Genome-on-Diet

Taming Large-Scale Genomic Analyses via Sparsified Genomics

Mohammed Alser, PhD

@mealser

RECOMB 2023 – BIO-Arch

14 April 2023

ETH zürich

SAFARI

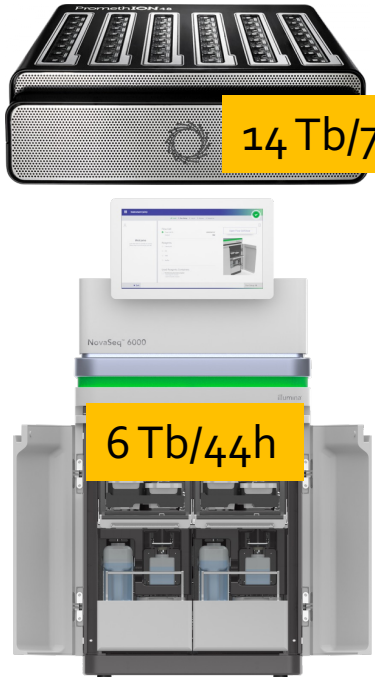
SAFARI Research Group

İSTANBUL 2023 RECOMB

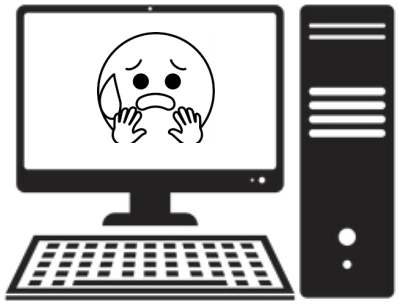
Genome Sequencing is Rapidly Growing

Specialized Machine
for Sequencing

Genome Sequencing is Rapidly Growing

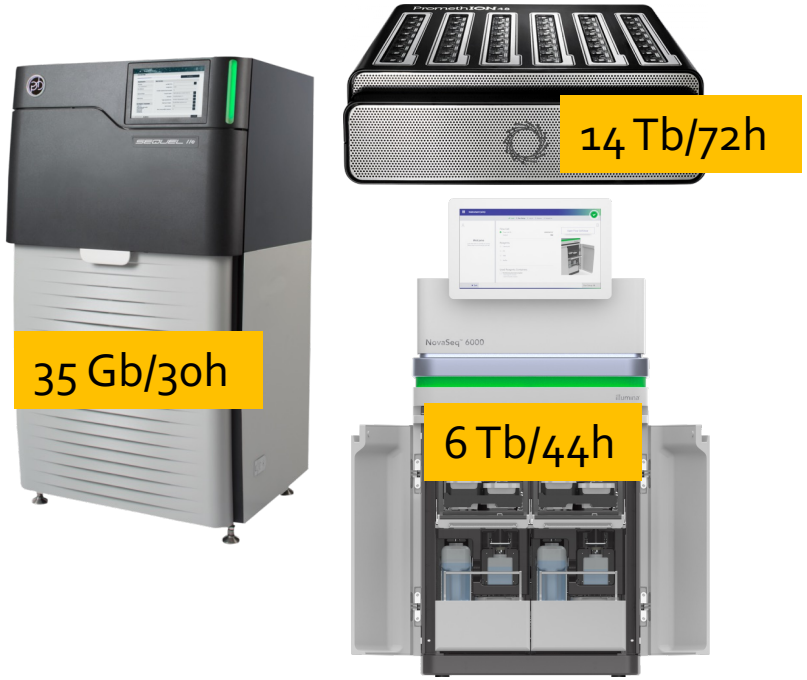


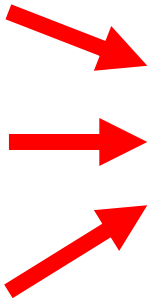
Specialized Machine
for Sequencing



General-Purpose Machine
for Analysis

Lack of Specialized Compute Capability





Specialized Machine for Sequencing

General-Purpose Machine for Analysis

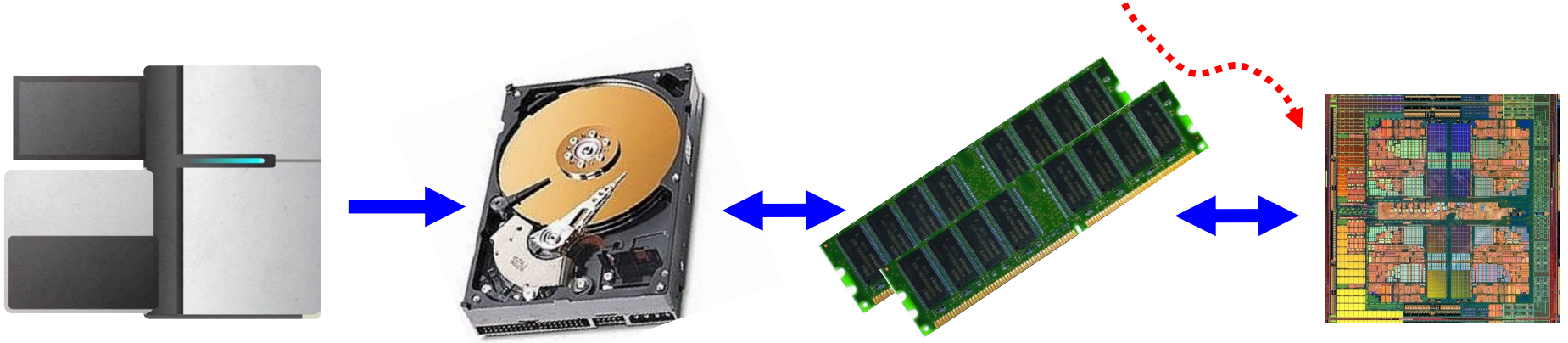
FAST

SLOW

Improving Processing via Accelerators

Specialized Genomic Accelerators (GPU, FPGA)

- **Scrooge [Bioinformatics 2023]**
- **RUBICON [arXiv 2022]**
- **GateKeeper-GPU [IPDPSW 2021]**
- **SneakySnake [Bioinformatics'20]**
- **Shouji [Bioinformatics 2019]**
- **MAGNET [AACBB 2018]**
- **GateKeeper [Bioinformatics 2017]**



