] = 29

Obtaining .fastq Files

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



COVID-19 is an emerging, rapidly evolving situation.

Public health information (CDC) | Research information (NIH) | SARS-CoV-2 data (NCBI) | Prevention and treatment information (HH

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

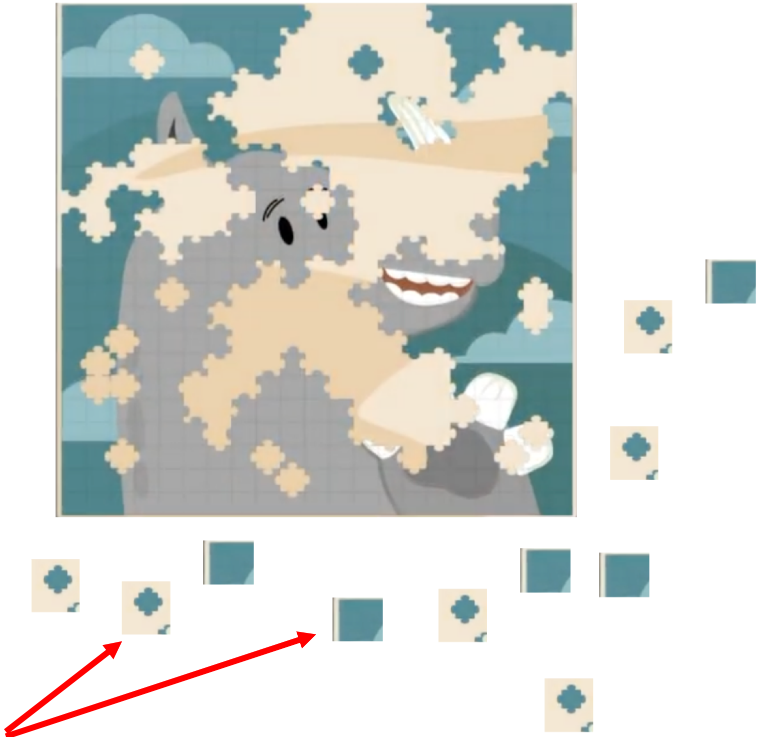
Run	# of Spots	# of Bases	Size	Published
ERR240727	4,093,747	818.7M	387.2Mb	2013-03-22

Solving the Puzzle

.FASTA file

Reference genome

.FASTQ file

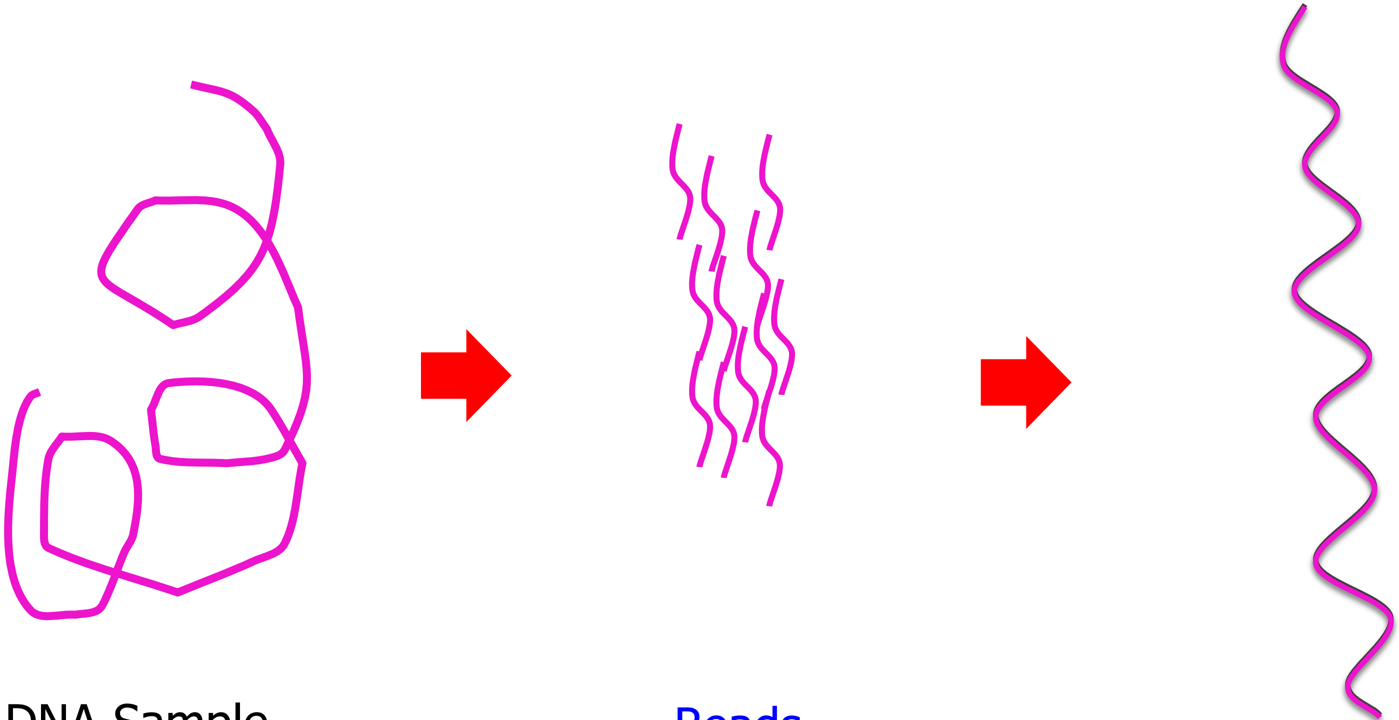


Reads

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

Read Mapping

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

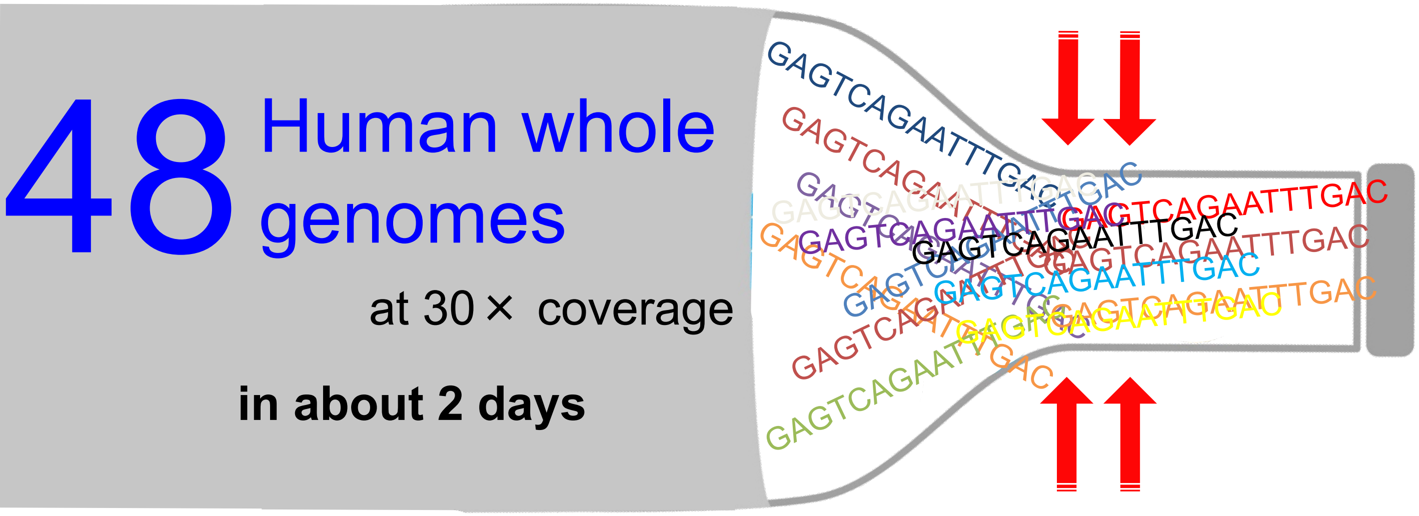


DNA Sample
“chemical format”

Reads
“text format”

Reference genome
Subject genome
“text format”

Bottlenecked in Read Mapping!!



48 Human whole genomes

at 30 × coverage

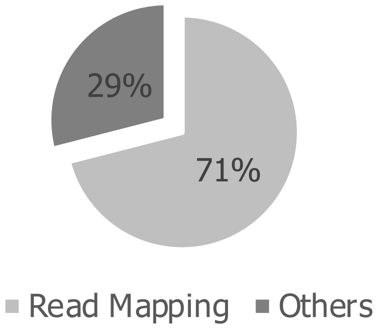
in about 2 days

Illumina NovaSeq 6000

1 Human genome

32 CPU hours

on a 48-core processor



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

What makes read mapper **slow**?

Let’s first learn how to map a read

Matching Each Read with Reference Genome

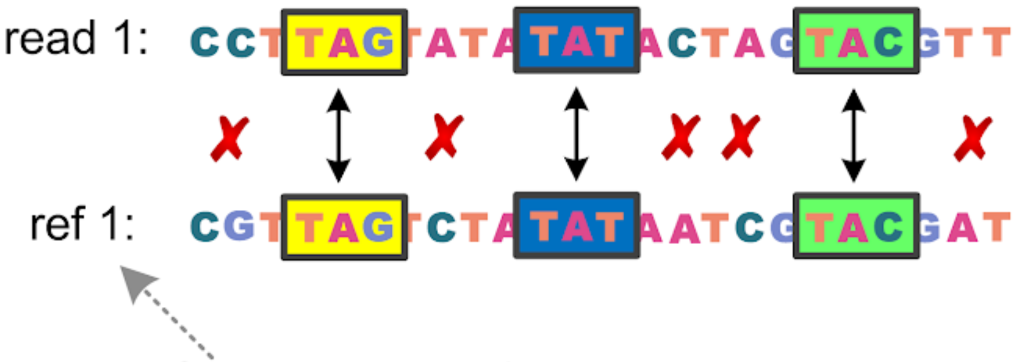
.FASTA file:

```
>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCT🟩TCATTGACATTTAAACTCTGGGCAGGT🟦GAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGGAACCGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACCGCCC🟨CCCCGGCCCGGCTCGGGCCCGCGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCGCCCCAAGTGGCCCCGGGGCTTGATTTTGCTTTTAAAAG
GAGGCATACAAAGATGGAAGCGAGTTACTGAGGGAGGGATAGGAAGGGGGGTGGAGGAGGGACTTGTCTT
TGCCGAGTGT🟦CAAAAGTAGCAA🟩CTCCTAA🟥TCCAGTCCGGCCCT
GAGCTGGGAGTAGGGGGCGGGAGTCTGCTGCTGCTGTCTGCTAAAGCCACTCGCGACCGCGAAAAATGCA
GGAGGTGGGGACGCACTTTGCATCCAGACCTCCTCTGCATCGCAGTTC🟨ACGCTTGGGAAAG
TCCGTACCCGCGCCT🟥AAAGACACCCTGCCGCGGGTCGGGCGAGGTGCAGCAGAAGTTTCCC
GCGGTTGCAAAGTGCAGATGGCTGGACCGCAACAAAGTCTAGAGATGGGGTTCGTTTCTCAGAAAGACGC
```

.FASTQ file:

```
@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
T🟦AATAAATCT🟩TTAGATN🟥NNNNNNNNTAG
+HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
efcfffffcfeefffcffffffddf`feed]`]_Ba_^__[YBBBBBBBBBRTT
```

Base-by-Base Comparison

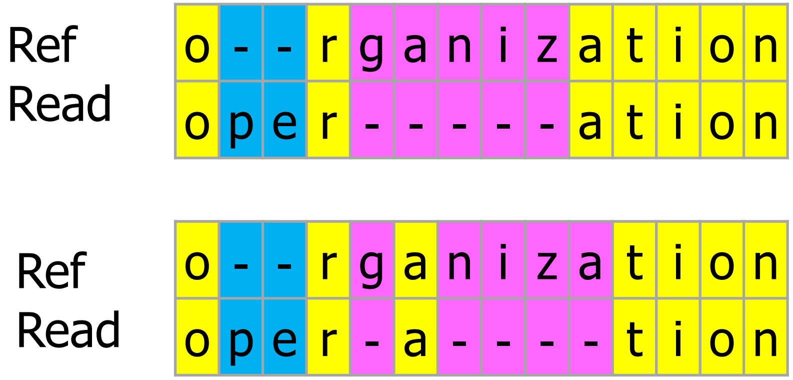


reference segment that spans locations (5, 7, and 9)

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

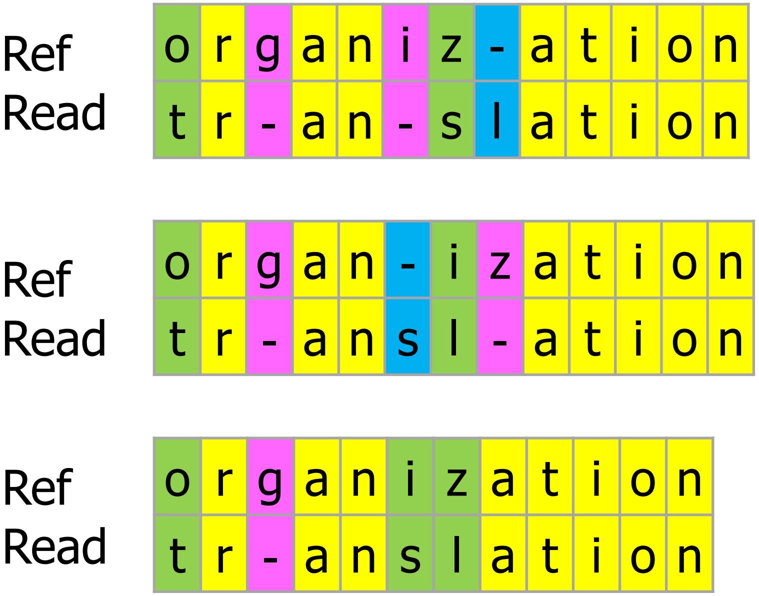


Ref	o	-	-	r	g	a	n	i	z	a	t	i	o	n
Read	o	p	e	r	-	-	-	-	-	a	t	i	o	n

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

organization x translation



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

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

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

Edit distance = 4

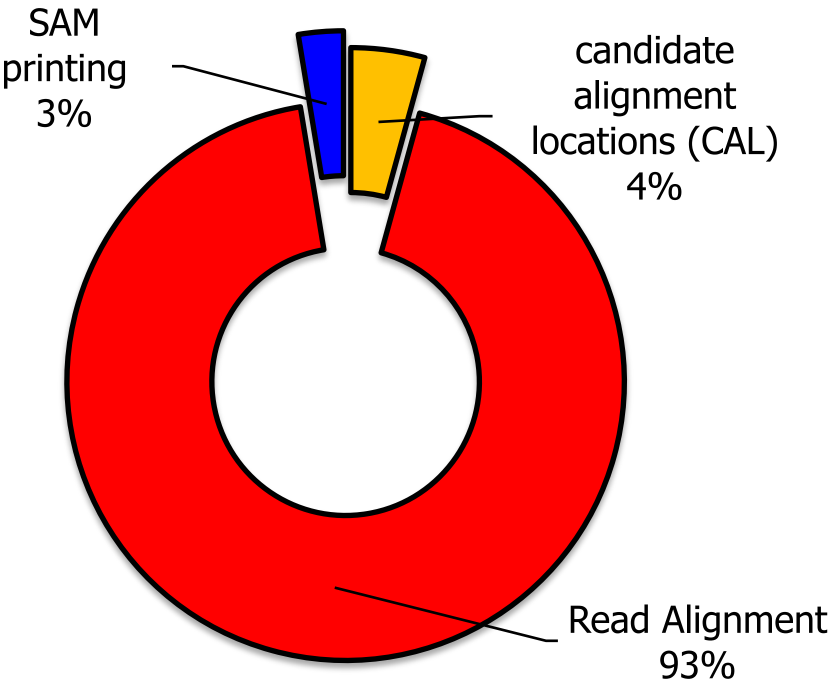
🟨 match
🟪 deletion
🟦 insertion
🟩 mismatch

What Makes Read Mapper Slow?

Key Observation # 1

93%

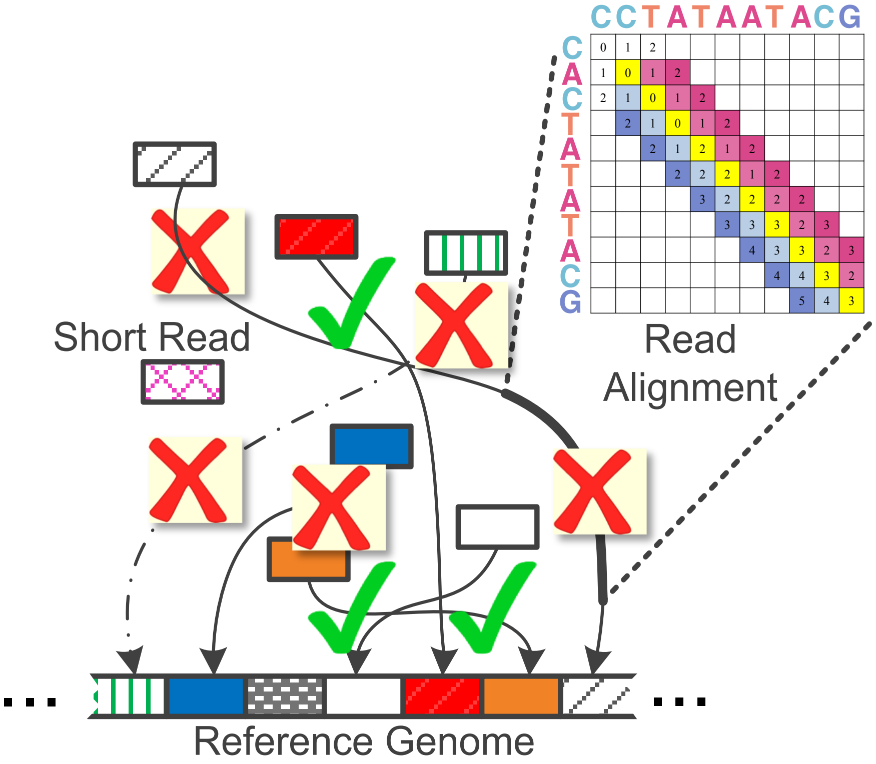
of the read mapper's execution time is spent in sequence alignment.



Alser et al, Bioinformatics (2017)

What Makes Read Mapper Slow? (cont'd)

Key Observation # 2



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

98%

of candidate locations

have high dissimilarity

with a given read.

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

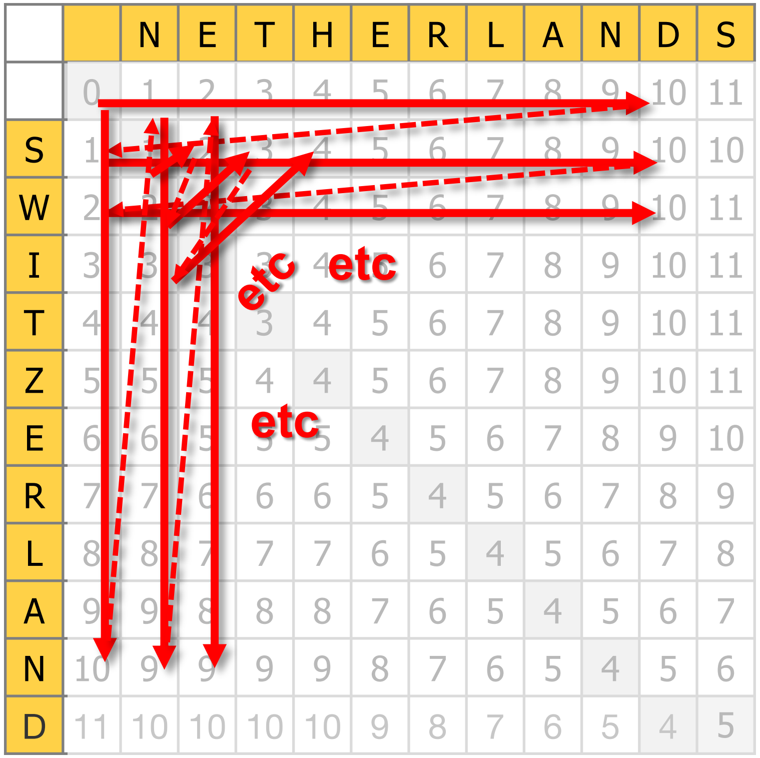
What Makes Read Mapper Slow? (cont'd)

Key Observation # 3

- Quadratic-time dynamic-programming algorithm **WHY?!**
  - Enumerating all possible prefixes

- NETHERLANDS x SWITZERLAND

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



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

What Makes Read Mapper Slow? (cont'd)

Key Observation # 3

- Quadratic-time dynamic-programming algorithm

  Enumerating all possible prefixes

- Data dependencies limit the computation parallelism

  Processing row (or column) after another

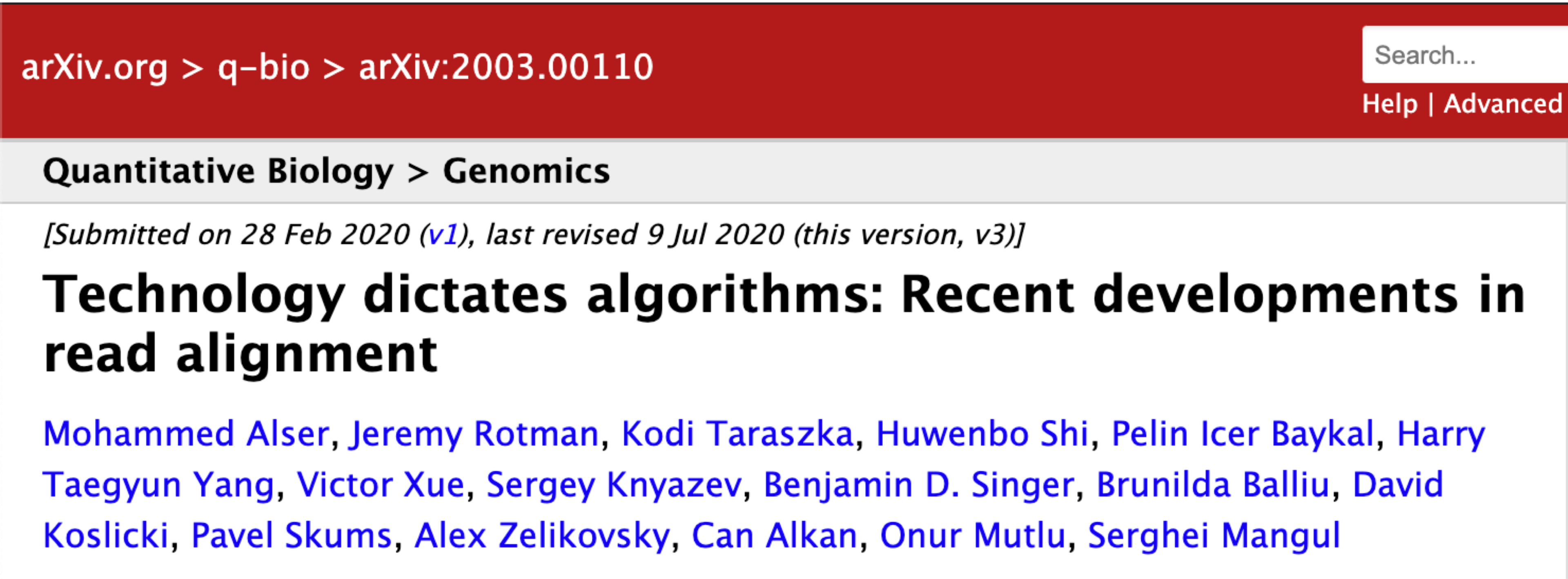
- Entire matrix is computed even though strings can be dissimilar.

