Accelerating Genome Analysis

A Primer on an Ongoing Journey

Onur Mutlu

omutlu@gmail.com

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

January 24, 2018

AACBB Keynote, Vienna

Systems@ETH zürich

SAFARI

ETH zürich

Overview

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

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

- Many bottlenecks exist in accessing and manipulating huge amounts of genomic data during analysis

- We will cover various recent ideas to accelerate read mapping
  - My personal journey since September 2006

Agenda

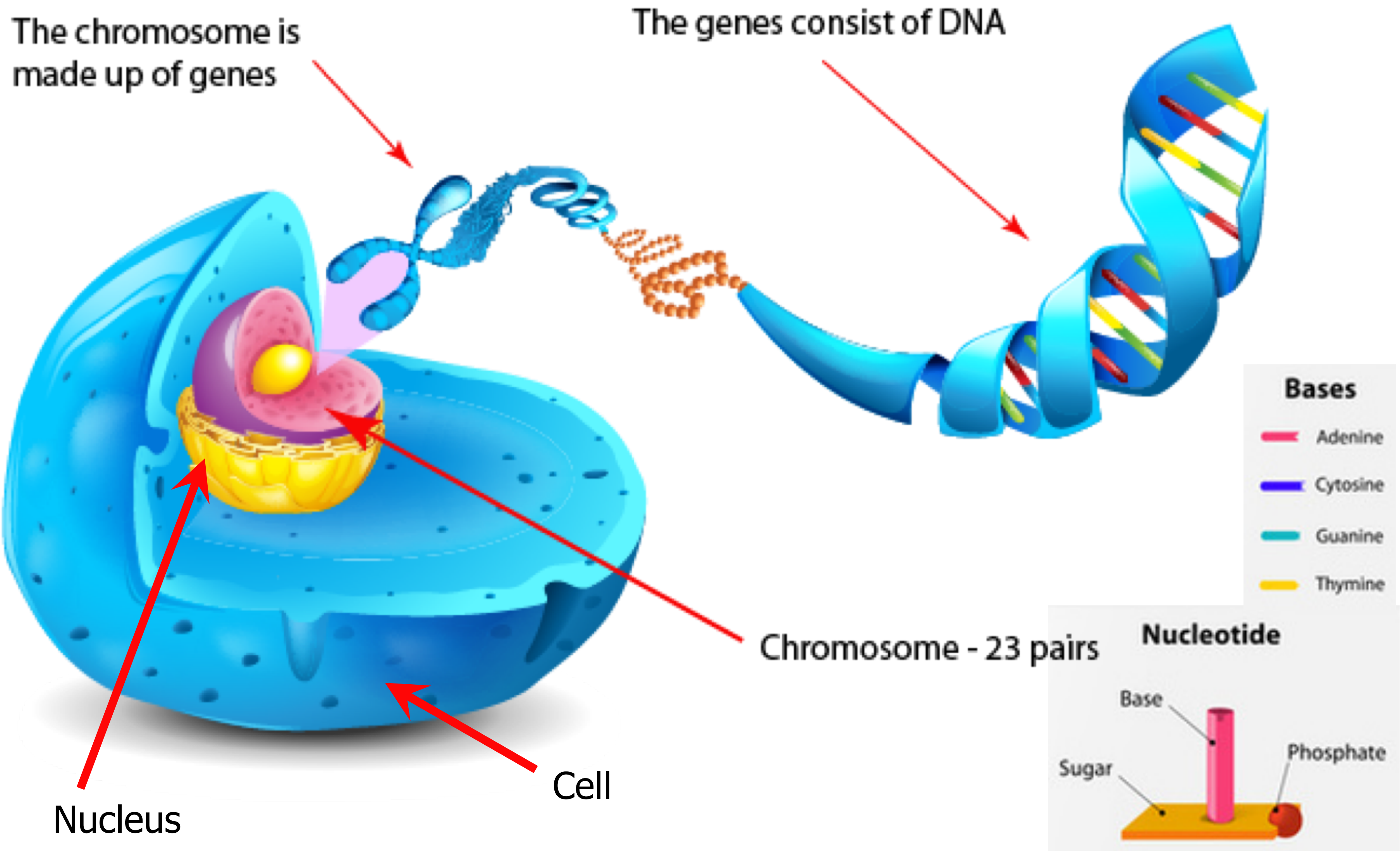
- The Problem: DNA Read Mapping
  - State-of-the-art Read Mapper Design

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

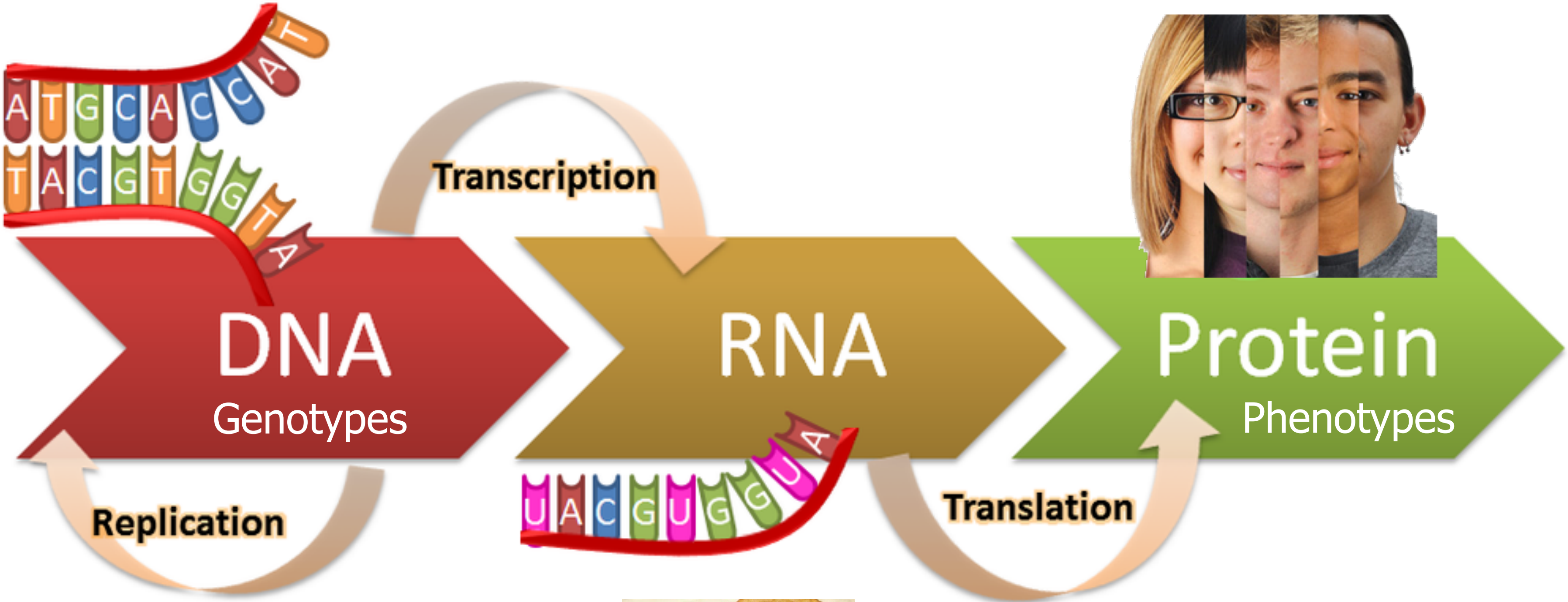
- Future Opportunities: New Sequencing Technologies

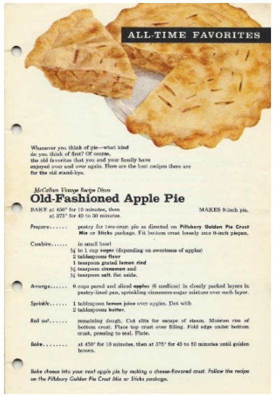
What Is a Genome Made Of?



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

The Central Dogma of Molecular Biology





ALL-TIME FAVORITES

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

McCallum Vintage Recipe Diva

Old-Fashioned Apple Pie

BAKE at 450° for 10 minutes, then at 375° for 40 to 50 minutes.

MAKES 9-inch pie.

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

Combine....... in small bowl
¾ to 1 cup **sugar** (depending on sweetness of apples)
2 tablespoons **flour**
1 teaspoon grated **lemon rind**
½ teaspoon **cinnamon** and
¼ teaspoon **salt**. Set aside.

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

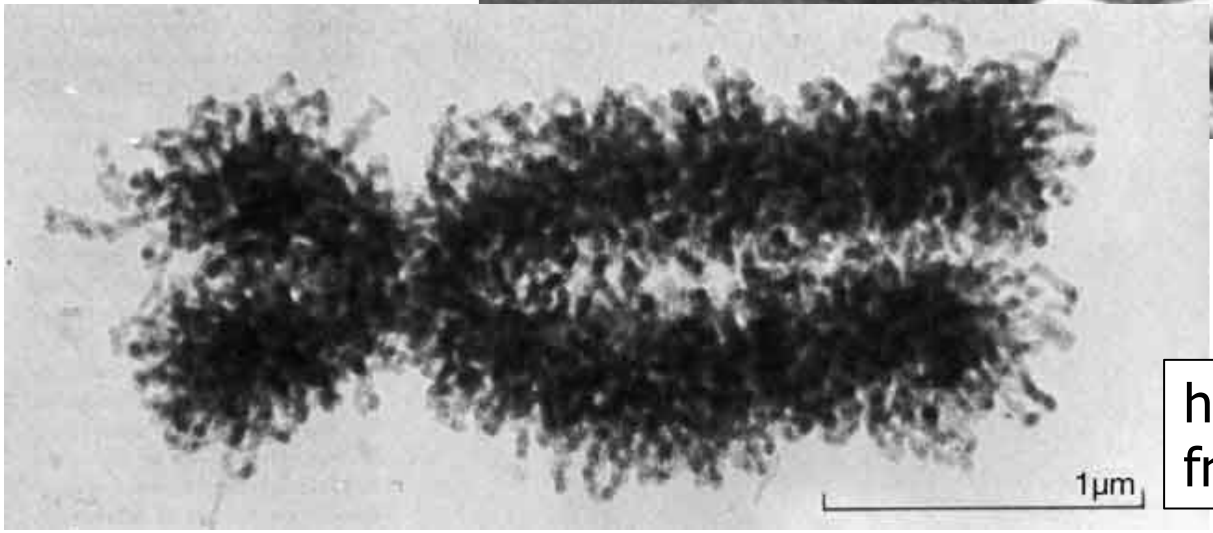
Sprinkle....... 1 tablespoon **lemon** juice over apples. Dot with
2 tablespoons **butter**.

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

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

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

DNA Under Electron Microscope



human chromosome #12 from HeLa’s cell

DNA Sequencing

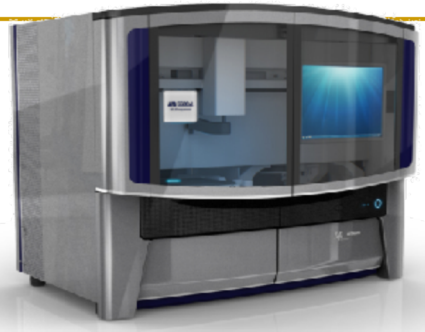
- Goal:
  - Find the complete sequence of A, C, G, T’s in DNA.

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

Untangling Yarn Balls & DNA Sequencing

Genome Sequencers

Roche/454



AB SOLiD

Illumina MiSeq

Complete Genomics

Oxford Nanopore MinION

Illumina NovaSeq 6000

Illumina HiSeq2000

Pacific Biosciences RS

Oxford Nanopore GridION

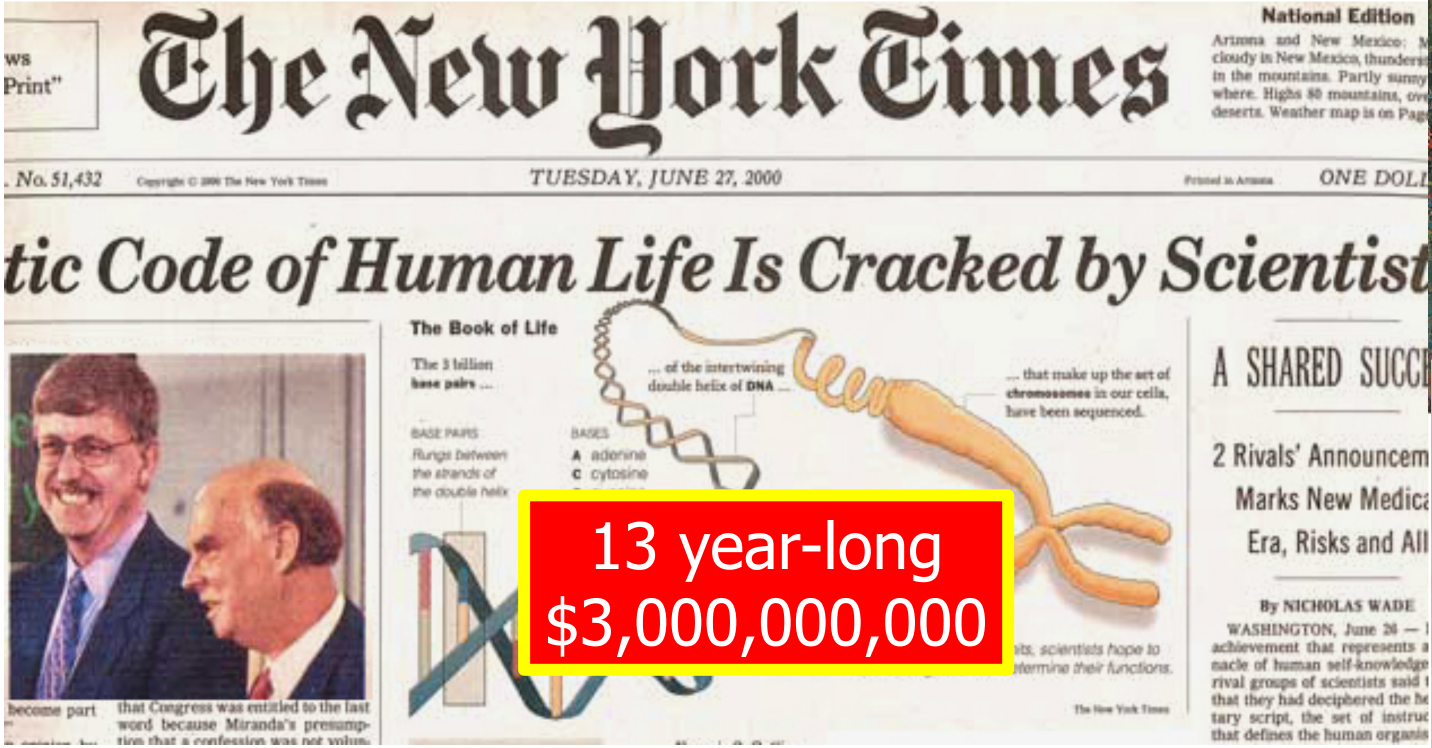
Ion Torrent PGM

Ion Torrent Proton

… and more! All produce data with different properties.

The Genomic Era

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



ws Print"

The New York Times

National Edition

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

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

tic Code of Human Life Is Cracked by Scientist

The Book of Life

The 3 billion **base pairs** ...

... of the intertwining double helix of **DNA** ...

... that make up the set of **chromosomes** in our cells, have been sequenced.

BASE PAIRS Rungs between the strands of the double helix

BASES **A** adenine **C** cytosine

...its, scientists hope to ...termine their functions.

The New York Times

become part

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

A SHARED SUCC

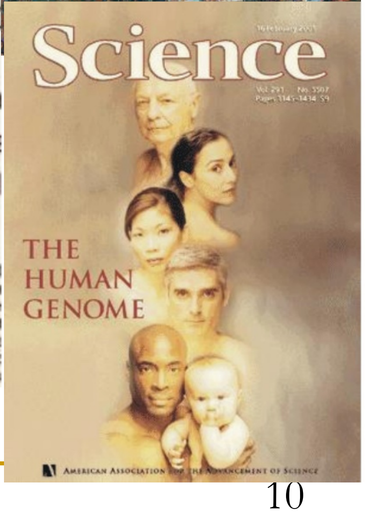
2 Rivals' Announcem Marks New Medica Era, Risks and All

By NICHOLAS WADE

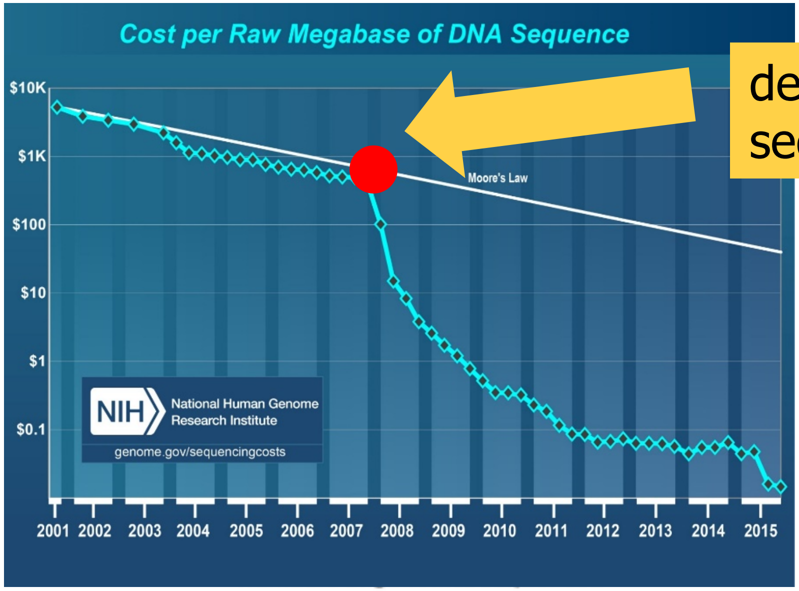
WASHINGTON, June 26 — achievement that represents a nacle of human self-knowledge rival groups of scientists said that they had deciphered the he tary script, the set of instruc that defines the human organis

13 year-long
$3,000,000,000

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

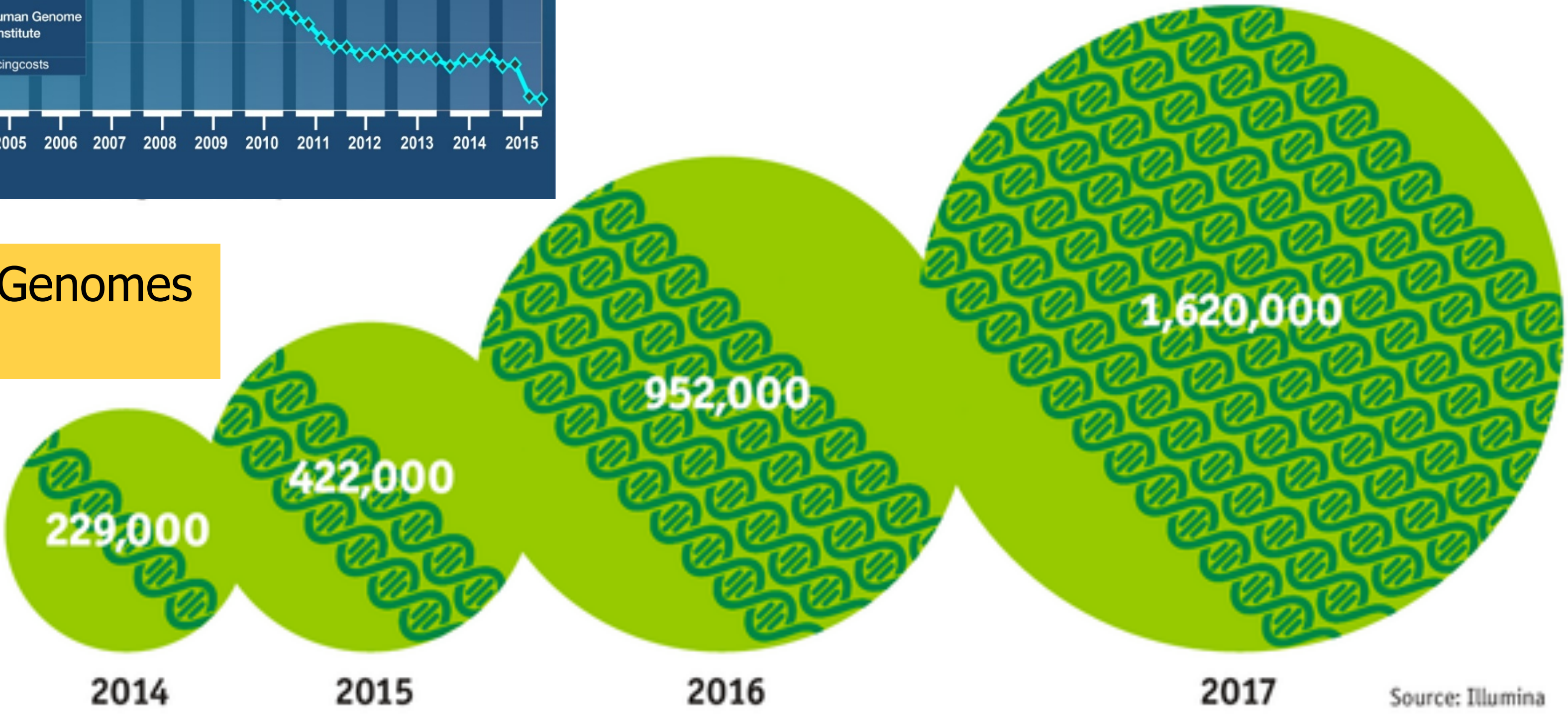


The Genomic Era (continued)



development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

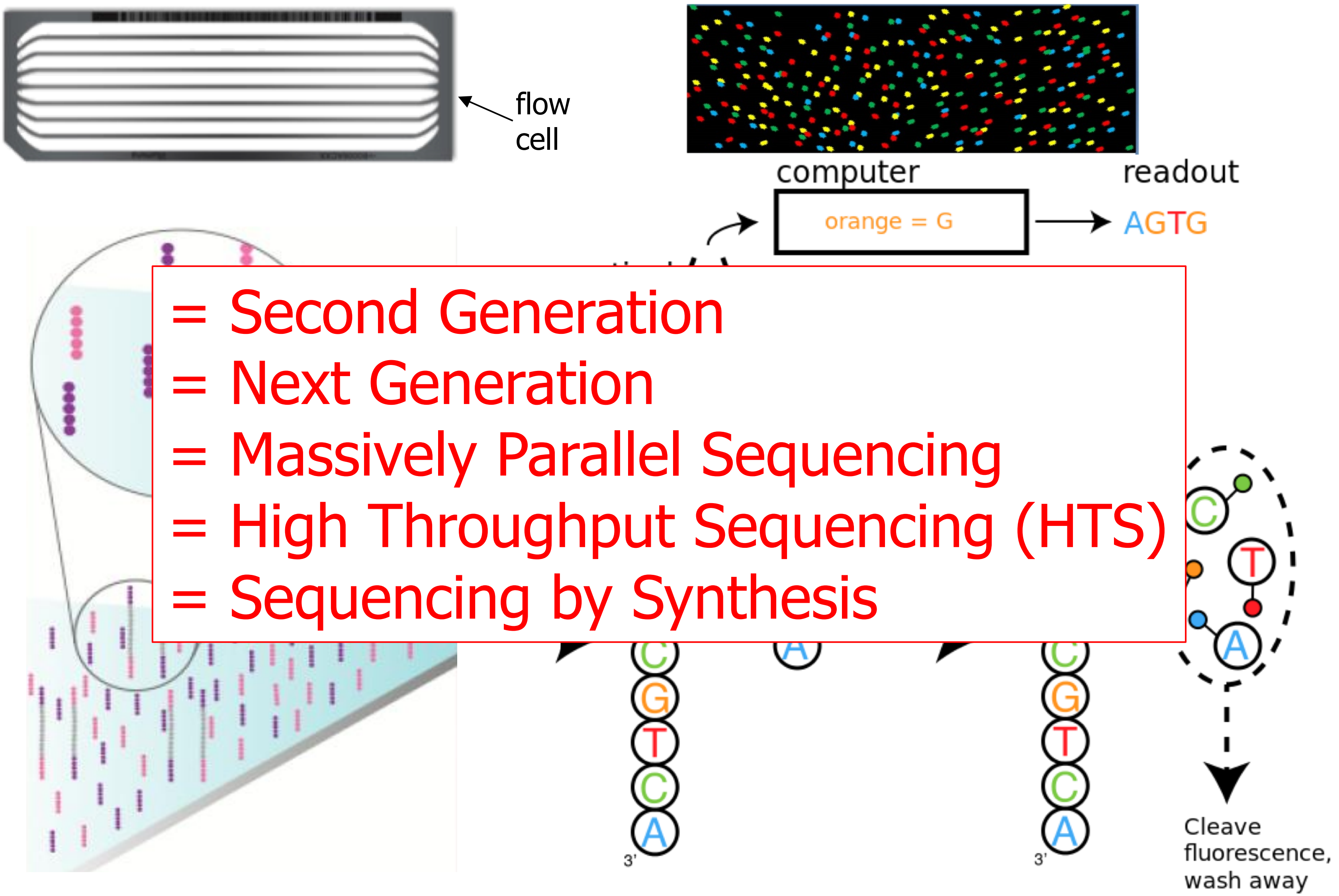


Source: Illumina

The Economist

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

High-Throughput Sequencing (HTS)



