# CMSC 829C: Algorithms, AI, and Hardware for Biological Data

## Lecture 4: Sequence Analysis in Genomics

Fall 2026

September 10, 2026

**Can Firtina** | canfirtina@gmail.com | cs.umd.edu/~firtina

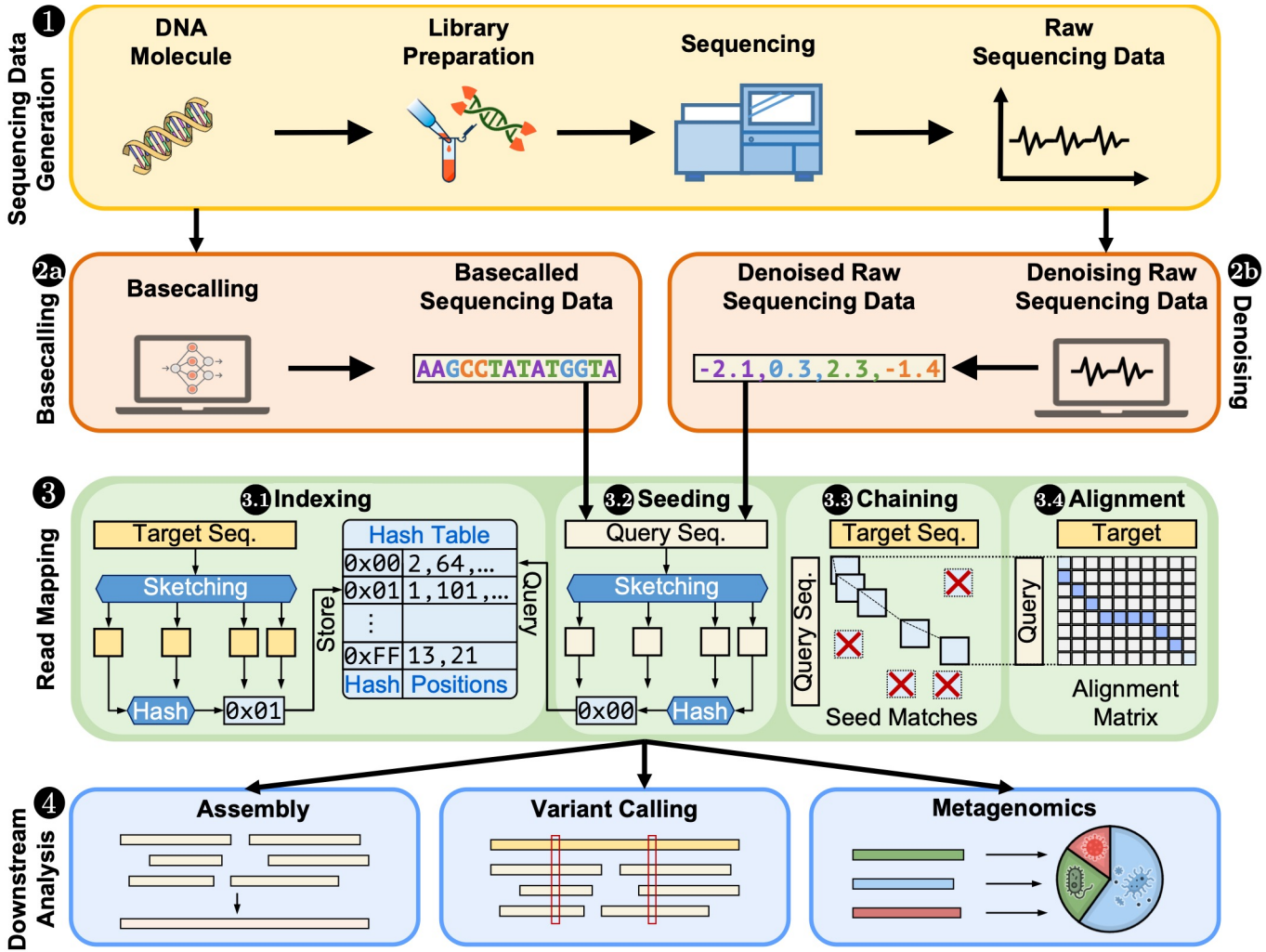
Assistant Professor of Computer Science at UMD