  Number of differences is computed only at the 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

Read Mapping in 111 pages!

Analyzing 107 read mappers (1988-2020) in depth



Quantitative Biology > Genomics

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

Technology dictates algorithms: Recent developments in read alignment

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

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

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

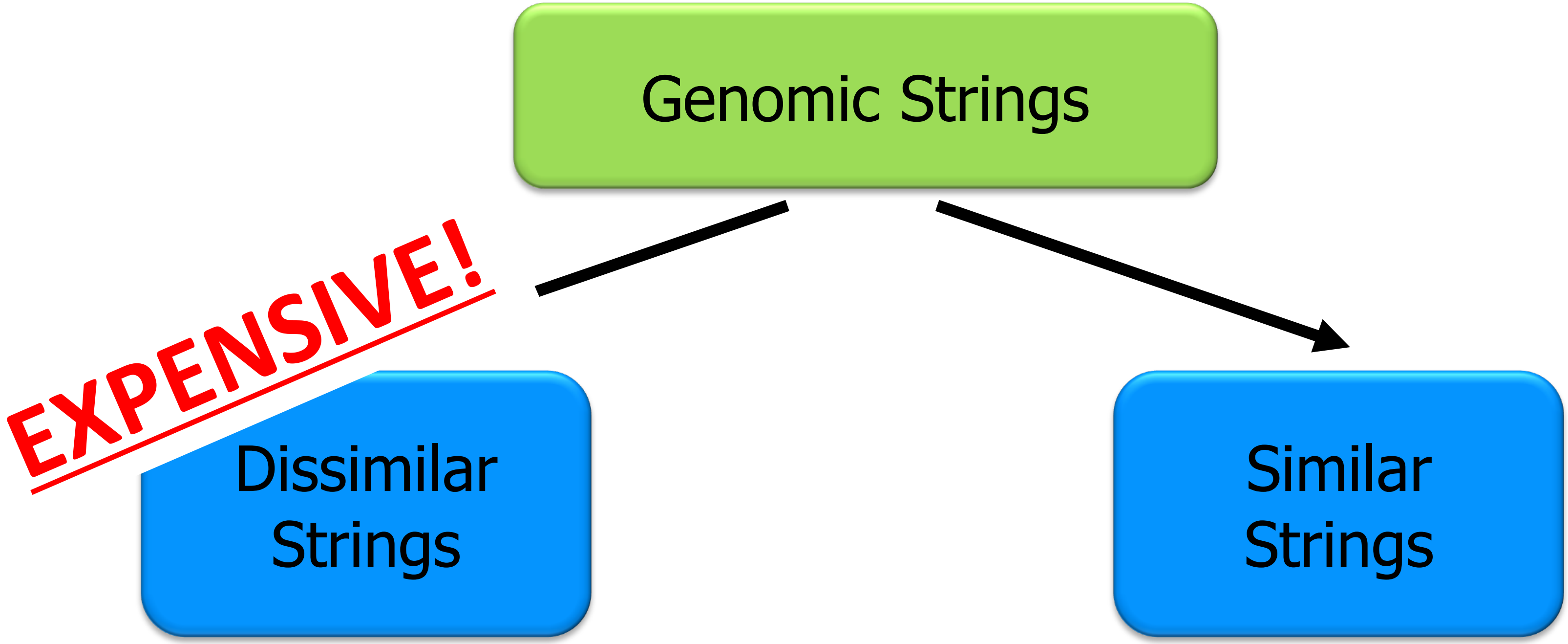
Goal: Minimizing Alignment Time

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

EXPENSIVE!

Dissimilar Strings

Similar Strings

- Ignore as number of differences exceeds a threshold.

- Find number and location of differences?

GateKeeper

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

- **Key ideas:**
  - Compute “Shifted Hamming Distance”: AND of 2*E*+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)

Proposed Solution: GateKeeper

Pre-Alignment Filter

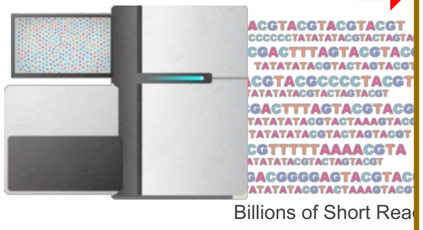
+

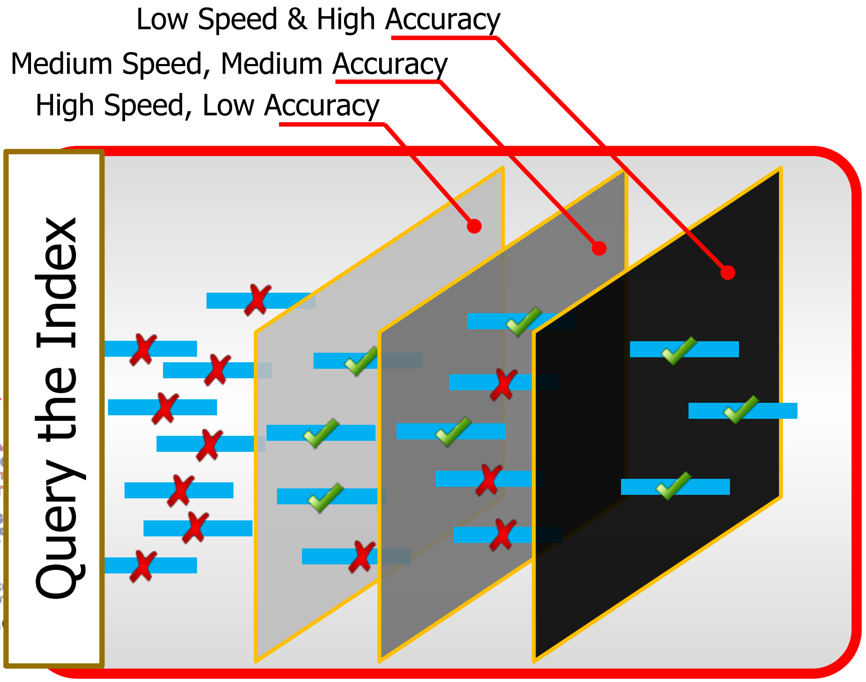
=

1st

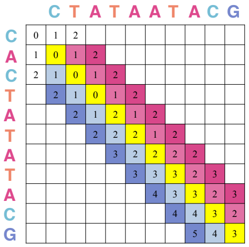
FPGA-based Alignment Filter

x10^{12} mappings





x10^{3} mappings



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

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

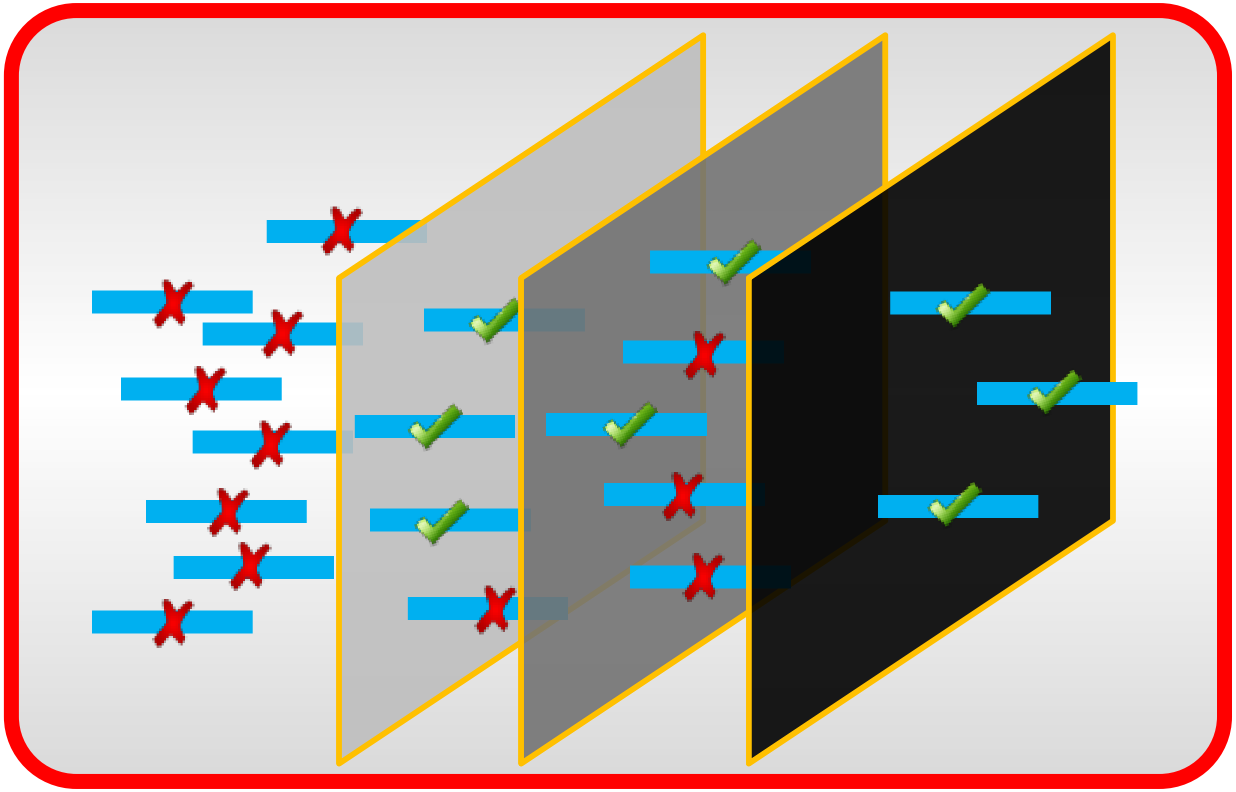
2 **Read Pre-Alignment Filtering**
Fast & Low False Positive Rate

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

Ideal Filtering Algorithm

Step 2

Query the Index



Step 3

Read Alignment

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

Mechanisms (in some detail)

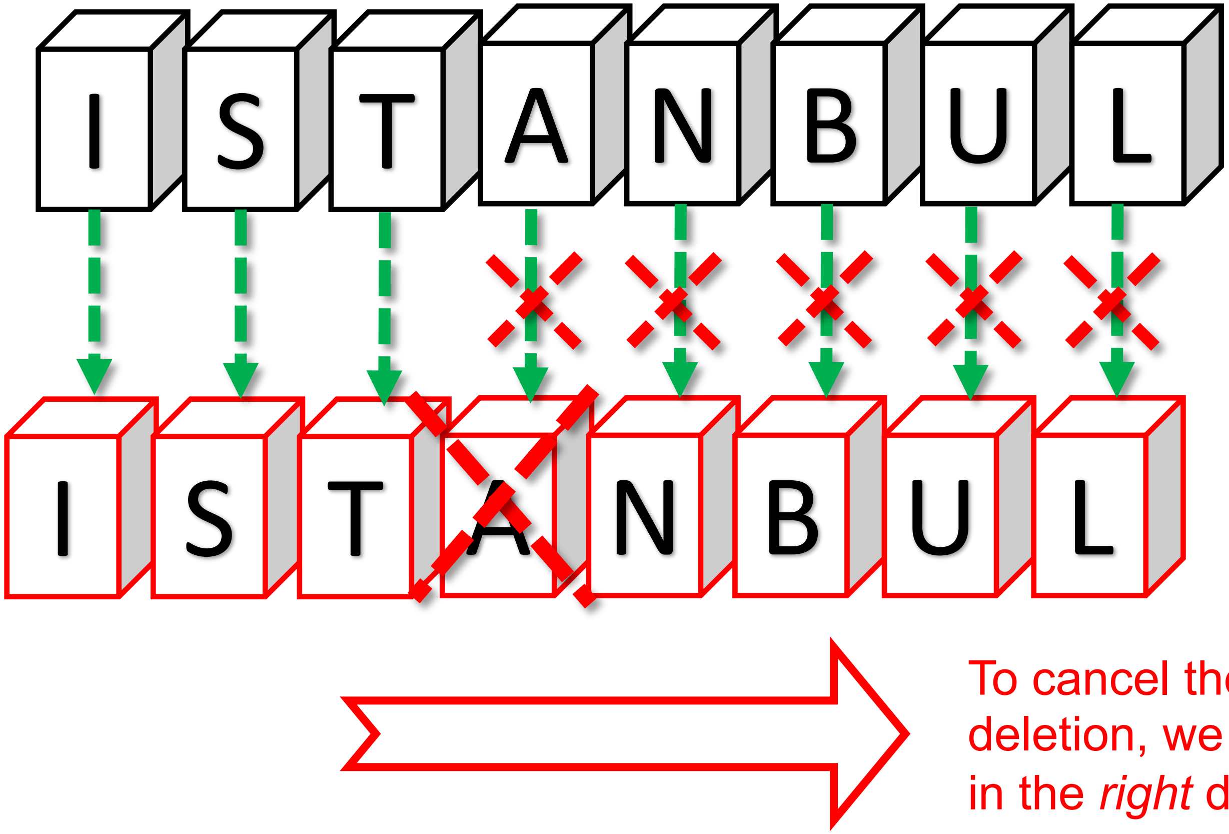
Mechanisms

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

Hamming Distance (∑⊕)

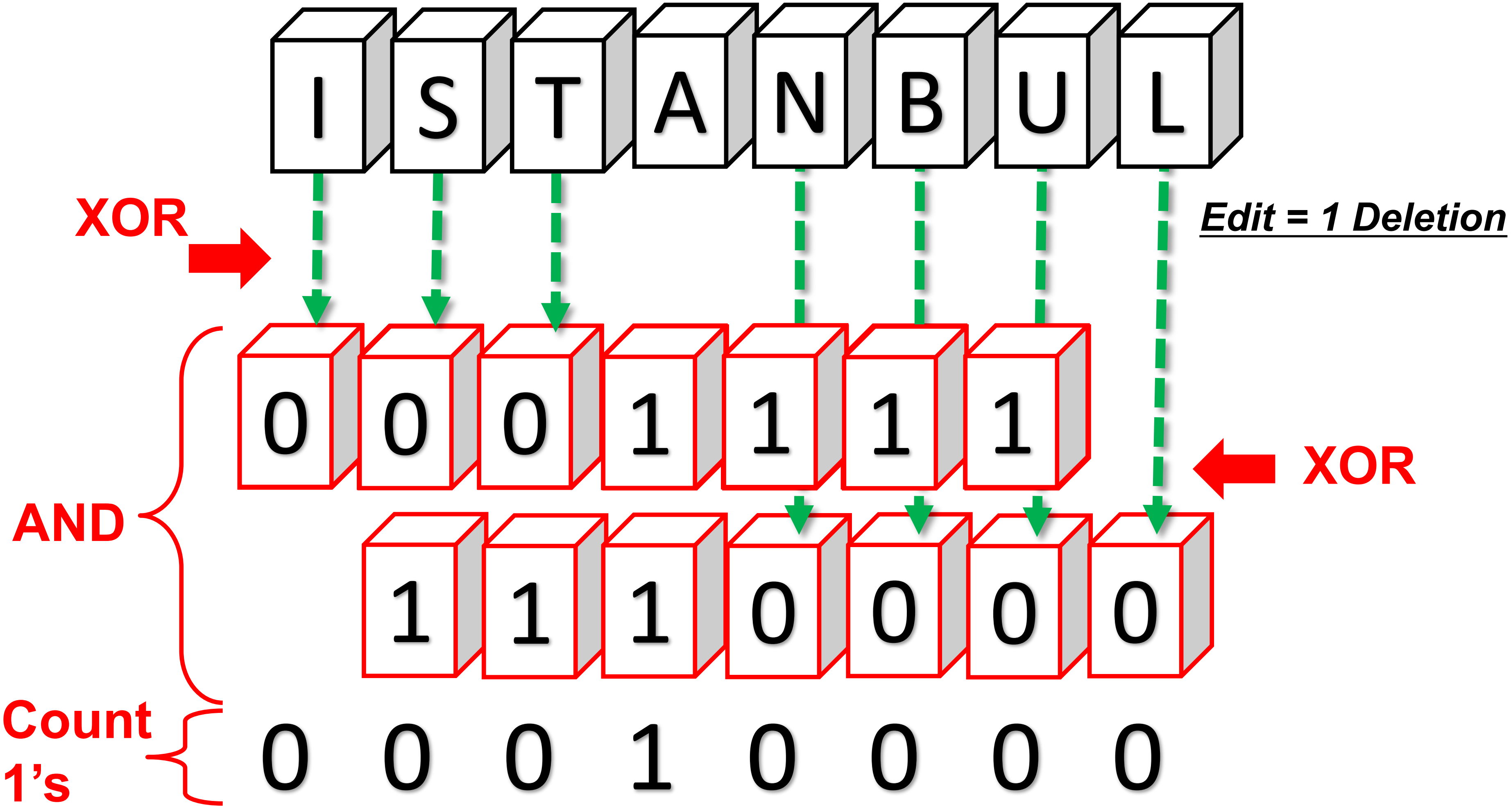
3 matches   5 mismatches

Edit = 1 Deletion



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

Shifted Hamming Distance (Xin+ 2015)



I S T A N B U L

XOR

Edit = 1 Deletion

0 0 0 1 1 1 1

XOR

AND

1 1 1 0 0 0 0

Count 1's

0 0 0 1 0 0 0 0

7 matches 1 mismatches

Mechanisms

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

- **Key ideas:**
  - Compute "Shifted Hamming Distance": AND of 2*E*+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' ≤ Threshold

```
            Query :GAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAACATTGTTGGGCCGGA
        Reference :GAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGAGACATTGTTGGGCCGG

     Hamming Mask :00000000001000000000000011111110111100011101101011011111111110001000000111101101001010 1
  1-Deletion Mask :111111111110011111011111100000000000000000000000000000000000000000001100000000000000000
  2-Deletion Mask :00000000101101110011111111111111011110001110110101101111111111000100010011101101001010
  3-Deletion Mask :1111111111101110110011011101110110001001001111111111111100101100110010110111011101111
  1-Insertion Mask :111111111110111110111111011101100010010011111111111111001011001100010101110111011111 0
  2-Insertion Mask :00000010011111100111111111100100011010101001101011111111111111011100111111100011110 1100
  3-Insertion Mask :11111110111011001100011111111101011011111100110010111011111111110110111101011100 1000

        AND Mask :000000000010000000000000100000000000000000000000000000000000000000000000000000000000000
```

```
1-D                                                                                                   0000
2-D                                                                                                   .110
3-D                                                                                                   .111
1-Ir                                                                                                  .110
2-Ir                                                                                                  .100
3-Ir                                                                                                  .000
                                                                                                      0000
```

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

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

Alignment Matrix vs. Neighborhood Map

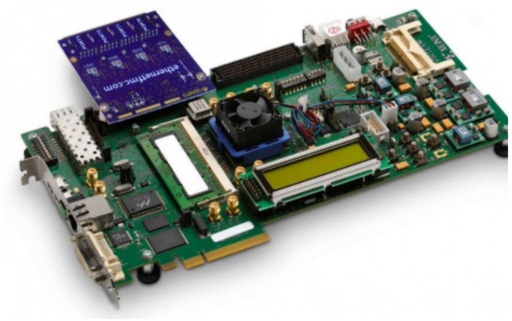
Needleman-Wunsch

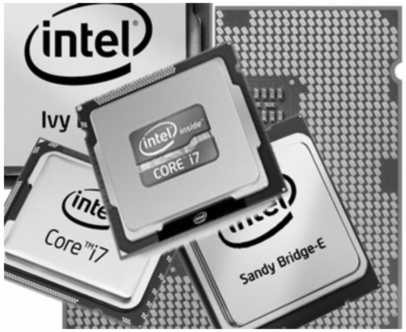
		C	T	A	T	A	A	T	A	C	G
	0	-1	-2								
A	-1	-1	-1	-2							
C	-2	-2	-2	-1	-2						
T		-2	-3	-2	-1	-2					
A			-3	-3	-2	-1	-2				
T				-4	-3	-2	-1	-2			
A					-4	-3	-2	-2	-2		

Neighborhood Map

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

Independent vectors can be processed in parallel using hardware technologies





Hardware Architecture

GateKeeper Walkthrough (cont'd)