= Second Generation

= Next Generation

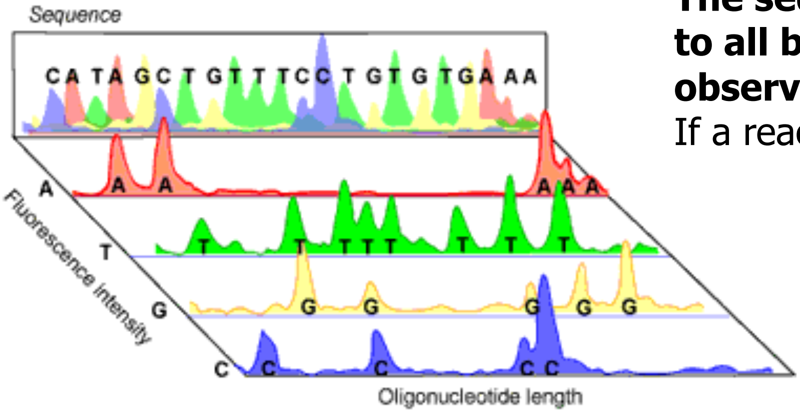
= Massively Parallel Sequencing

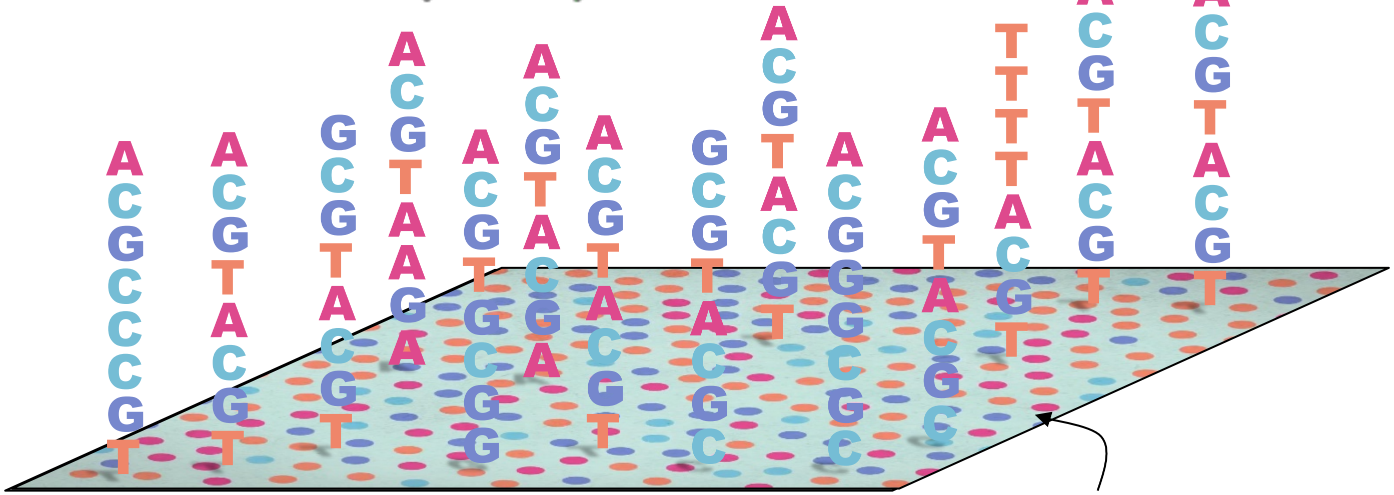
= High Throughput Sequencing (HTS)

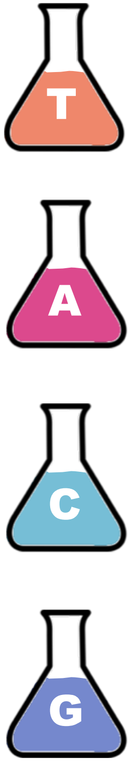
= Sequencing by Synthesis

High-Throughput Sequencing (HTS)

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







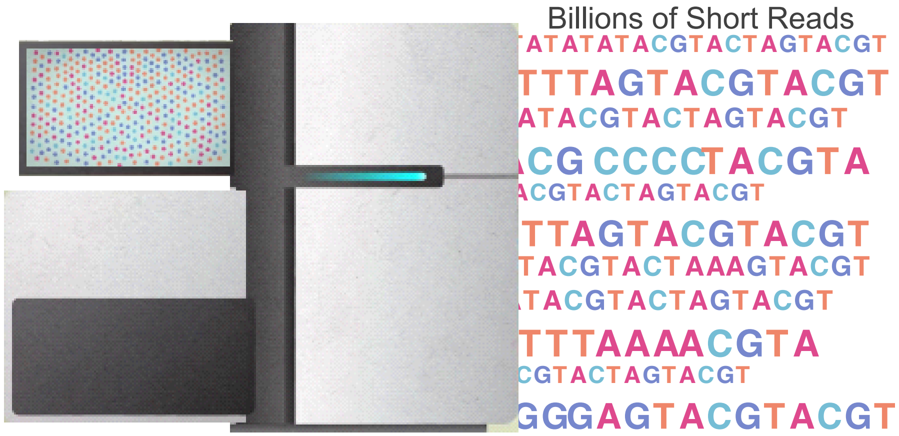
As a workaround, HTS technologies sequence random short DNA fragments (75-300 basepairs long) of copies of the original molecule.

High-Throughput Sequencing

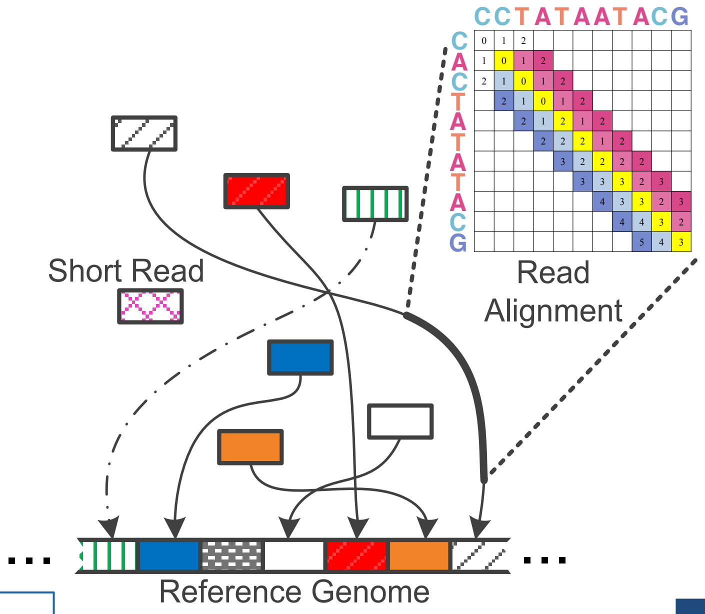
- Massively parallel sequencing technology
  - Illumina, Roche 454, Ion Torrent, SOLID…

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

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



1 Sequencing



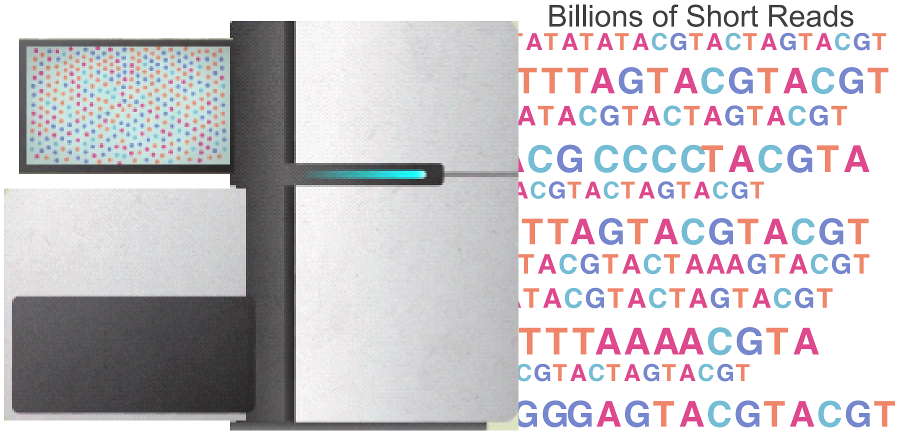
2 Read Mapping

Genome Analysis

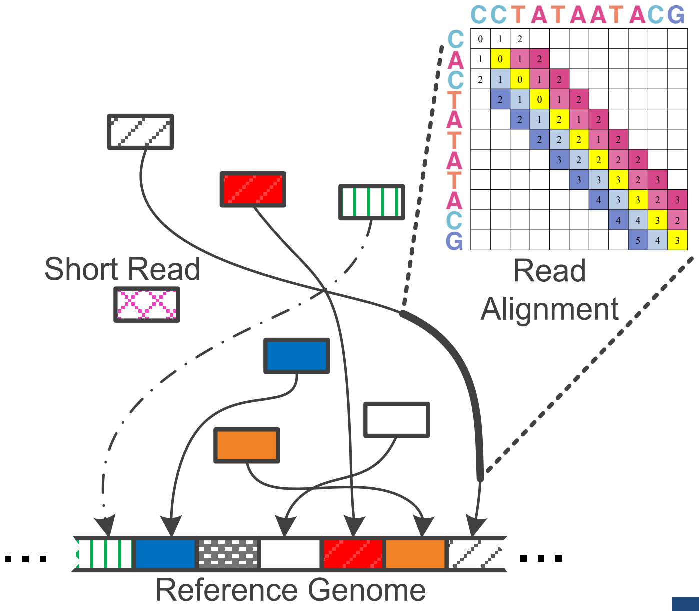
```
reference: TTTATCGCTTCCATGACGCAG
read1:        ATCGCATCC
read2:       TATCGCATC
read3:            CATCCATGA
read4:          CGCTTCCAT
read5:               CCATGACGC
read6:             TTCCATGAC
```

3 Variant Calling

4 Scientific Discovery

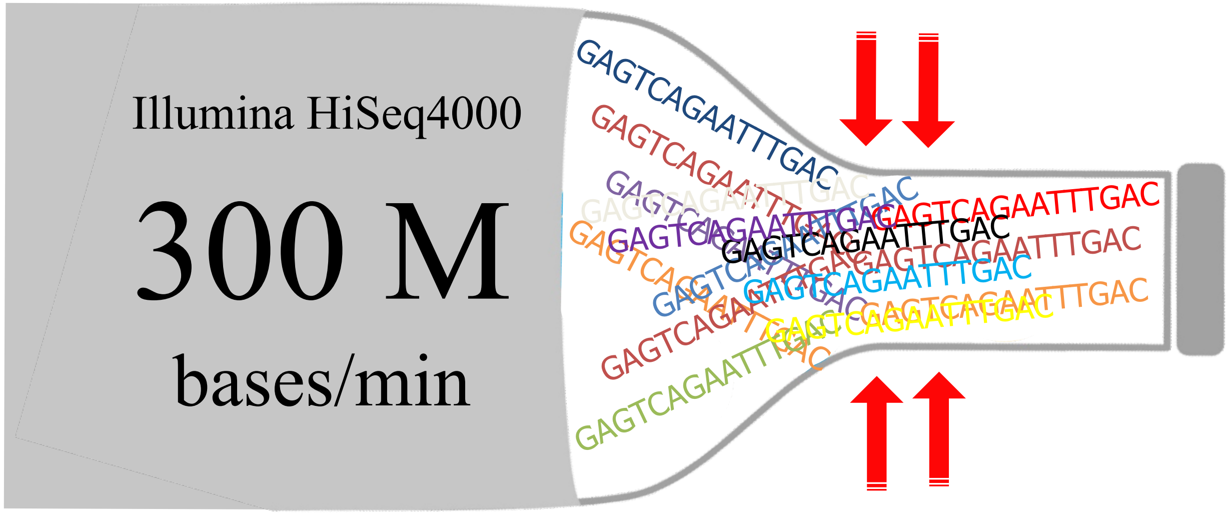


1 Sequencing



Read Mapping 2

Bottlenecked in Mapping!!



Illumina HiSeq4000

300 M

bases/min

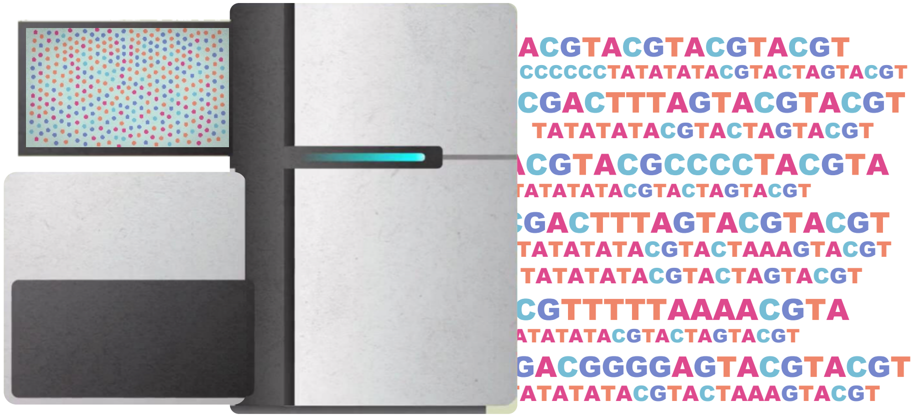
on average

2 M

bases/min

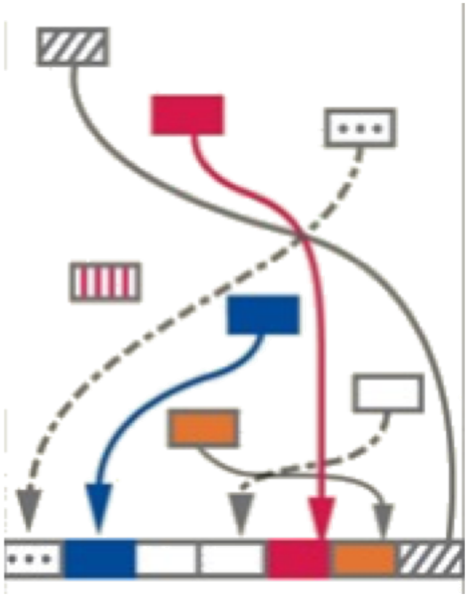
(0.6%)

The Read Mapping Bottleneck



Illumina HiSeq4000

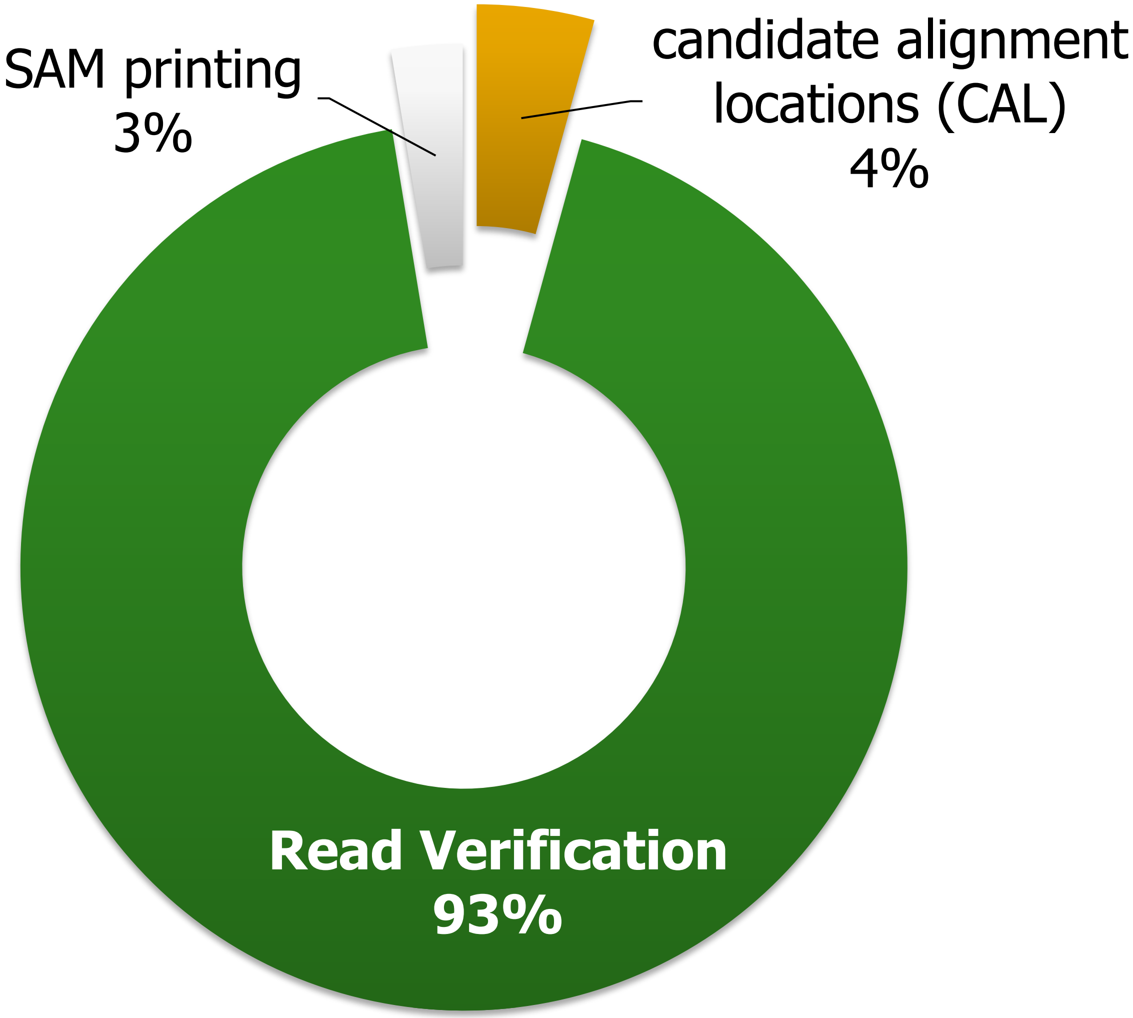
300 Million bases/minute



2 Million bases/minute

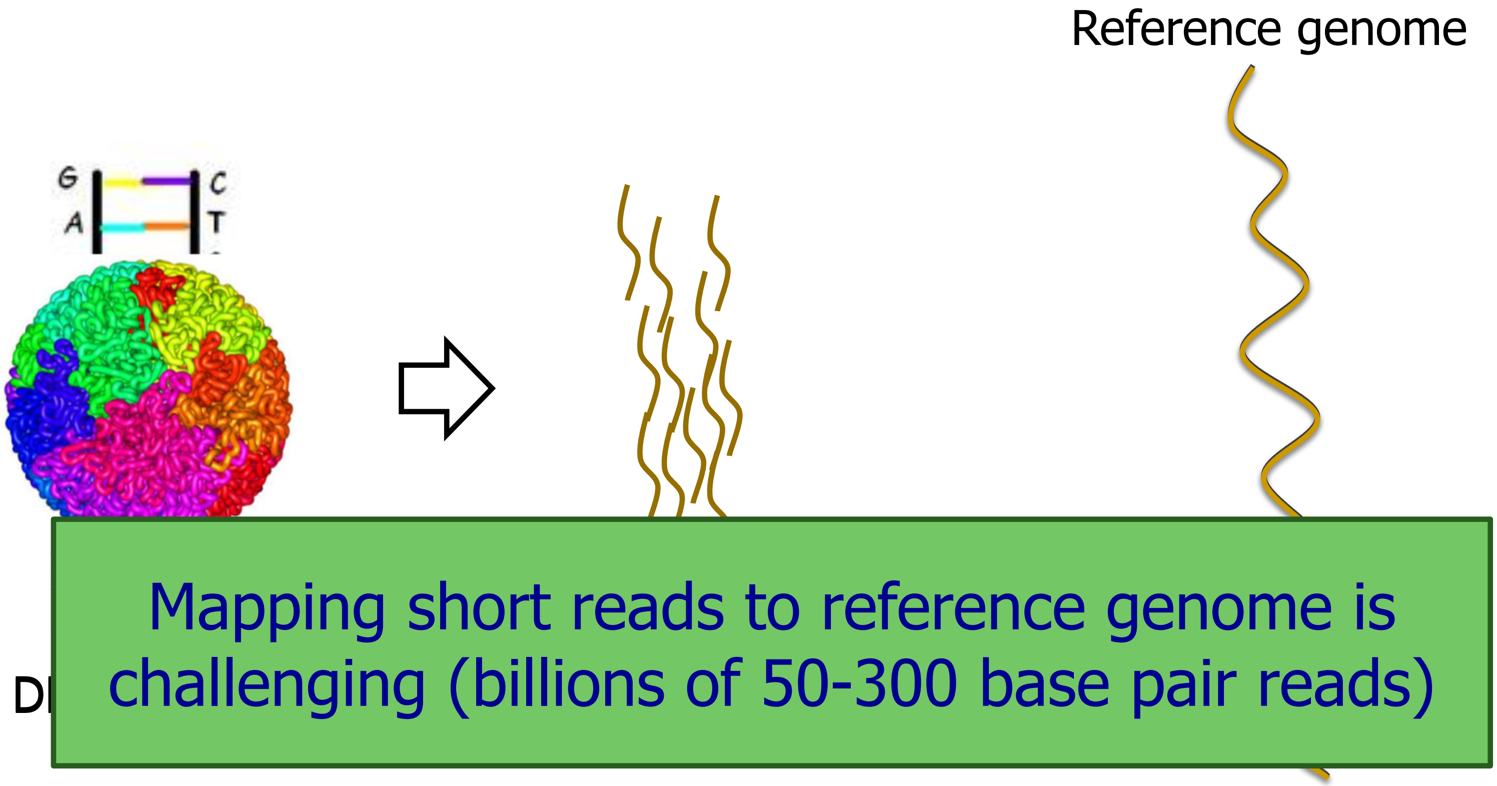
150X slower

Read Mapping Execution Time Breakdown



Read Mapping

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



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

Challenges in Read Mapping

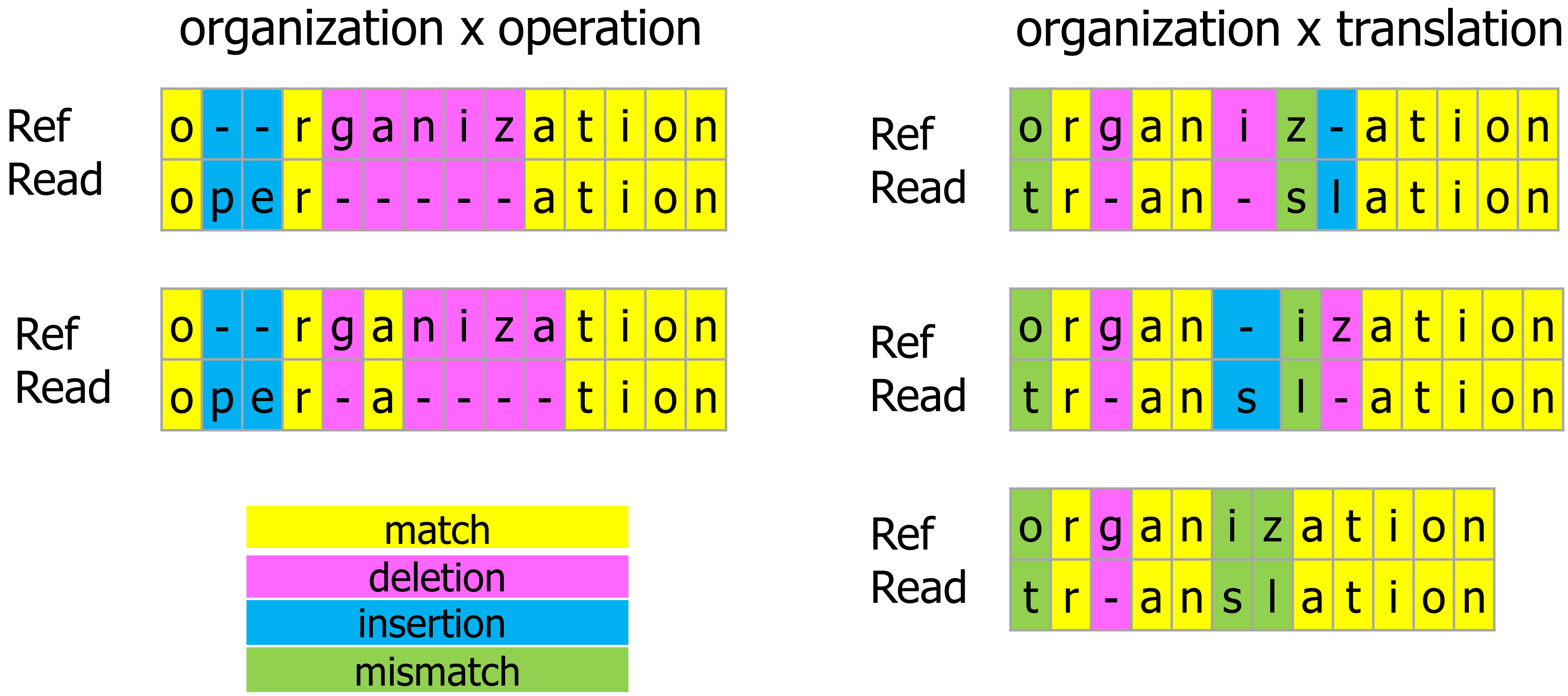
- Need to find many mappings of each read
  - A short read may map to many locations, especially with High-Throughput DNA Sequencing technologies
  - How can we find all mappings efficiently?