UNIVERSITY OF MARYLAND

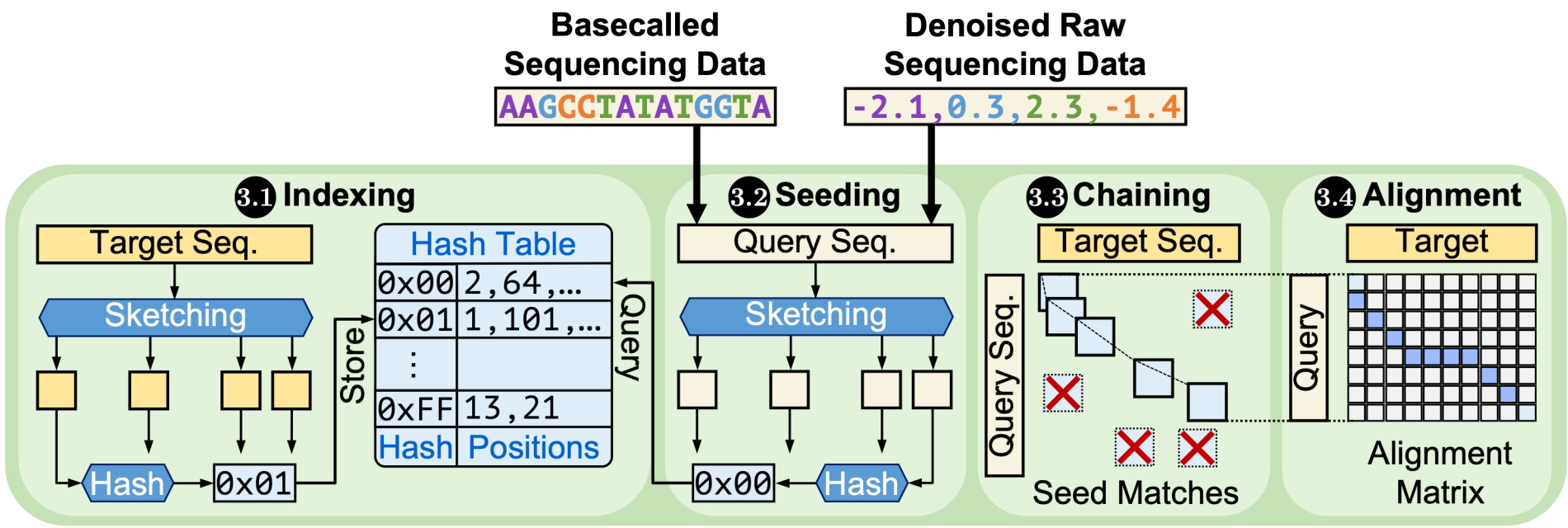
# Agenda

- Read mapping
  - Indexing, seeding, chaining, alignment
- Pangenome graphs: representing and searching alternatives