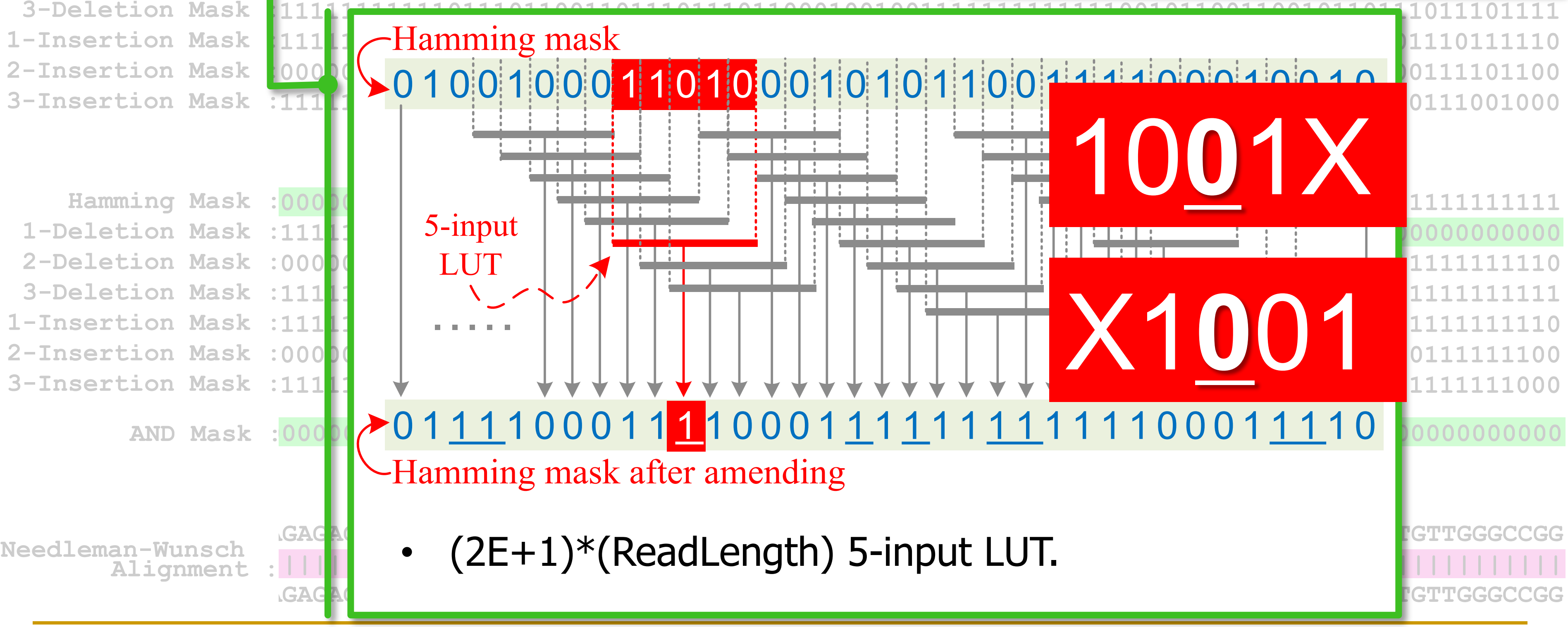
Generate 2E+1 masks

Amend random zeros: 101 → 111 & 1001 → 1111

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

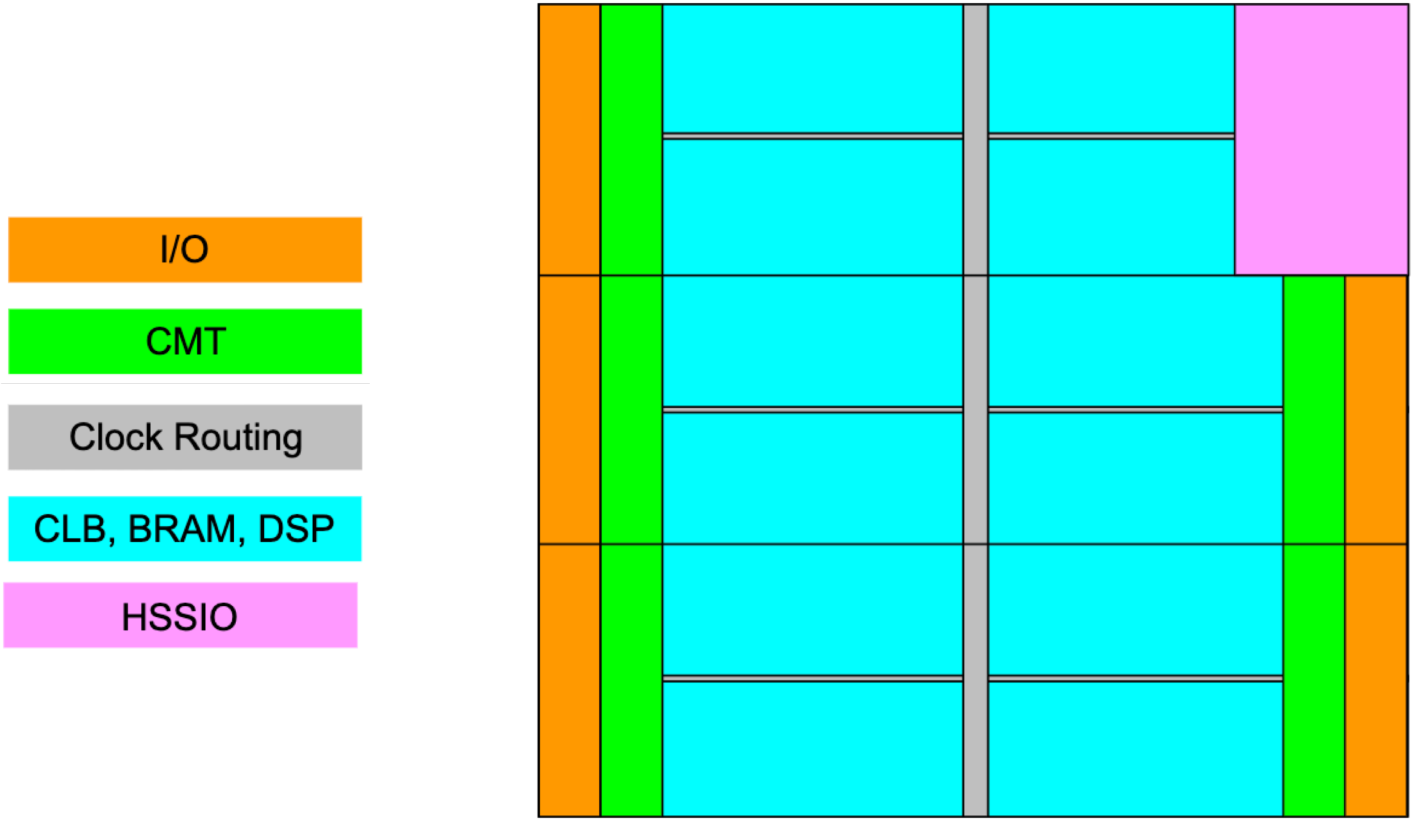
- E right-shift registers (length=ReadLength)
- E left-shift registers (length=ReadLength)
- (2E+1) * (ReadLength) 2-XOR operations.

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



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

Virtex-7 FPGA Layout



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

"7 Series FPGAs Configurable Logic Block", User Guide, Xilinx 2016

Virtex-7 FPGA Layout

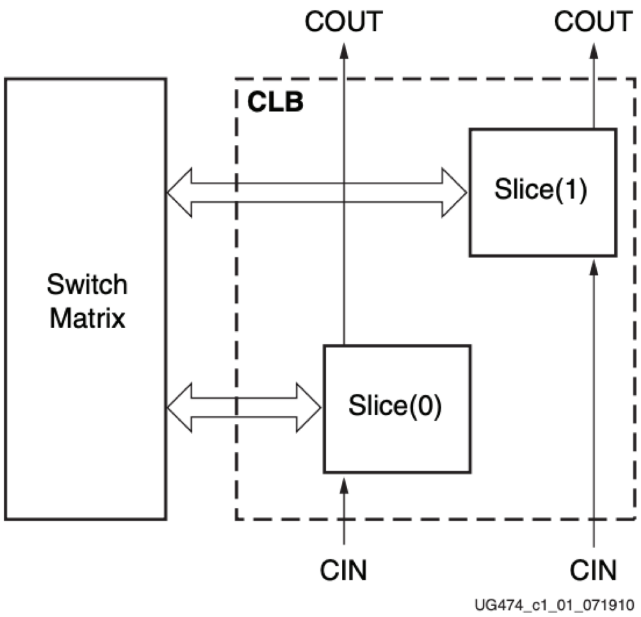
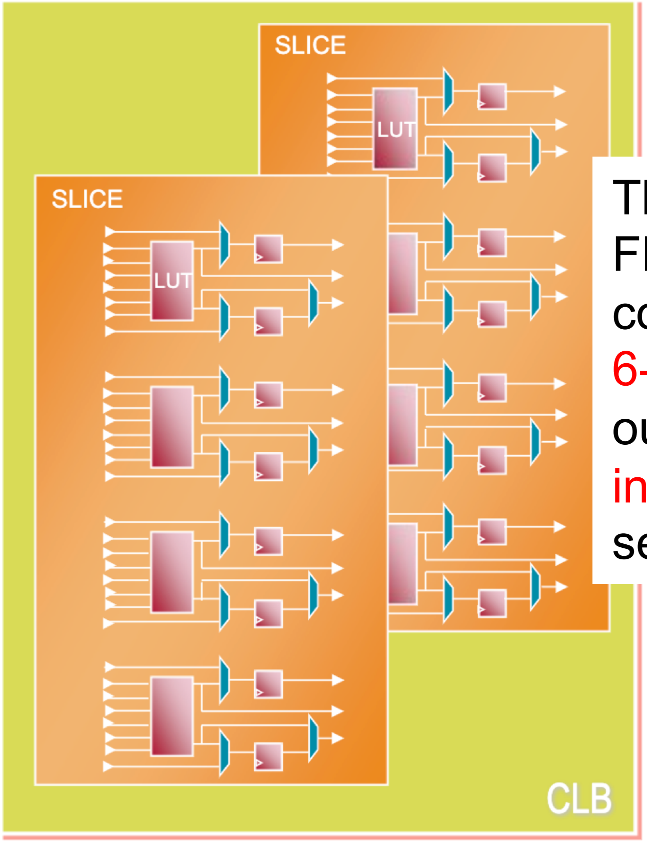


Figure 1-1: **Arrangement of Slices within the CLB**



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[1]	Shift Registers[1]
2	8	16	2	256 bits	128 bits

"7 Series FPGAs Configurable Logic Block", User Guide, Xilinx 2016

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 (≤ 128 bp)

VC709 Resource Utilization

Theoretically:

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

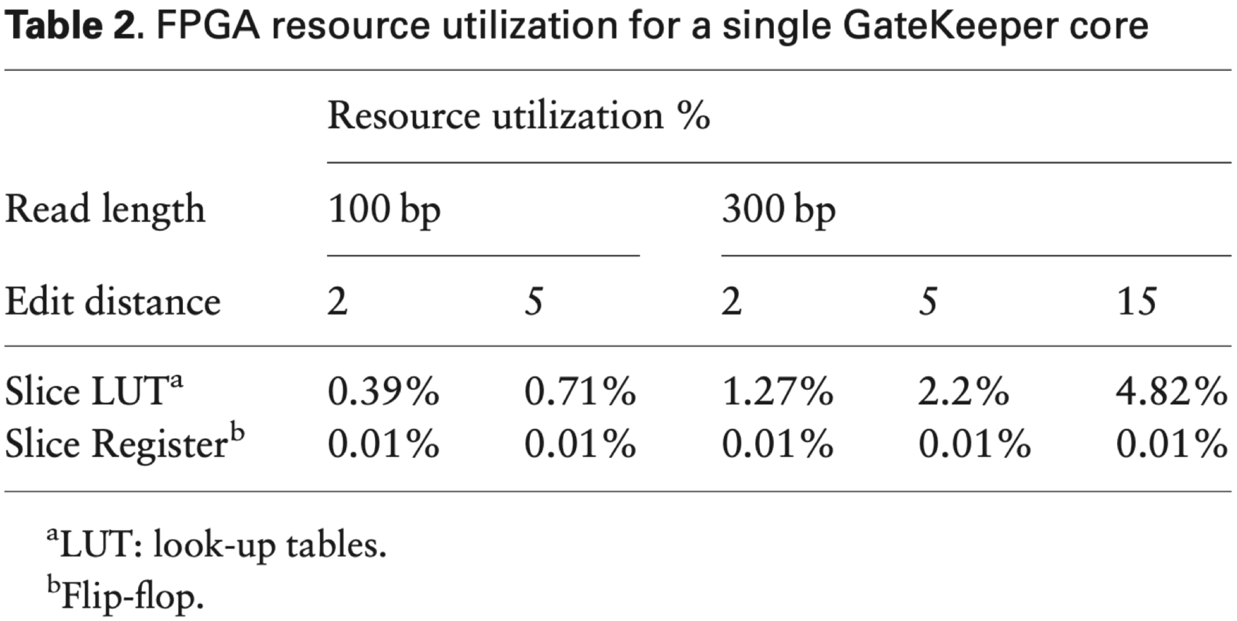


Table 2. FPGA resource utilization for a single GateKeeper core

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

[a] LUT: look-up tables.

[b] 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.

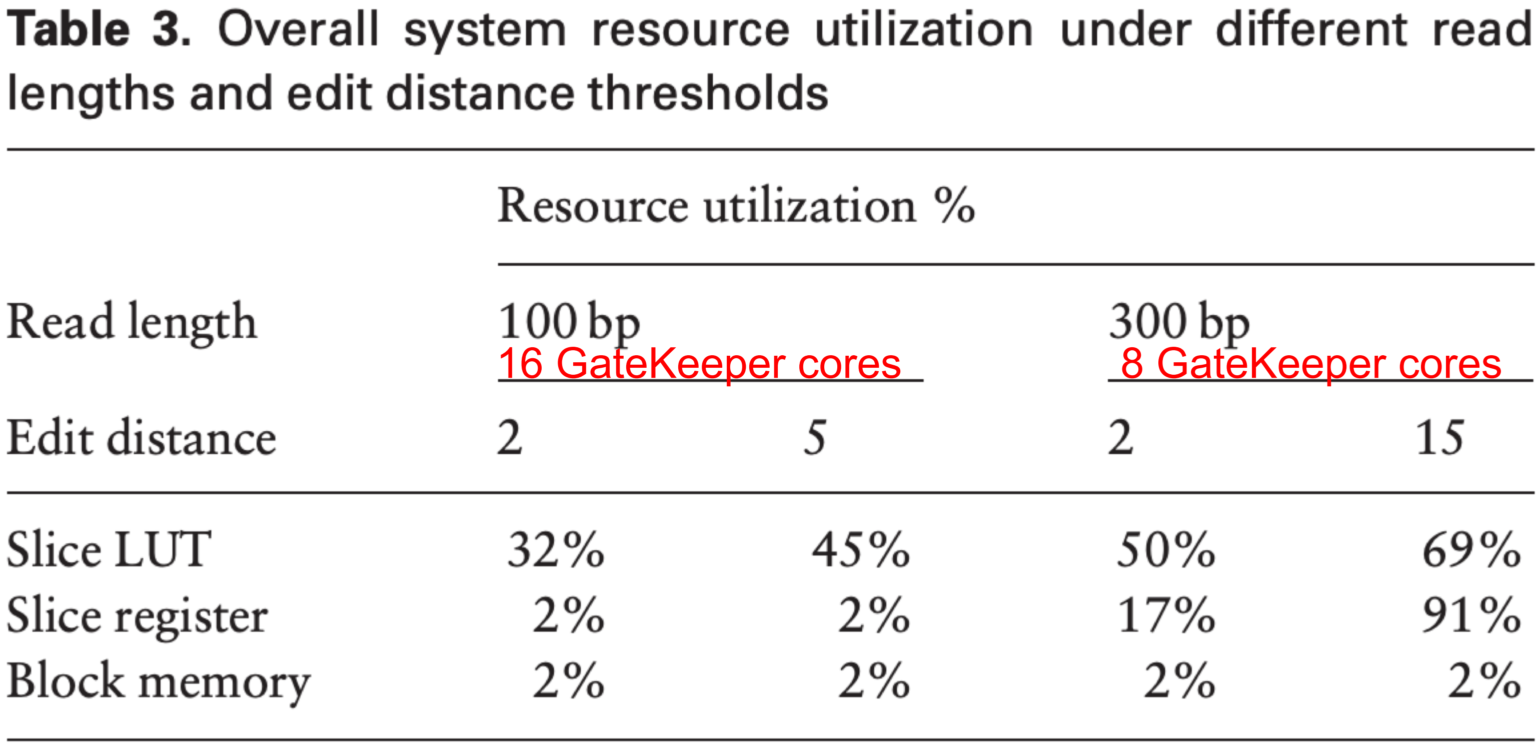
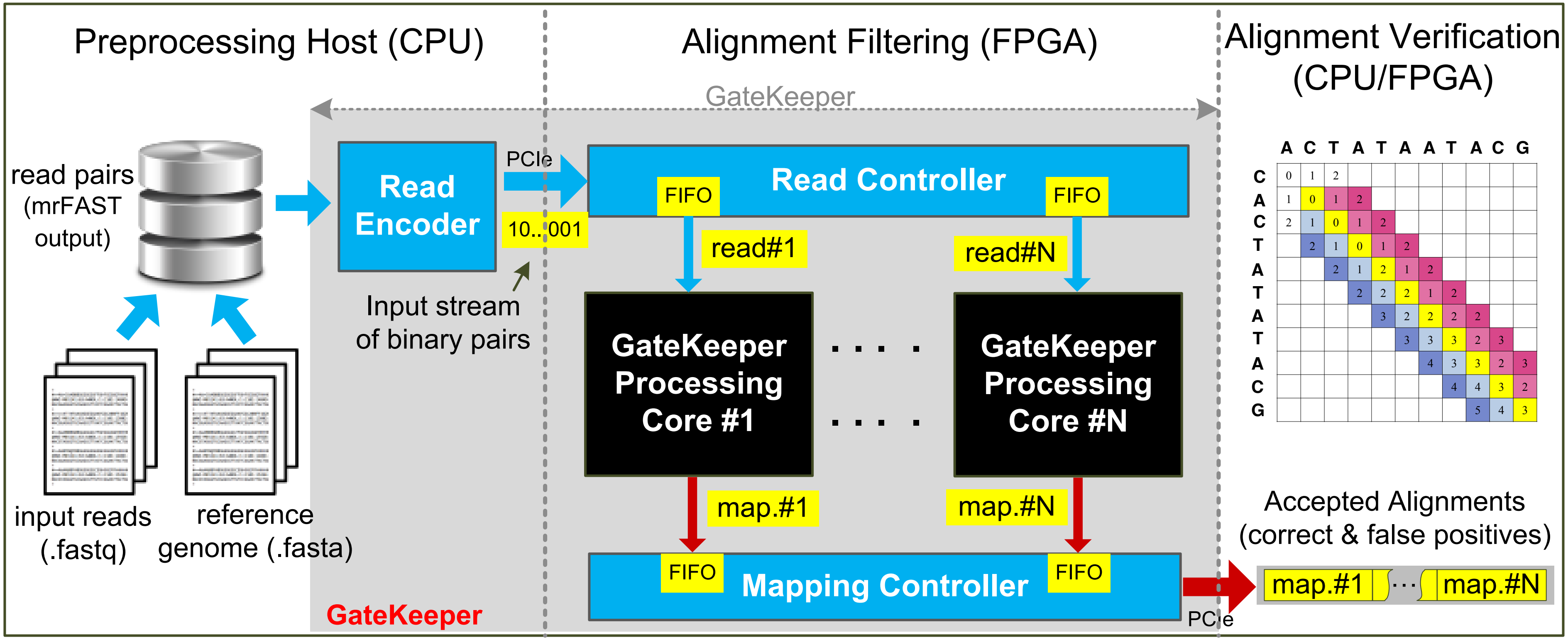


Table 3. Overall system resource utilization under different read lengths and edit distance thresholds

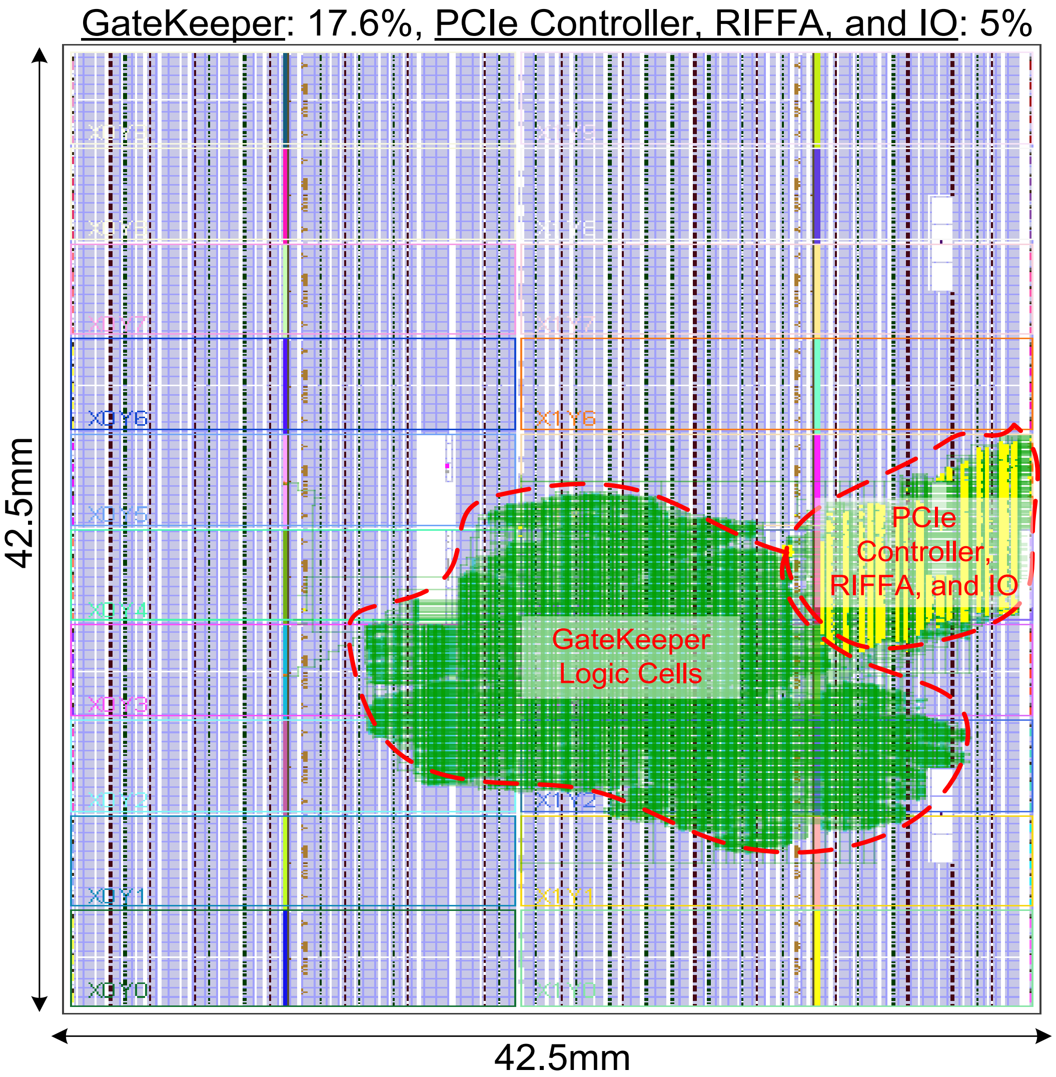
	Resource utilization %			
Read length	100 bp 16 GateKeeper cores		300 bp 8 GateKeeper cores	
Edit distance	2	5	2	15
Slice LUT	32%	45%	50%	69%
Slice register	2%	2%	17%	91%
Block memory	2%	2%	2%	2%

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


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

FPGA Chip Layout


GateKeeper: 17.6%, PCIe Controller, RIFFA, and IO: 5%
42.5mm
42.5mm
PCIe Controller, RIFFA, and IO
GateKeeper Logic Cells
X0Y6
X1Y6
X0Y5
X0Y4
X0Y3
X0Y2
X1Y2
X0Y1
X1Y1
X0Y0
X1Y0

300 bp

E=15

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

Summary

Executive Summary

- **Problem:** There is a significant performance gap between high-throughput DNA sequencers and read mapper

- **Observations:** **Sequence alignment** is **computationally expensive** and **unavoidable**

- **Goal:** provide the **first hardware accelerator architecture** (as a **pre-alignment** filter) for **quickly** rejecting **dissimilar sequences**

- **Key Results:**
  - Provides a huge speedup of up to 130x compared to the previous state of the art software solution.