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

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

Read Alignment/Verification

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



organization x operation

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

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

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

match
deletion
insertion
mismatch

Agenda

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

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

Read Mapping Algorithms: Two Styles

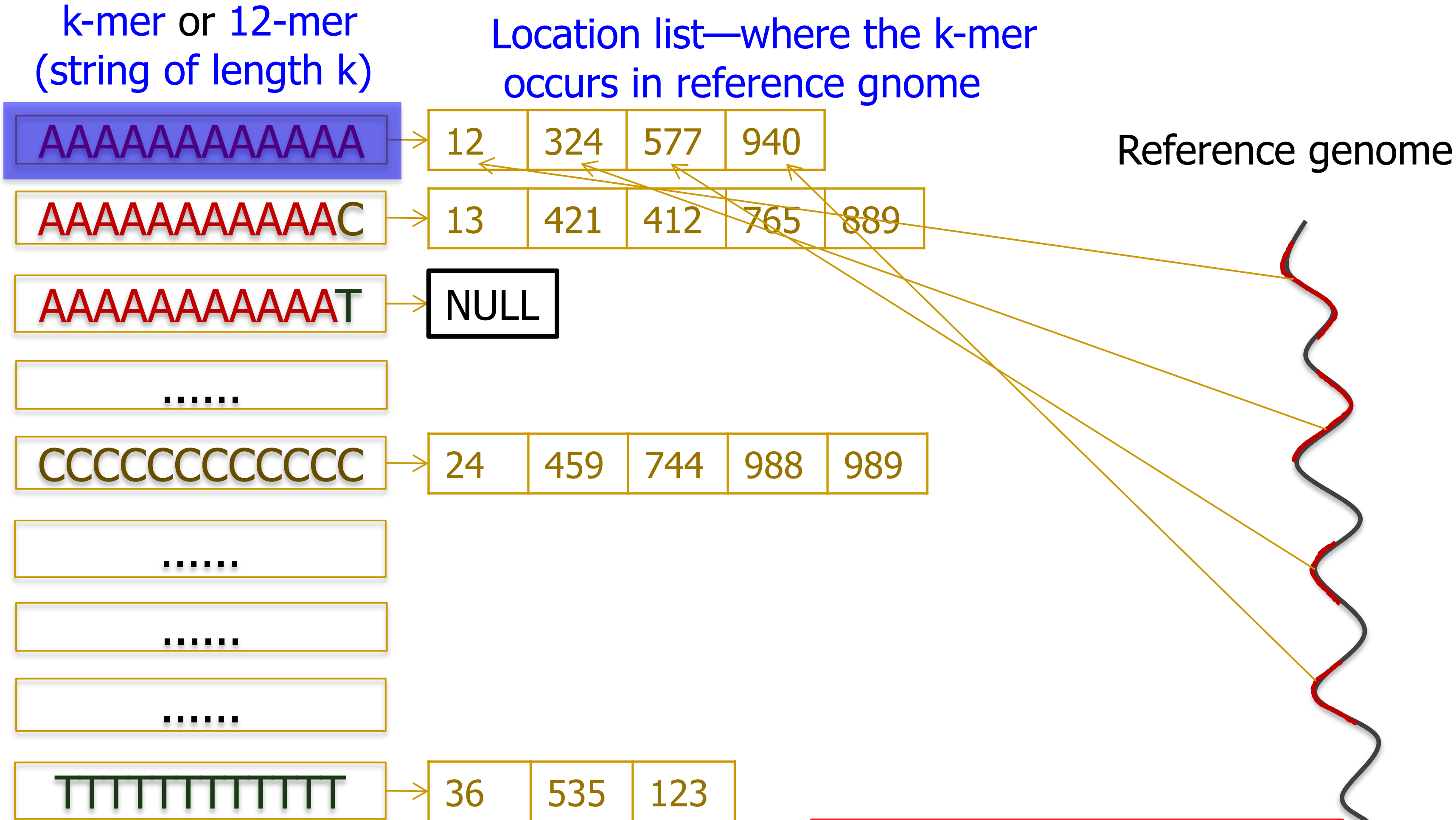
- Hash based seed-and-extend (hash table, suffix array, suffix tree)
  - Index the “k-mers” in the genome into a hash table (pre-processing)
  - When searching a read, find the location of a k-mer in the read; then extend through alignment
  - More sensitive, but slow
  - Requires large memory; this can be reduced with cost to run time

- Burrows-Wheeler Transform & Ferragina-Manzini Index based aligners
  - BWT is a compression method used to compress the genome index
  - Perfect matches can be found very quickly, memory lookup costs increase for imperfect matches
  - Reduced sensitivity

Hash Table Based Read Mappers

- Key Idea
  - Preprocess the reference into a *Hash Table*
  - Use *Hash Table* to map reads

Hash Table-Based Mappers [Alkan+ Nature Gen'09]



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

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

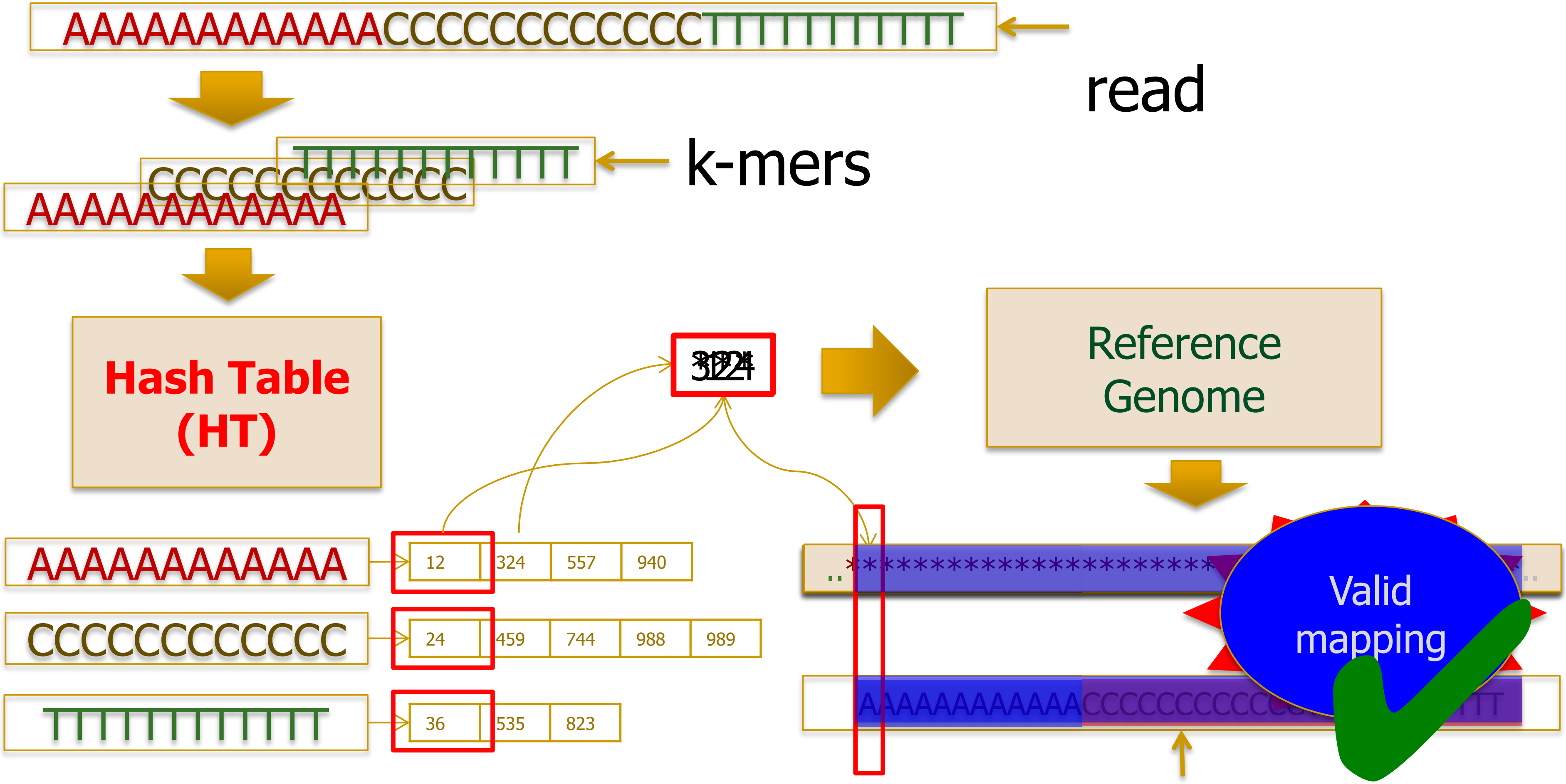
AAAAAAAAAAAA	12	324	577	940	
AAAAAAAAAAAC	13	421	412	765	889
AAAAAAAAAAAT	NULL				
......					
CCCCCCCCCCCC	24	459	744	988	989
......					
......					
......					
TTTTTTTTTTTT	36	535	123		

Once for a reference

Hash Table Based Read Mappers

- Key Idea
  - Preprocess the reference into a *Hash Table*
  - Use *Hash Table* to map reads

Hash Table-Based Mappers [Alkan+ Nature Gen’09]



AAAAAAAAAAAACCCCCCCCCCCCTTTTTTTTTTTT

read

k-mers

Hash Table (HT)

Reference Genome

AAAAAAAAAAAA	12	324	557	940	
CCCCCCCCCCCC	24	459	744	988	989
TTTTTTTTTTTT	36	535	823		

read

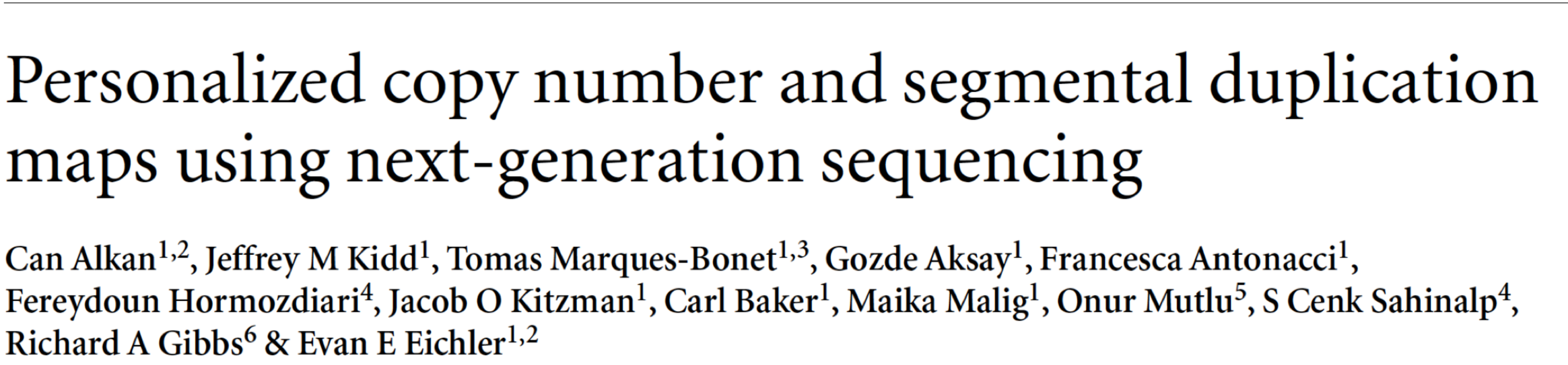
Verification/Local Alignment

Advantages of Hash Table Based Mappers

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

nature genetics

http://mrfast.sourceforge.net/



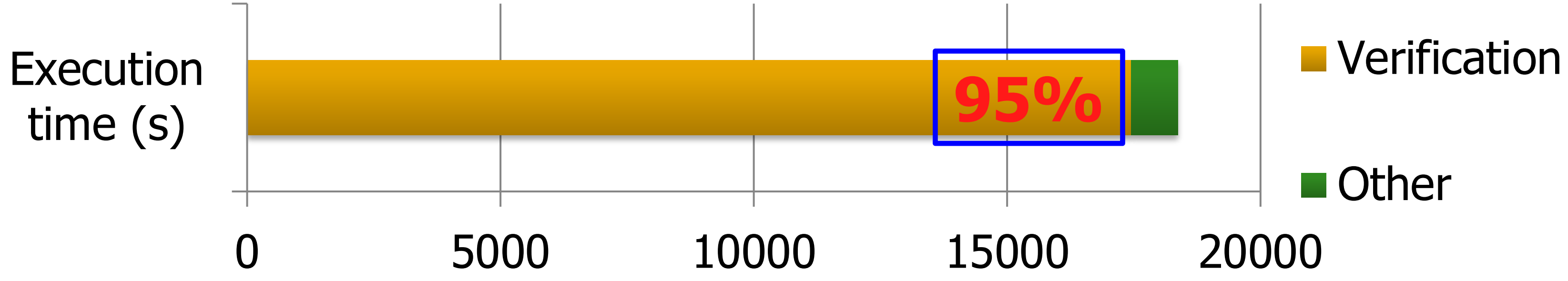
Personalized copy number and segmental duplication maps using next-generation sequencing

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

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

Problem and Goal

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



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

Agenda

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

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

Reducing the Cost of Verification

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

- We observe that we can get rid of unnecessary verification calculations by
  - *Detecting and rejecting early* invalid mappings (filtering)
  - *Reducing* the *number* of potential mappings

Key Observations [Xin+, BMC Genomics 2013]

- Observation 1
  - Adjacent k-mers in the read should also be adjacent in the reference genome
  - Read mapper can quickly reject mappings that do **not** satisfy this property

- Observation 2
  - Some k-mers are cheaper to verify than others because they have shorter location lists (they occur less frequently in the reference genome)
    - Mapper needs to examine only *e+1* k-mers' locations to tolerate *e* errors
  - Read mapper can choose the cheapest *e+1* k-mers and verify their locations

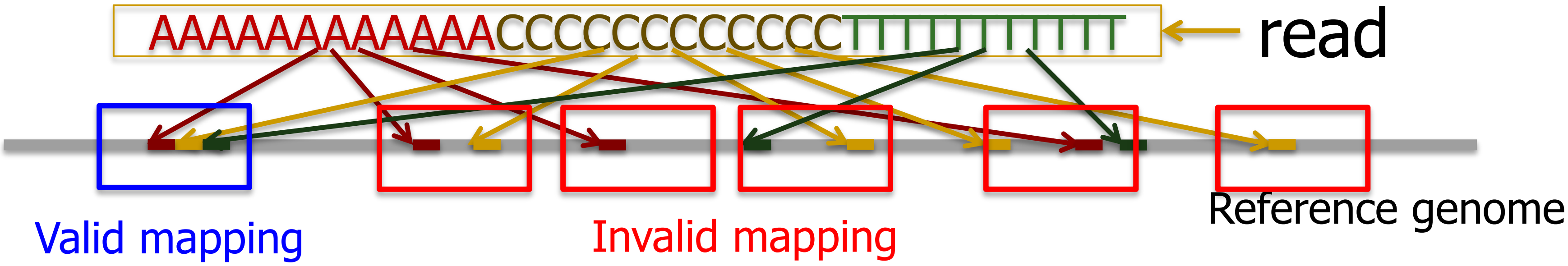
FastHASH Mechanisms [Xin+, BMC Genomics 2013]

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

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

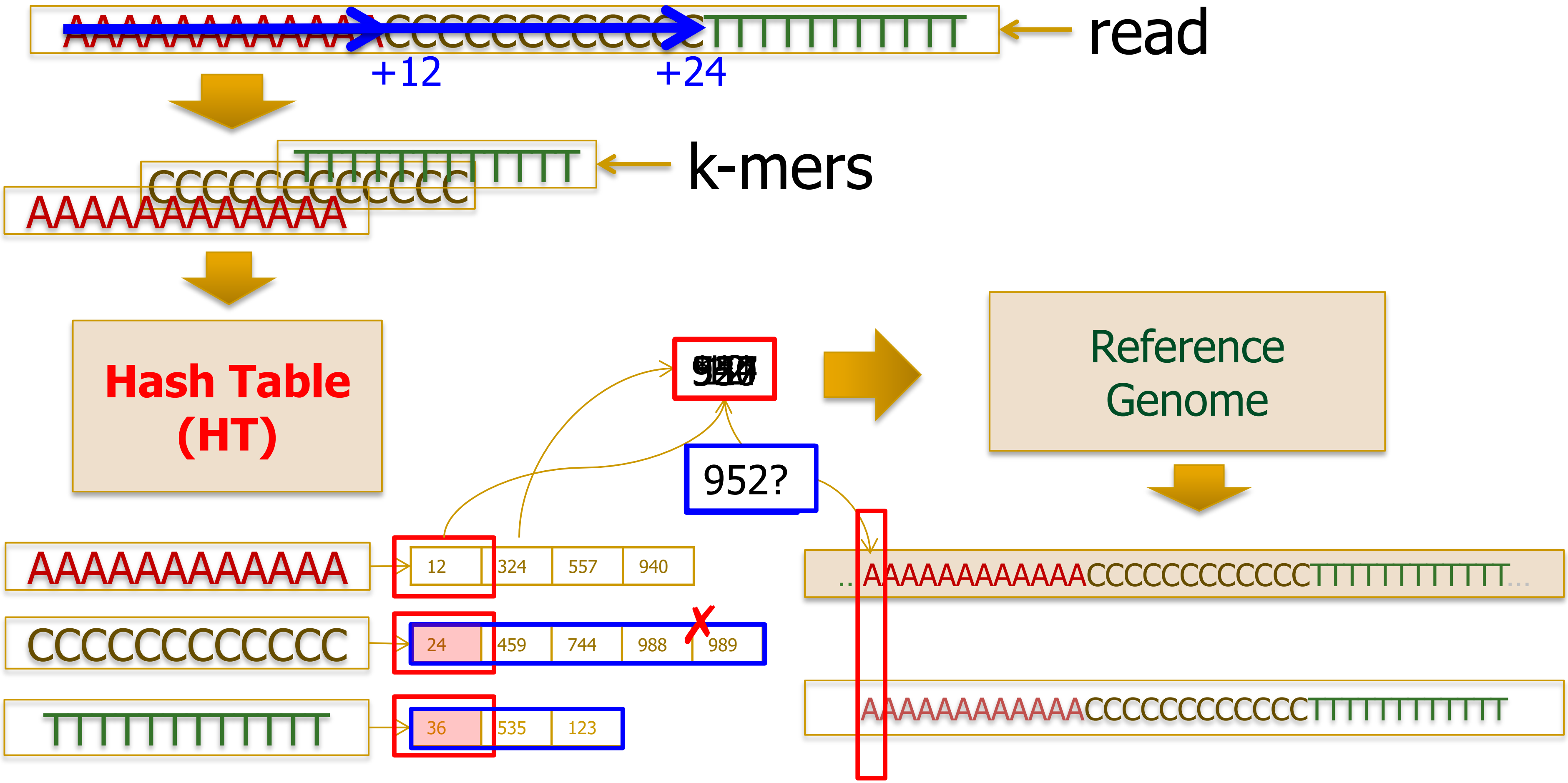
Adjacency Filtering (AF)

- **Goal:** detect and filter out invalid mappings at early stage
- **Key Insight:** For a valid mapping, adjacent k-mers in the read are also adjacent in the reference genome



- **Key Idea:** search for adjacent locations in the k-mers' location lists
  - If more than *e* k-mers fail → there must be more than e errors → invalid mapping

Adjacency Filtering (AF)



FastHASH Mechanisms [Xin+, BMC Genomics 2013]

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

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

Cheap K-mer Selection (CKS)

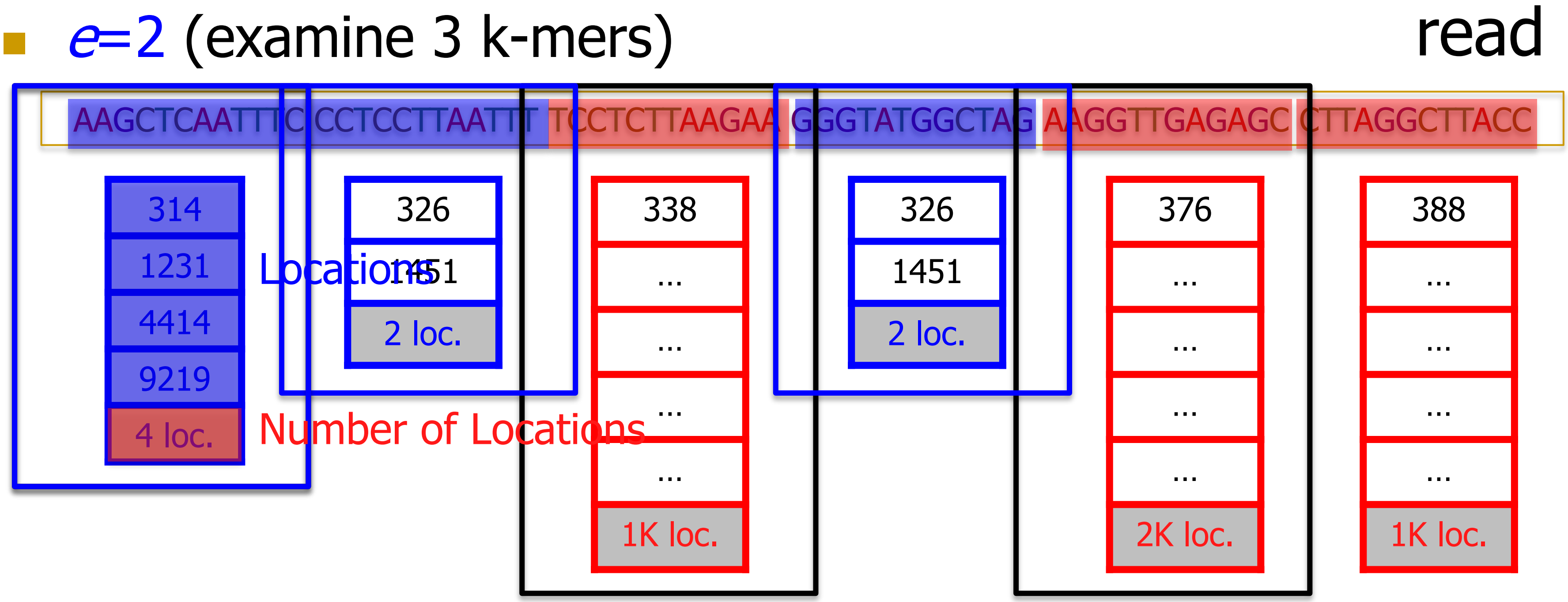
- **Goal:** Reduce the number of potential mappings

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

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

Cheap K-mer Selection

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



Expensive 3 k-mers

Cheapest 3 k-mers

Previous work needs to verify:

3004 locations

FastHASH verifies only:

8 locations

Methodology

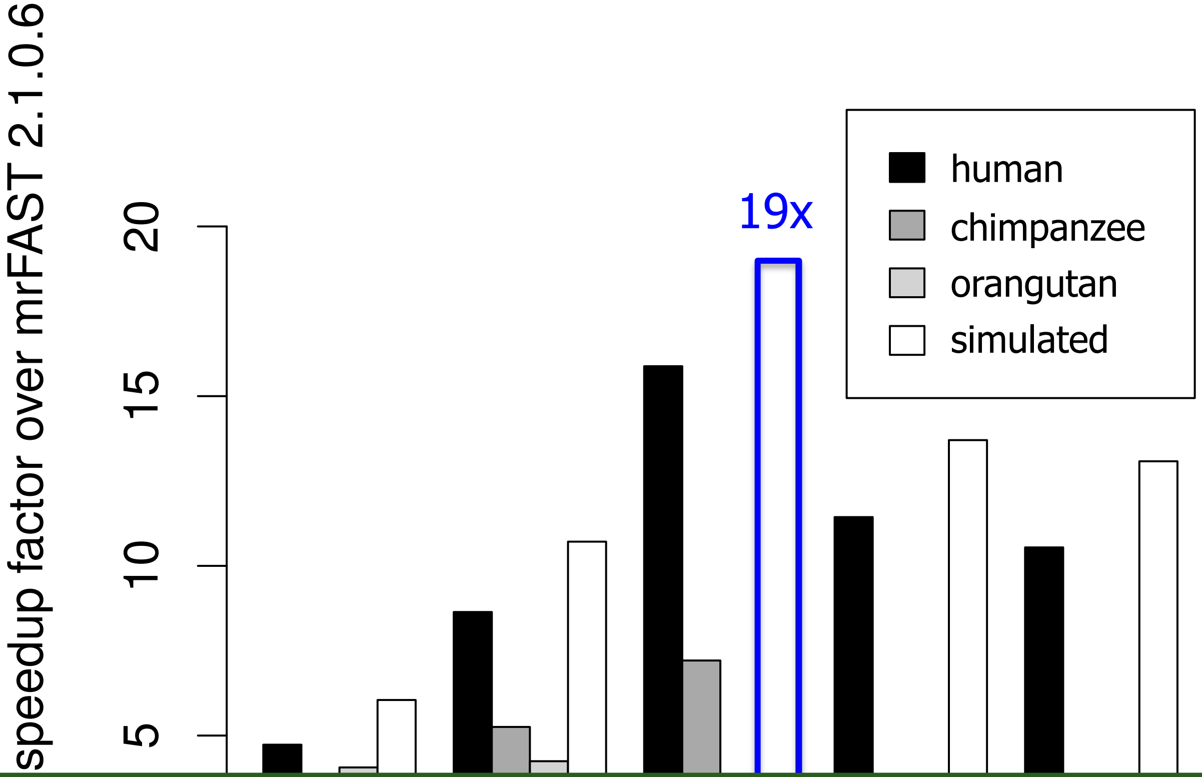
- Implemented FastHASH on top of state-of-the-art mapper: mrFAST
  - New version mrFAST-2.5.0.0 over mrFAST-2.1.0.6