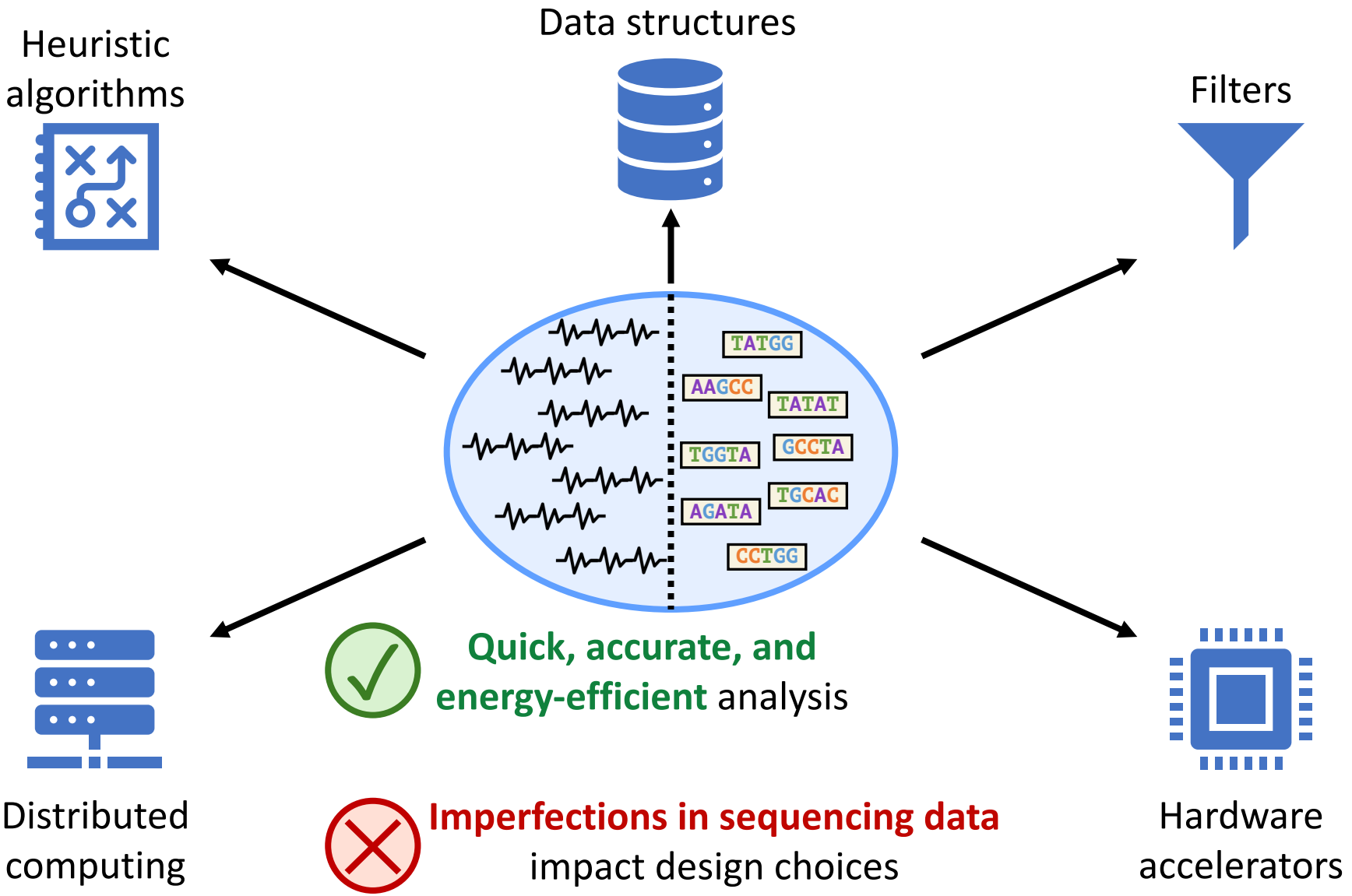
# Recall: Genome Analysis Pipeline



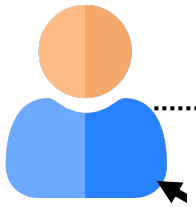
# Read Mapping



# Sequencing Data Analysis



# Recall: Challenges in Genomic Data Analysis



**Human (reference) genome:** 3.2 billion bps (haploid)

CGAAATGCC...AGATTAAACGCT...TGCCCTAA...GAATGGCGT

?