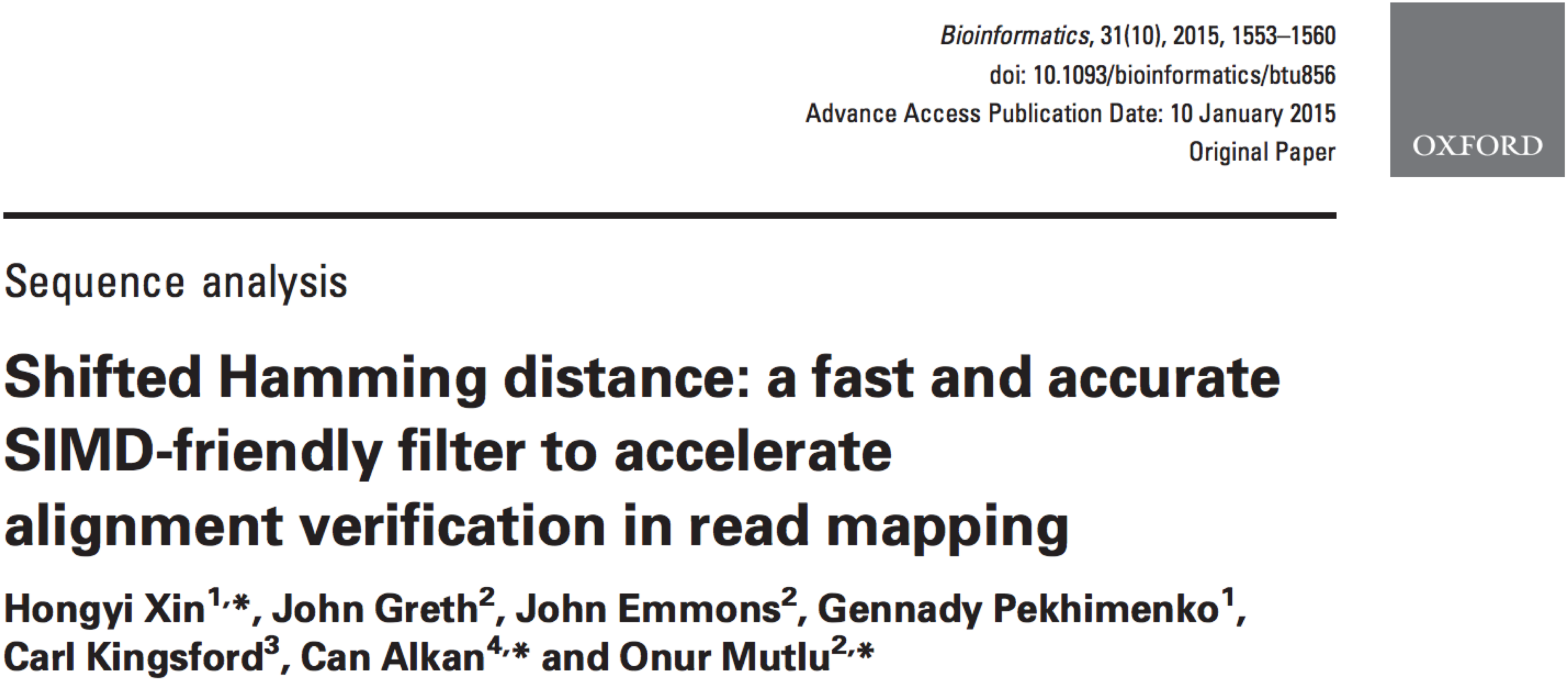
GateKeeper Conclusions

- 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

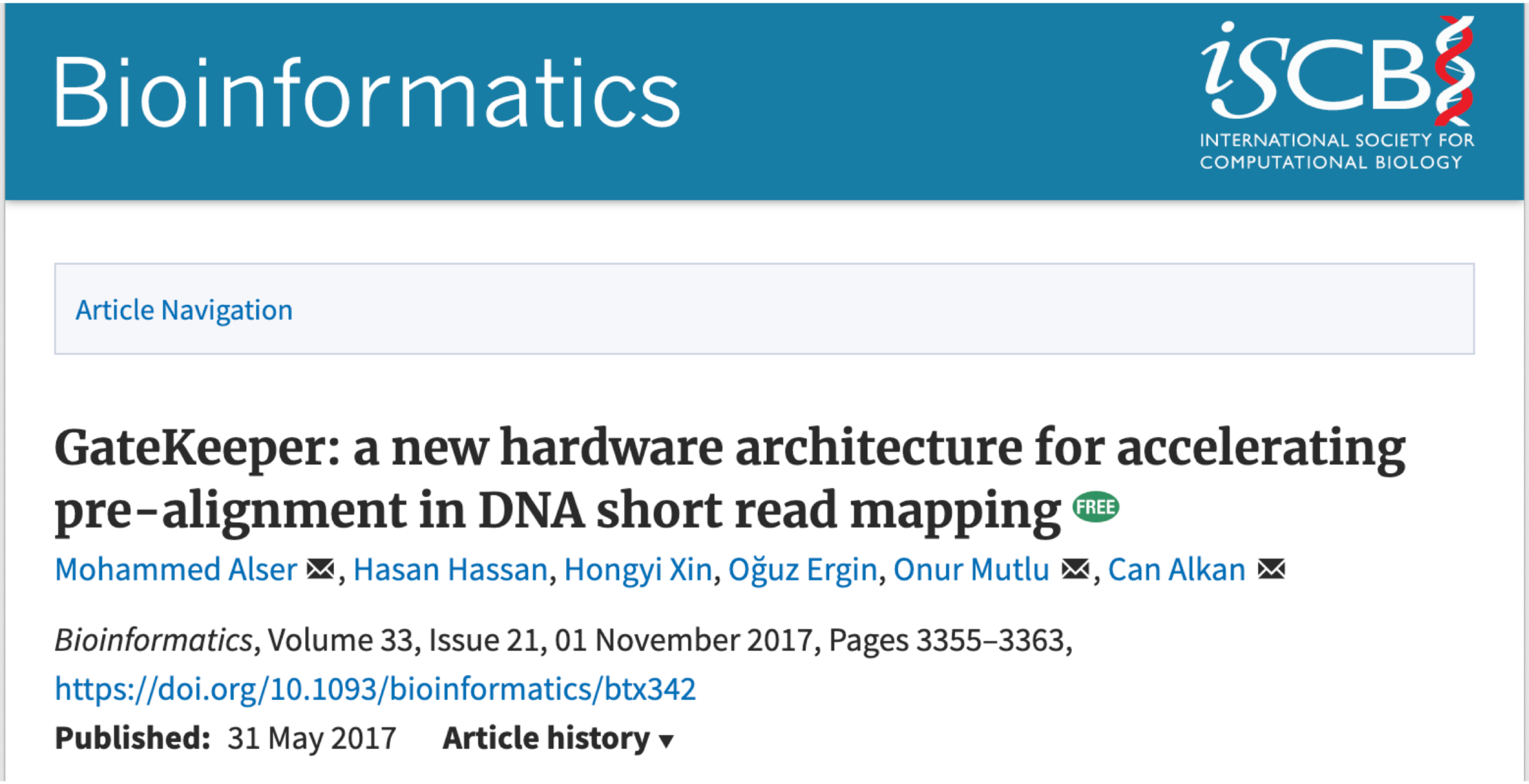
Sequence analysis

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

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

More on GateKeeper

- Download and test for yourself https://github.com/BilkentCompGen/GateKeeper



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

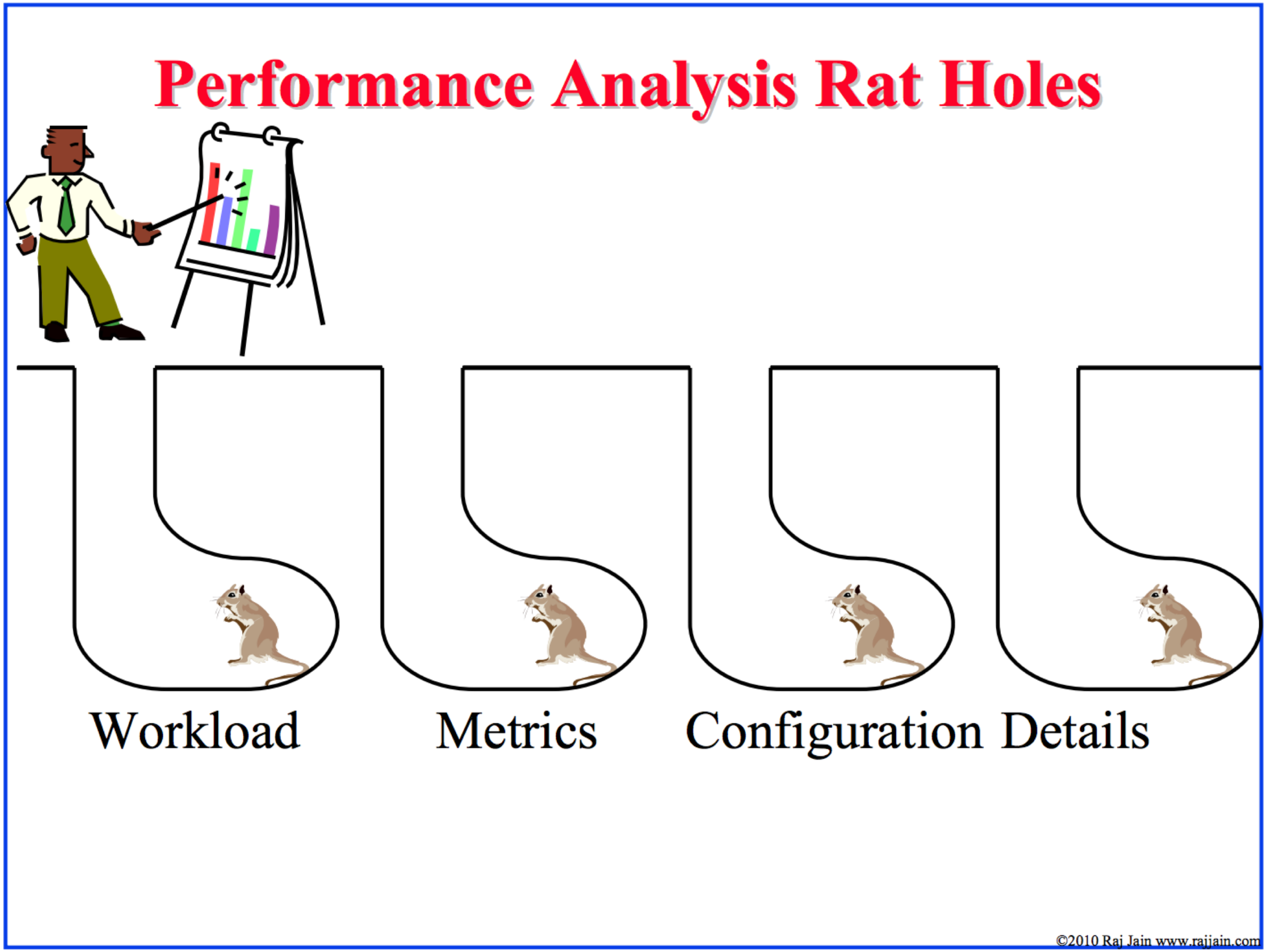
- New and simple solution to a critical problem. New algorithm and hardware architecture.
- GateKeeper does not sacrifice any of the aligner capabilities, as it does not modify or replace the alignment step.
- Design is scalable; could add more processing cores in the future.
- Some sequencers use FPGAs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



Performance Analysis Rat Holes

Workload Metrics Configuration Details

©2010 Raj Jain www.rajjain.com

Source: https://www.cse.wustl.edu/~jain/iucee/ftp/k_10adp.pdf

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
  - 3rd generation sequencing machines produce much longer reads

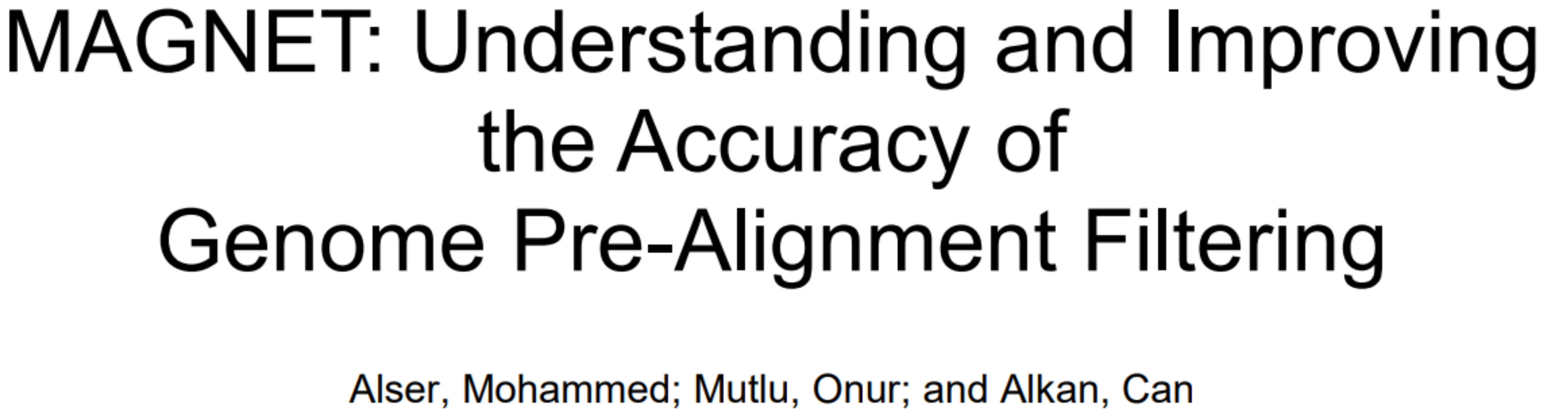
Thoughts and Ideas

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]

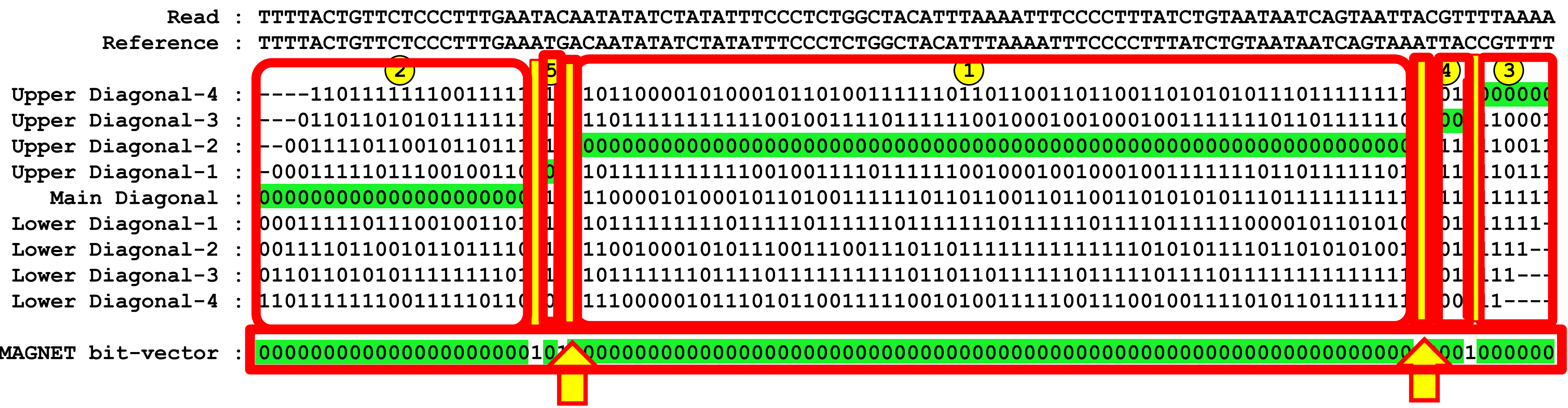


MAGNET: Understanding and Improving the Accuracy of Genome Pre-Alignment Filtering

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

MAGNET Walkthrough

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



```
             Read : TTTTACTGTTCTCCCTTTGAATACAATATATCTATATTTCCCTCTGGCTACATTTAAAATTTCCCCTTTATCTGTAATAATCAGTAATTACGTTTTAAAA
        Reference : TTTTACTGTTCTCCCTTTGAAATGACAATATATCTATATTTCCCTCTGGCTACATTTAAAATTTCCCCTTTATCTGTAATAATCAGTAAATTACCGTTTT
 Upper Diagonal-4 : ----11011111111001111 1 1011000010100010110100111111011011001101100110101010111011111111 01 000000
 Upper Diagonal-3 : ---0110110101011111111 1 1101111111111100100111011111110010001001000100111111101101111110 00 10001
 Upper Diagonal-2 : --00111101100101101111 1 0000000000000000000000000000000000000000000000000000000000000000 11 10011
 Upper Diagonal-1 : -00011111011100100110 0 1011111111111001001111011111110010001001000100111111101101111101 11 10111
    Main Diagonal : 00000000000000000000 1 1100001010001011010011111101101100110110011010101011101111111111 11 11111
 Lower Diagonal-1 : 000111110111001001101 1 1011111111101111101111110111111011111101111011111100001011010 10 01 1111-
 Lower Diagonal-2 : 0011110110010110111110 1 110010001010111001110011101101111111111111101010111011010101001 01 111--
 Lower Diagonal-3 : 011011010101111111101 1 1011111110111101111111111011011011111101111101111011111111111111 01 11---
 Lower Diagonal-4 : 110111111100111110110 0 1110000010111010110011111001010011111001110010011110101101111111 00 1----

MAGNET bit-vector : 0000000000000000000000101000000000000000000000000000000000000000000000000000000000000000000001000000
```

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

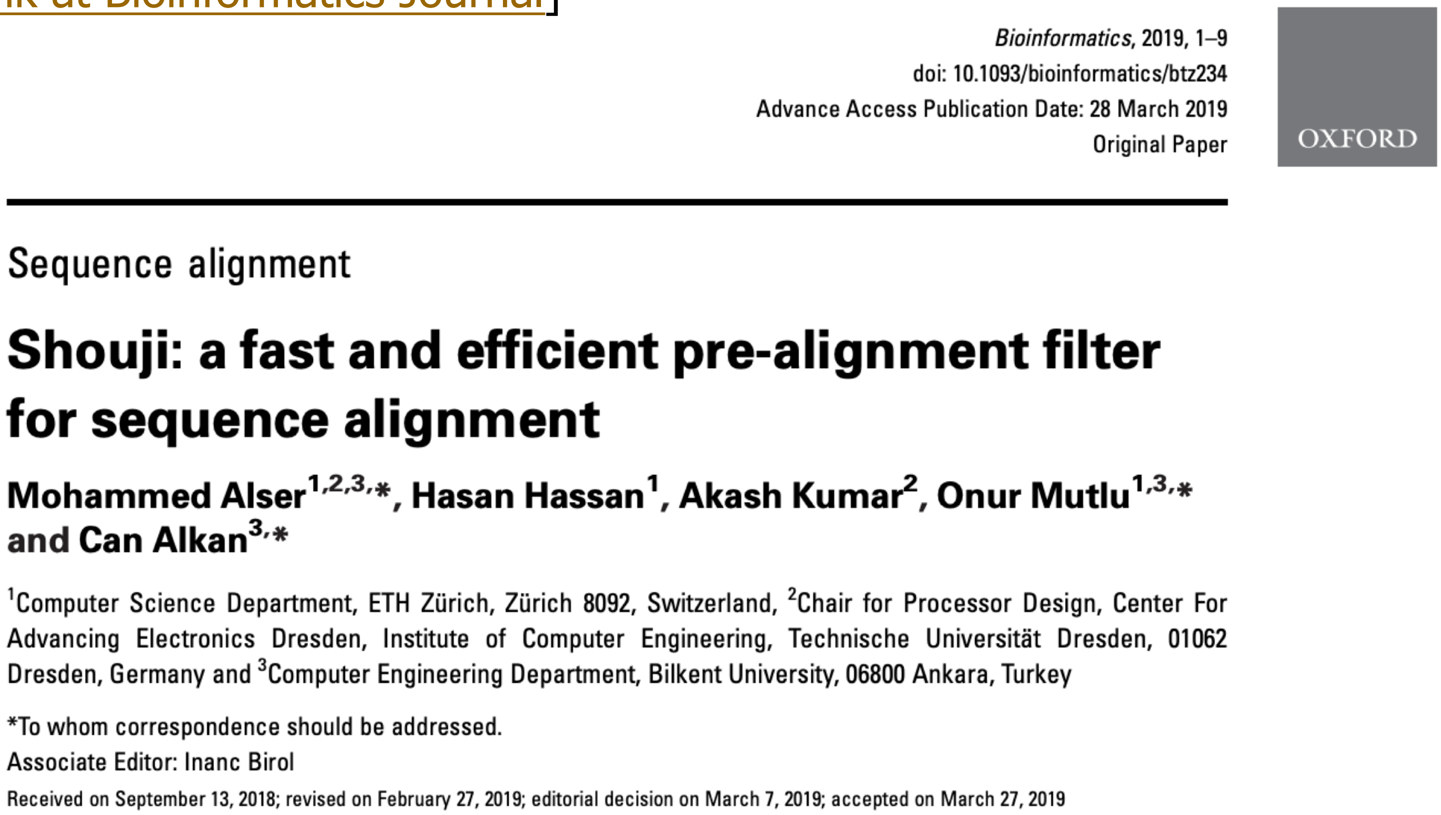
Extensions

- 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

OXFORD

Sequence alignment

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

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

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

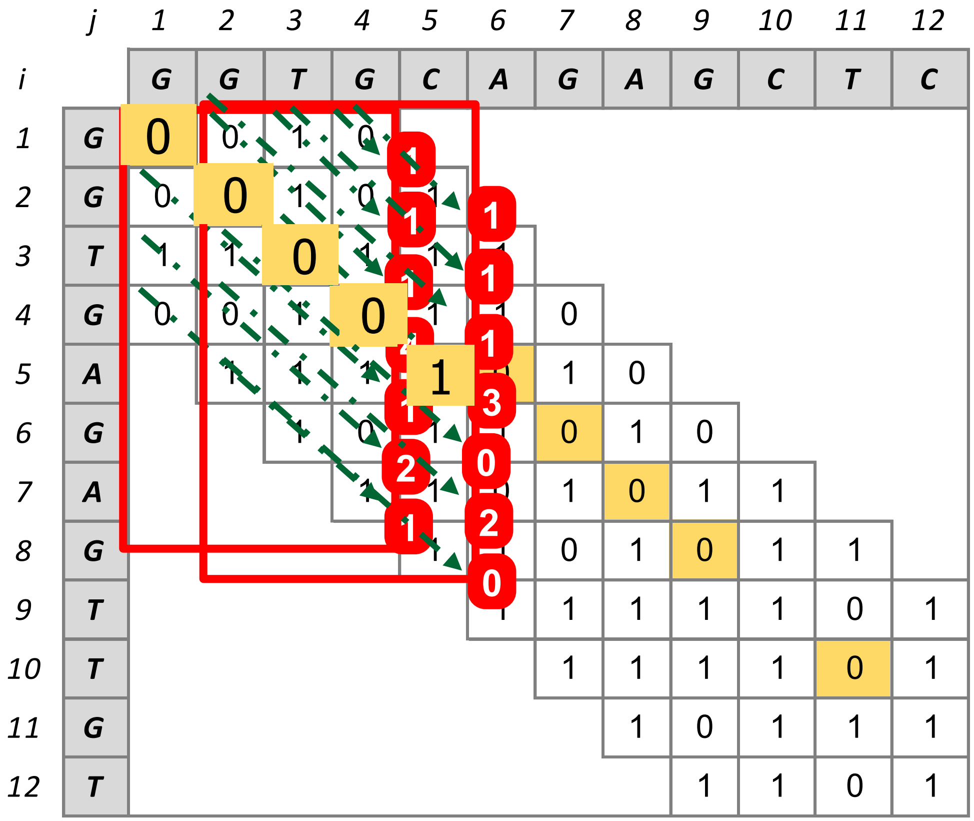
*To whom correspondence should be addressed.
Associate Editor: Inanc Birol

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

Shouji 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’ ≤ Threshold

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

Shouji Walkthrough

Building the Neig

j		1	2	3	4	5	6	7	8	9	10	11	12
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 @ ctor

0	0	0	0	**1**	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

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].