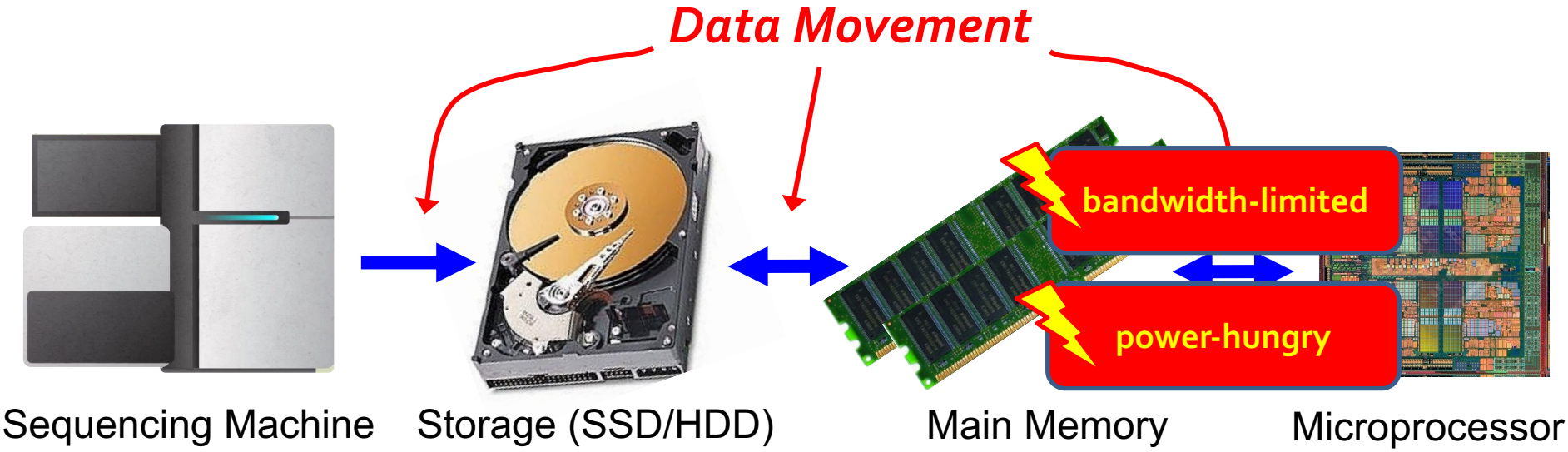
Sequencing Machine | Storage (SSD/HDD) | Main Memory | Microprocessor

Data Movement Bottleneck

Data movement is a major bottleneck
in modern computer architectures

Over 60% of the total system energy is spent on data movement

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



Improving Processing via Paradigm Shift

Near-memory/In-memory Genomic Accelerators

AIM [Bioinformatics 2023]

SeGraM [ISCA 2022]

Specialized Genomic Accelerators (GPU, FPGA)

Scrooge [Bioinformatics 2023]

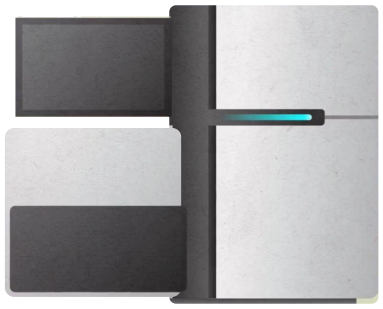
RUBICON [arXiv 2022]

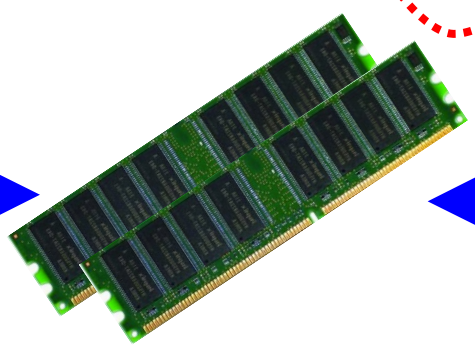
GateKeeper [Bioinformatics 2017]

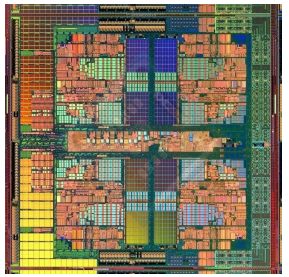
Improving **performance** and **energy efficiency** by 1-3 orders of magnitude

In-Storage Sequence Alignment

GenStore [ASPLOS 2022]