CGAAATGCC...AGATTAAATGGT...TGCCCTAA...GAATGGCGT

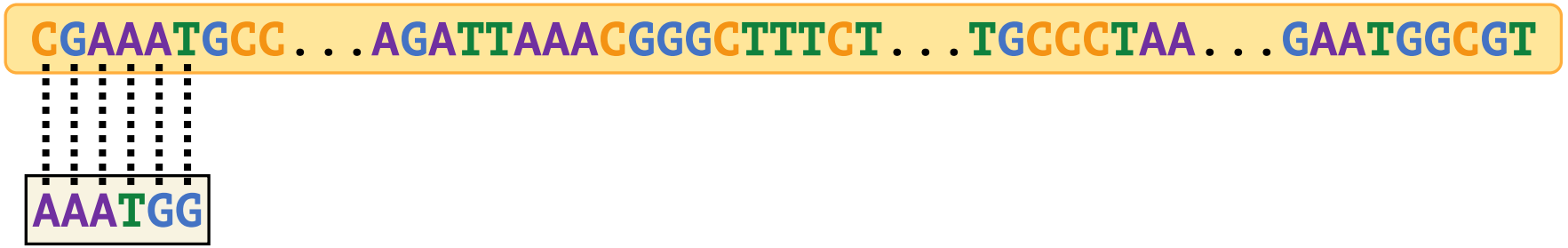
**Entire genomic sequence of an organism**

- **Small pieces (reads)**
- **Unknown origins**
- **Non-identical reference genome**
- **Large volume of data to analyze**