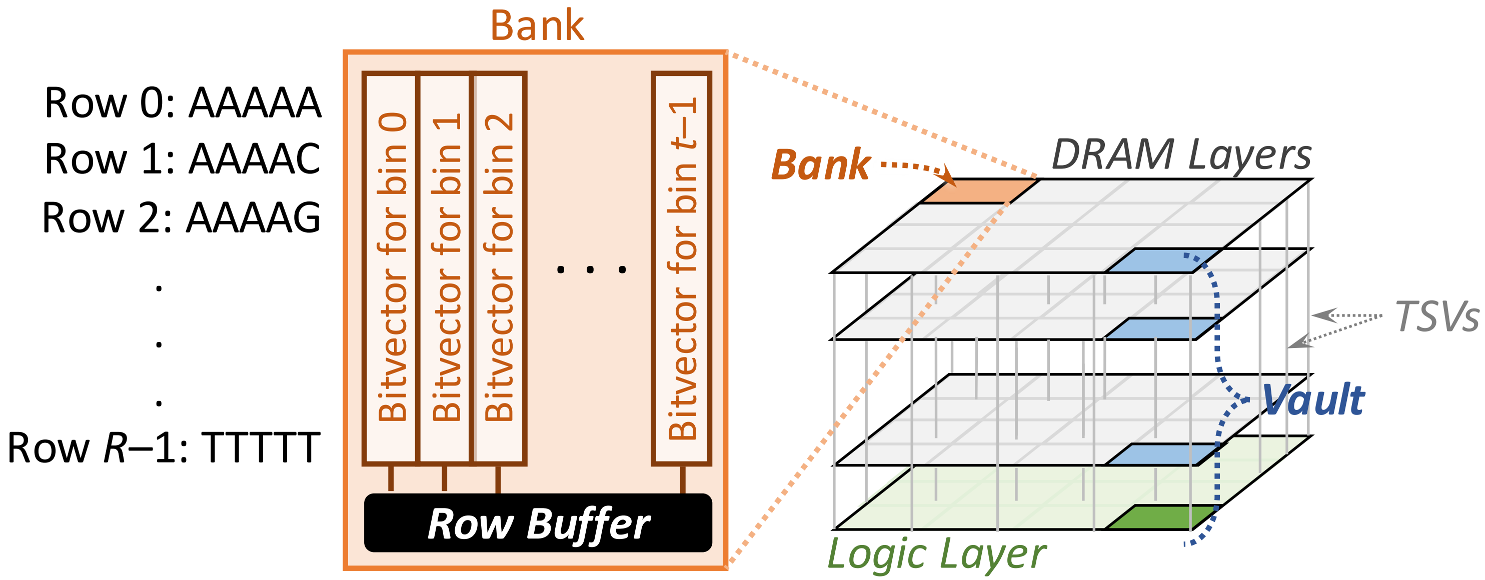
GRIM-Filter [Kim+, BMC Genomics 2018]

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

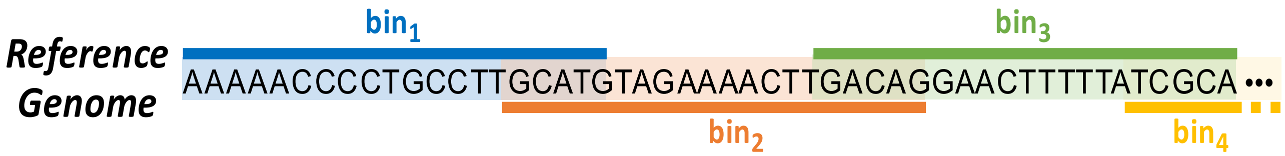
GRIM-Filter in 3D-Stacked DRAM

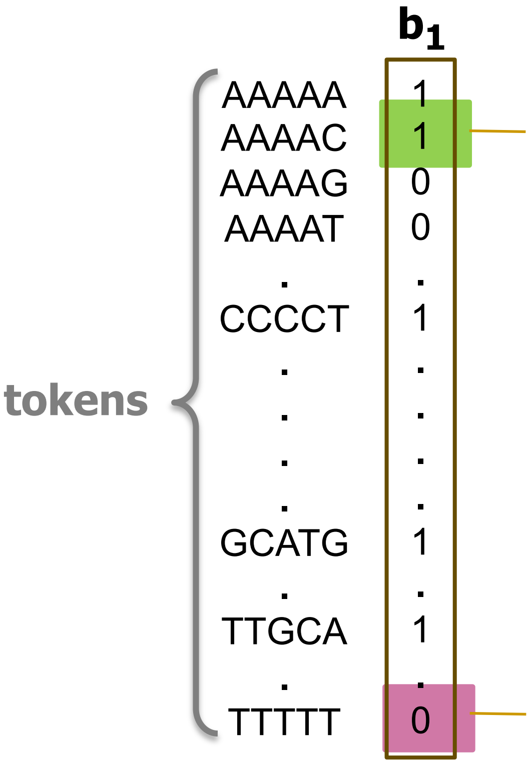


- Each DRAM layer is organized as an array of **banks**
  - A **bank** is an array of cells with a row buffer to transfer data

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

GRIM-Filter: Bitvectors





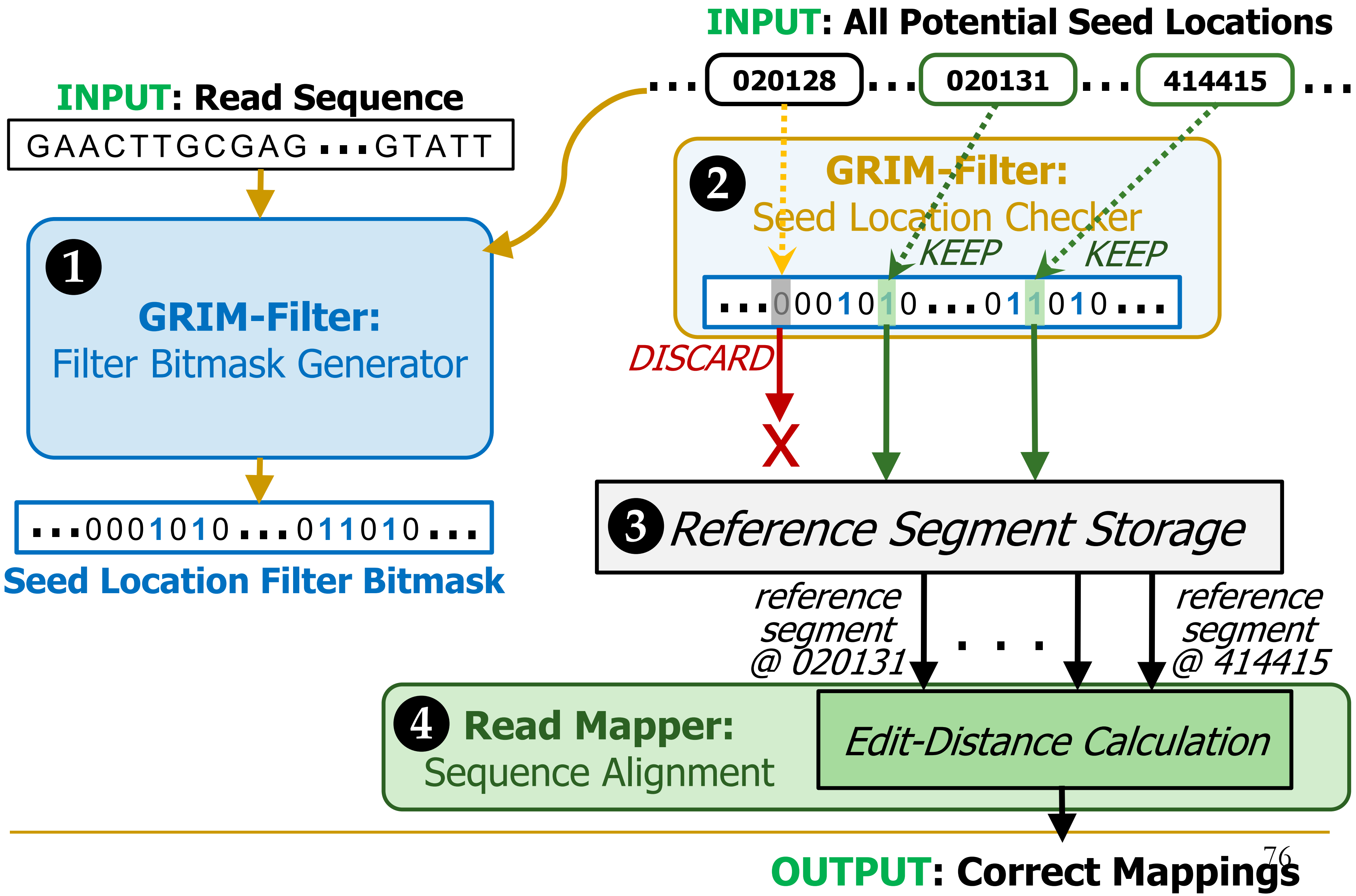
AAAAC exists in bin 1

CCCCT doesn't exist in bin 1

- Represent each bin with a **bitvector** that holds the occurrence of all permutations of a small string (**token**) in the bin

- To account for matches that straddle bins, we employ overlapping bins
  - A read will now always completely fall within a single bin

Integrating GRIM-Filter into a Read Mapper



INPUT: All Potential Seed Locations
...020128...020131...414415...
INPUT: Read Sequence
GAACTTGCGAG...GTATT
1 GRIM-Filter: Filter Bitmask Generator
...0001010...011010...
Seed Location Filter Bitmask
2 GRIM-Filter: Seed Location Checker
KEEP
KEEP
...0001010...011010...
DISCARD
X
3 Reference Segment Storage
reference segment @ 020131
. . .
reference segment @ 414415
4 Read Mapper: Sequence Alignment
Edit-Distance Calculation
OUTPUT: Correct Mappings

Can We Do Better?

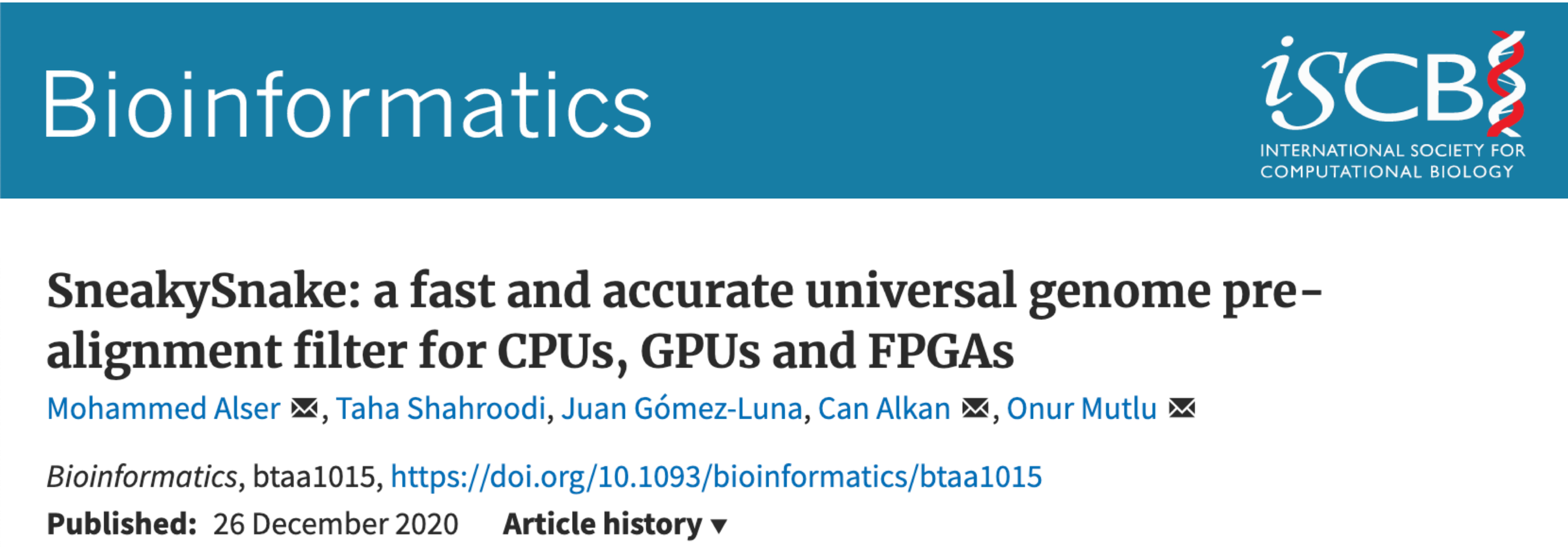
Faster, More Accurate,

More Scalable

Pre-Alignment Filtering

SneakySnake [Alser+, Bioinformatics 2020]

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



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

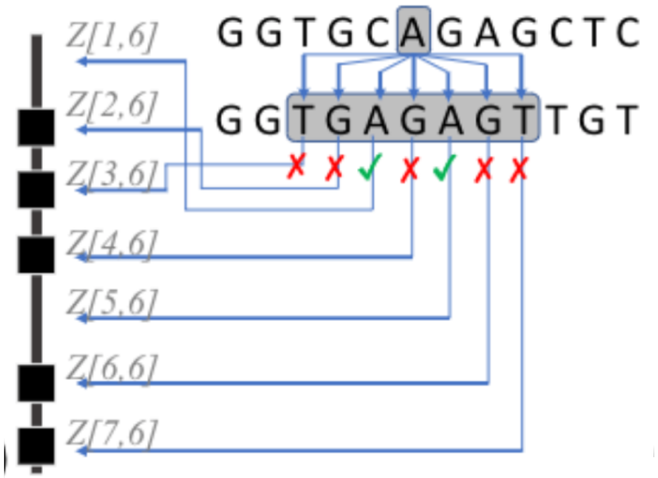
Mohammed Alser ✉, Taha Shahroodi, Juan Gómez-Luna, Can Alkan ✉, Onur Mutlu ✉

Bioinformatics, btaa1015, https://doi.org/10.1093/bioinformatics/btaa1015

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

SneakySnake 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 $R[1 \ldots m]$ and a query sequence $Q[1 \ldots m]$, 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:

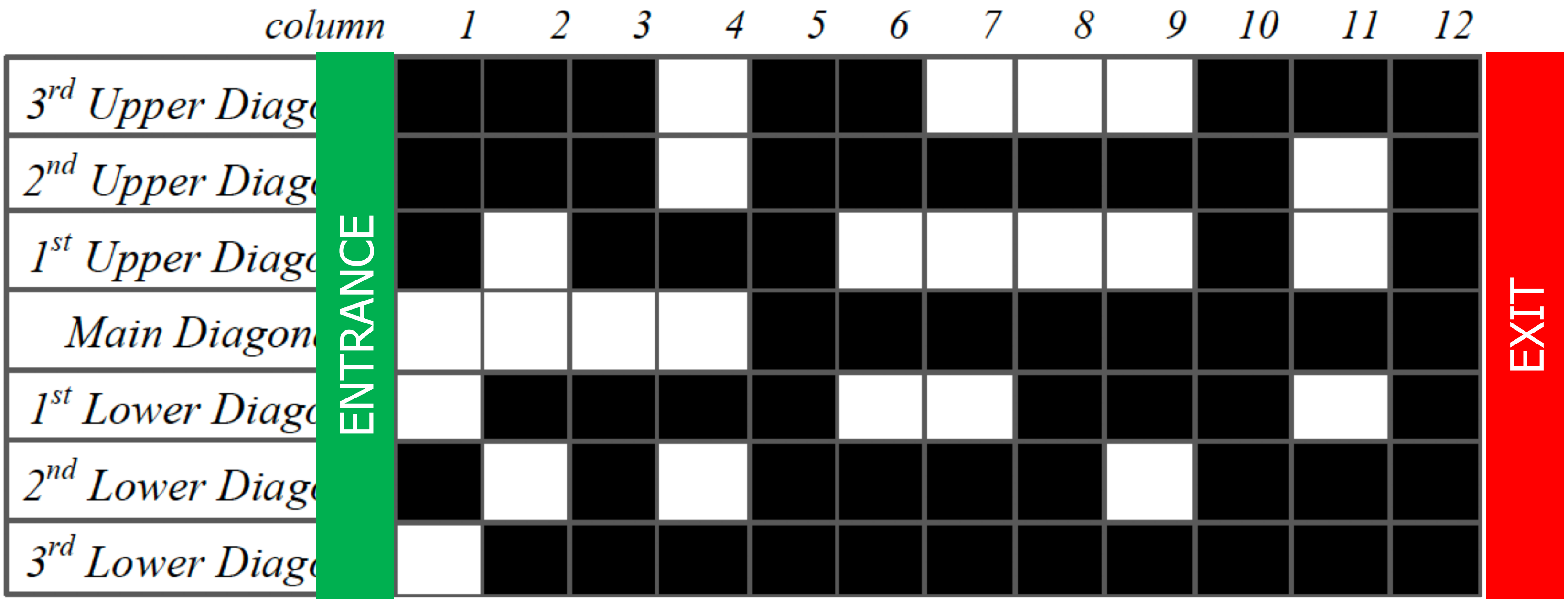
$$Z[i,j] = \begin{cases} 0, & if \;\; i = E+1, \; Q[j] = R[j], \\ 0, & if \;\; 1 \leq i \leq E, \; Q[j-i] = R[j], \\ 0, & if \;\; i > E+1, \; Q[j+i-E-1] = R[j], \\ 1, & otherwise \end{cases} \qquad (1)$$

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

SneakySnake Walkthrough

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

$E = 3$

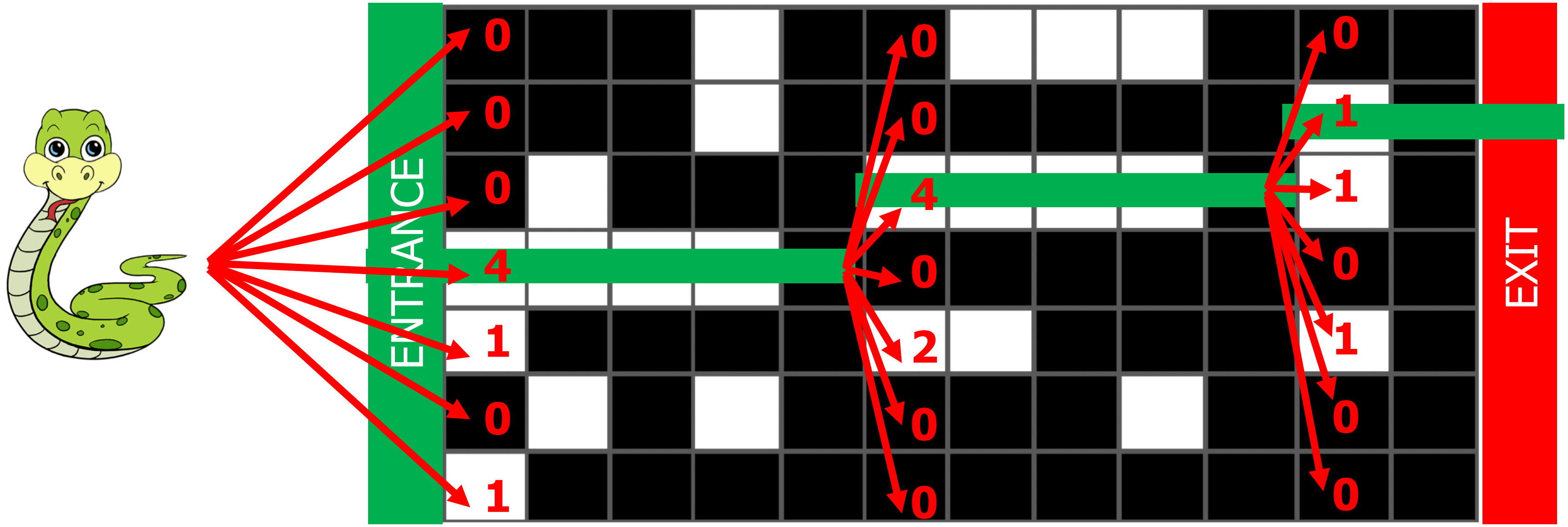


column	1	2	3	4	5	6	7	8	9	10	11	12
3rd Upper Diag	■	■	■	□	■	■	□	□	□	■	■	■
2nd Upper Diag	■	■	■	□	■	■	■	■	■	■	□	■
1st Upper Diag	■	□	■	■	■	□	□	□	□	■	□	■
Main Diagon	□	□	□	□	■	■	■	■	■	■	■	■
1st Lower Diag	□	■	■	■	■	□	□	■	■	■	□	■
2nd Lower Diag	■	□	■	□	■	■	■	■	□	■	■	■
3rd Lower Diag	□	■	■	■	■	■	■	■	■	■	■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