- Tested with real read sets generated from Illumina platform
  - 1M reads of a human (160 base pairs)
  - 500K reads of a chimpanzee (101 base pairs)
  - 500K reads of a orangutan (70 base pairs)

- Tested with simulated reads generated from reference genome
  - 1M simulated reads of human (180 base pairs)

- Evaluation system
  - Intel Core i7 Sandy Bridge machine
  - 16 GB of main memory

FastHASH Speedup

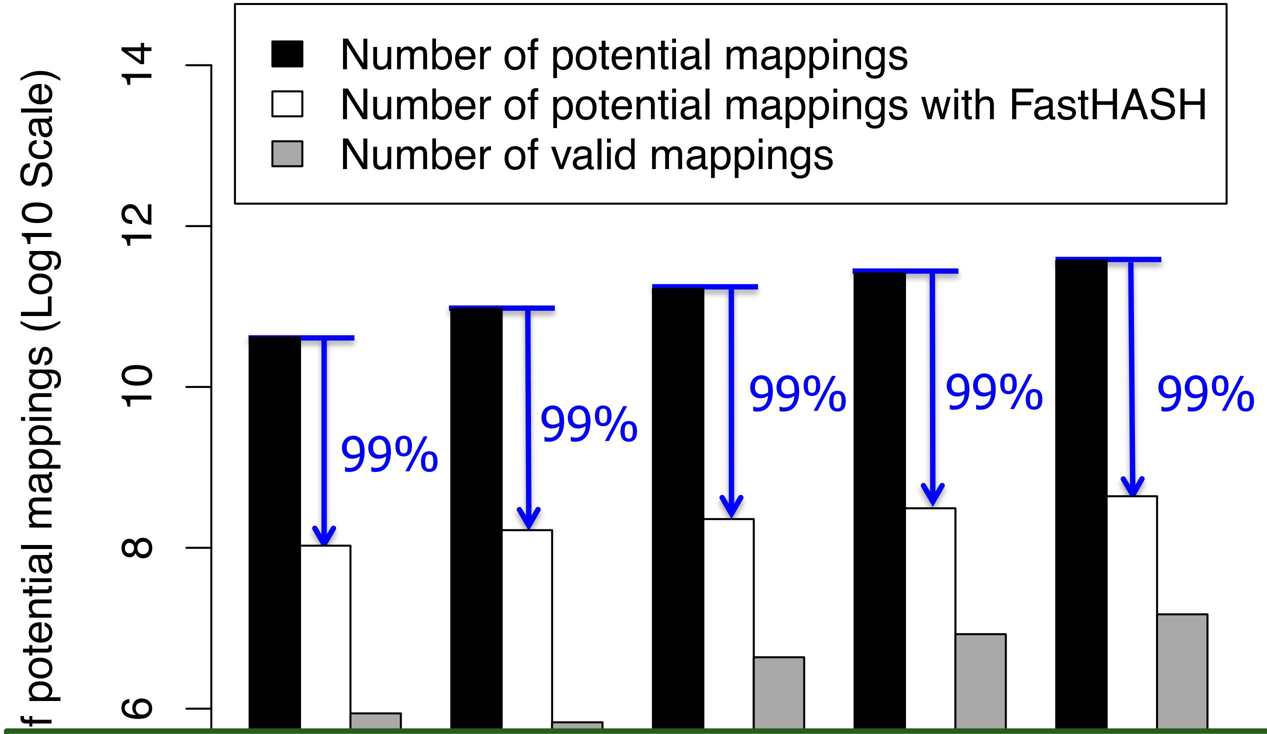


With FastHASH, new mrFAST obtains up to 19x speedup over previous version, without losing valid mappings

e: edit distance

Analysis

- Reduction of potential mappings with FastHASH



FastHASH filters out over 99% of the potential mappings without sacrificing any valid mappings

Reduction of potential mappings with FastHASH

FastHASH Conclusion

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

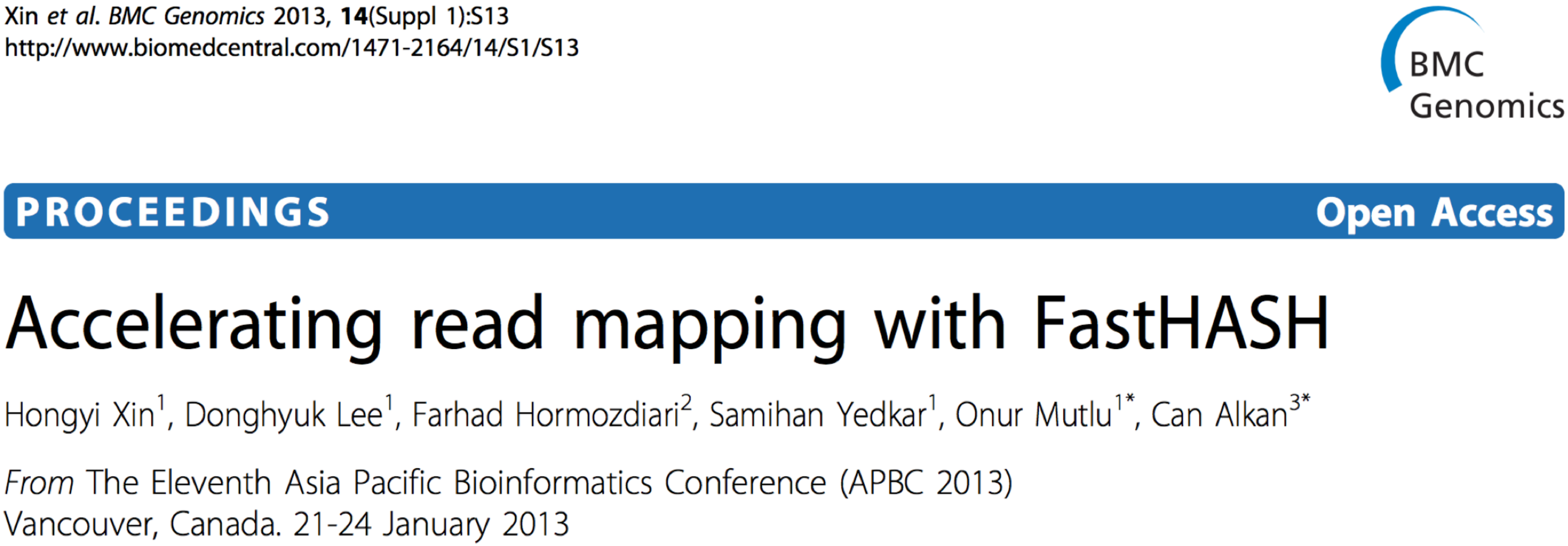
- Observation: Most of the verification calculations are unnecessary → filter them out

- Key Idea: To reduce the cost of unnecessary verification
  - Reject invalid mappings early (Adjacency Filtering)
  - Reduce the number of possible mappings to examine (Cheap K-mer Selection)

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

More on FastHASH

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



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

BMC Genomics

PROCEEDINGS — Open Access

Accelerating read mapping with FastHASH

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

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

Agenda

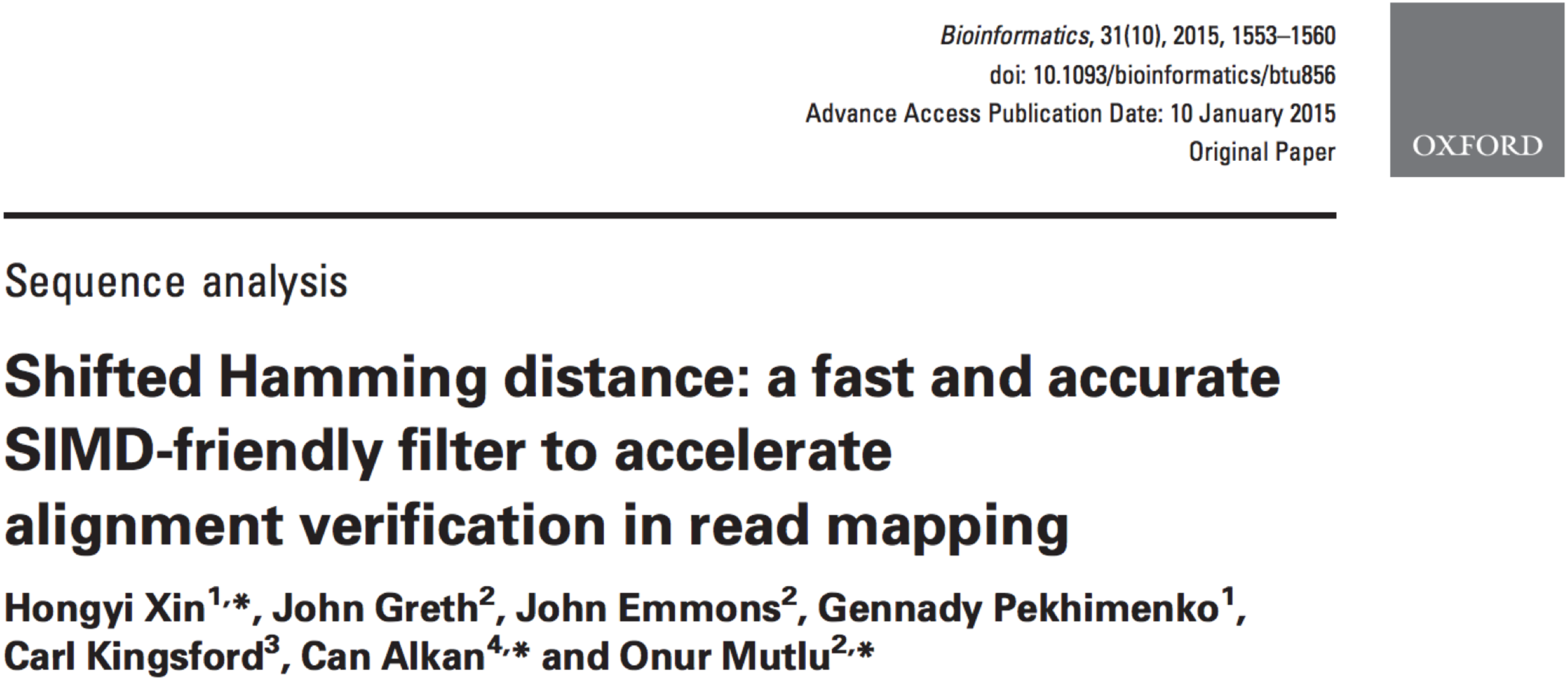
- The Problem: DNA Read Mapping
  - State-of-the-art Read Mapper Design

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

An Example: Shifted Hamming Distance



Bioinformatics, 31(10), 2015, 1553–1560
doi: 10.1093/bioinformatics/btu856
Advance Access Publication Date: 10 January 2015
Original Paper

OXFORD

Sequence analysis

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

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

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

Shifted Hamming Distance

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

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

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

New Bottleneck: Filtering (Pre-Alignment)

Sequencing generates many reads, each of which potentially mapping to many locations

→

Filtering (Pre-alignment) eliminates the need to verify/align read to invalid mapping locations

→

Alignment/verification (costly edit distance computation) is performed **only** on reads that pass the filter)

- New bottleneck in read mapping becomes the "filtering (pre-alignment)" step

Agenda

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

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

Location Filtering

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

- M t
  f
  1
  1
  2
  out mismatches quickly

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

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

Ideal Filtering Algorithm

Minimal False Accept Rate

Maximal True Reject Rate

Filter out all incorrect mappings

Zero False Reject Rate

Faster Than Mapper

Do not filter out any correct mappings

Alignment vs. Pre-alignment (Filtering)

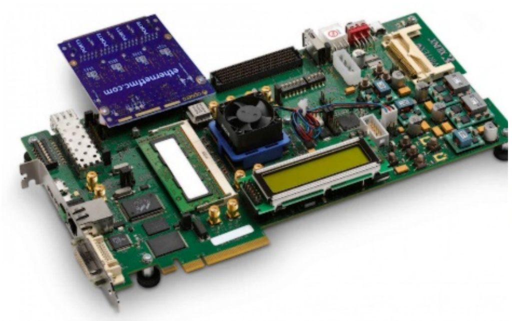
Needleman-Wunsch

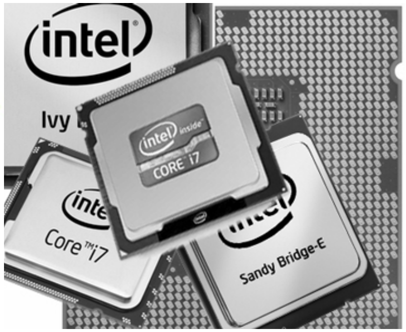
		C	T	A	T	A	A	T	A	C	G
	0	1	2								
A	1	0	1	2							

GateKeeper

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

- Independent vectors can be processed in parallel using hardware technologies





```
               |dp[i][j-1]  // Inser.
dp[i][j]=1+max|dp[i-1][j]  // Del.
               |dp[i-1][j-1]// Subs.
```

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

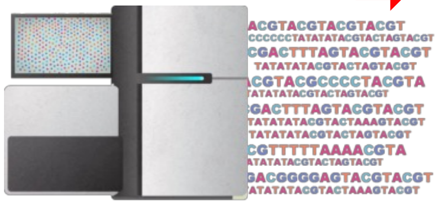
Each cell depends on three pre-computed cells!

No data dependencies!

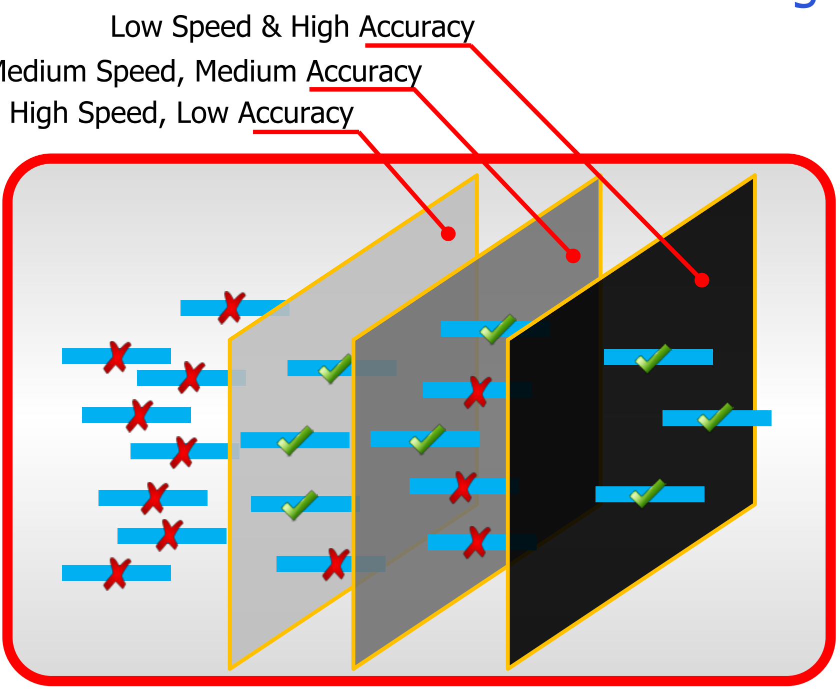
Our Solution: GateKeeper

Alignment Filter + = 1^{st} FPGA-based Alignment Filter.

$x10^{12}$ mappings



Billions of Short Reads



$x10^{3}$ mappings

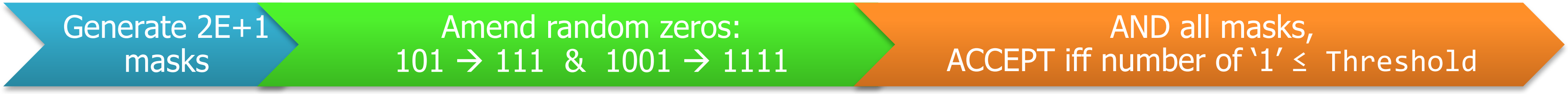
		C	T	A	T	A	A	T	A	C	G
C	0	1	2								
A	1	0	1	2							
C	2	1	0	1	2						
T		2	1	0	1	2					
A			2	1	2	1	2				
T				2	2	2	1	2			
A					3	2	2	2	2		
T						3	3	3	2	3	
A							4	3	3	2	3
C								4	4	3	2
G									5	4	3

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

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

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

GateKeeper Walkthrough



Generate 2E+1 masks

Amend random zeros: 101 → 111 & 1001 → 1111

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

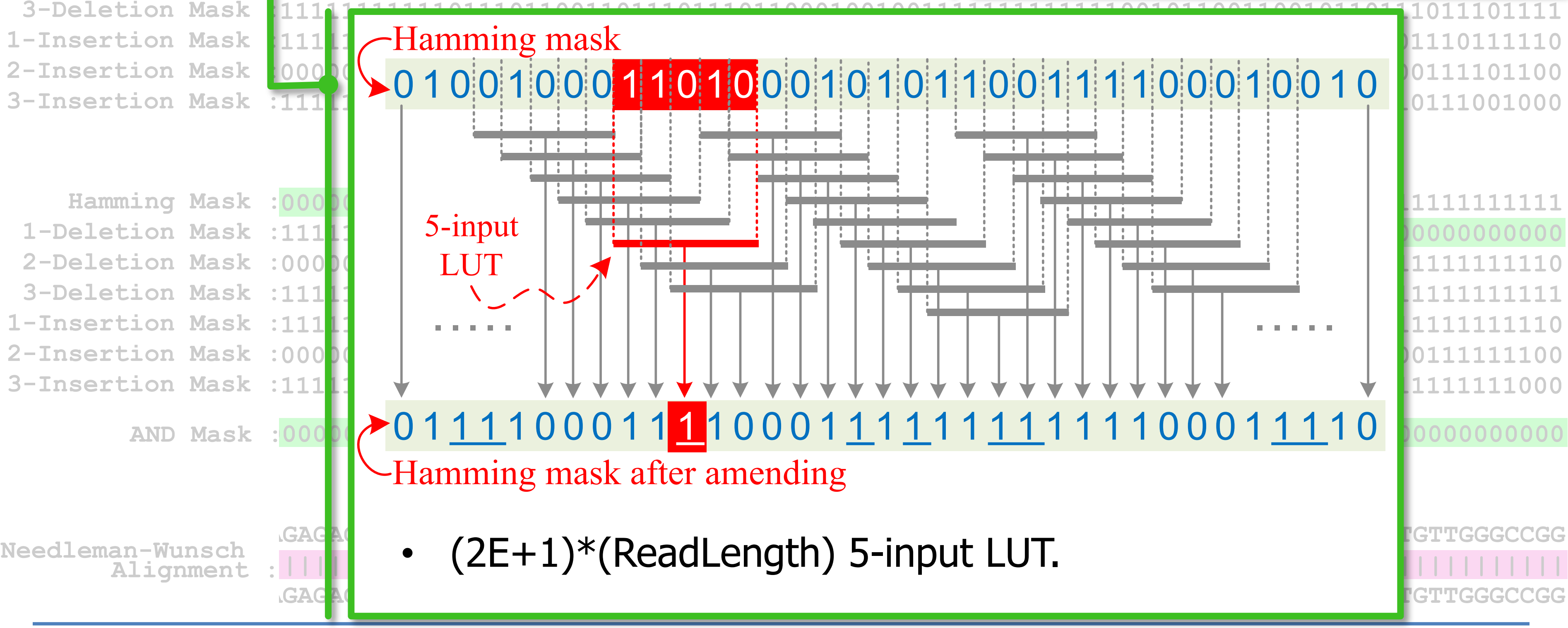
```
          Query :GAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGGAACATTGTTGGGCCGGA
      Reference :GAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGGCCGG

   Hamming Mask :0000000000100000000000011111101111000111011010110111111111110001000001111011010010101
 1-Deletion Mask :1111111111100111110111110000000000000000000000000000000000000000000011000000000000000
 2-Deletion Mask :0000000010110111001111111111110111100011101101011011111111111000100010011101101001010
 3-Deletion Mask :1111111111101110110011011101110110001001001111111111111100101100110010110111011101111
1-Insertion Mask :1111111111101111101111110111011000100100111111111111110010110011000101011101110111110
2-Insertion Mask :0000001001111100111111111001000110101010011010111111111111110111001111111000111101100
3-Insertion Mask :1111111011101100110001111111111010110111111100110010111011111111011101111010111001000

                                --- Masks after amendment ---

   Hamming Mask :0000000000100000000000011111111111000111111110111111111111110001000001111111111111111
 1-Deletion Mask :1111111111111111111111110000000000000000000000000000000000000000000011000000000000000
 2-Deletion Mask :0000000011111111111111111111111111100011111111111111111111111000100011111111111111110
 3-Deletion Mask :1111111111111111111111111111111110001111111111111111111111111111111111111111111111111
1-Insertion Mask :1111111111111111111111111111111000111111111111111111111111111111000111111111111111110
2-Insertion Mask :0000001111111111111111111111100011111111111111111111111111111111111111111000111111100
3-Insertion Mask :1111111111111111110001111111111111111111111111111111111111111111111111111111111111000

       AND Mask :0000000000100000000000010000000000000000000000000000000000000000000001000000000000000


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

GateKeeper Walkthrough (cont'd)

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

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

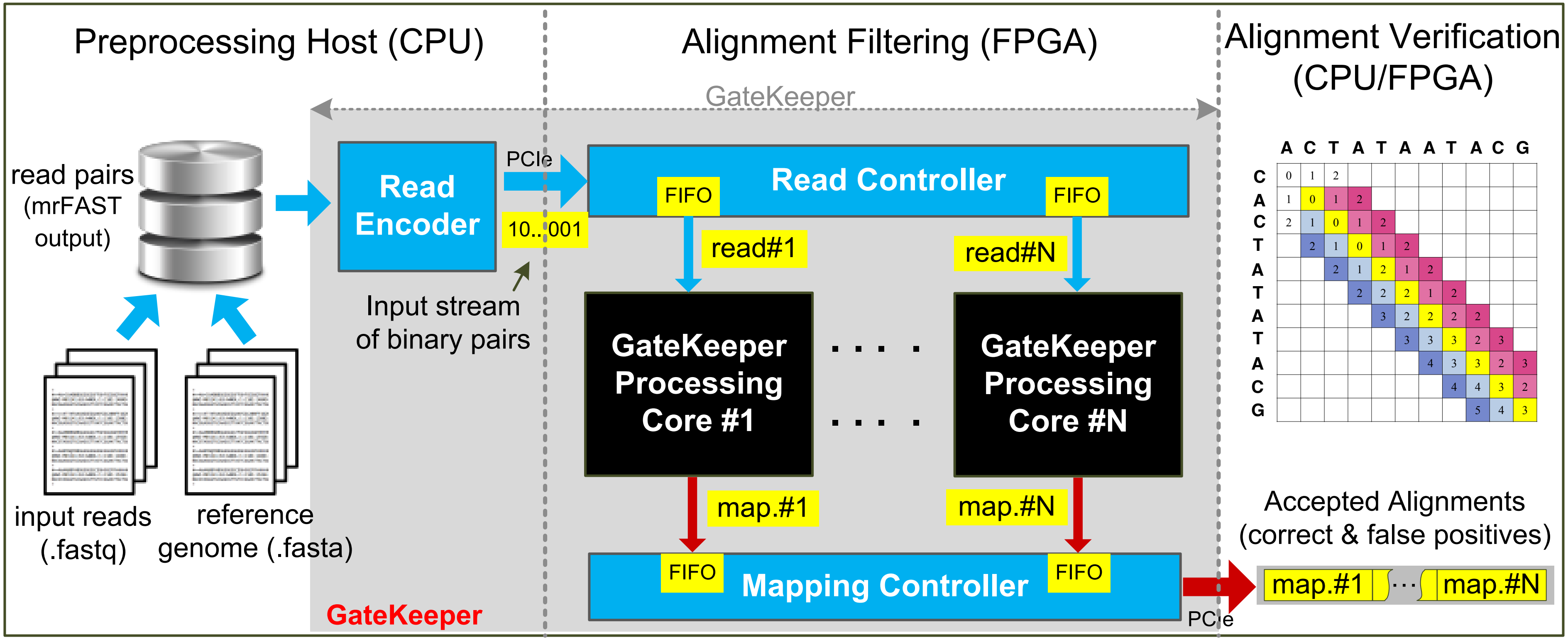
- (2E)*(ReadLength) 2-AND operations.
- (ReadLength/4) 5-input LUT.
- log_2ReadLength-bit counter.



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

GateKeeper Accelerator Architecture

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


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

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

GateKeeper vs. SHD

GateKeeper	SHD
■ FPGA (Xilinx VC709)	■ Intel SIMD
■ Multi-core (parallel)	■ Single-core (sequential)
■ Examines a single mapping @ 125 MHz	■ Examines a single mapping @ ~2MHz
■ Limited to PCIe Gen3(4x) transfer rate (128 bits @ 250MHz)	■ Limited to a read length of 128 bp (SSE register size)
■ Amending requires: ❑ (2E+1) 5-input LUT.	■ Amending requires: ❑ 4(2E+1) bitwise OR. ❑ 4(2E+1) packed shuffle. ❑ 3(2E+1) shift.

GateKeeper: Speed & Accuracy Results

90x-130x faster filter

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

4x lower false accept rate

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

10x speedup in read mapping

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

Freely available online

github.com/BilkentCompGen/GateKeeper

Conclusions

- FPGA-based pre-alignment greatly speeds up read mapping
  - 10x speedup of a state-of-the-art mapper (mrFAST)

- FPGA-based pre-alignment can be integrated with the sequencer
  - It can help to hide the complexity and details of the FPGA
  - Enables real-time filtering while sequencing

More on GateKeeper

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

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

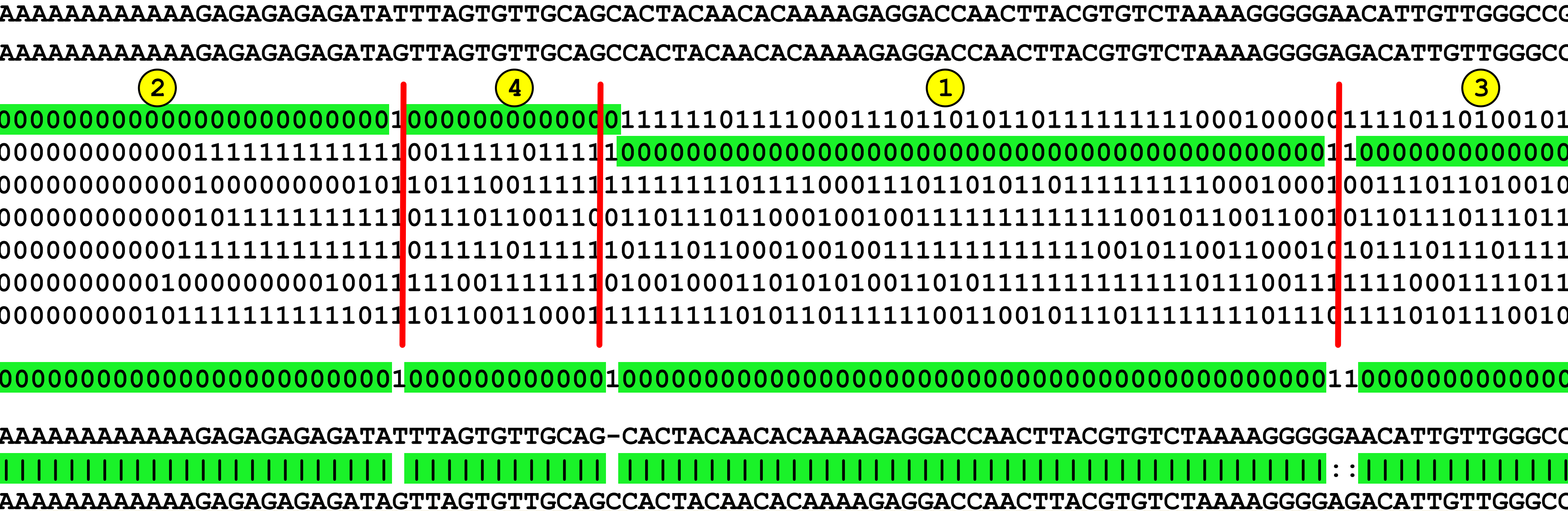
Sequence analysis

GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping

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

Next Talk: MAGNET

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