→ **Dealing with massive and noisy datasets (TBs to PBs)**

# Recall: Solving the Puzzle in a Naïve Way

**Human (Reference) Genome:** 3.2 Billion bps (haploid)




**Pairwise Approximate String Matching (Alignment)**
$O(n^2)$

→

**Entire Genome**
$O(n^2m)$

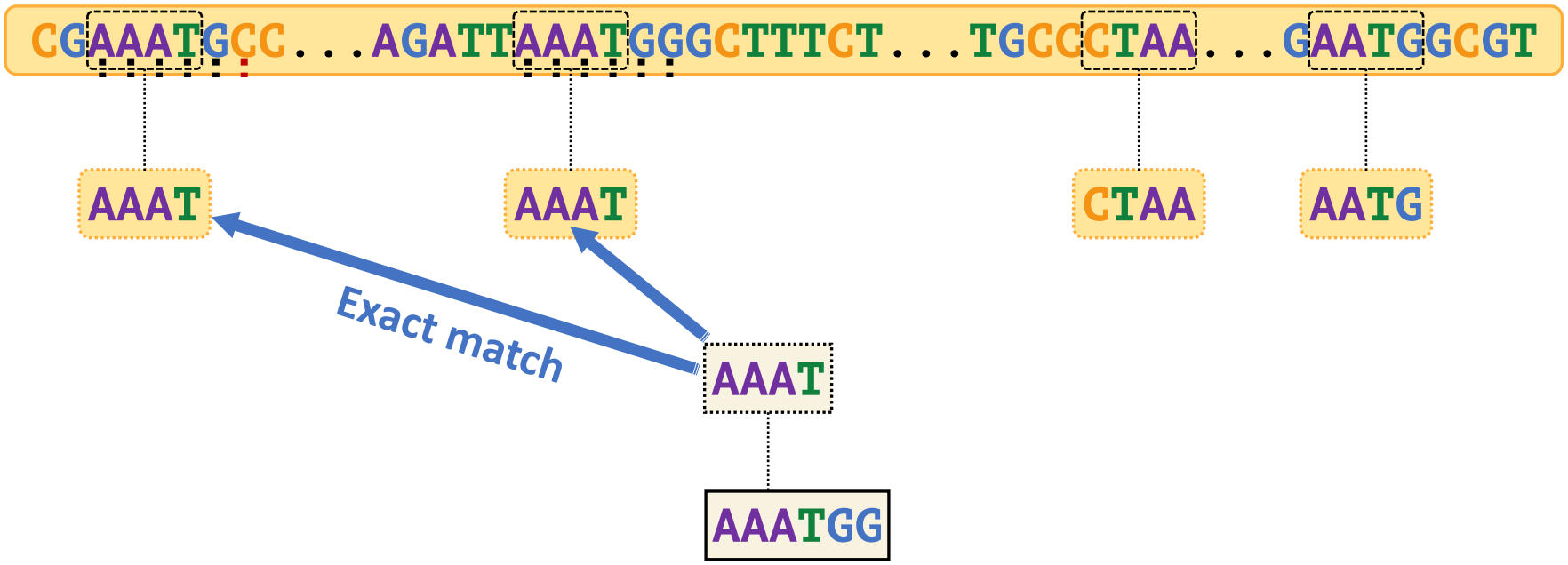
→

**For All Reads**
$O(n^2mk)$
$\cong O(10^{20})$

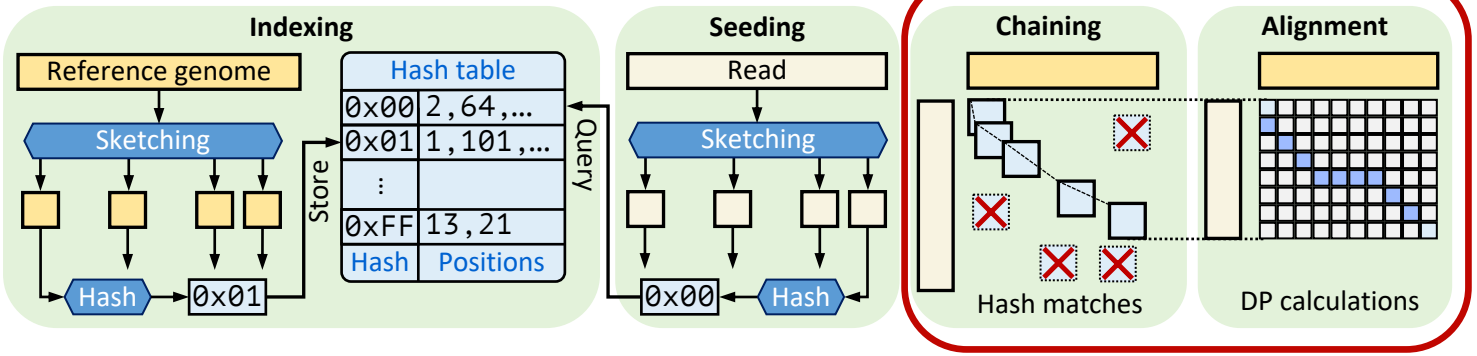
**≅Years**

# Solving the Puzzle: The Practical Way

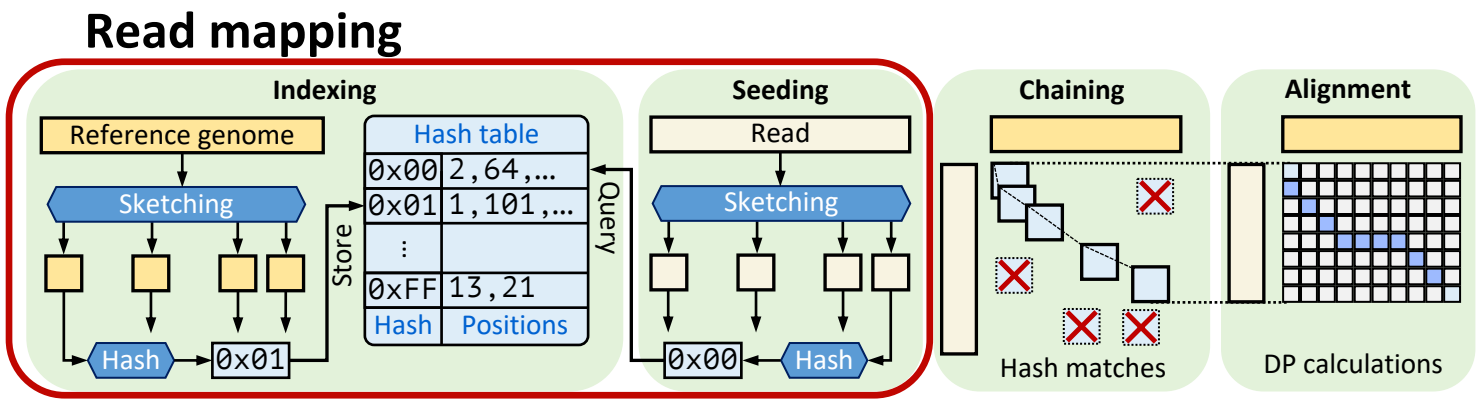
**Goal:** quickly and accurately reduce the search space