Sequencing Machine | Storage (SSD/HDD) | Main Memory | Microprocessor

Our Goal

To further reduce the execution time

and memory/storage footprint of genomic

analyses via **sparsified genomics**

Sparsifying Genomic Data

ACCCTAACCCTAACCCTAACCCTAACCCTAA

Exact Match

ACCCTAACCCTAACCCTAACCCTAACCCTAA

Sparsifying Genomic Data

ACCCTAACCCTAACCCTAACCCTAACCCTAA — N Bytes

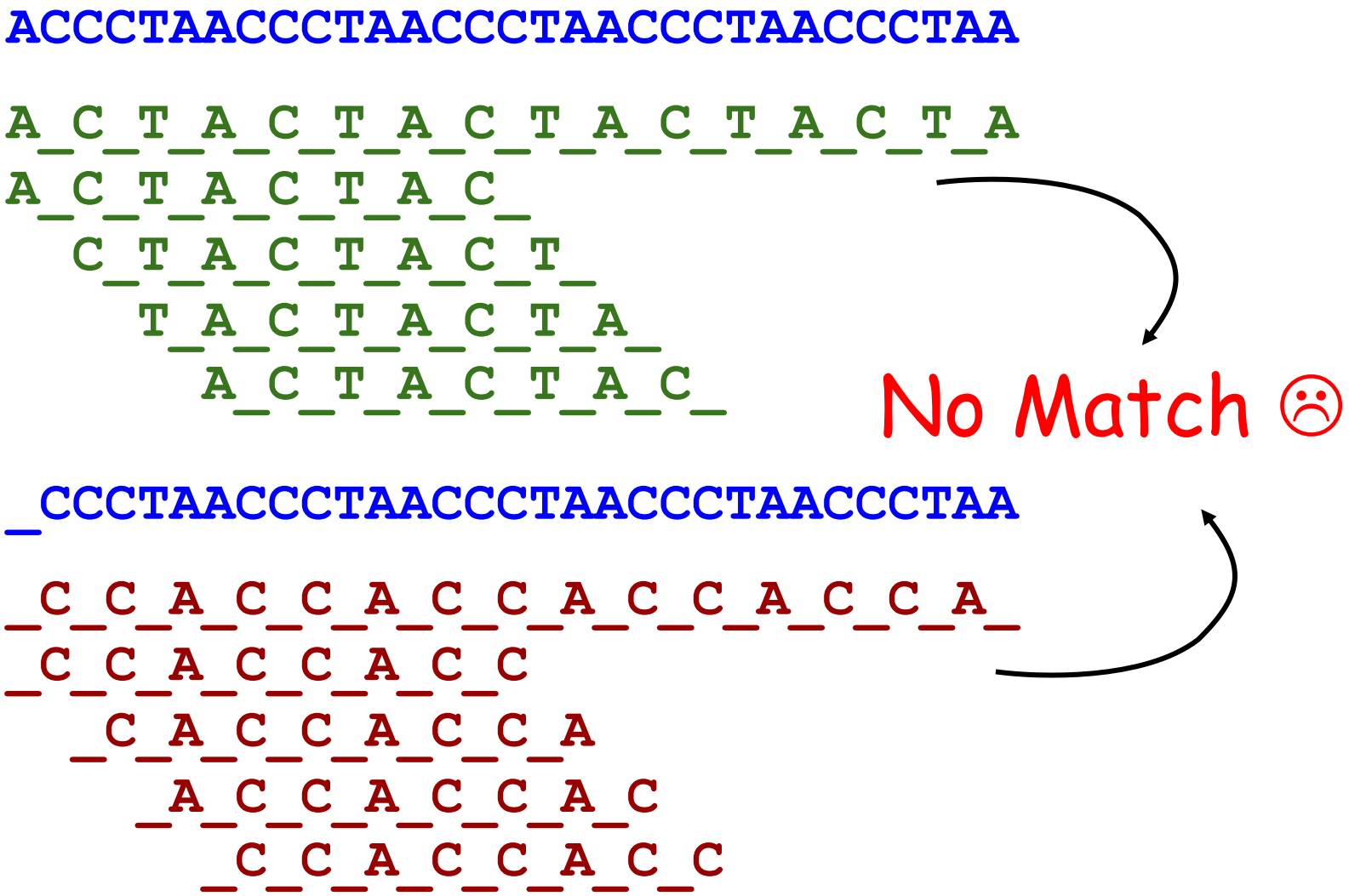
A_C_T_A_C_T_A_C_T_A_C_T_A_C_T_A — N/2 Bytes