```
AAAAAAAAAAAAGAGAGAGAGATATTTAGTGTTGCAGCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAACATTGTTGGCCG
AAAAAAAAAAAAGAGAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGCC
             2                      4                          1                                3
0000000000000000000000001000000000000011111101111000111011101011011111111100010000011110110100101
0000000000001111111111111001111101111100000000000000000000000000000000000000000000011000000000000
0000000000001000000000101101110011111111111111011110001110110101101111111110001000100111011010010
0000000000001011111111111011101100110001101110110001001001111111111111001011001100101101110111011
0000000000001111111111111011111011111101110110001001001111111111111001011001100010101110111011111
0000000000100000000010011111001111111010010001101010100110101111111111111110111001111111000111011
0000000001011111111111011101100110001111111110101101111111001100101110111111111011100111101011100100

0000000000000000000000001000000000000100000000000000000000000000000000000000000000000110000000000000

AAAAAAAAAAAAGAGAGAGAGATATTTAGTGTTGCAG-CACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAACATTGTTGGCC
||||||||||||||||||||||||| ||||||||||||| ||||||||||||||||||||||||||||||||||||||||||||::|||||||||||||
AAAAAAAAAAAAGAGAGAGAGATAGTTAGTGTTGCAGCCACTACAACACAAAAGAGGACCAACTTACGTGTCTAAAAGGGGAGACATTGTTGGCC
```

Agenda

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

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

Read Mapping & Filtering

- Problem: Heavily bottlenecked by Data Movement

- GateKeeper performance limited by DRAM bandwidth [Alser+, Bioinformatics 2017]

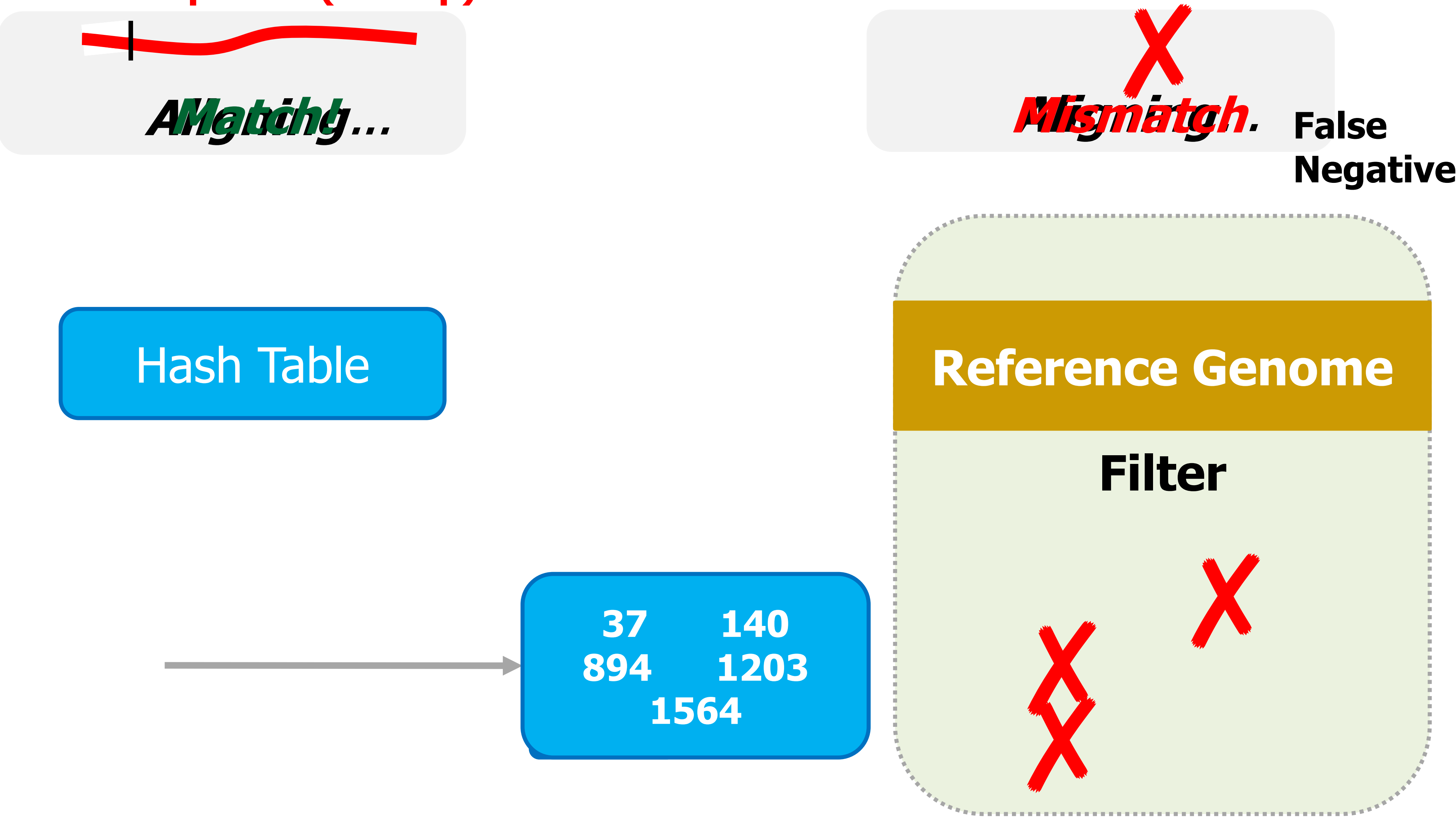
- Ditto for SHD [Xin+, Bioinformatics 2015]

- Solution: Processing-in-memory can alleviate the bottleneck

- However, we need to design mapping & filtering algorithms to fit processing-in-memory

Hash Tables in Read Mapping

Read Sequence (100 bp)



Aligning...

Match!

Aligning.

Mismatch

False Negative

Hash Table

37 140
894 1203
1564

Reference Genome

Filter

SAFARI

Read Mapping & Filtering in Memory

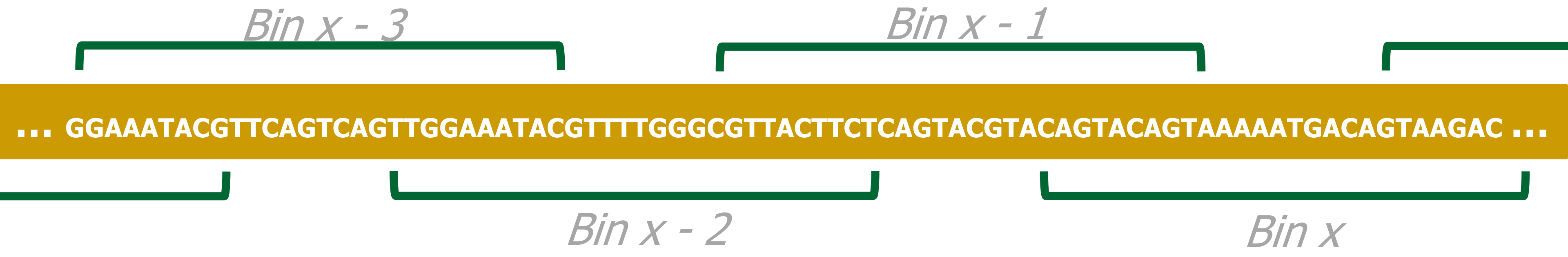
We need to design
mapping & filtering algorithms
that fit processing-in-memory

Our Proposal: GRIM-Filter

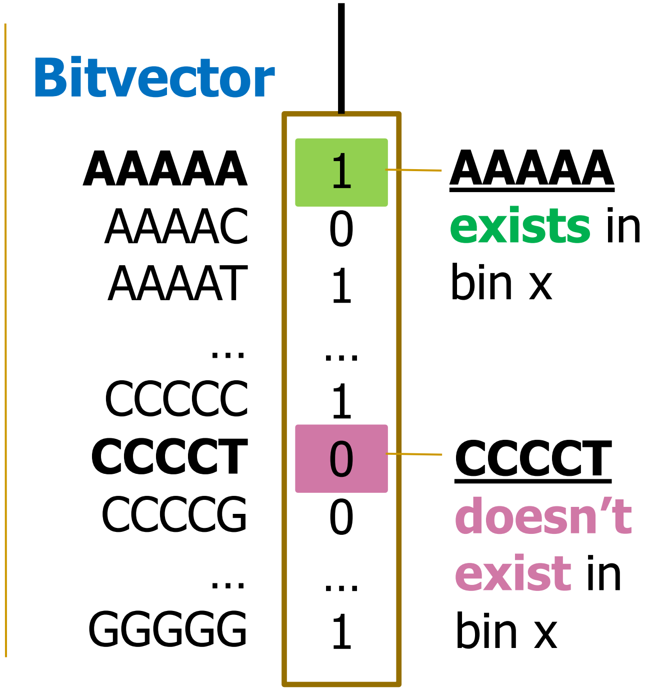
1. **Data Structures: Bins & Bitvectors**
2. Checking a Bin
3. Integrating GRIM-Filter into a Mapper

GRIM-Filter: Bins

- We partition the genome into large sequences (**bins**).


Bin x - 3
Bin x - 1
… GGAAATACGTTCAGTCAGTTGGAAATACGTTTTGGGCGTTACTTCTCAGTACGTACAGTACAGTAAAAATGACAGTAAGAC …
Bin x - 2
Bin x

- Represent each bin with a **bitvector** that holds the occurrence of all permutations of a small string (**token**) in the bin
- To account for matches that straddle bins, we employ overlapping bins
  - A read will now always completely fall within a single bin

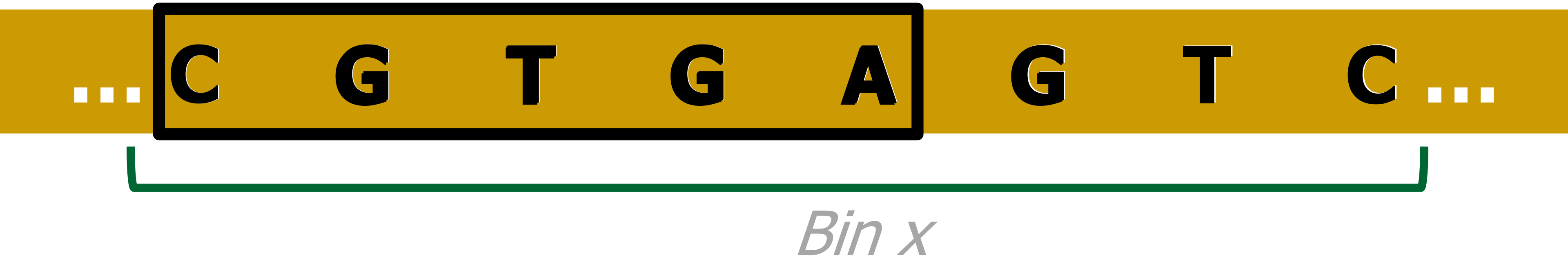

Bitvector

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

AAAAA **exists** in bin x

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

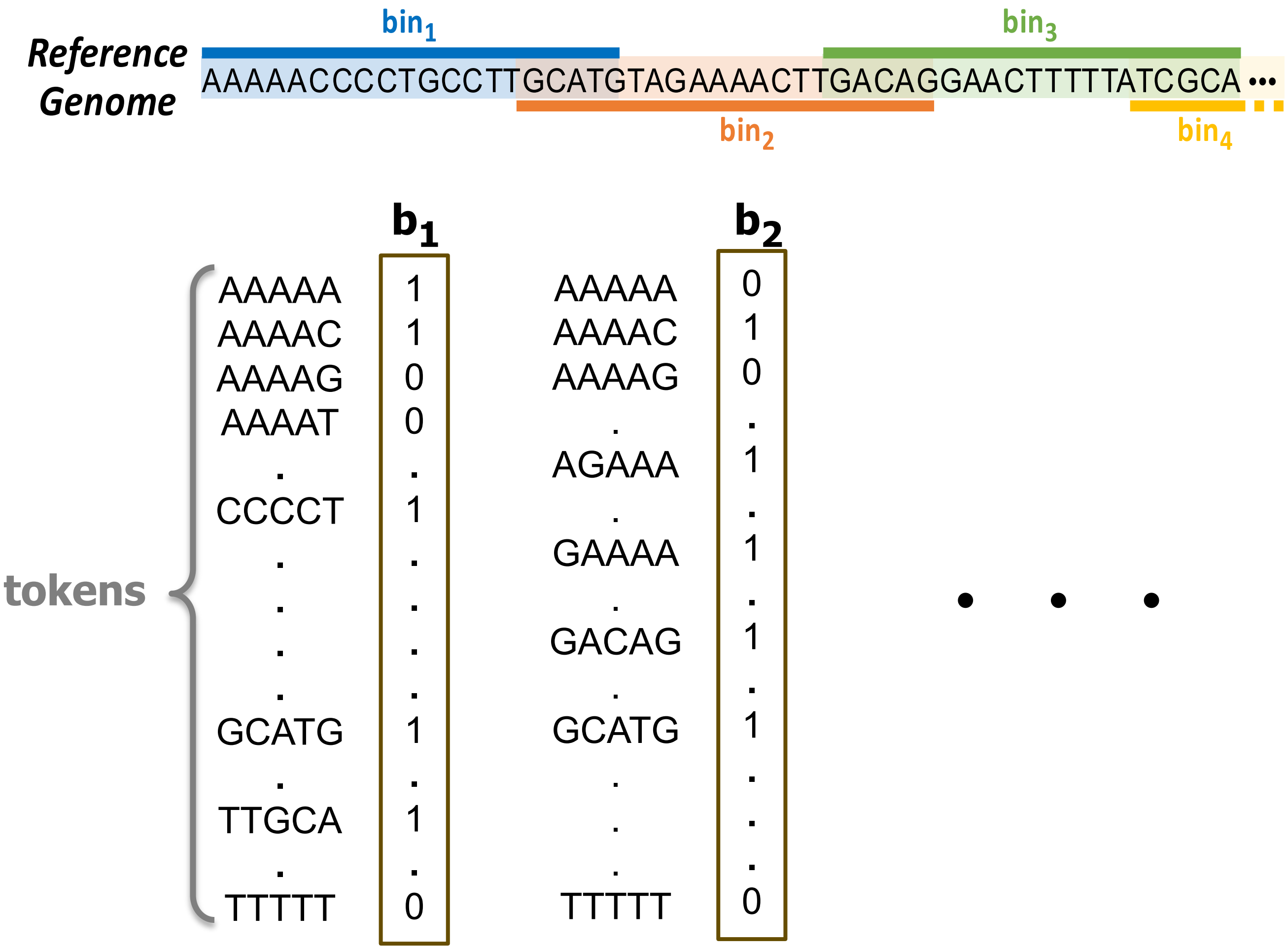
GRIM-Filter: Bitvectors



Bin x Bitvector

AAAAA	0
…	…
CGTGA	0
…	…
TGAGT	0
…	…
GAGTC	0
…	…
GTGAG	0
…	…

GRIM-Filter: Bitvectors



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

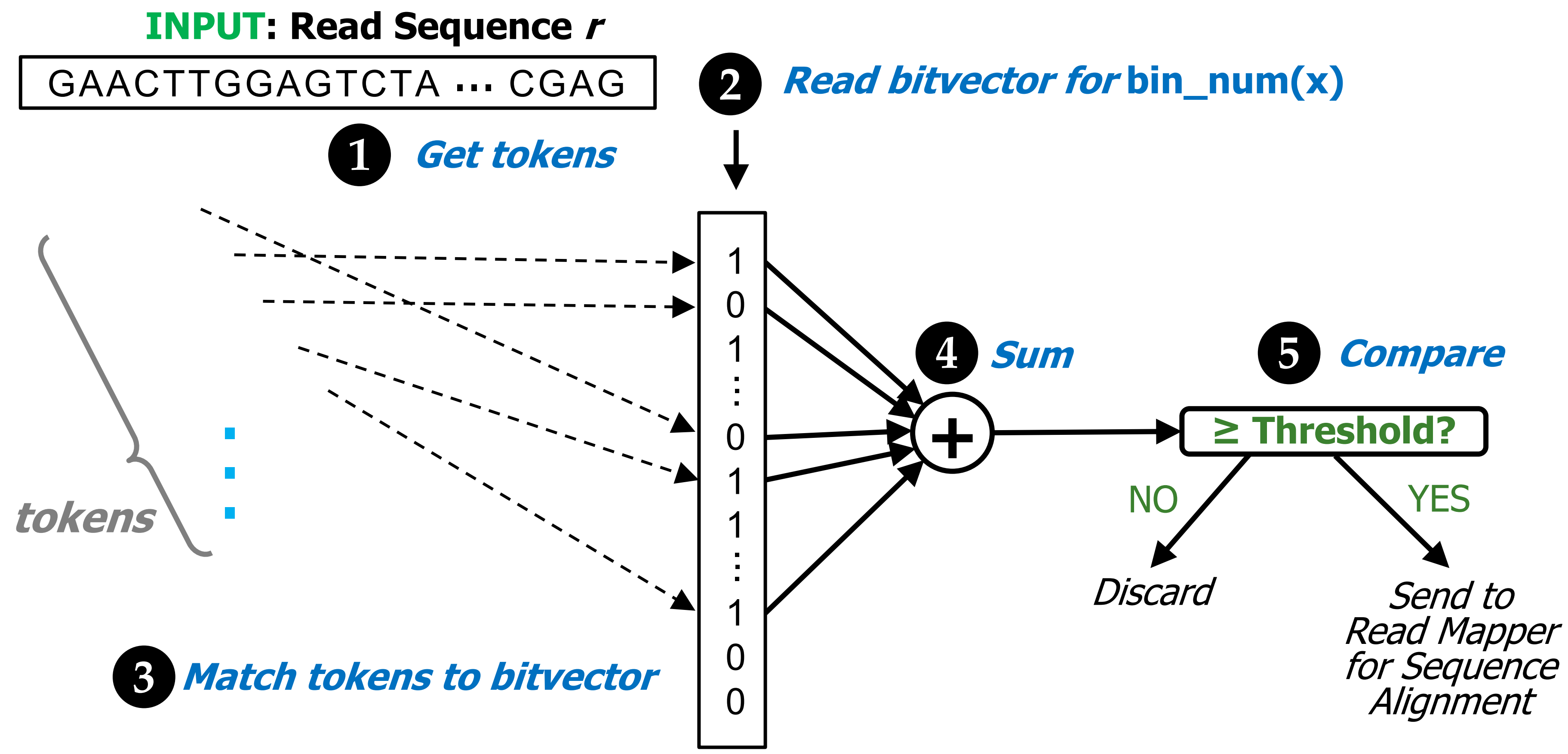
For **bin size** ~200, and **n** = 5, **memory footprint** ~3.8 GB

Our Proposal: GRIM-Filter

1. Data Structures: Bins & Bitvectors
2. **Checking a Bin**
3. Integrating GRIM-Filter into a Mapper

GRIM-Filter: Checking a Bin

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



INPUT: Read Sequence *r*

GAACTTGGAGTCTA ... CGAG

1 *Get tokens*

2 *Read bitvector for* **bin_num(x)**

3 *Match tokens to bitvector*

4 *Sum*

5 *Compare*

SAFARI

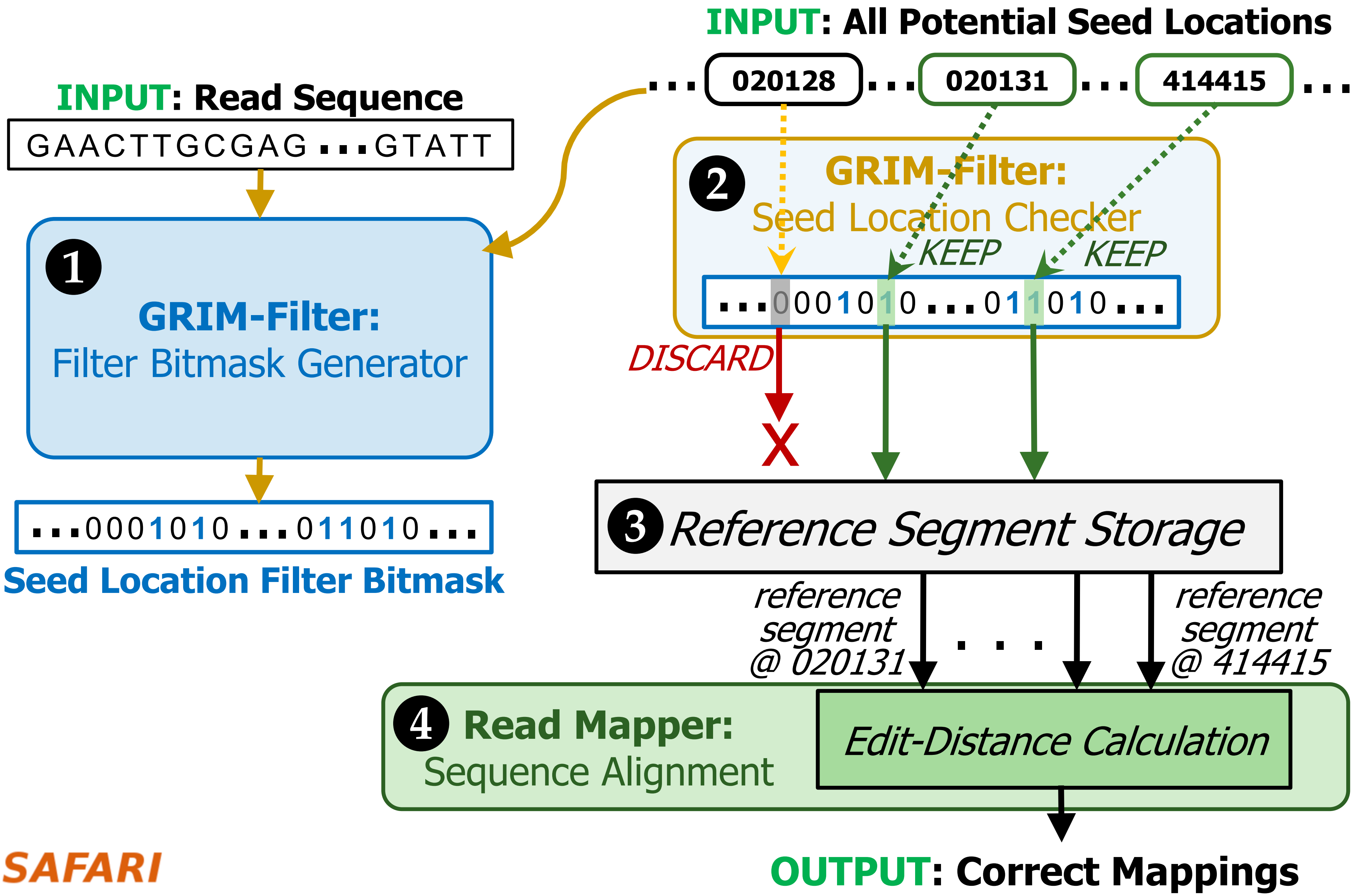
Our Proposal: GRIM-Filter

1. Data Structures: Bins & Bitvectors
2. Checking a Bin
3. Integrating GRIM-Filter into a Mapper

Our Proposal: GRIM-Filter

1. Data Structures: Bins & Bitvectors
2. Checking a Bin
3. **Integrating GRIM-Filter into a Mapper**

Integrating GRIM-Filter into a Read Mapper



INPUT: Read Sequence

GAACTTGCGAG...GTATT

1 **GRIM-Filter:** Filter Bitmask Generator

...0001010...011010...

Seed Location Filter Bitmask

INPUT: All Potential Seed Locations

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

2 **GRIM-Filter:** Seed Location Checker

KEEP *KEEP*

...0001010...011010...

DISCARD

X

3 *Reference Segment Storage*

reference segment @ 020131

. . .

reference segment @ 414415

4 **Read Mapper:** Sequence Alignment

Edit-Distance Calculation

OUTPUT: Correct Mappings

Key Properties of GRIM-Filter

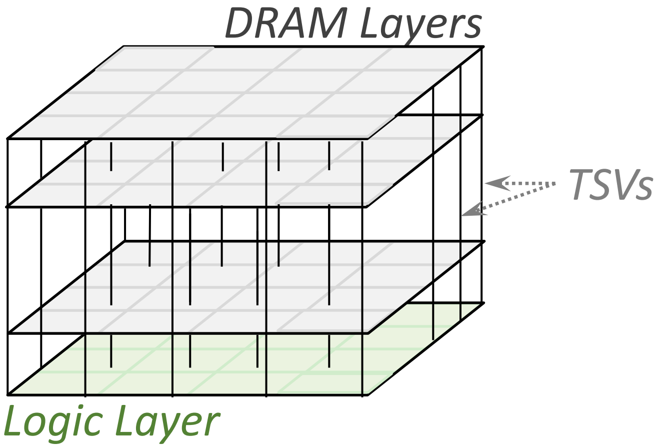
1. **Simple Operations**:
   - To check a given bin, find the **sum** of all bits corresponding to each token in the read
   - **Compare** against threshold to determine whether to align

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

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

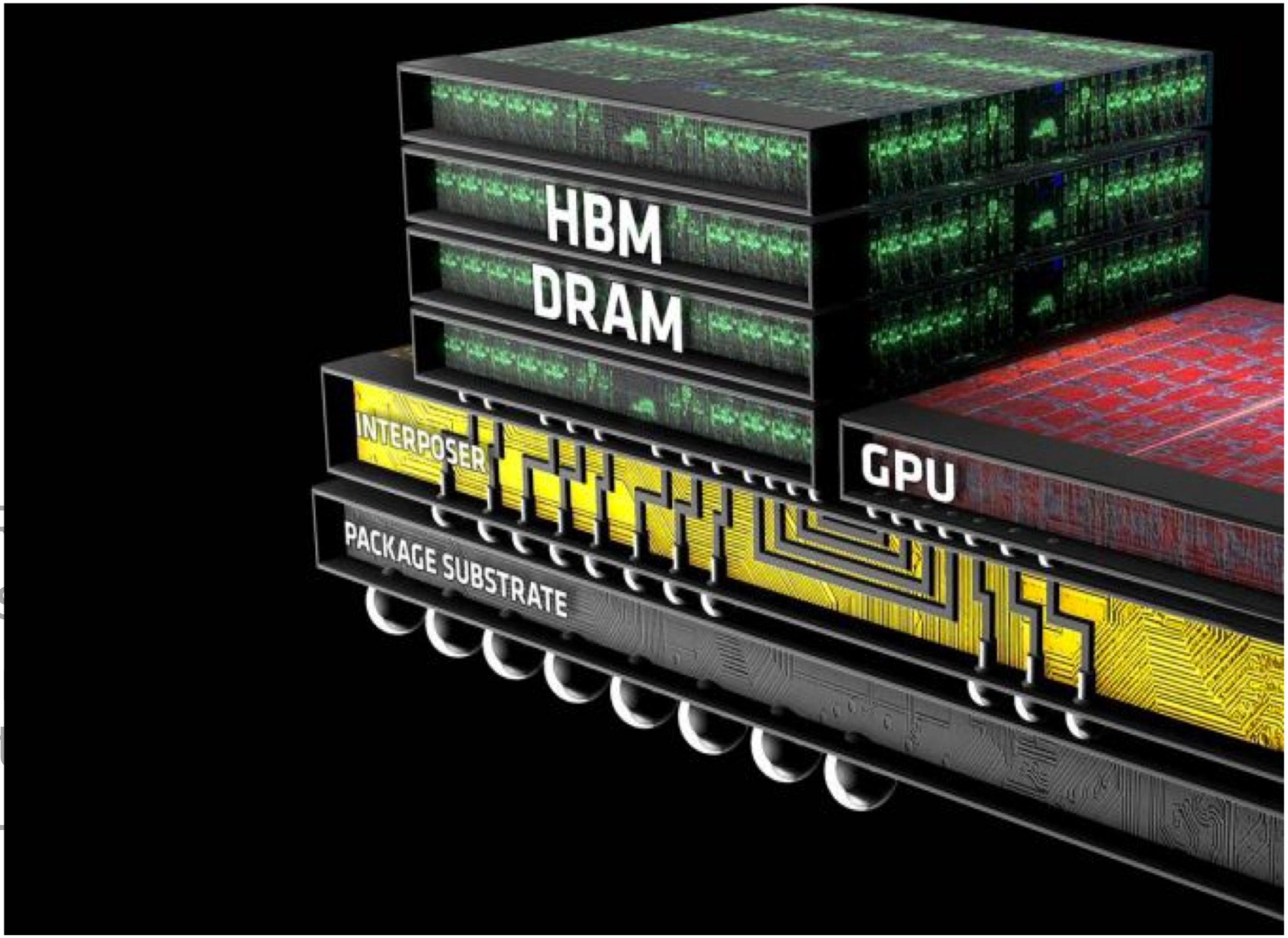
These properties together make GRIM-Filter a good algorithm to be run in 3D-Stacked DRAM

3D-Stacked Memory



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

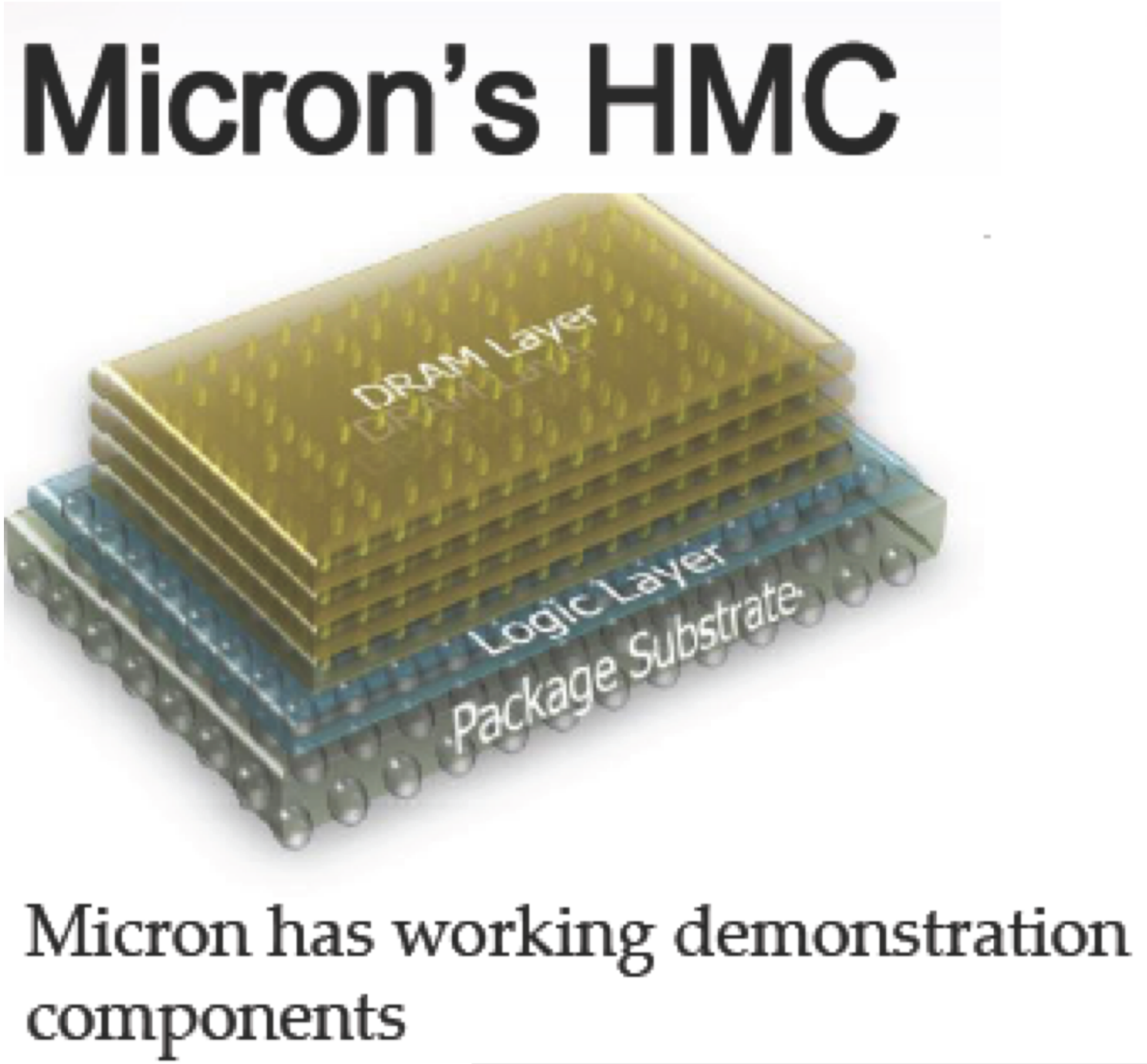
3D-Stacked Memory



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

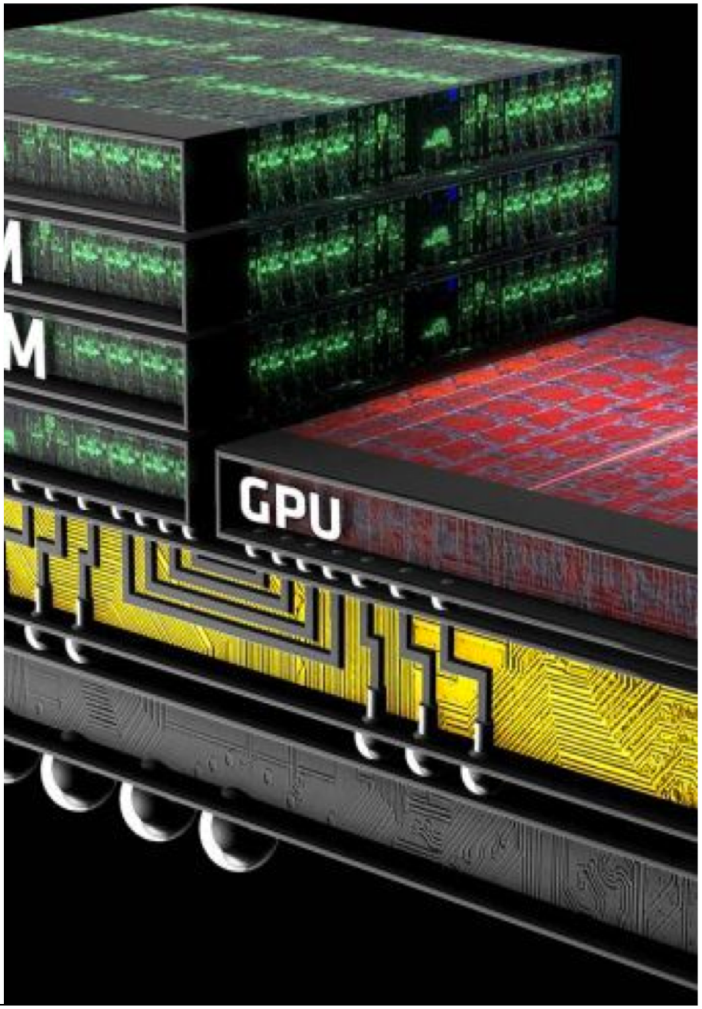
http://i1-news.softpedia-static.com/images/news2/Micron-and-Samsung-Join-Force-to-Create-Next-Gen-Hybrid-Memory-2.png

3D-Stacked Memory



Micron's HMC

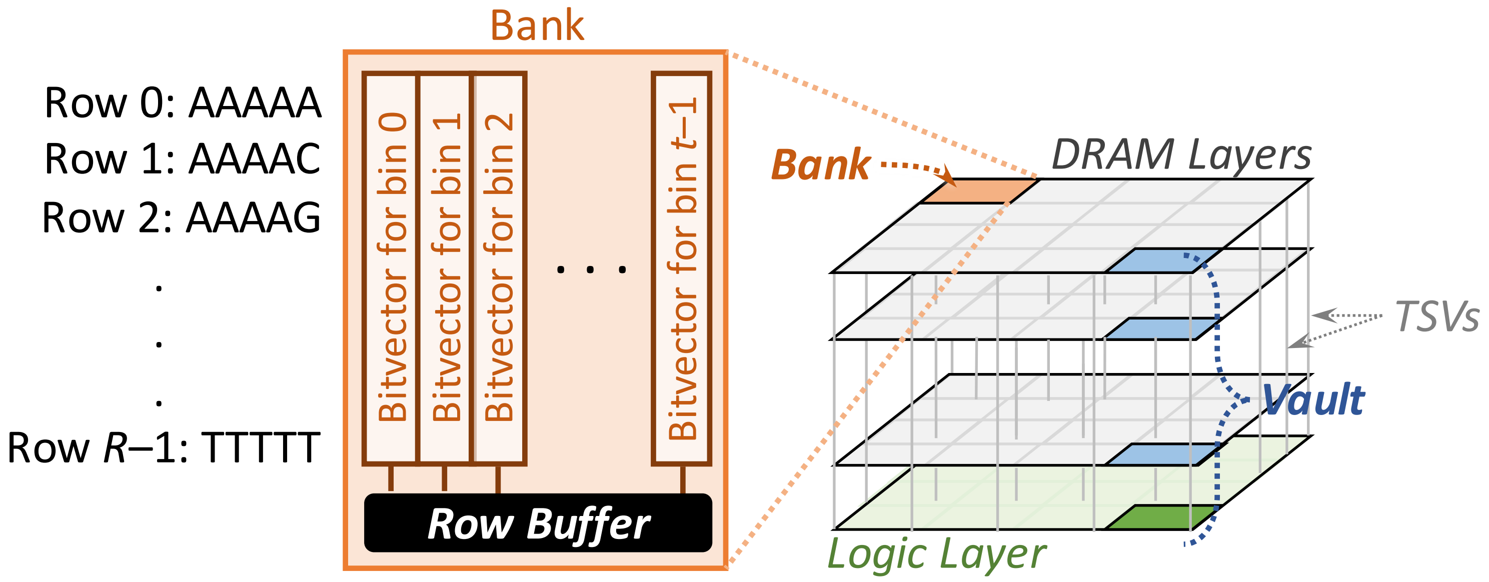
Micron has working demonstration components



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

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

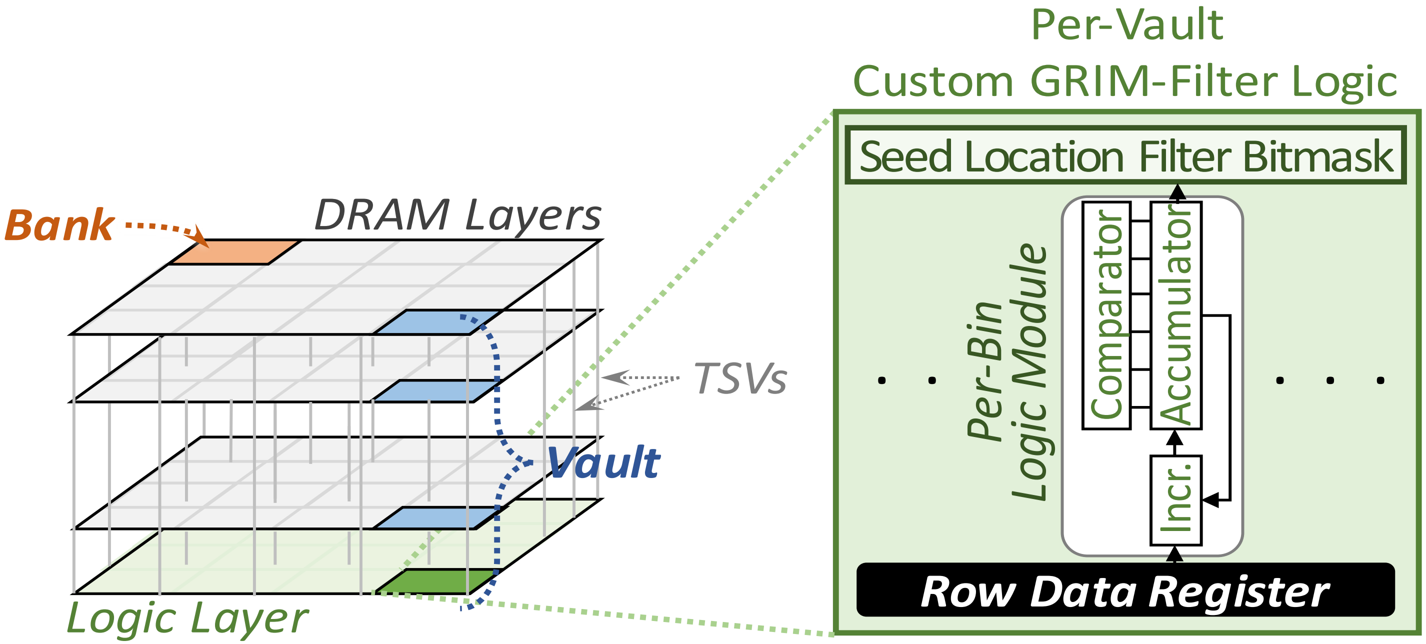
GRIM-Filter in 3D-Stacked DRAM



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

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

GRIM-Filter in 3D-Stacked DRAM



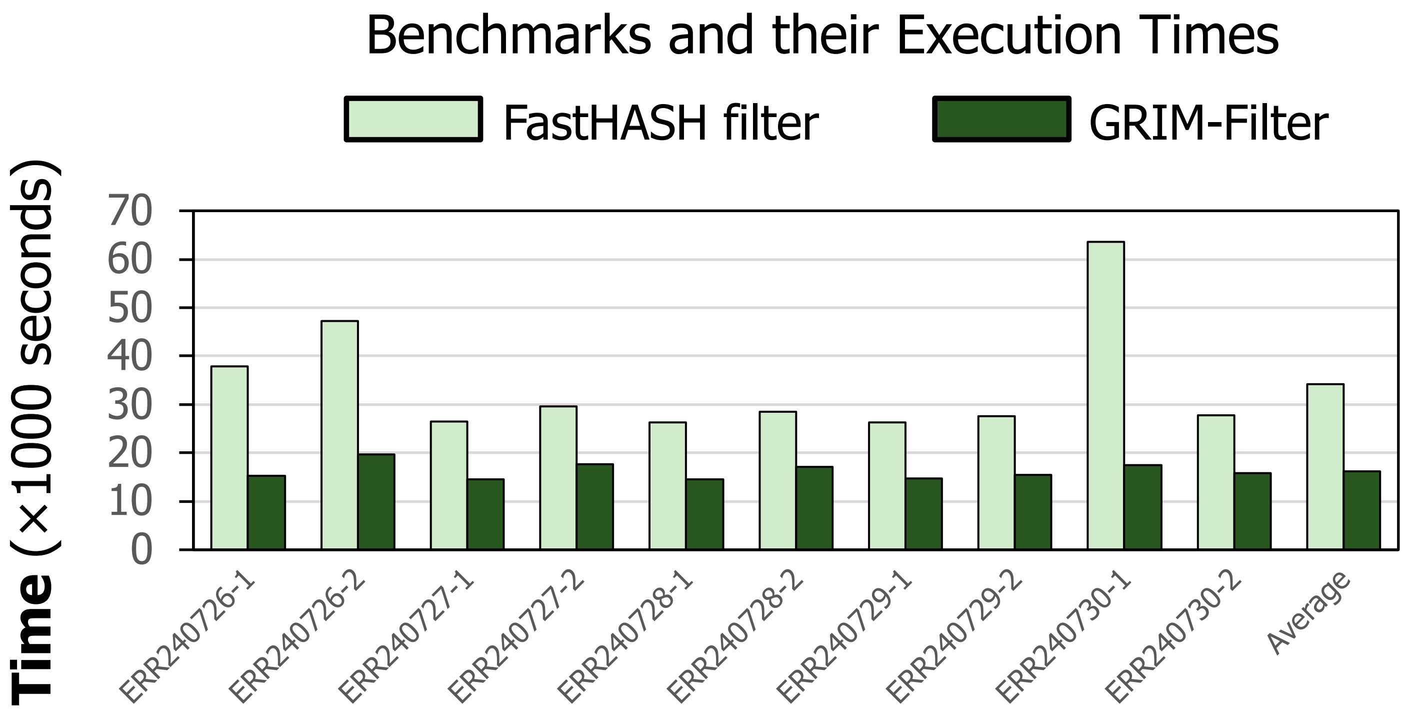
- Customized logic for accumulation and comparison per genome segment
  - Low area overhead, simple implementation
  - For HBM2, we use 4096 incrementer LUTs, 7-bit counters, and comparators in logic layer

Details are in [Kim+, BMC Genomics 2018]

Methodology

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

GRIM-Filter Performance



Sequence Alignment Error Tolerance (*e*)

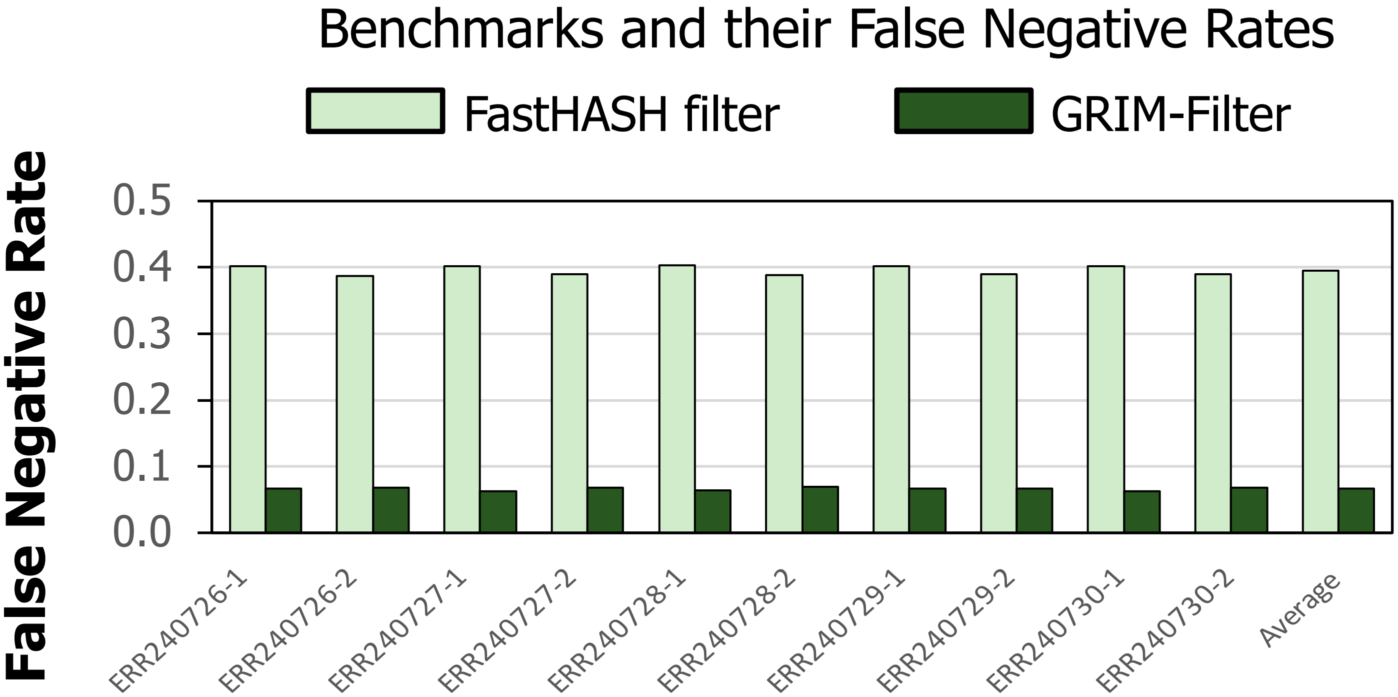
***e* = 0.05**

1.8x-3.7x performance benefit across real data sets

2.1x average performance benefit

GRIM-Filter gets performance due to its hardware-software co-design

GRIM-Filter False Negative Rate



Sequence Alignment Error Tolerance (*e*)

***e* = 0.05**

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

6.0x average reduction in False Negative Rate

GRIM-Filter utilizes more information available in the read to filter

More on GRIM-Filter