SneakySnake Walkthrough

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

■ 0

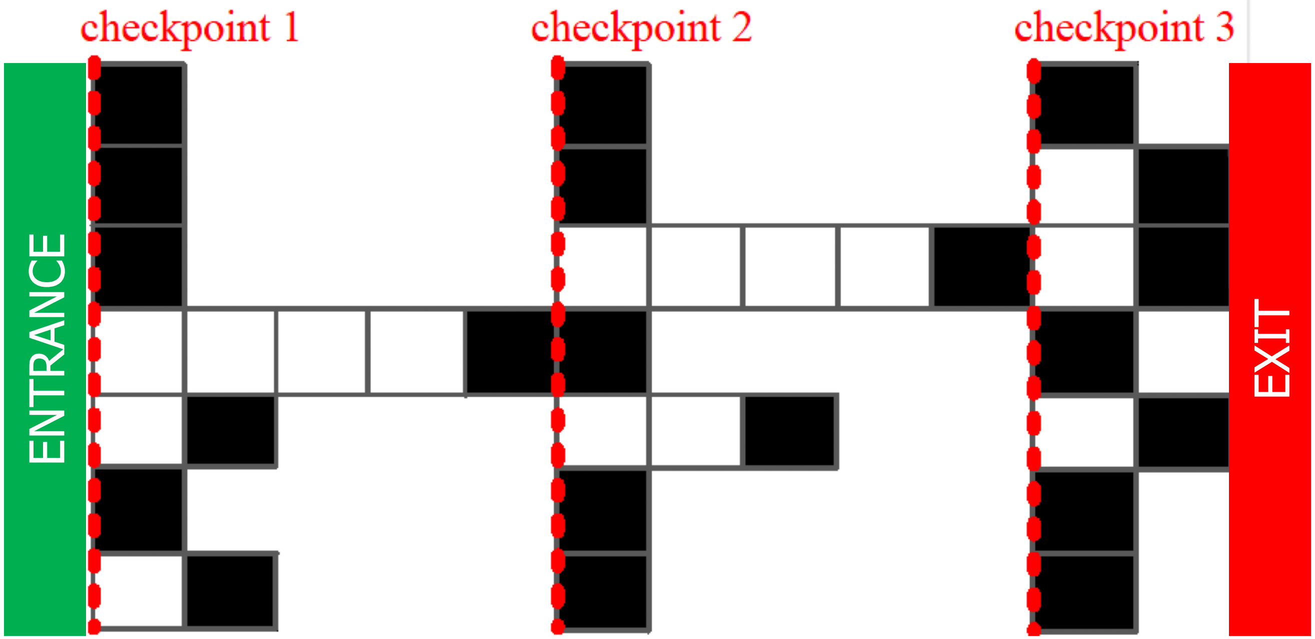


SneakySnake Walkthrough

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

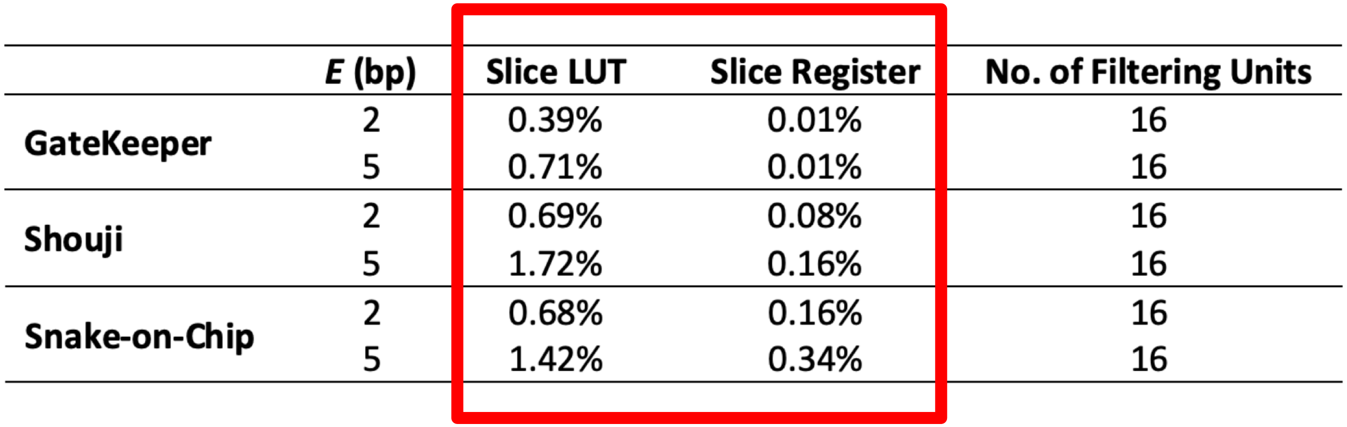
■ 3

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



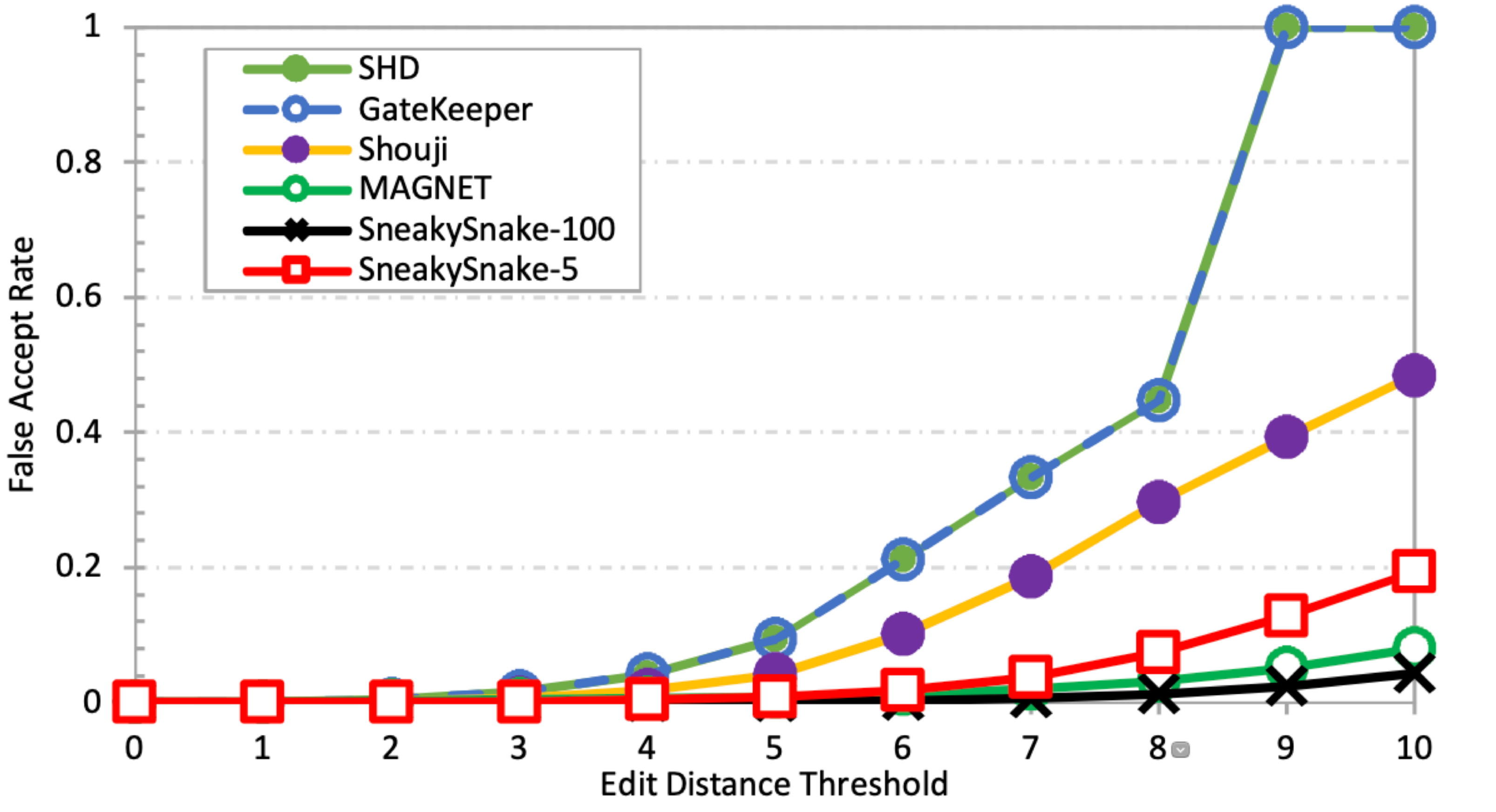
FPGA Resource Analysis

- FPGA resource usage for a single filtering unit of GateKeeper, Shouji, and Snake-on-Chip for a sequence length of 100 and under different edit distance thresholds (E).



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

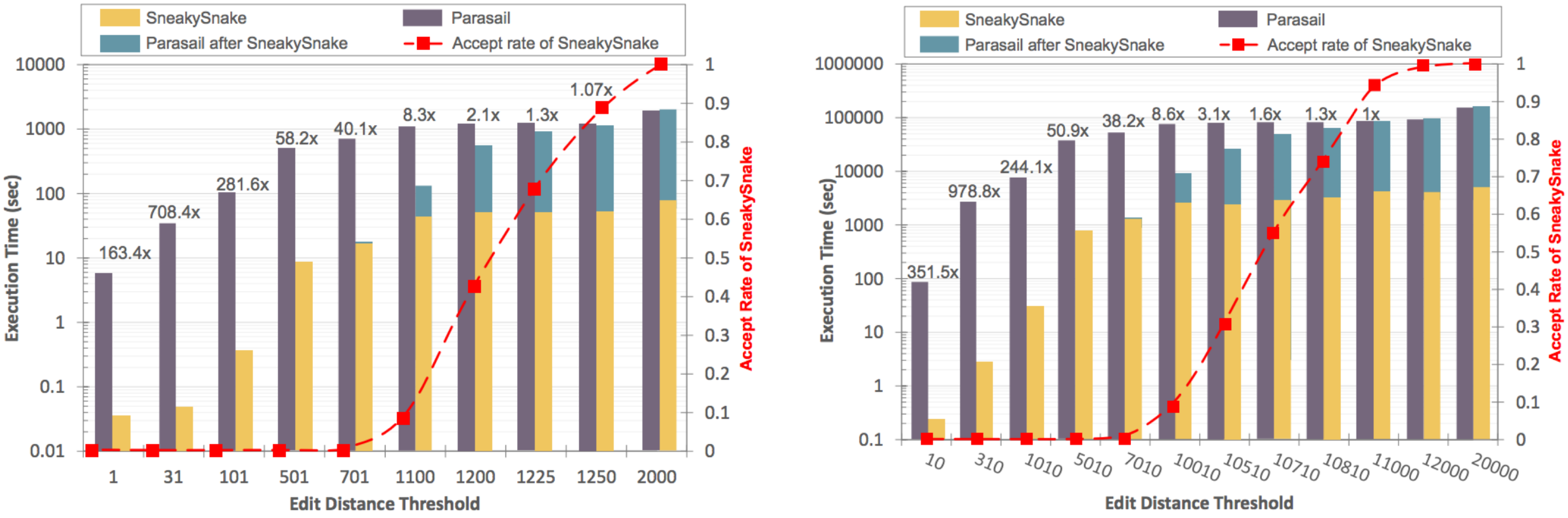
Filtering Accuracy



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

Long Read Mapping (SneakySnake vs Parasail)

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

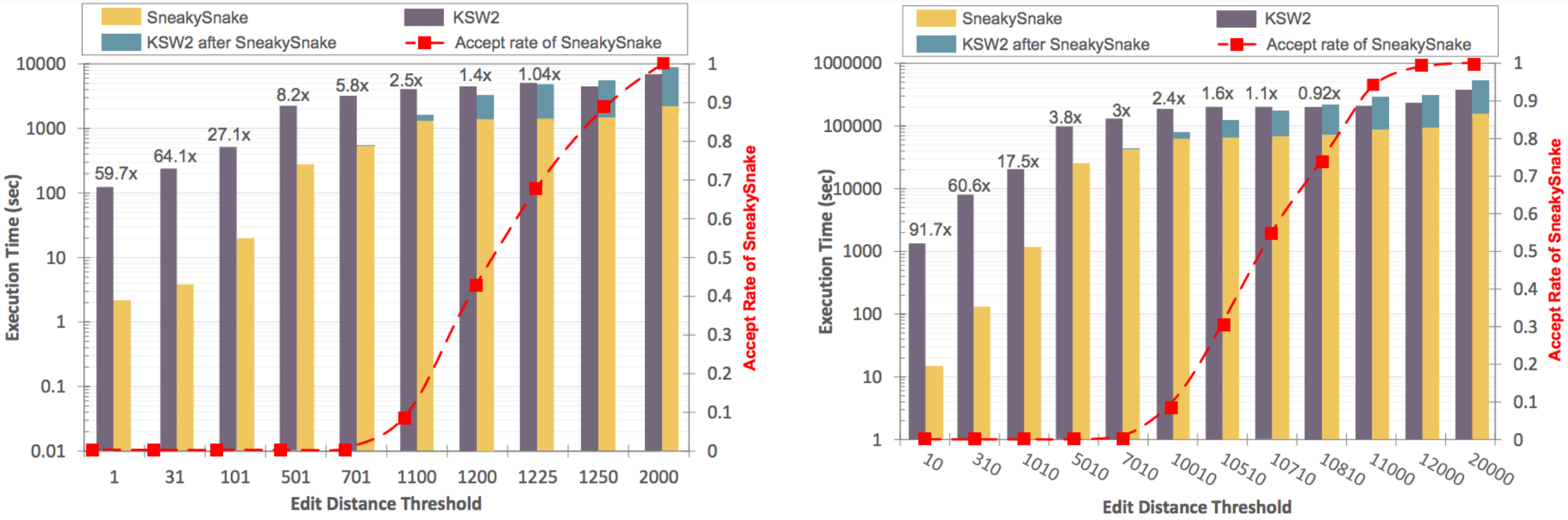


(a) **(b)**

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

Long Read Mapping (SneakySnake vs KSW2)

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



(a) (b)

Fig. 11: The execution time of SneakySnake, KSW2, and SneakySnake integrated with KSW2 using long sequences, (a) 10Kbp and (b) 100Kbp, and a single CPU thread. The left y-axes of (a) and (b) are on a logarithmic scale. For each edit distance threshold value, we provide in the right y-axes of (a) and (b) the rate of accepted pairs (out of 100,000 pairs for 10Kbp and out of 74,687 pairs for 100Kbp) by SneakySnake that are passed to KSW2. We present the end-to-end speedup values obtained by integrating SneakySnake with KSW2.

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)

- 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?

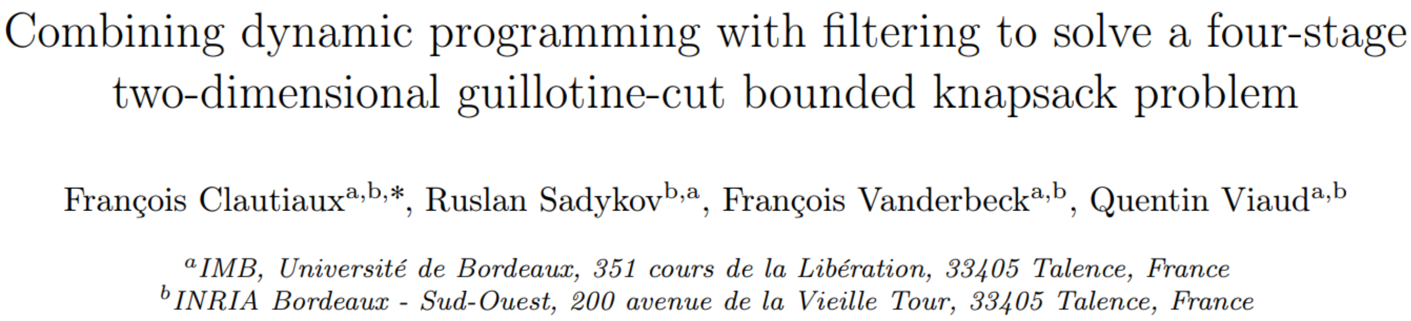
Discussion Starters (II)

- 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

Illumina DRAGEN Bio-IT Platform

Discussion Starters (III)

- Can you think of fields that could be similarly in need of string alignment as 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

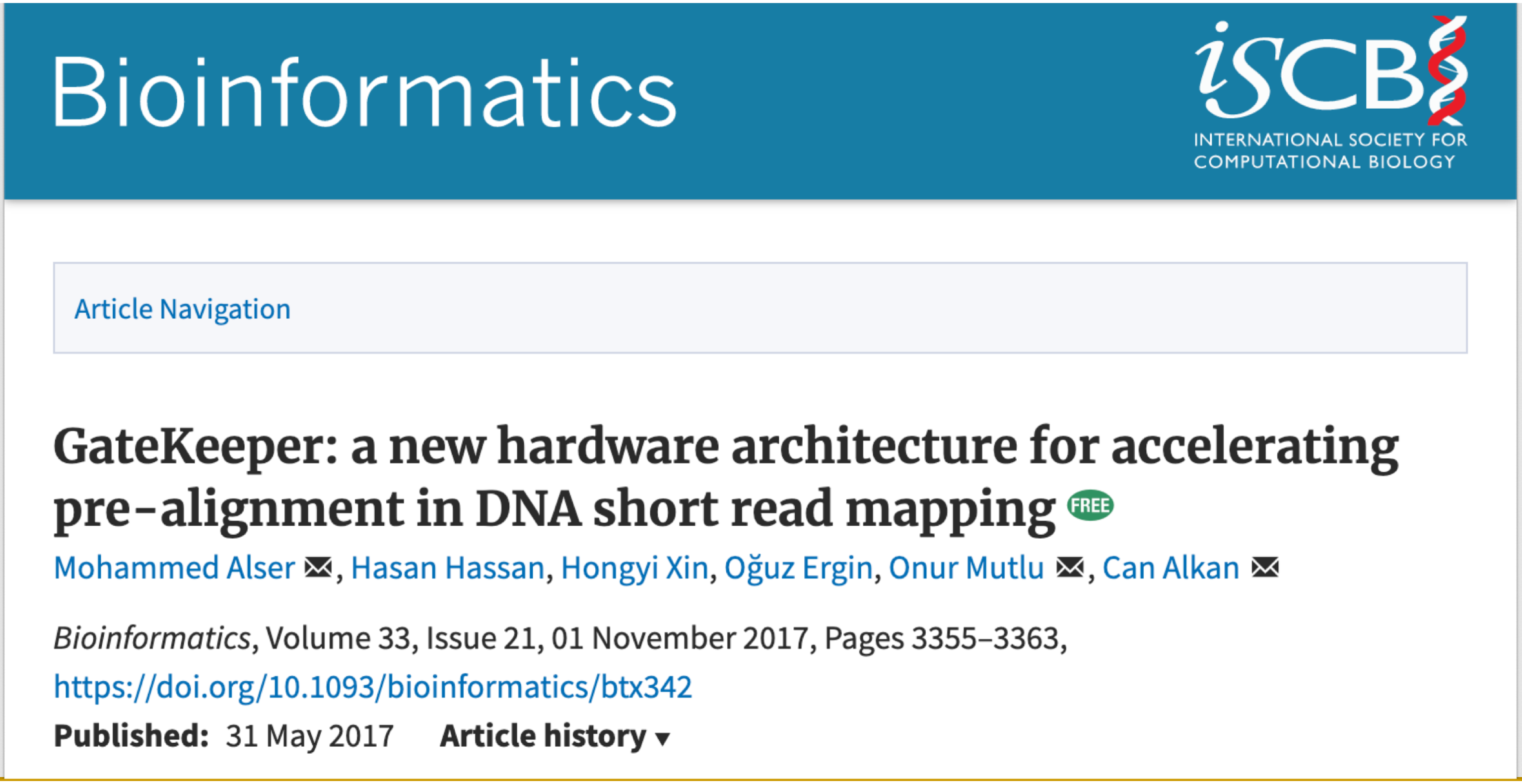
François Clautiaux[a,b,*], Ruslan Sadykov[b,a], François Vanderbeck[a,b], Quentin Viaud[a,b]

[a] *IMB, Université de Bordeaux, 351 cours de la Libération, 33405 Talence, France*
[b] *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.

More Details 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]



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

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

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

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

GateKeeper: Accelerating Pre-Alignment in DNA Read Mapping

Mohammed Alser[1], Hasan Hassan[2,3], Hongyi Xin[4], Oğuz Ergin[2], Onur Mutlu[1,3,4], Can Alkan[1]

Bioinformatics, 2017

1 Bilkent University

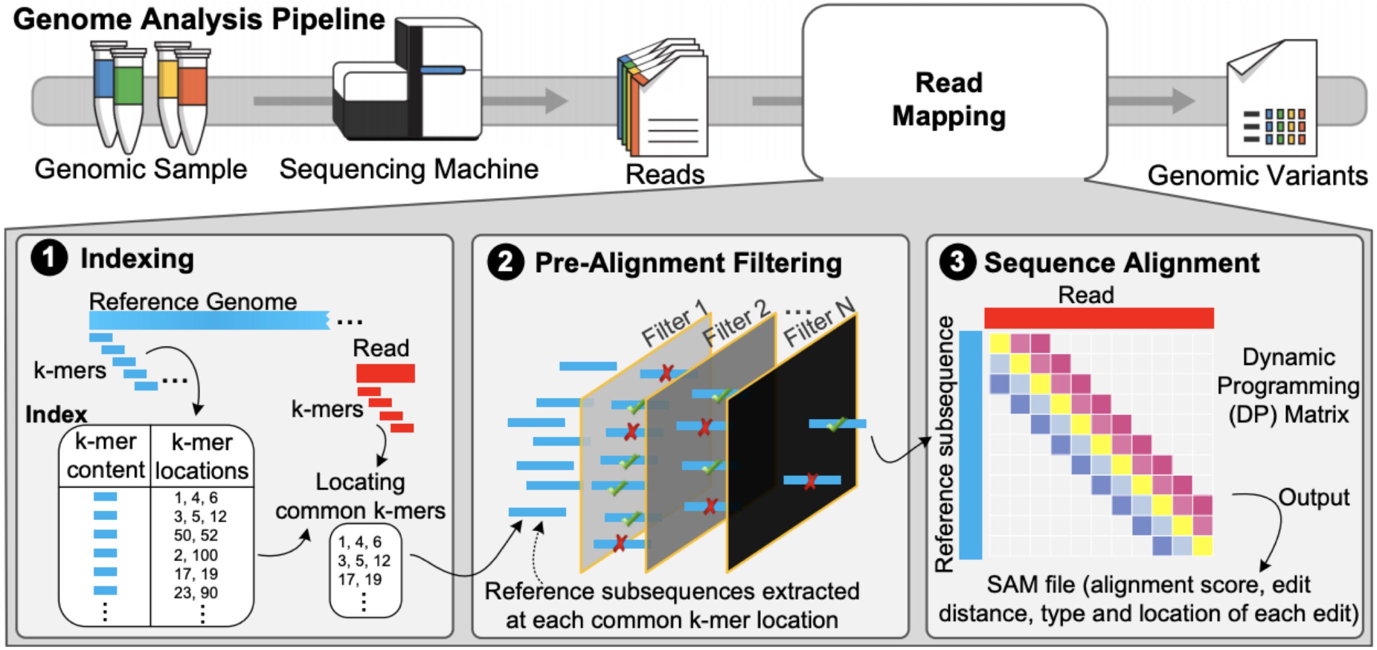
2 TOBB University of Economics & Technology

3 ETH zürich

4 Carnegie Mellon

What else can be done?

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

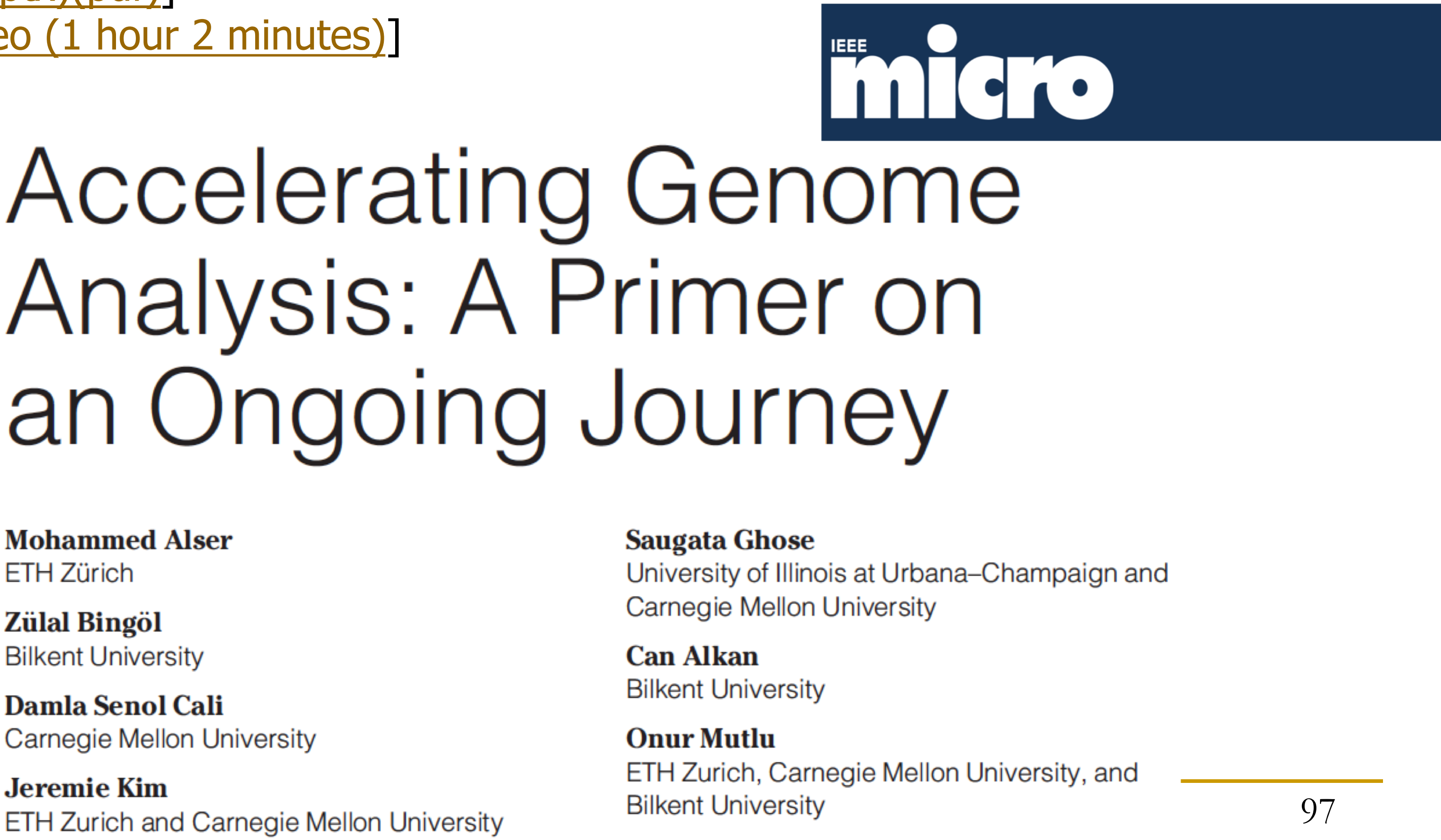
Accelerating Alignment

- Accurate alignment accelerators
- Heuristic-based alignment accelerators

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

Accelerating Genome Analysis: Overview

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


IEEE micro

Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser
ETH Zürich

Zülal Bingöl
Bilkent University

Damla Senol Cali
Carnegie Mellon University

Jeremie Kim
ETH Zurich and Carnegie Mellon University

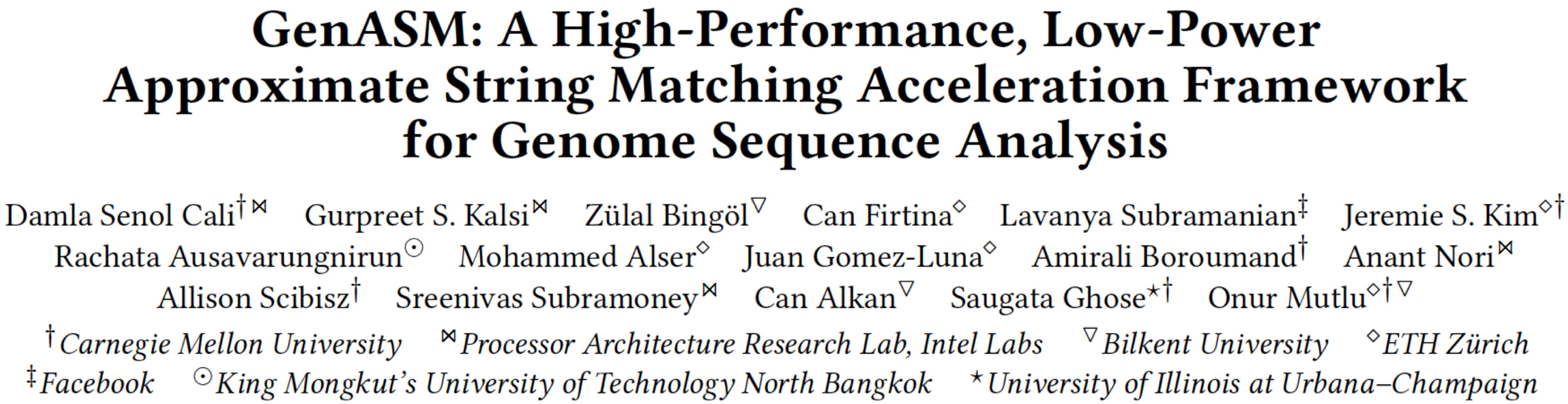
Saugata Ghose
University of Illinois at Urbana–Champaign and Carnegie Mellon University