Still Exact Match

ACCCTAACCCTAACCCTAACCCTAACCCTAA

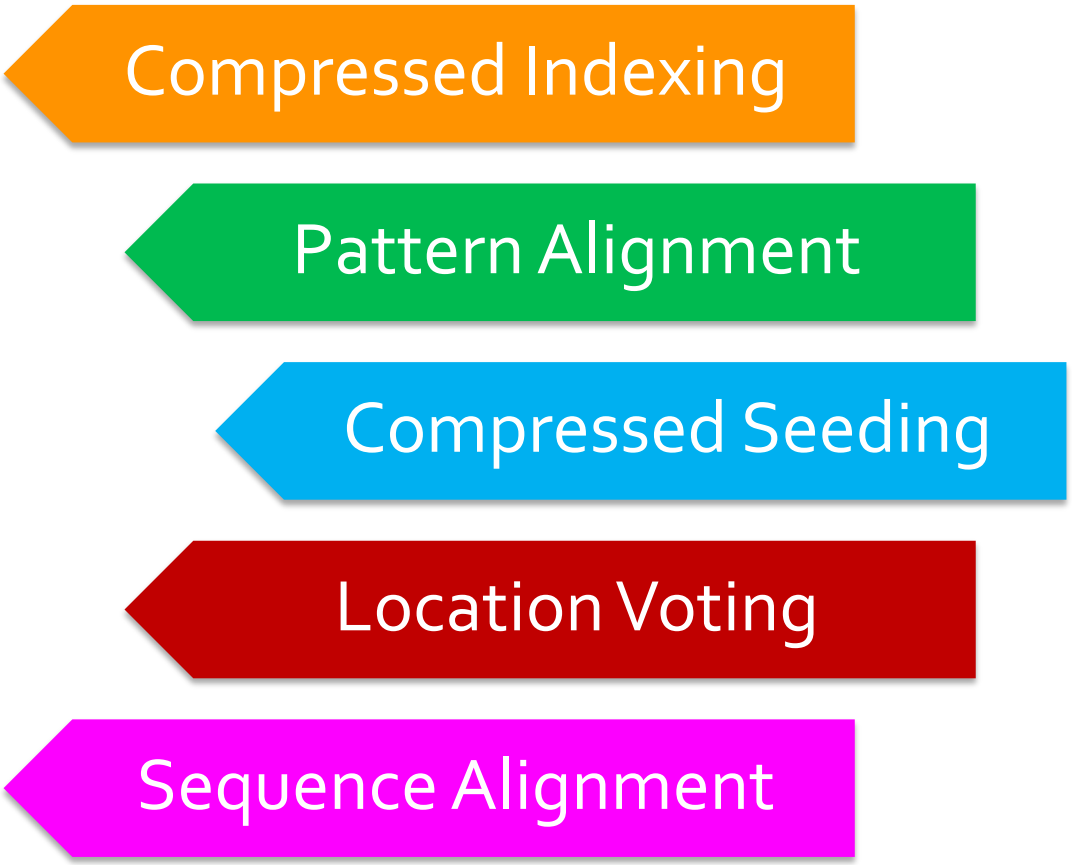
A_C_T_A_C_T_A_C_T_A_C_T_A_C_T_A

Sparsifying Genomic Data Is Challenging



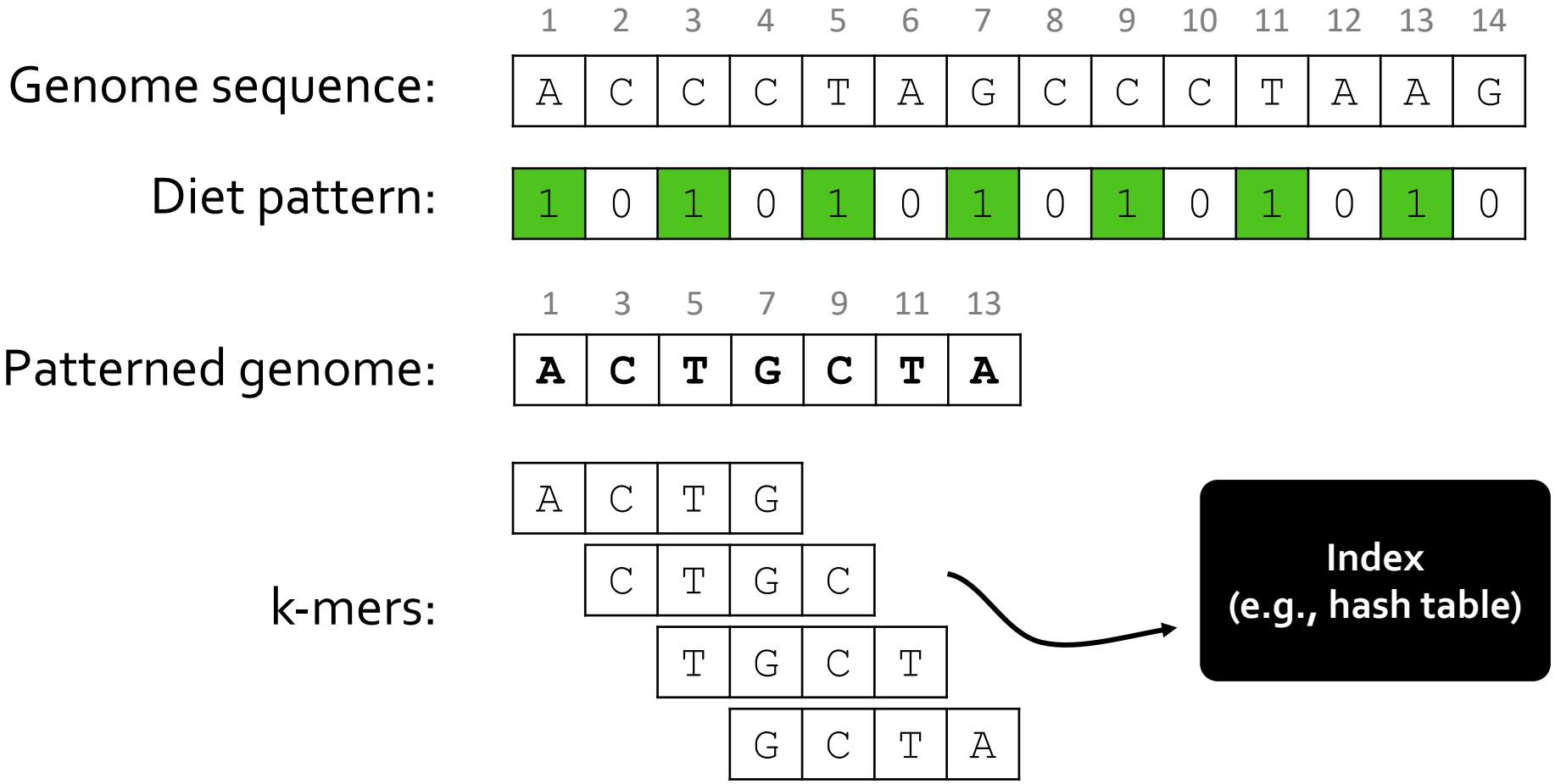
How It Works?