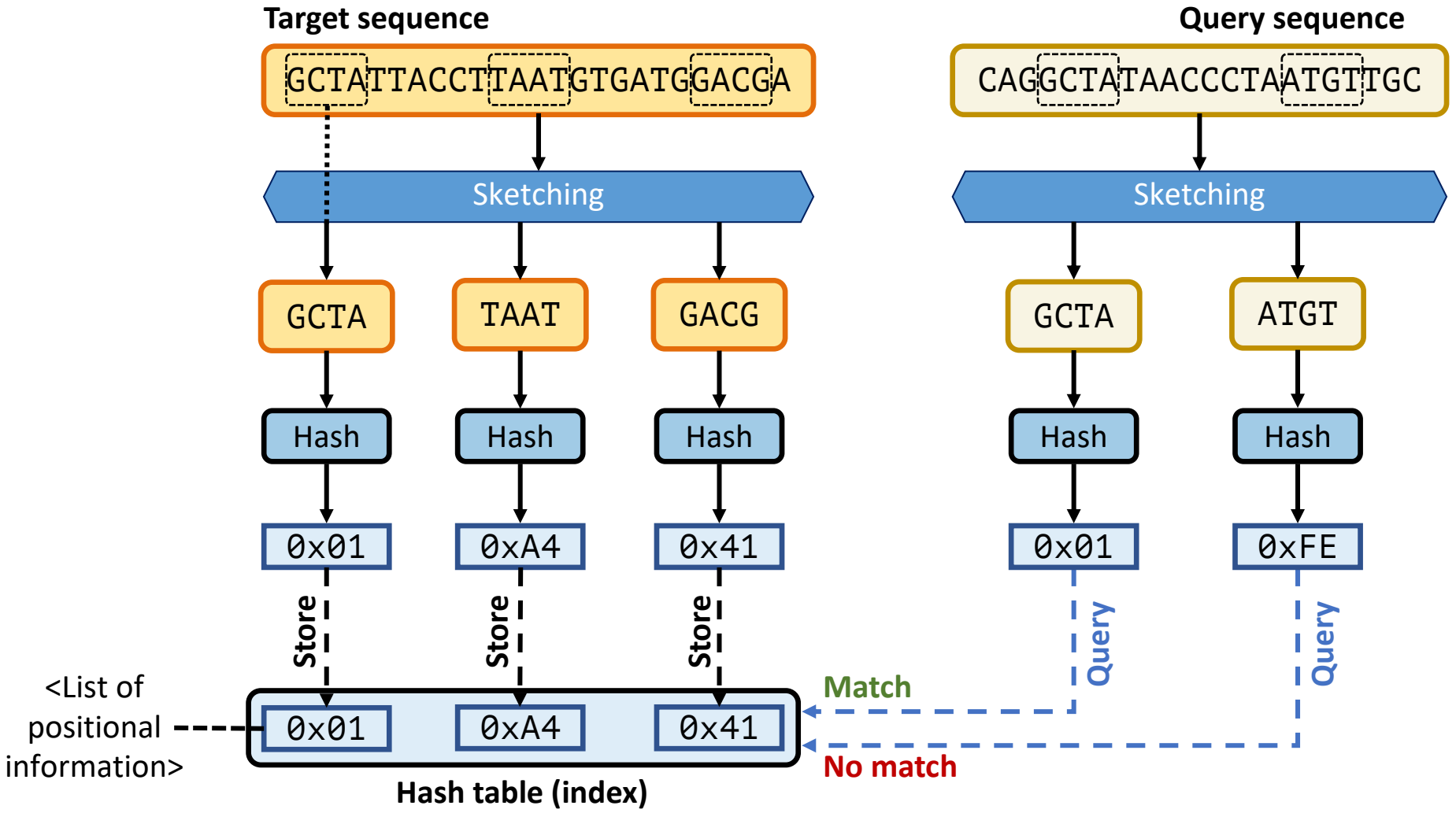

**Read mapping**



# Steps in Read Mapping – Indexing and Seeding

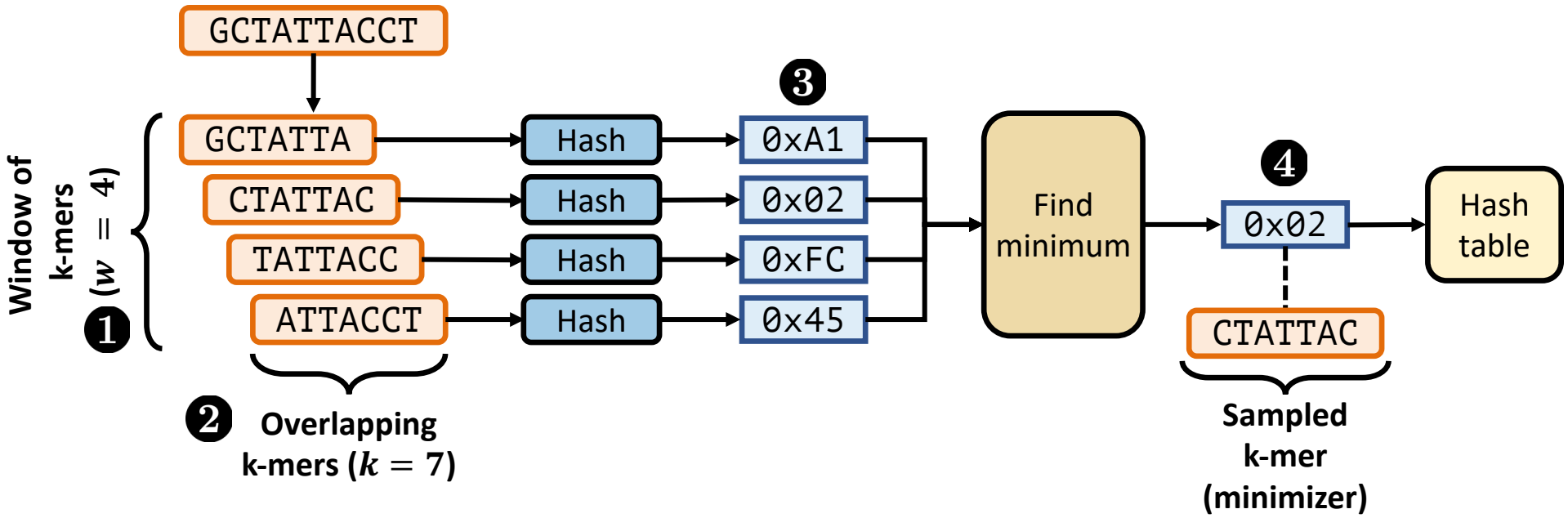


# Hash-Based Sketching and Seed Matching