- Jeremie S. Kim, Damla Senol Cali, Hongyi Xin, Donghyuk Lee, Saugata Ghose, Mohammed Alser, Hasan Hassan, Oguz Ergin, Can Alkan, and Onur Mutlu, **"GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies"**
  *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]

Agenda

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

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

Recall: High-Throughput Sequencing

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

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

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

Nanopore Sequencing Technology

- **Nanopore sequencing** is an emerging and a promising single-molecule DNA sequencing technology
  - No amplification → Less limit on read length → Longer read length

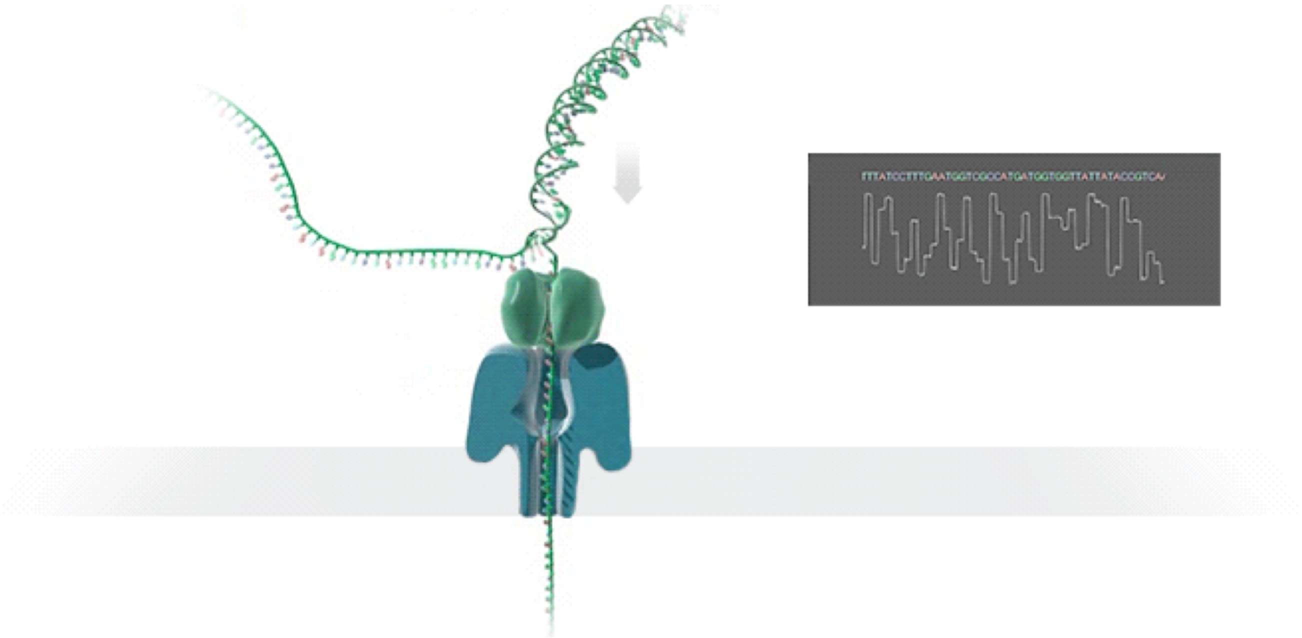
- First nanopore sequencing device, **MinION**, made commercially available by **Oxford Nanopore Technologies** (ONT) in **May 2014.**
  - Inexpensive
  - Long read length (> 882K bp)
  - Portable: Pocket-sized
  - Produces data in real-time

Nanopore Sequencing Technology

an emerging and a promising
ncing technology
read length → Longer read length

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

Nanopore Sequencing



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

Advantages of Nanopore Sequencing

Nanopores:

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

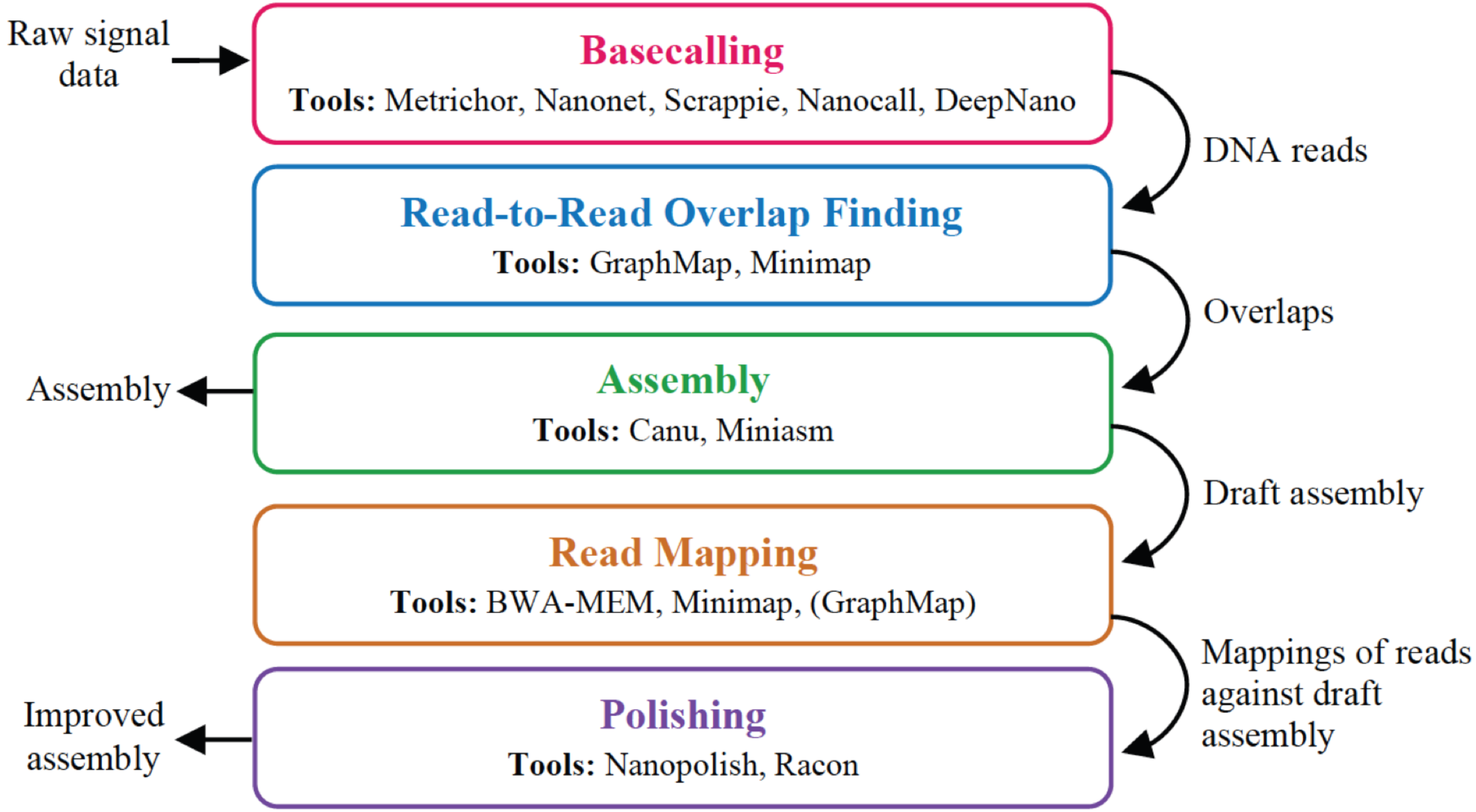
Challenges of Nanopore Sequencing

- One major drawback: high error rates

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

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

Nanopore Genome Assembly Pipeline



Raw signal data

Basecalling

Tools: Metrichor, Nanonet, Scrappie, Nanocall, DeepNano

DNA reads

Read-to-Read Overlap Finding

Tools: GraphMap, Minimap

Overlaps

Assembly

Tools: Canu, Miniasm

Assembly

Draft assembly

Read Mapping

Tools: BWA-MEM, Minimap, (GraphMap)

Mappings of reads against draft assembly

Polishing

Tools: Nanopolish, Racon

Improved assembly

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

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

More on Nanopore Sequencing & Tools

Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks, and Future Directions

Damla Senol Cali [1,*], Jeremie Kim [1,3], Saugata Ghose [1], Can Alkan [2*] and Onur Mutlu [3,1*]

[1] Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA
[2] Department of Computer Engineering, Bilkent University, Bilkent, Ankara, Turkey
[3] Department of Computer Science, Systems Group, ETH Zürich, Zürich, Switzerland

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

Agenda

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

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

- Hardware Acceleration
  - Specialized Architectures
  - Processing in Memory

- Future Opportunities: New Sequencing Technologies

Conclusion

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

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

- We covered various recent ideas to accelerate read mapping
  - My personal journey since September 2006

- **Many future opportunities exist**
  - **Especially with new sequencing technologies**

Acknowledgments

- Prof. Can Alkan, Bilkent University

- Many students
  - Mohammed Alser, Damla Senol Cali, Jeremie Kim
  - Hasan Hassan
  - Hongyi Xin
  - …

Accelerating Genome Analysis

A Primer on an Ongoing Journey

Onur Mutlu

omutlu@gmail.com

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

January 24, 2018

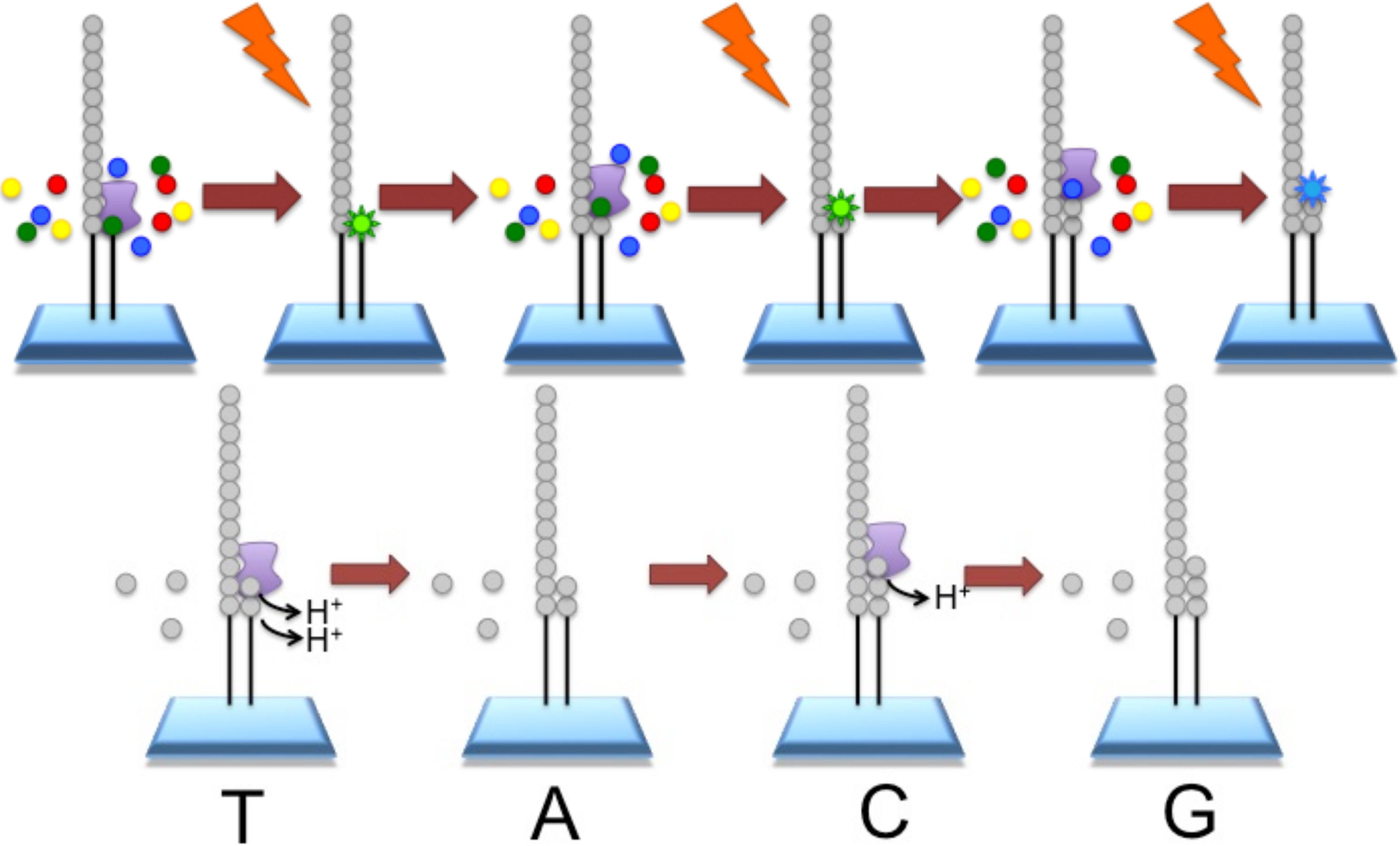
AACBB Keynote, Vienna

Systems@ETH zürich

SAFARI

ETH zürich

High-Throughput Sequencing



Nanopore Sequencing

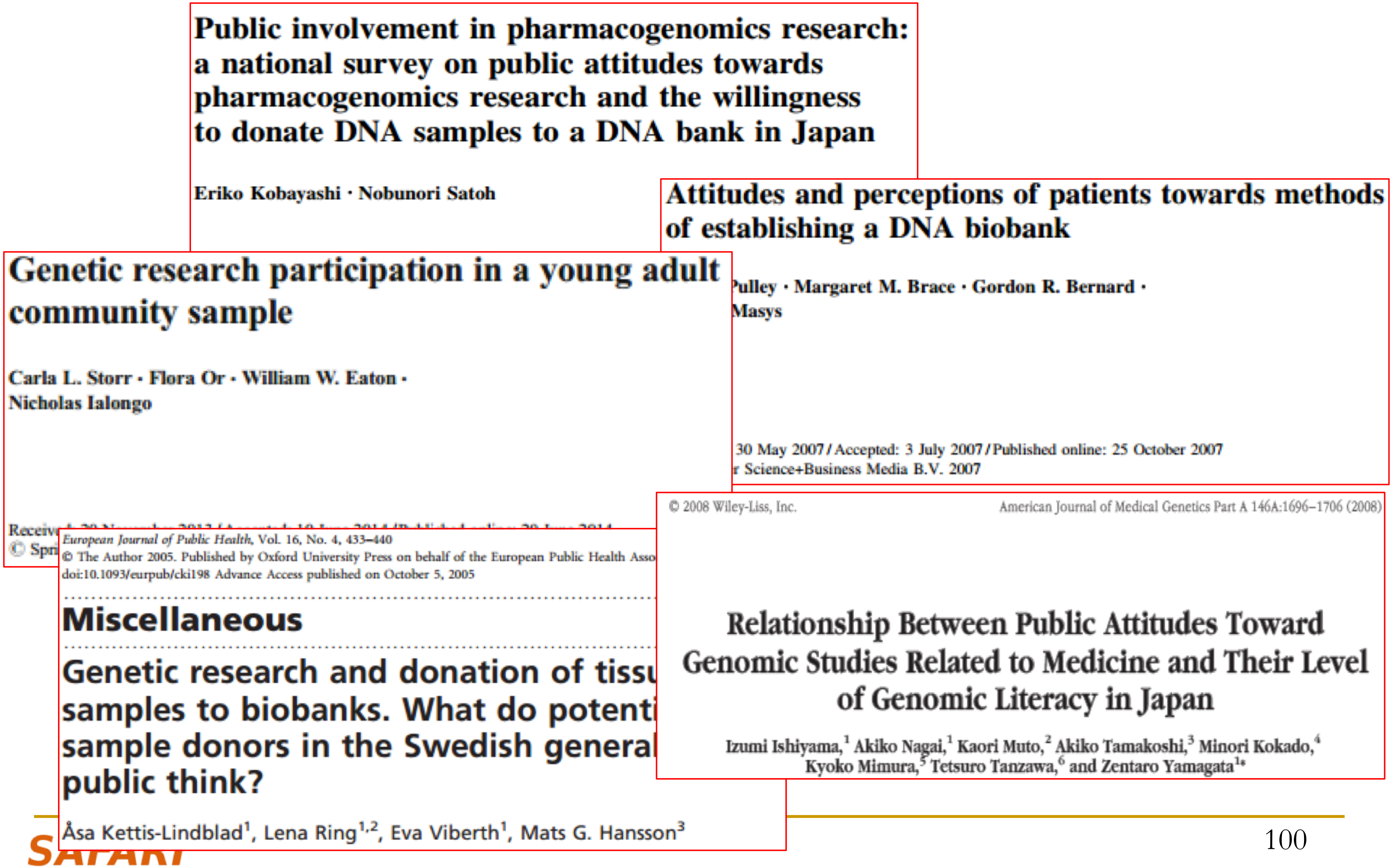
- **Basecalling** translates the raw signal output of the nanopore sequencer into bases (A, C, G, T) to generate DNA reads.
  - 1) The raw current signal is divided into discrete blocks (events).
  - 2) Each event is decoded into a most-likely set of bases.

- **Deletions** are the dominant error of nanopore sequencing.
  - In the ideal case, each consecutive event should differ by one base. However, in practice, this is not the case because of the non-stable speed of the translocation.
  - Determining the correct length of the homopolymers (*i.e.*, repeating stretches of one kind of base, *e.g.*, AAAAAAA) is challenging.

The Importance of Genome Analysis?

Helps, for example, to answer the following 3 questions:

69–92% of the respondents in these studies had positive attitudes towards genomics research and donating their DNA samples.



Public involvement in pharmacogenomics research: a national survey on public attitudes towards pharmacogenomics research and the willingness to donate DNA samples to a DNA bank in Japan

Eriko Kobayashi · Nobunori Satoh

Attitudes and perceptions of patients towards methods of establishing a DNA biobank

ulley · Margaret M. Brace · Gordon R. Bernard ·
Masys

30 May 2007 / Accepted: 3 July 2007 / Published online: 25 October 2007
r Science+Business Media B.V. 2007

Genetic research participation in a young adult community sample

Carla L. Storr · Flora Or · William W. Eaton ·
Nicholas Ialongo

Receiv
© Spri

European Journal of Public Health, Vol. 16, No. 4, 433–440
© The Author 2005. Published by Oxford University Press on behalf of the European Public Health Asso
doi:10.1093/eurpub/cki198 Advance Access published on October 5, 2005

Miscellaneous

Genetic research and donation of tissu samples to biobanks. What do potenti sample donors in the Swedish general public think?

Åsa Kettis-Lindblad[1], Lena Ring[1,2], Eva Viberth[1], Mats G. Hansson[3]

© 2008 Wiley-Liss, Inc. American Journal of Medical Genetics Part A 146A:1696–1706 (2008)

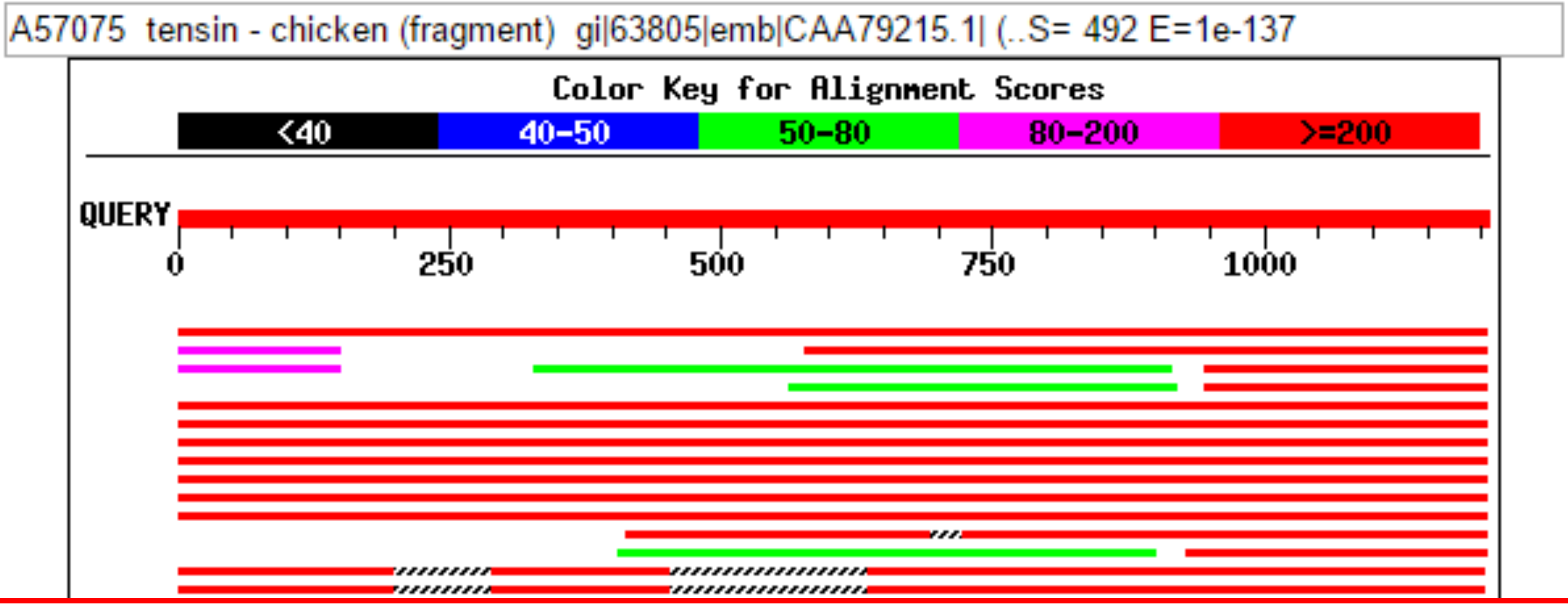
Relationship Between Public Attitudes Toward Genomic Studies Related to Medicine and Their Level of Genomic Literacy in Japan

Izumi Ishiyama,[1] Akiko Nagai,[1] Kaori Muto,[2] Akiko Tamakoshi,[3] Minori Kokado,[4] Kyoko Mimura,[5] Tetsuro Tanzawa,[6] and Zentaro Yamagata[1*]

Pairwise sequence alignment

Distribution of 116 Blast Hits on the Query Sequence

BLAST



A57075 tensin - chicken (fragment) gi|63805|emb|CAA79215.1| (..S= 492 E=1e-137

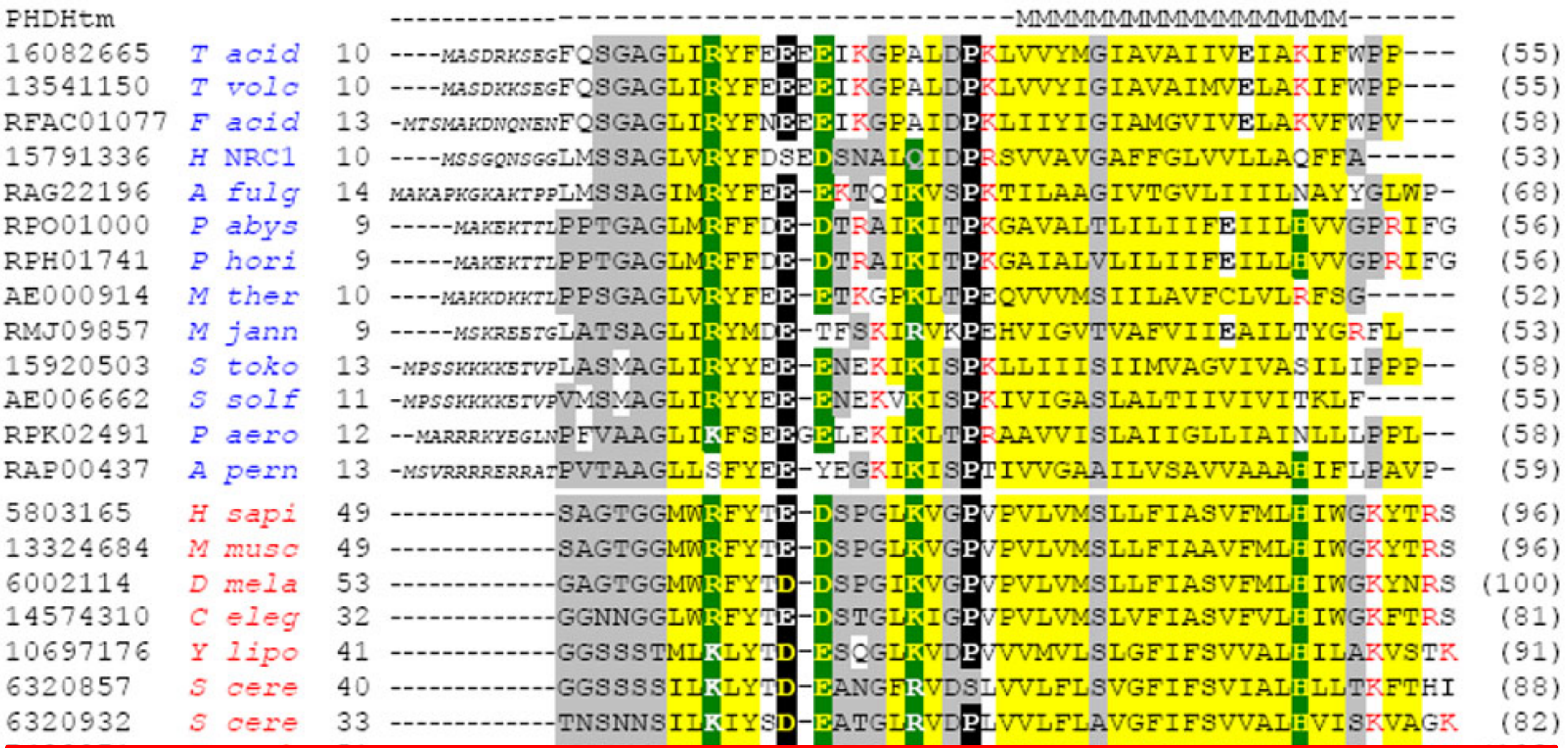
Question #1: If I give you a gene sequence, tell me which of the billions of known sequences is most similar to it.

CODIS: Combined DNA Index System

- FBI's program of support for criminal justice.
- CODIS defines 13 human DNA regions (loci) to be stored in the database for personal identification purposes.
- Stored 14.5 million DNA profiles (for offenders, arrestees ..)
- As of September 2016, CODIS has produced over 346,880 hits assisting in more than 332,776 investigations.

https://www.fbi.gov/services/laboratory/biometric-analysis/codis/ndis-statistics

Multiple sequence alignment

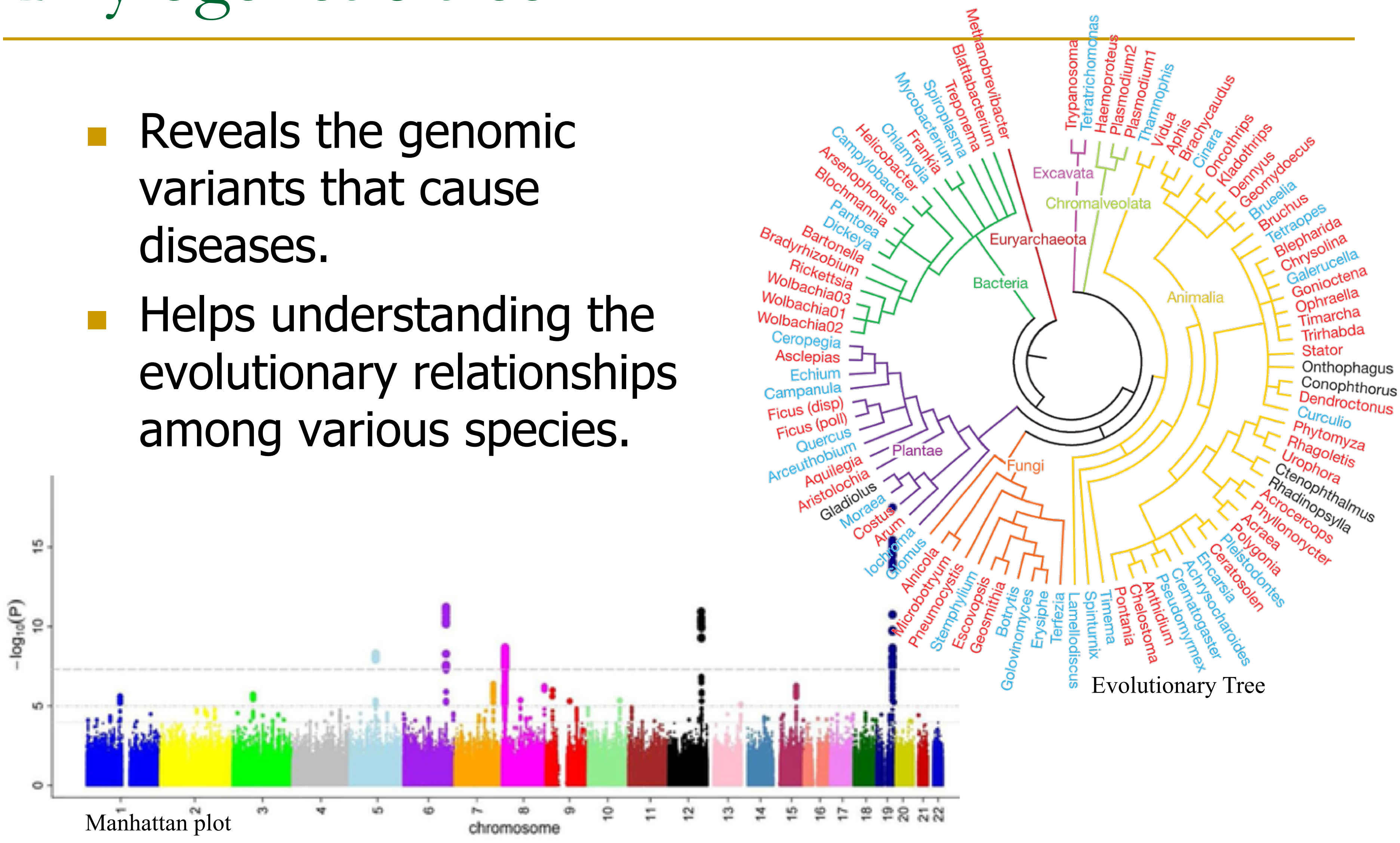