Genome-on-Diet Steps



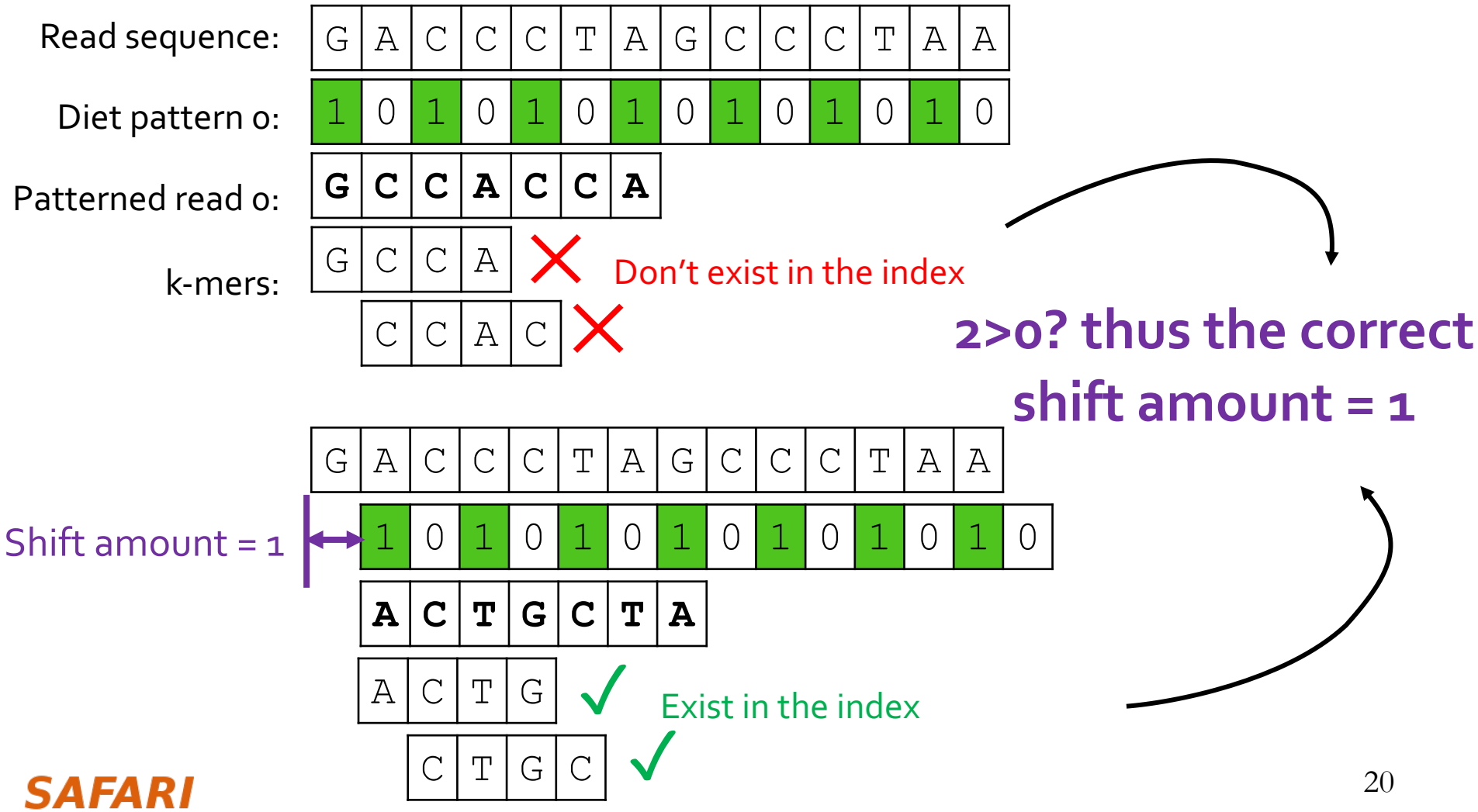
Step 1: Compressed Indexing

- We use a user-defined binary pattern to identify the location and number of the to-be-dropped bases.



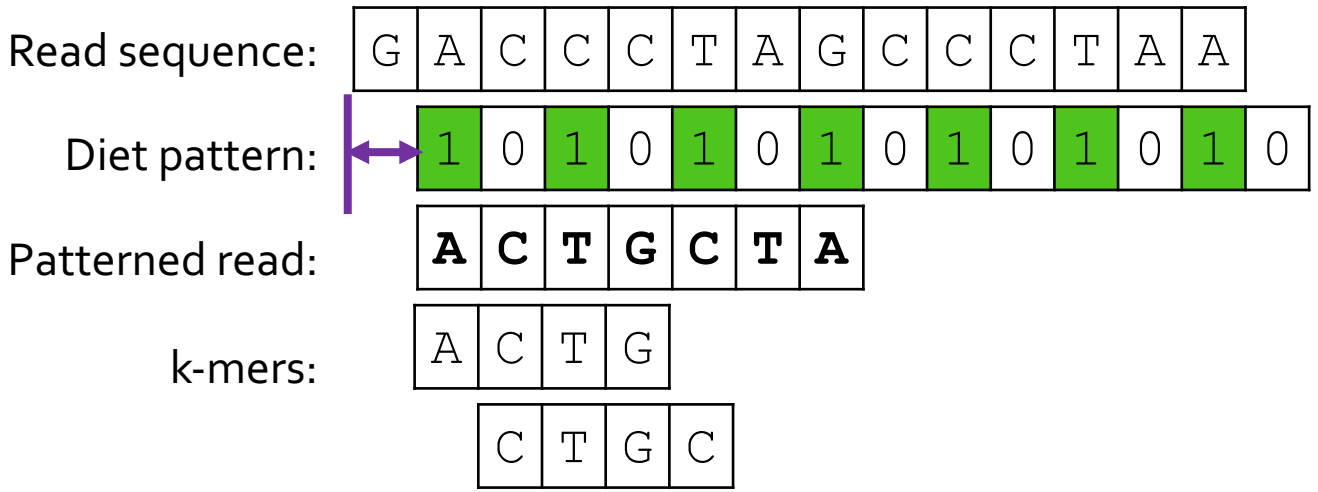
Step 2: Pattern Alignment

- Deciding where in the read to apply the pattern essential for the correctness of Genome-on-Diet



Step 3: Compressed Seeding

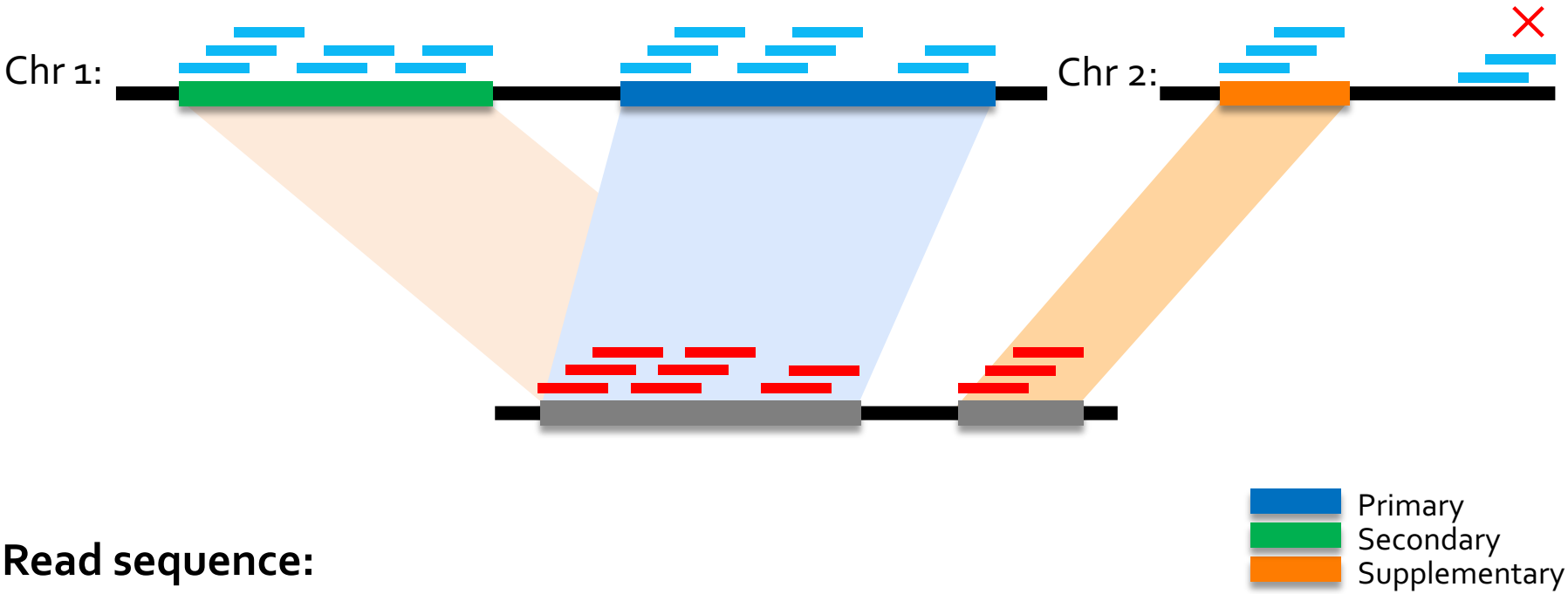
- Use the calculated shift amount to correctly extract seeds from the read sequence.
- Now both the reference genome and the read are half in length and their seeds can be still correctly matched