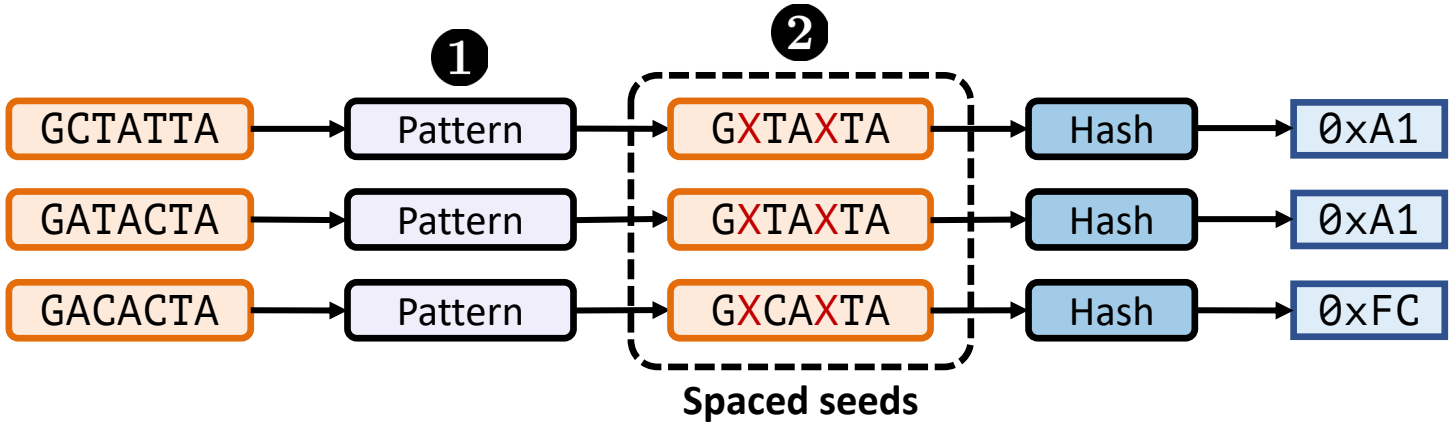
# Minimizer Sketching

- Sketching Goal: Selectively use *some* subsequences for similarity estimation
  - Trade-offs: (Usually) reduced storage and