Can Alkan
Bilkent University

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

GenASM Framework **[MICRO 2020]**

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



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

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

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

Problem & Our Goal

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

Our Goal:

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

GenASM: ASM Framework for GSA

Our Goal:

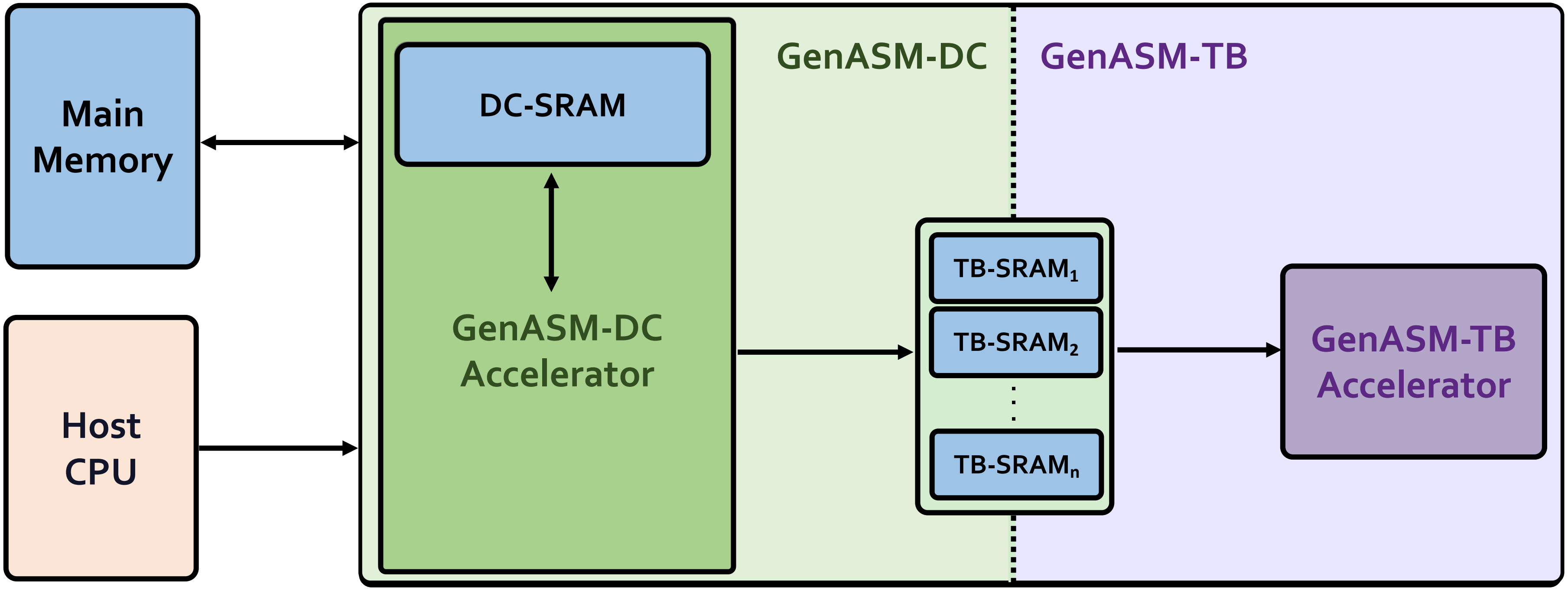
Accelerate approximate string matching

by designing a fast and flexible framework,

which can accelerate *multiple steps* of genome sequence analysis

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

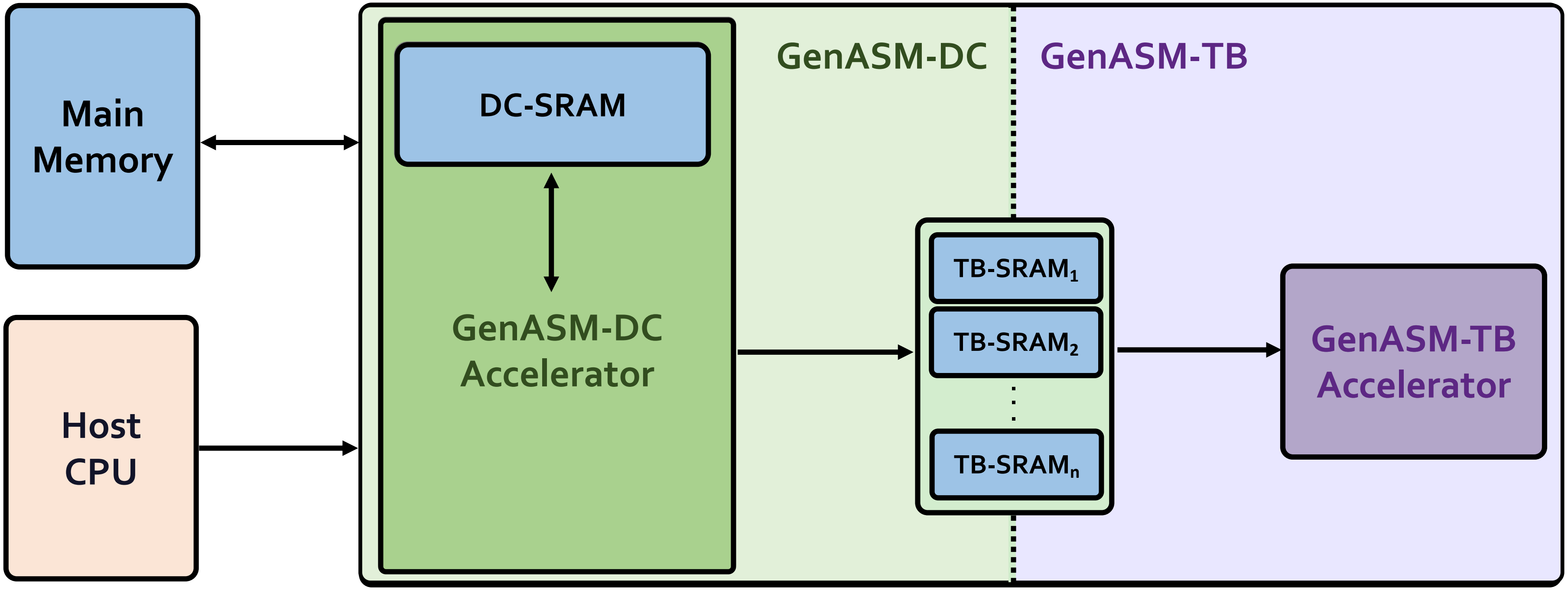
GenASM: Hardware Design



GenASM-DC:
generates bitvectors
and performs edit
Distance **C**alculation

GenASM-TB:
performs **T**race**B**ack
and assembles the
optimal alignment

GenASM: Hardware Design



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

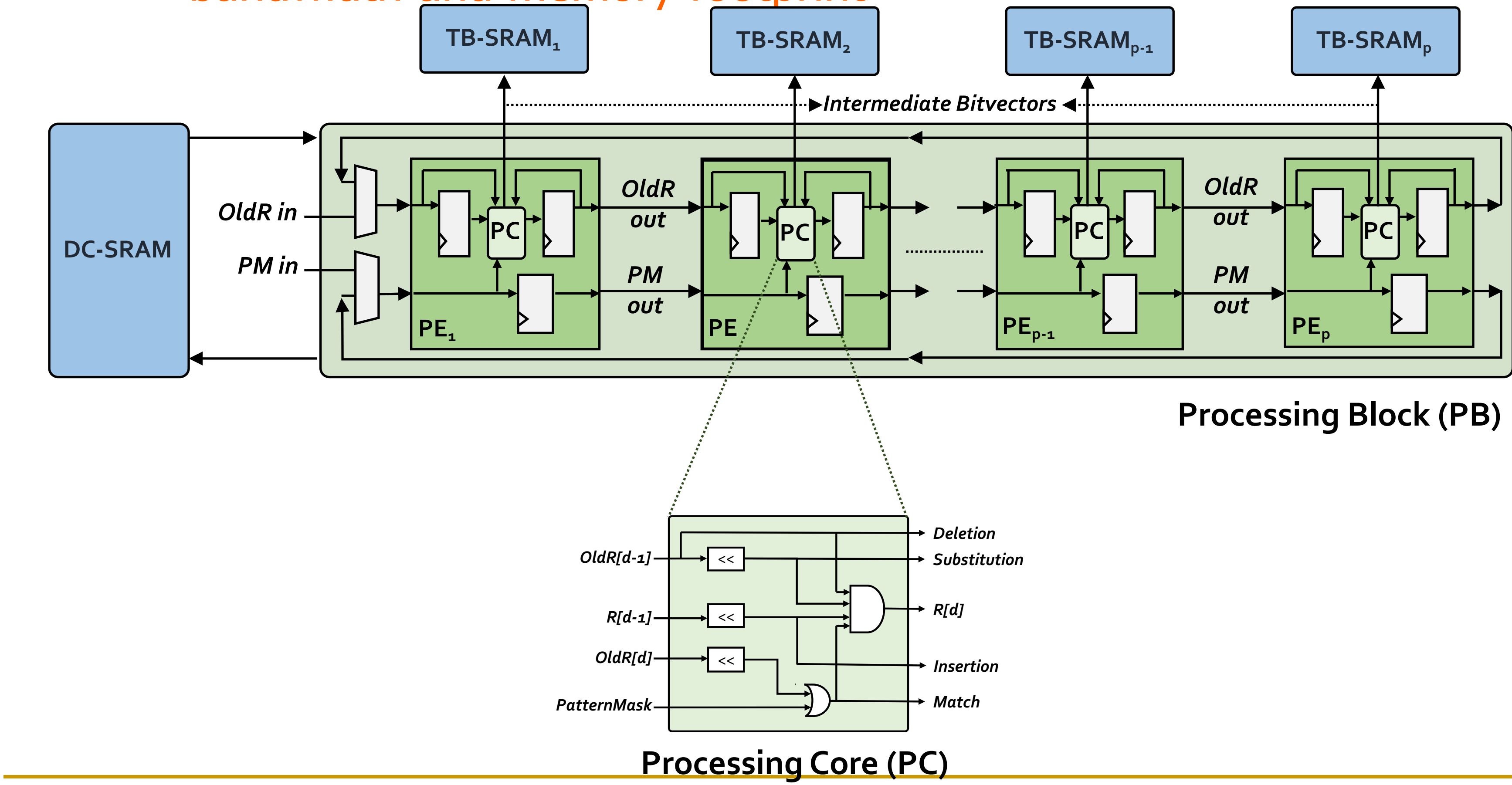
→ Match the rate of computation with memory capacity and bandwidth

→ Achieve high performance and power efficiency

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

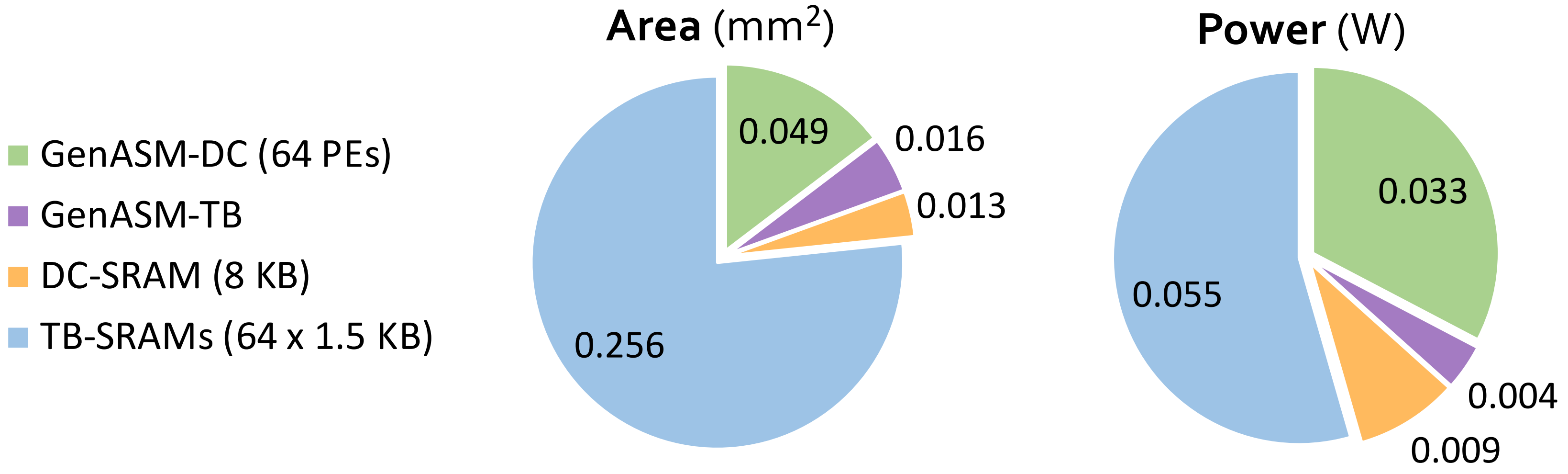
GenASM-DC: Hardware Design

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



Key Results – Area and Power

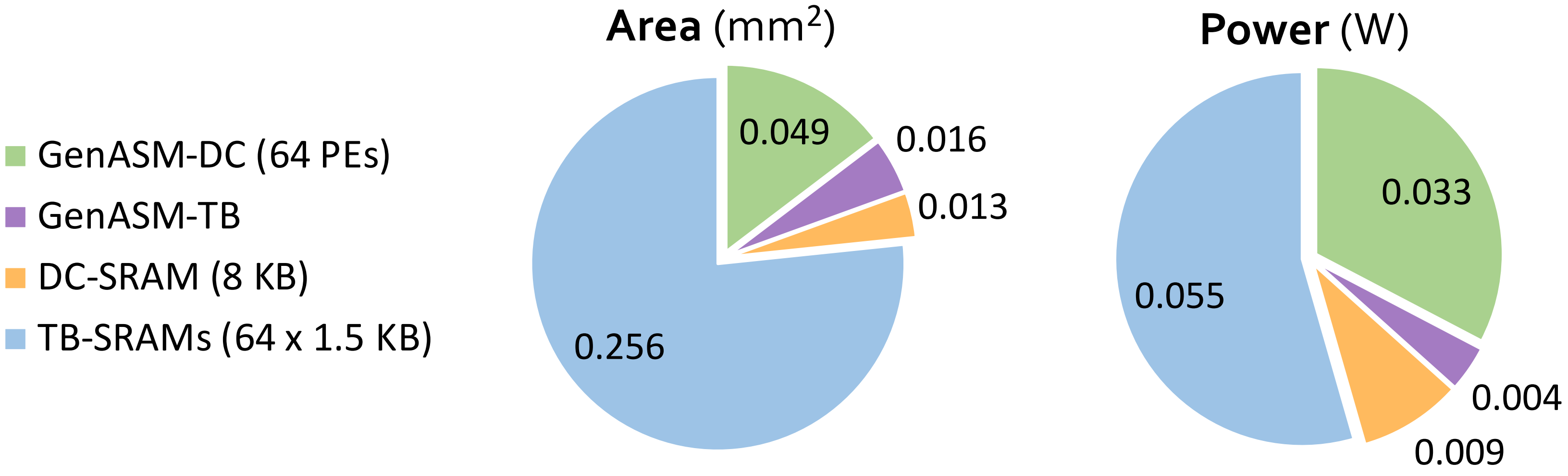
- Based on our **synthesis** of **GenASM-DC** and **GenASM-TB** accelerator datapaths using the Synopsys Design Compiler with a **28nm** LP process:
  - Both GenASM-DC and GenASM-TB operate **@ 1GHz**



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

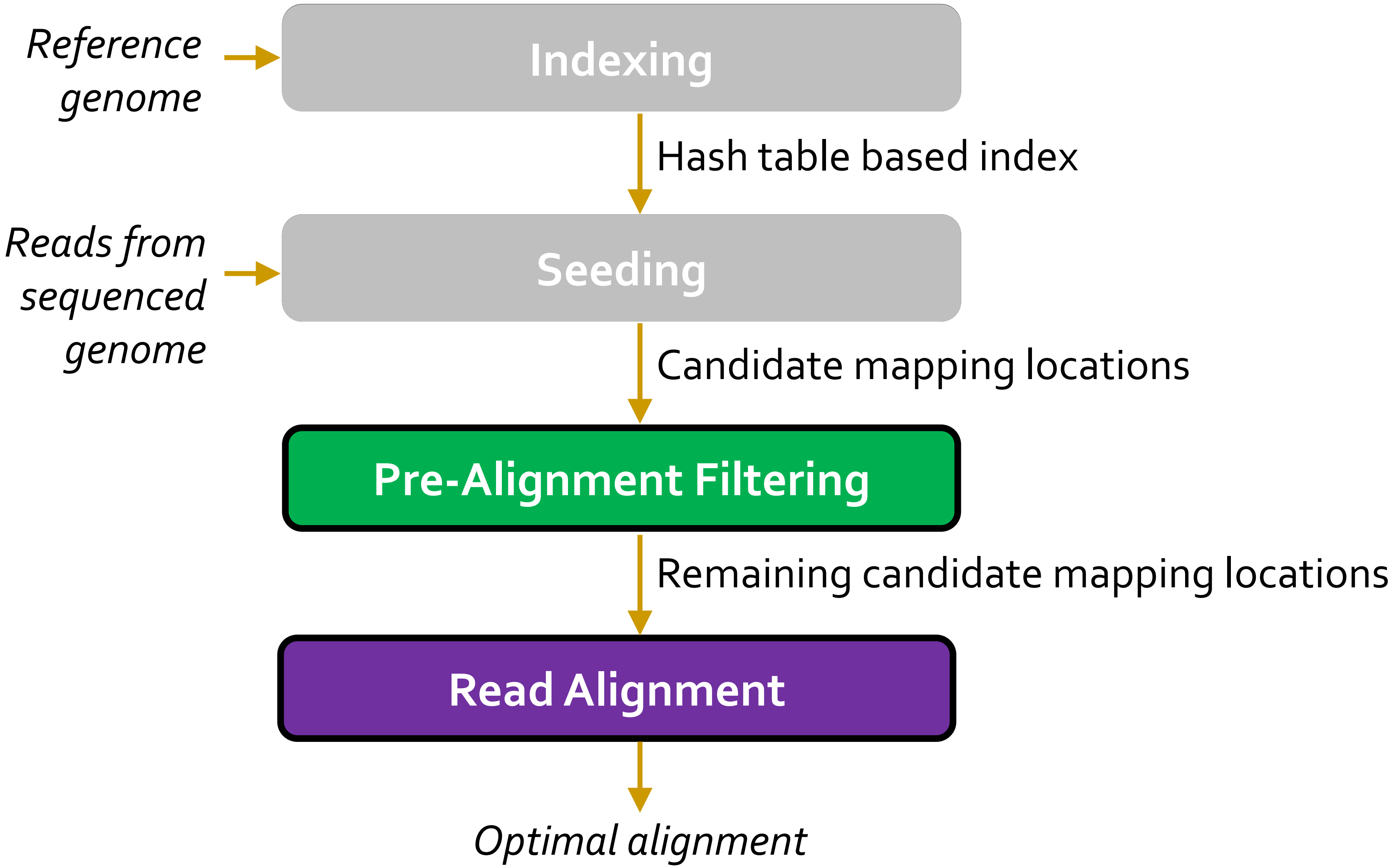
Key Results – Area and Power

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



GenASM has low area and power overheads

Use Cases of GenASM



Use Cases of GenASM (cont'd.)