Step 4: Location Voting

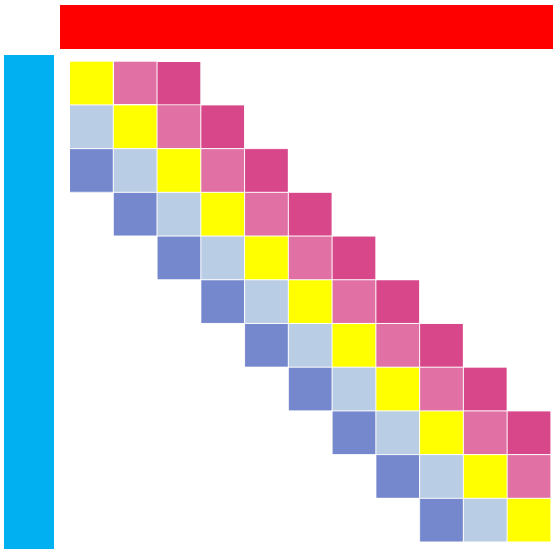
- Seeds are sparsified and thus cannot be directly chained, instead to detect mapping locations we use the number of matching seeds in a region

Genome sequence:



Read sequence:

Step 5: Sequence Alignment



Dynamic programming matrix

nature computational science

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › nature computational science › brief communications › article

Brief Communication | Published: 28 February 2022

Accelerating minimap2 for long-read sequencing applications on modern CPUs

Saurabh Kalikar ✉, Chirag Jain ✉, Md Vasimuddin ✉ & Sanchit Misra ✉

Nature Computational Science **2**, 78–83 (2022) | Cite this article

BMC Bioinformatics

Home About Articles Submission Guidelines Join The Board

Methodology | Open Access | Published: 19 February 2018

Introducing difference recurrence relations for faster semi-global alignment of long sequences

Hajime Suzuki & Masahiro Kasahara ✉

BMC Bioinformatics **19**, Article number: 45 (2018) | Cite this article

CPU Implementation

Genome-on-Diet is implemented on top of minimiap2
(2.24-r1122 version as of 11 November 2022)

Introducing Four Optimization Strategies

- Accelerating Indexing & Seeding with **SIMD Instructions**
  - Calculating 8 (512/(32*2)) overlapping k-mers along with their hash values in parallel
- **Sorting** Seed Locations
  - Merge sort instead of Radix and Heap sort algorithms
- **Rescuing** Mapping Location
  - Based on two voting thresholds
- Handling **Exactly-Matching** Short Reads
  - It is observed that 80% of short reads usually exactly match to the reference genome

Applications of Sparsified Genomics

- Applications that compare sequences for similarity
  - Genome similarity & genomic distance
  - Prealignment filtering
  - Containment search

They may or may not require building index

- Applications that generate huge index
  - Taxonomic profiling
  - Pangenomics

Index can be up to 21.25x larger in size than a single (2-bit encoded) indexed genome

- Applications that require building index during the analysis
  - Read mapping for many assembly ve
  - Identifying *de novo* variations by con members
  - Identifying somatic variations by con healthy and tumor cells of the same patient

Indexing and seeding time account for

97% Taxonomic Profiling

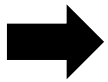
10%-27% Read Mapping

- And many more …

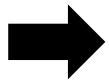
Genome-on-Diet vs. Spaced Seeding?

Spaced Seeding:

```
ACCCTAACCCTAACCCTAACCCTAA..
ACCCTAACCC
 CCCTAACCCT
  CCTAACCCTA
   CTAACCCTAA
    TAACCCTAAC
     AACCCTAACC
```



1000101001



```
A___T_A__C
 C___A_C__T
  C___A_C__A
   C___C_C__A
    T___C_T__C
     A___C_A__C
```

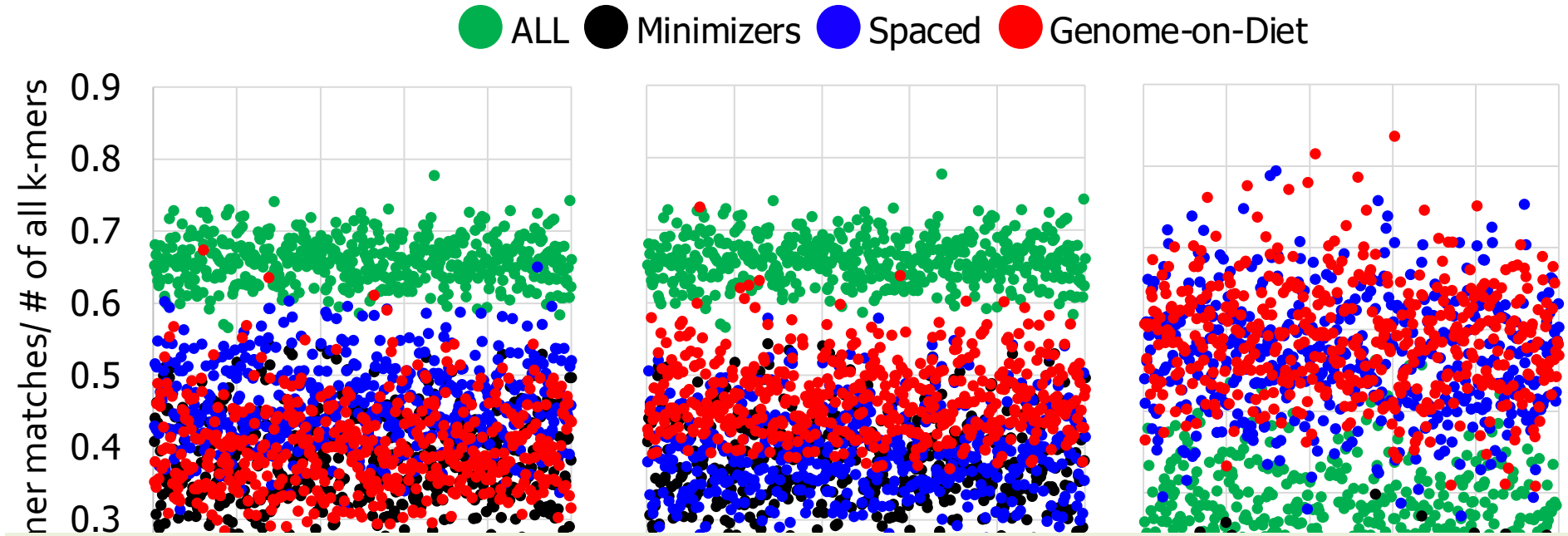
Increased execution time
No effect on peak memory footprint
No effect on the number of seeds
All seeds have the same pattern