```
PHDHtm                          -------------------------------------MMMMMMMMMMMMMMMMMMM------
16082665  T acid     10  ----MASDRKSEGFQSGAGLIRYFEEEEIKGPALDPKLVVYMGIAVAIIVEIAKIFWPP---  (55)
13541150  T volc     10  ----MASDKKSEGFQSGAGLIRYFEEEEIKGPALDPKLVVYIGIAVAIMVELAKIFWPP---  (55)
RFAC01077 F acid     13  -MTSMAKDNQNENFQSGAGLIRYFNEEEIKGPAIDPKLIIYIGIAMGVIVELAKVFWPV---  (58)
15791336  H NRC1     10  ----MSSGQNSGGLMSSAGLVRYFDSEDSNALQIDPRSVVAVGAFFGLVVLLAQFFA-----  (53)
RAG22196  A fulg     14  MAKAPKGKAKTPPLMSSAGIMRYFEE-EKTQIKVSPKTILAAGIVTGVLIIIILNAYYGLWP-  (68)
RPO01000  P abys      9  -----MAKEKTTLPPTGAGLMRFFDE-DTRAIKITPKGAVALTLILIIFEIILHVVGPRIFG  (56)
RPH01741  P hori      9  -----MAKEKTTLPPTGAGLMRFFDE-DTRAIKITPKGAIALVLILIIFEILLHVVGPRIFG  (56)
AE000914  M ther     10  ----MAKKDKKTLPPSGAGLVRYFEE-ETKGPKLTPEQVVVMSIILAVFCLVLRFSG-----  (52)
RMJ09857  M jann      9  -----MSKREETGLATSAGLIRYMDE-TFSKIRVKPEHVIGVTVAFVIIEAILTYGRFL---  (53)
15920503  S toko     13  -MPSSKKKKETVPLASMAGLIRYYEE-ENEKIKISPKLLIIISIIMVAGVIVASILIPPP--  (58)
AE006662  S solf     11  -MPSSKKKKETVPVMSMAGLIRYYEE-ENEKVKISPKIVIGASLALTIIVIVITKLF-----  (55)
RPK02491  P aero     12  --MARRRKYEGLNPFVAAGLIKFSEEGELEKIKLTPRAAVVISLAIIGLLIAINLLLPPL--  (58)
RAP00437  A pern     13  -MSVRRRRERRATPVTAAGLLSFYEE-YEGKIKISPTIVVGAAILVSAVVAAAHIFLPAVP-  (59)