(1) Read Alignment Step of Read Mapping

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

(2) Pre-Alignment Filtering for Short Reads

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

(3) Edit Distance Calculation

- Measure the similarity or distance between two sequences

- We also discuss other possible use cases of GenASM in our paper:
  - Read-to-read overlap finding, hash-table based indexing, whole genome alignment, generic text search

Key Results

(1) Read Alignment

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

(2) Pre-Alignment Filtering

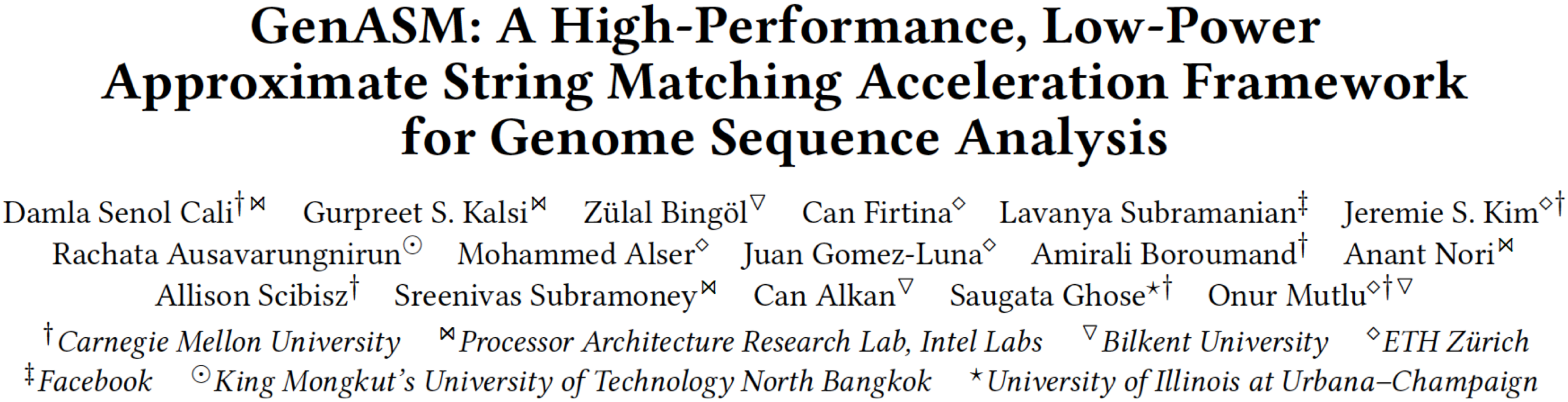
- **3.7×** speedup, **1.7×** less power than **Shouji** (state-of-the-art **HW**)

(3) Edit Distance Calculation

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

More on GenASM Framework [MICRO 2020]

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



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

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

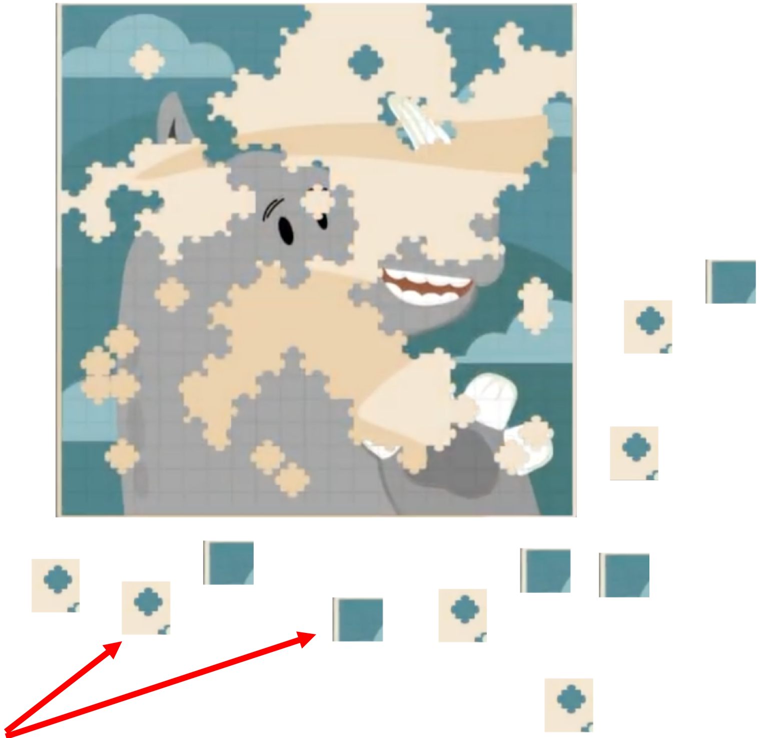
[†]*Carnegie Mellon University* [⋈]*Processor Architecture Research Lab, Intel Labs* [▽]*Bilkent University* [◇]*ETH Zürich*
[‡]*Facebook* [⊙]*King Mongkut's University of Technology North Bangkok* [⋆]*University of Illinois at Urbana–Champaign*

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

.FASTA file

Reference genome

.FASTQ file

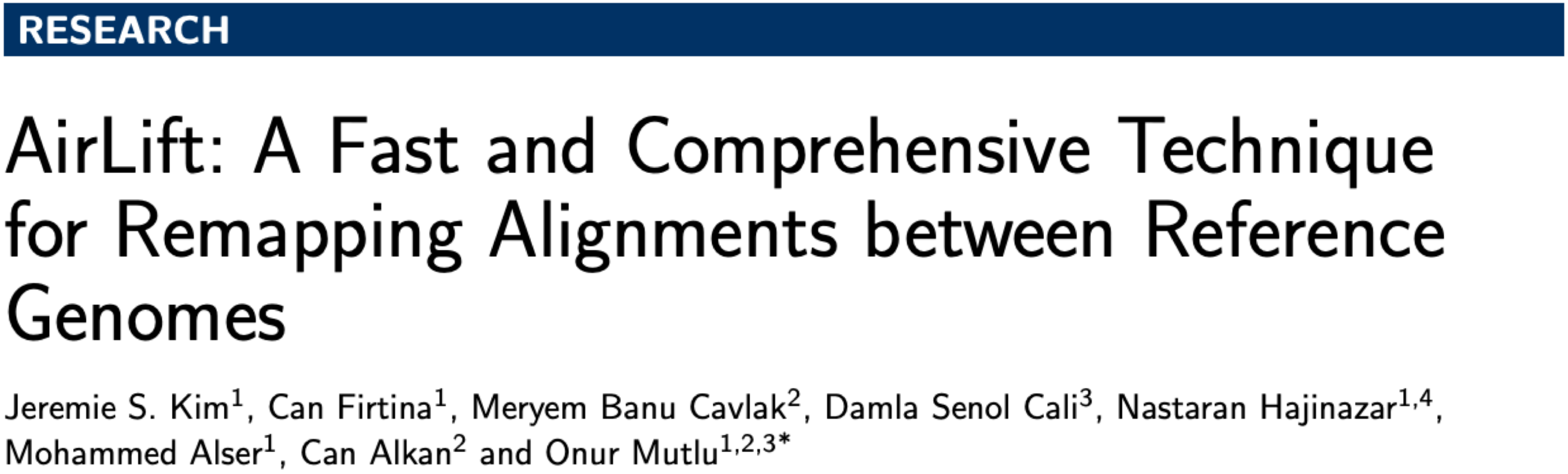


Reads

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[1], Can Firtina[1], Meryem Banu Cavlak[2], Damla Senol Cali[3], Nastaran Hajinazar[1,4], Mohammed Alser[1], Can Alkan[2] and Onur Mutlu[1,2,3*]

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

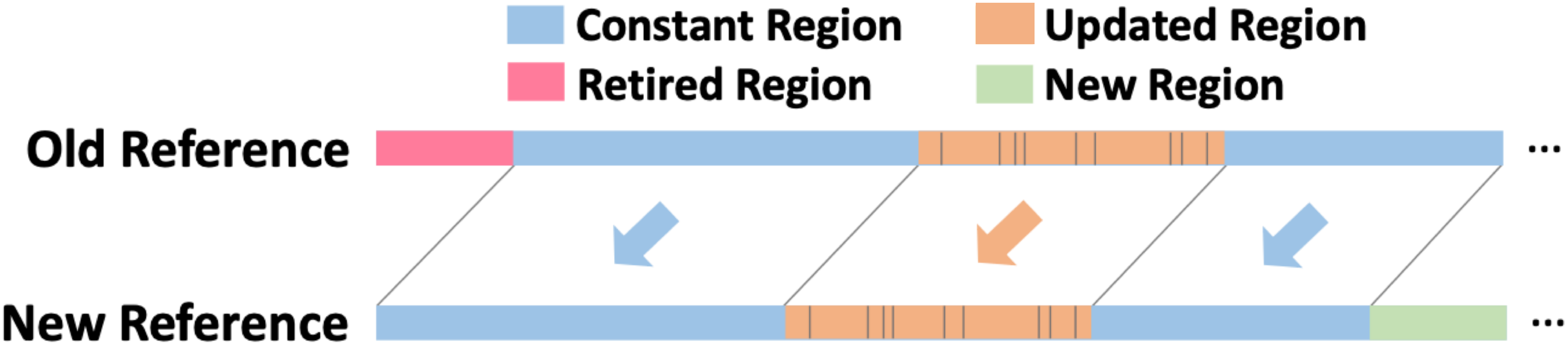
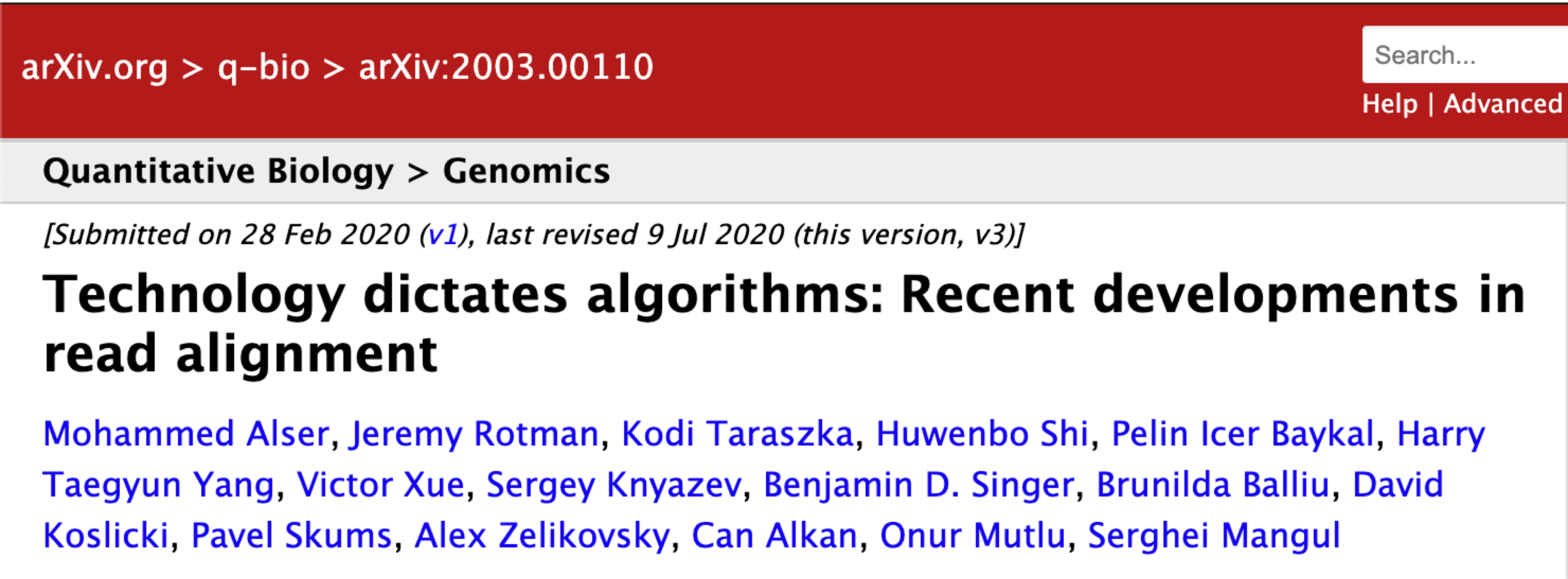


Fig. 2. Reference Genome Regions.

Read Mapping in 111 pages!

Analyzing 107 read mappers (1988-2020) in depth



Quantitative Biology > Genomics

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

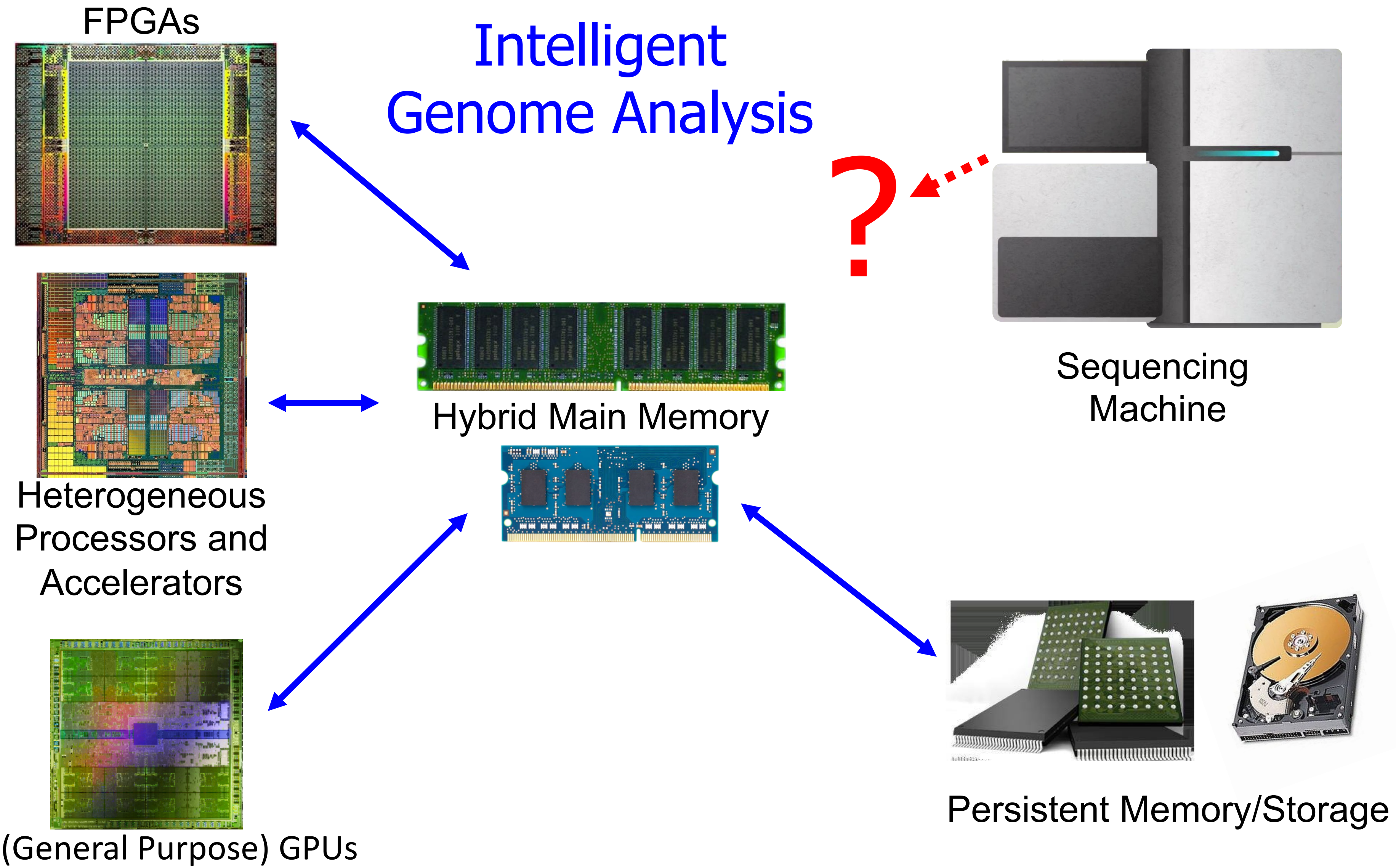
Technology dictates algorithms: Recent developments in read alignment

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

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

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

Processing Genomic Data Where it Makes Sense



Intelligent Genome Analysis

FPGAs

Heterogeneous Processors and Accelerators

(General Purpose) GPUs

Hybrid Main Memory

?

Sequencing Machine

Persistent Memory/Storage

What is Intelligent Genome Analysis?

- Fast genome analysis — Bandwidth
  - *Real-time analysis*

- Using intelligent architectures — Energy-efficiency & Latency
  - *Specialized HW with less data movement*

- DNA is a valuable asset — Privacy
  - *Controlled-access analysis*

- Population-scale genome analysis — Scalability
  - *Sequence anywhere at large scale!*

- Avoiding erroneous analysis — Accuracy
  - *E.g., your father is not your father*

Achieving Intelligent Genome Analysis?

How and where to enable

fast, accurate, cheap,

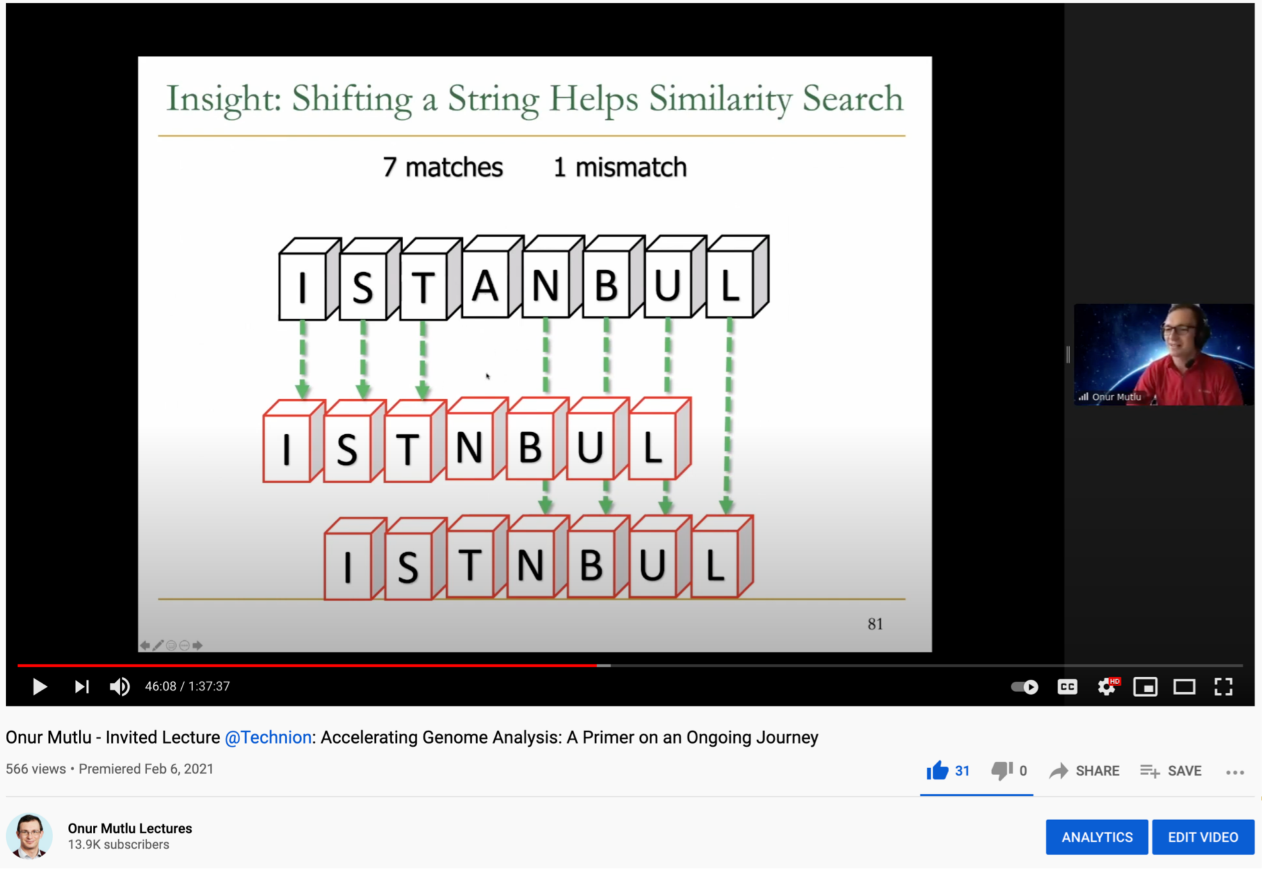
privacy-preserving, and exabyte scale

analysis of genomic data?

Most speedup comes from parallelism enabled by novel architectures and algorithms

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)]

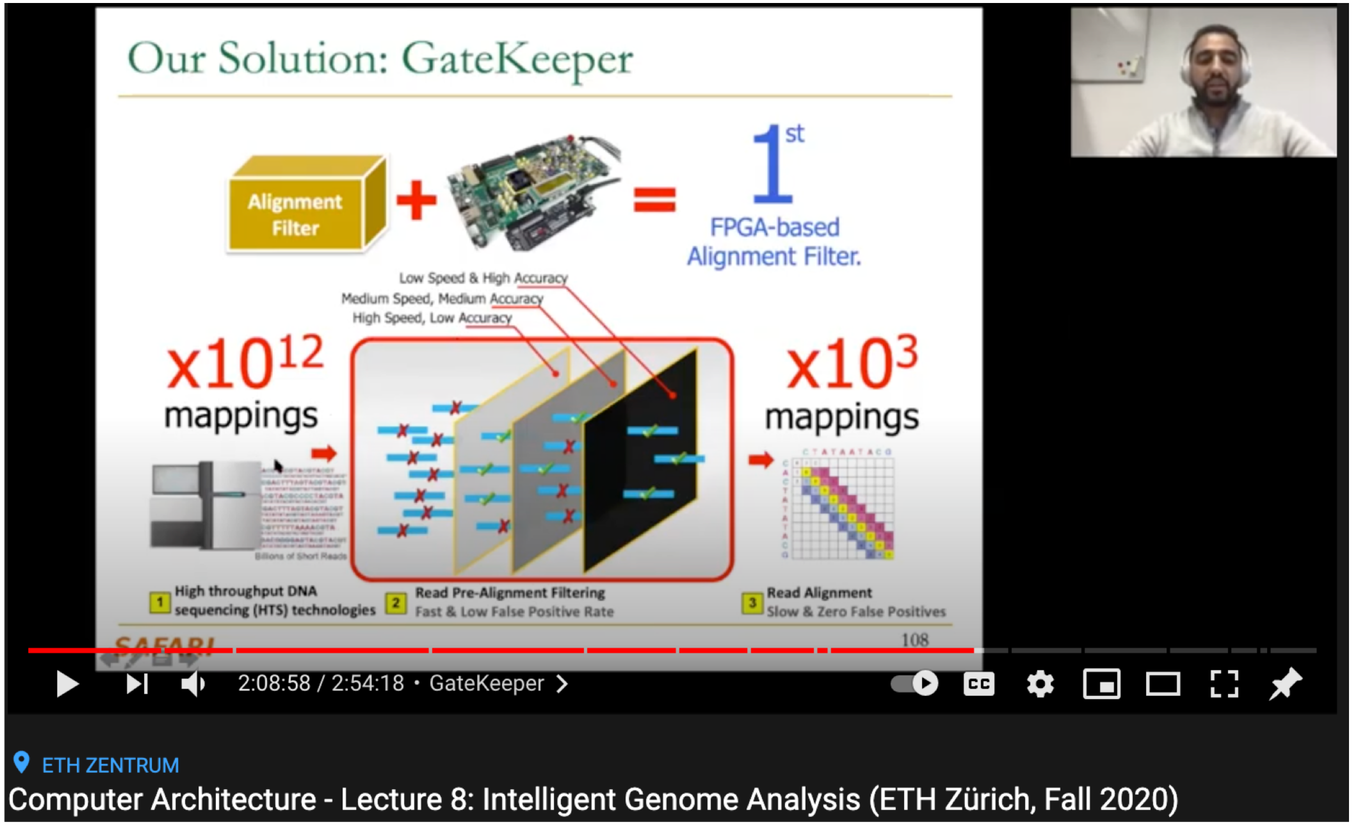


Insight: Shifting a String Helps Similarity Search

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

566 views • Premiered Feb 6, 2021

More on Intelligent Genome Analysis …



Our Solution: GateKeeper

Alignment Filter
+
=
1st
FPGA-based Alignment Filter.
Low Speed & High Accuracy
Medium Speed, Medium Accuracy
High Speed, Low Accuracy
x10^12 mappings
x10^3 mappings
Billions of Short Reads
1 High throughput DNA sequencing (HTS) technologies
2 Read Pre-Alignment Filtering Fast & Low False Positive Rate
3 Read Alignment Slow & Zero False Positives
SAFARI
108
2:08:58 / 2:54:18 • GateKeeper
CC

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.

Openings @ SAFARI

- We are **hiring** enthusiastic and motivated students and researchers at all levels.

- Join us now: safari.ethz.ch/apply

Thank you. Questions?

Seminar in Computer Architecture

Meeting 4: GateKeeper

Dr. Mohammed Alser

ALSERM@ethz.ch

ETH Zürich

Spring 2021

18 March 2021