Genome-on-Diet:

```
ACCCTAACCCTAACCCTAACCCTAA..
10001010011000101001000101001
A___T_A__CT___C_T__CC___A..
A___T_A__C
   T_A__CT
     A__CT___C
```

Reduced execution time
Reduced peak memory footprint
Reduced number of seeds
Each seed may have its own pattern

Increasing Common k-mers Rate



Genome-on-Diet provides the same or higher sensitivity compared to spaced seeding, while Genome-on-Diet is always faster and more memory efficient

Evaluation Results

Great Benefits by Sparsified Genomics

Read Mapping

- Genome-on-Diet is 1.13-6.28x faster and has 2.1x smaller memory footprint, and 2x smaller index size compared to minimap2, for performing read mapping.

Containment Search

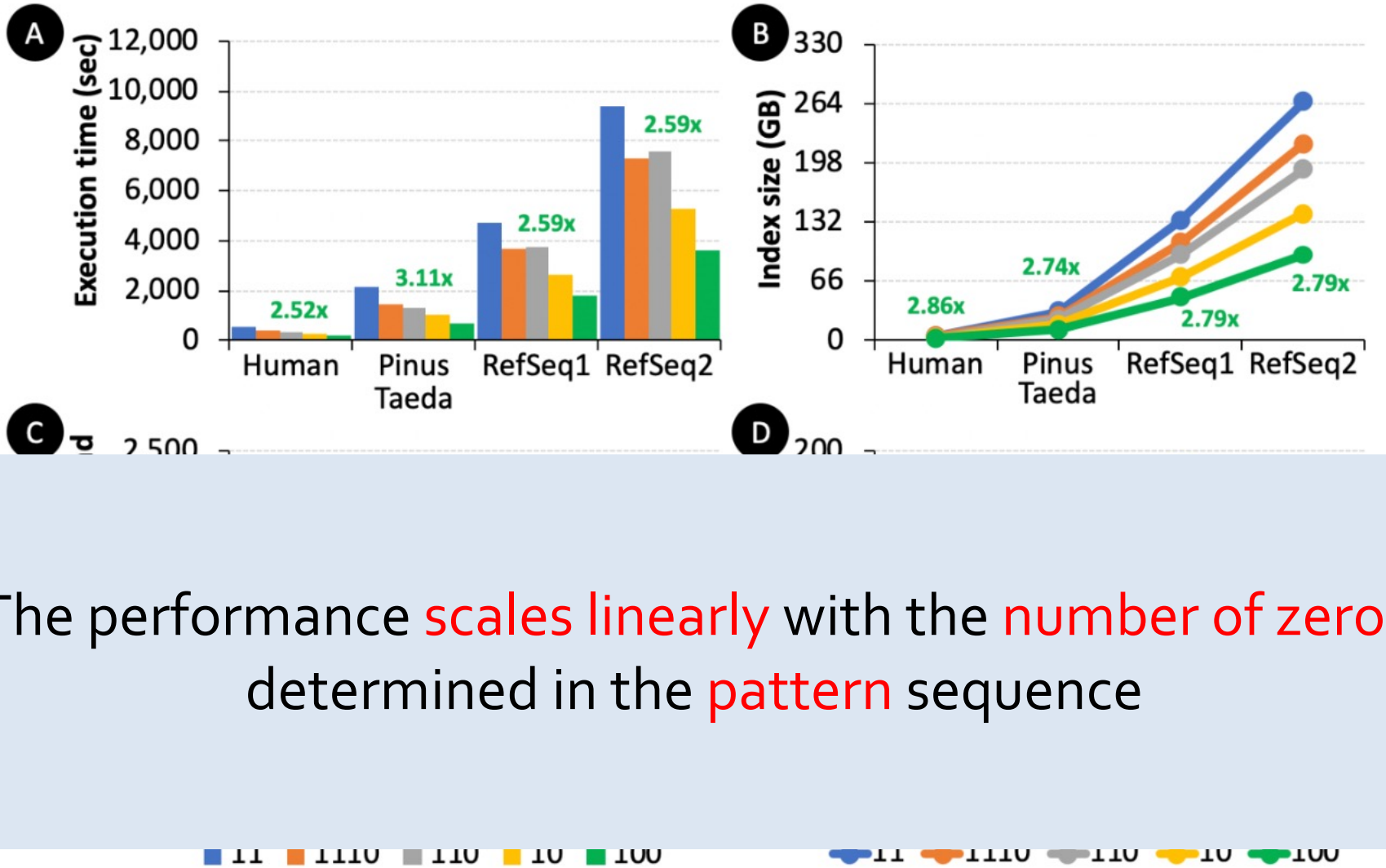
- Genome-on-Diet is 72.7-75.88x faster and 723.3x more storage-efficient than KMC3 combined with CMash, for performing containment search.

Metagenomic Profiling

- Genome-on-Diet is 54.15-61.88x faster and 720x more storage-efficient than Metalign, for performing taxonomic profiling of metagenomic samples.