5803165   H sapi     49  -------------SAGTGGMWRFYTE-DSPGLKVGPVPVLVMSLLFIASVFMLHIWGKYTRS  (96)
13324684  M musc     49  -------------SAGTGGMWRFYTE-DSPGLKVGPVPVLVMSLLFIAAVFMLHIWGKYTRS  (96)
6002114   D mela     53  -------------GAGTGGMWRFYTD-DSPGIKVGPVPVLVMSLLFIASVFMLHIWGKYNRS (100)
14574310  C eleg     32  -------------GGNNGGLWRFYTE-DSTGLKIGPVPVLVMSLVFIASVFVLHIWGKFTRS  (81)
10697176  Y lipo     41  -------------GGSSSTMLKLYTD-ESQGLKVDPVVVMVLSLGFIFSVVALHILAKVSTK  (91)
6320857   S cere     40  -------------GGSSSSILKLYTD-EANGFRVDSLVVLFLSVGFIFSVIALHLLTKFTHI  (88)
6320932   S cere     33  -------------TNSNNSILKIYSD-EATGLRVDPLVVLFLAVGFIFSVVALHVISKVAGK  (82)
```

Question #2: If I give you a bunch of sequences, tell me where they are the same and where they are different.

Phylogenetic tree

- Reveals the genomic variants that cause diseases.
- Helps understanding the evolutionary relationships among various species.



Evolutionary Tree

Manhattan plot

http://www.nature.com/nature/journal/v465/n7300/fig_tab/nature09113_F3.html

The genetic similarity between species

Human ~ Human

99.9%

Human ~ Chimpanzee

96%

Human ~ Cat

90%

Human ~ Cow

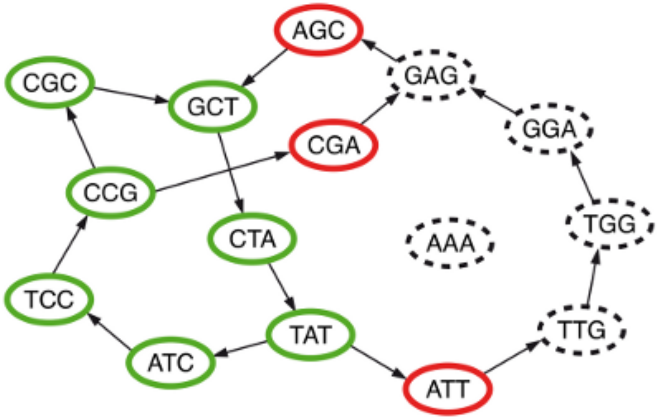
80%

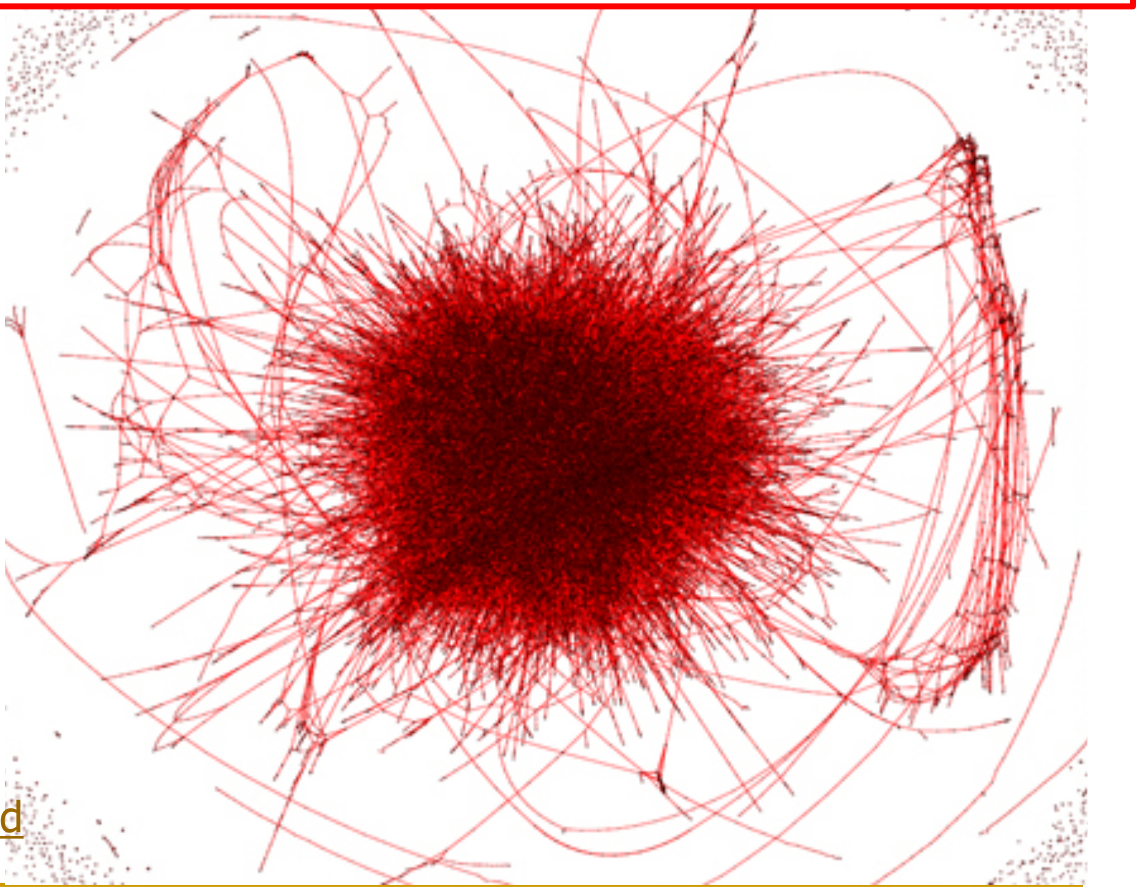
Human ~ Banana

50-60%

Metagenomics, genome assembly, de novo sequencing

Question #3: Given a bunch of short sequences, Can you identify the approximate species cluster for genomically unknown organisms (bacteria)?



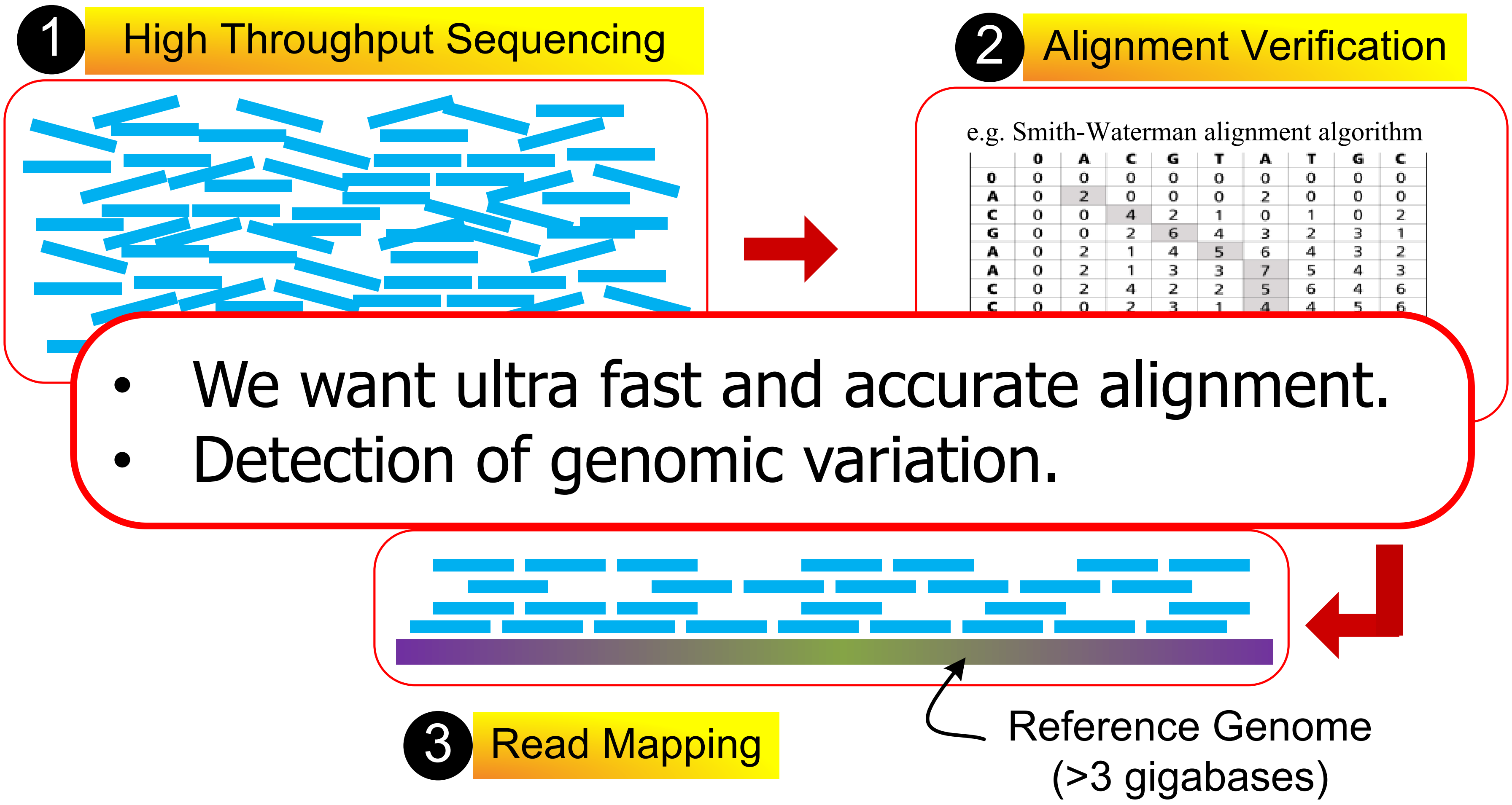


uncleaned de Bruijn graph

http://math.oregonstate.edu/~koslickd

ANALYZING THE PROBLEM

Read Mapping



1 High Throughput Sequencing

2 Alignment Verification

e.g. Smith-Waterman alignment algorithm

	0	A	C	G	T	A	T	G	C
0	0	0	0	0	0	0	0	0	0
A	0	2	0	0	0	2	0	0	0
C	0	0	4	2	1	0	1	0	2
G	0	0	2	6	4	3	2	3	1
A	0	2	1	4	5	6	4	3	2
A	0	2	1	3	3	7	5	4	3
C	0	2	4	2	2	5	6	4	6
C	0	0	2	3	1	4	4	5	6

- We want ultra fast and accurate alignment.
- Detection of genomic variation.

3 Read Mapping

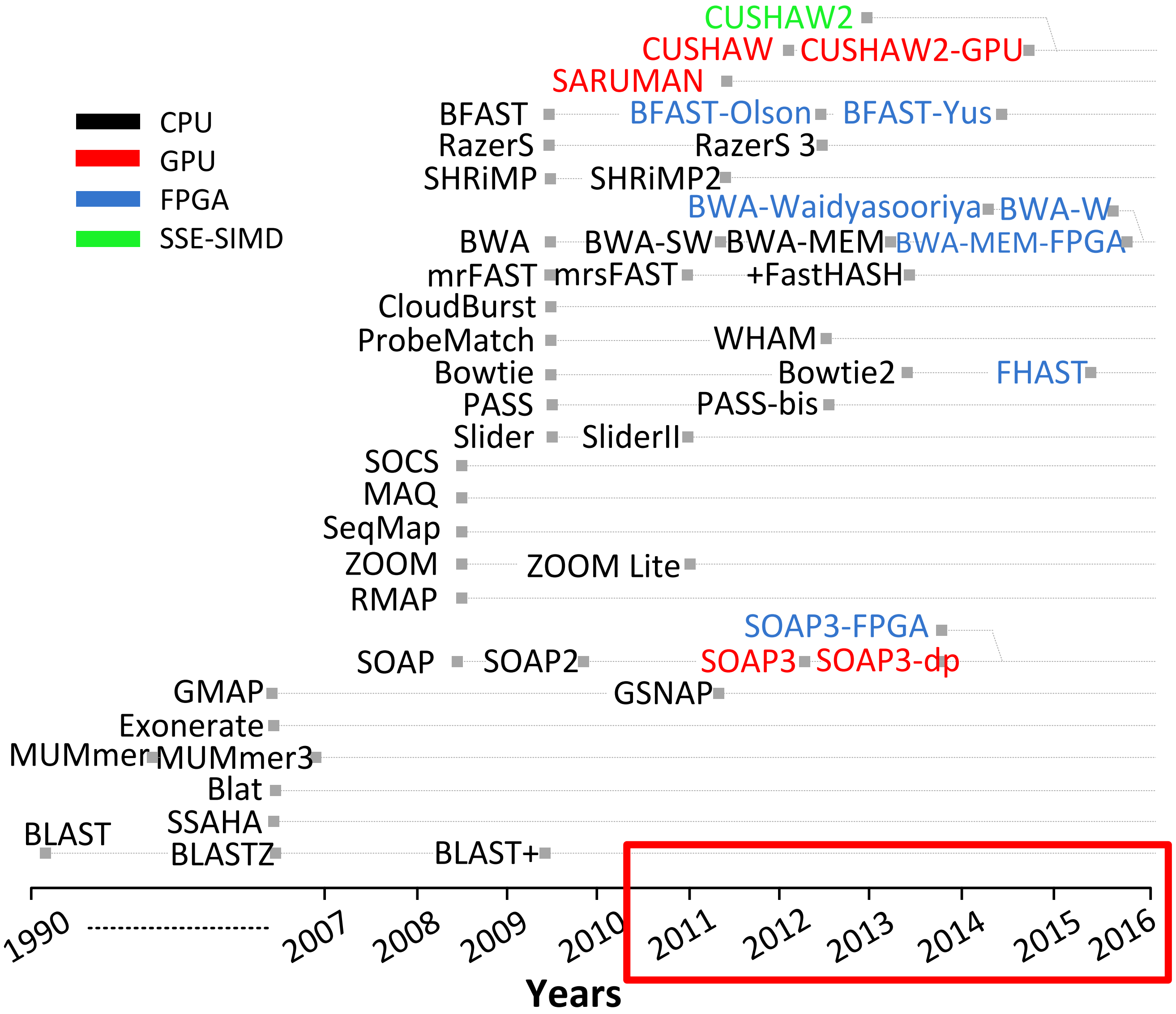
Reference Genome
(>3 gigabases)

Key Observations:

- Alignment Verification → **90%** of mapper's execution time.
- **>98%** of candidate locations have high dissimilarity with a given read.

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

Read Mappers Timeline



Filters, Alignment, and Mappers

Year	Purpose	Architecture	Platform	alignments#/1sec
2015	Filter	Shifted Hamming Distance	Intel SSE	3x faster 3583
	Alignment	Myers's bit-vector [45]	Intel SSE	409
	Alignment	Smith-Waterman [40]	Intel SSE	38
	Mapp			16
2014	M			13
	A			1
				60
2013				17
				4
			7)	15
	Mapper	BWT-FM	FPGA(Virtex6)	1092
2012	Mapper	BWT-FM	GPU	17
	Mapper	Hash-Based (BFAST)	FPGA(Virtex6)	35
	Alignment	Smith-Waterman	FPGA(Virtex4) GPU Cell BE CPU	 131 16 41

ideal filter → fast & accurate to compensate the computation overhead

Alignment performance for various state-of-the-art mappers and filters for 100 bp reads with at most 2% mismatch rate.