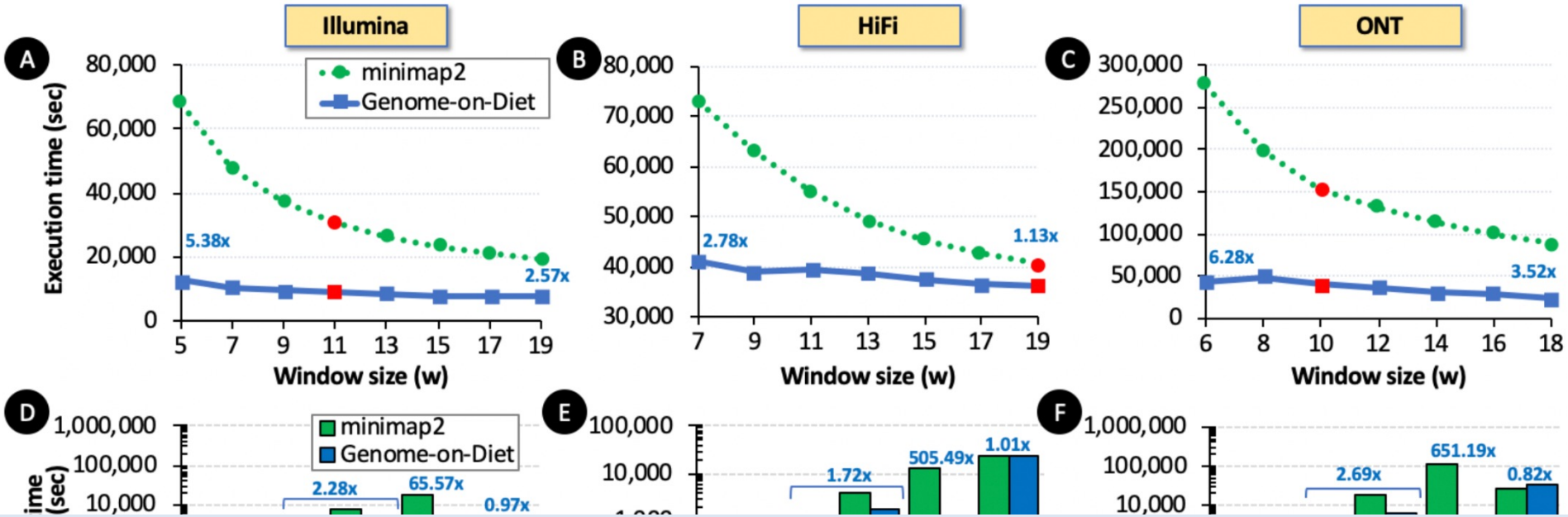
Effect of Using Different Patterns

For performing both containment indexing and k-mer intersection



The performance scales linearly with the number of zeros determined in the pattern sequence

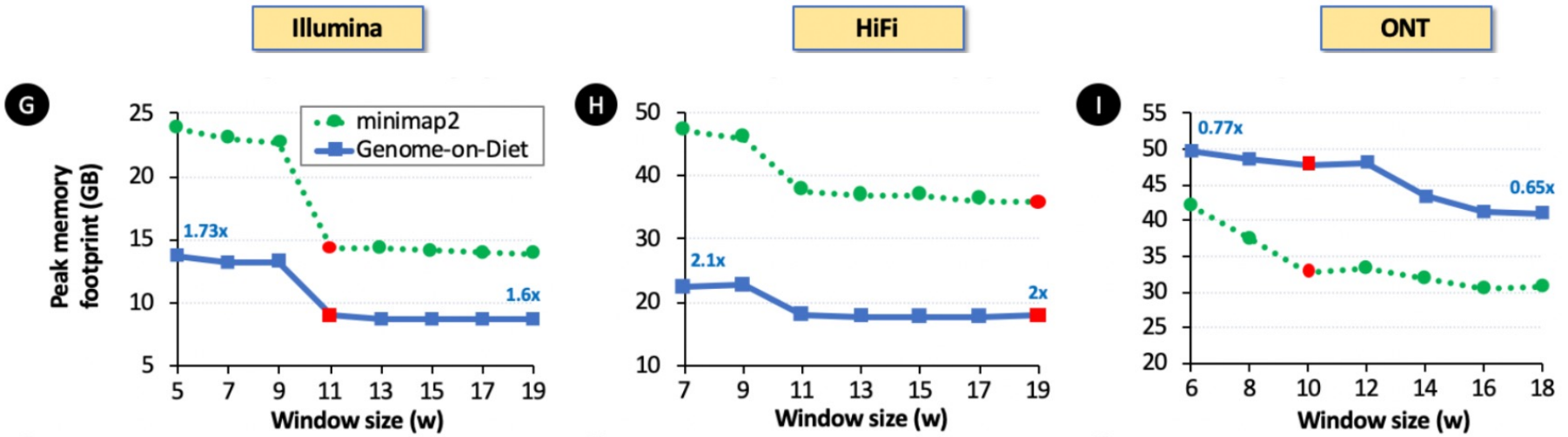
See Our Paper for Many More Analyses and Results



Genome-on-Diet performance is not affected by the value of minimizer window.

Location voting step is much faster than seed chaining

See Our Paper for Many More Analyses and Results



J

		G-on-Diet	minimap2
Indels / SNPs	Correctly detected	**19,105**	18,178
	Incorrectly detected	787	**14**
	Missed	**2,119**	3,046
Execution time (sec)		**9,178.78**	31,358.75
Peak memory (GB)		**8.62**	14.38

K

		G-on-Diet	minimap2
Indels / SNPs	Correctly detected	**17,795**	17,297
	Incorrectly detected	2,028	**938**
	Missed	**3,429**	3,927
SVs	Correctly detected	**193**	185
	Incorrectly detected	57	**11**
	Missed	**23**	31
Execution time (sec)		**37,787.72**	41,016.12

L

		G-on-Diet	minimap2
Indels / SNPs	Correctly detected	**13,432**	5,717
	Incorrectly detected	5,620	**3,862**
	Missed	**7,792**	15,507
SVs	Correctly detected	**147**	41
	Incorrectly detected	**61**	226
	Missed	**69**	175
Execution time (sec)		**40,348.88**	161,545.61

Genome-on-Diet leads to the detection of a higher number of SNPs, indels, and SVs compared to minimap2

Preprint and Source Code

Computer Science > Data Structures and Algorithms

[Submitted on 15 Nov 2022 (v1), last revised 18 Jan 2023 (this version, v2)]

Genome-on-Diet: Taming Large-Scale Genomic Analyses via Sparsified Genomics

Mohammed Alser, Julien Eudine, Onur Mutlu

https://arxiv.org/abs/2211.08157

https://github.com/CMU-SAFARI/Genome-on-Diet

Contributors

Mohammed Alser

Julien Eudine

Onur Mutlu

https://safari.ethz.ch

Genome-on-Diet

Taming Large-Scale Genomic Analyses via Sparsified Genomics

Mohammed Alser, PhD

@mealser

RECOMB 2023 – BIO-Arch

14 April 2023

ETH zürich

SAFARI

SAFARI Research Group

İSTANBUL 2023 RECOMB