computation requirements vs. sensitivity



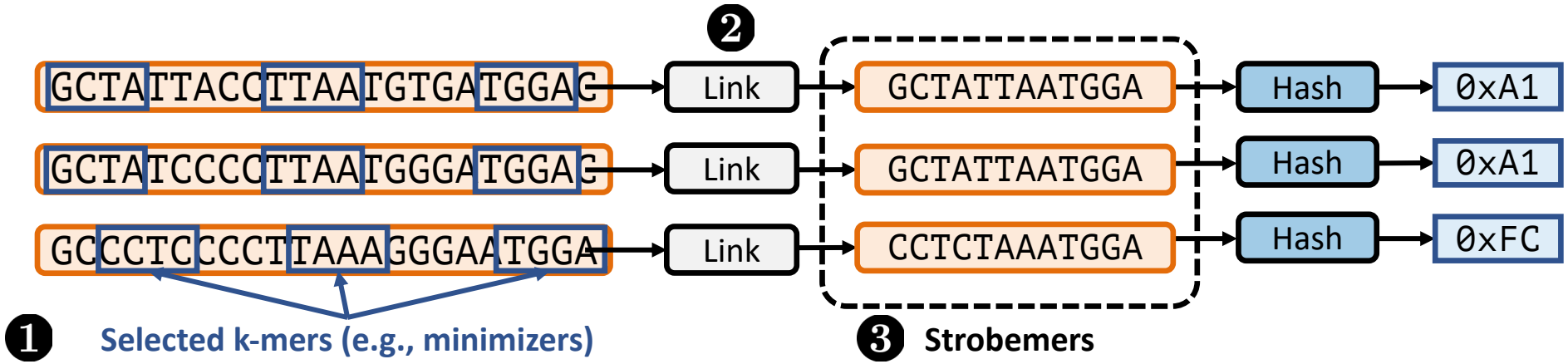
# Spaced Seeding

- Can we tolerate certain mismatches?



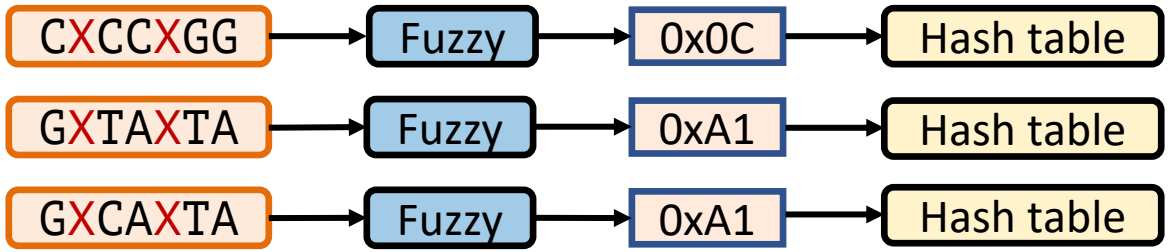
# Strobemer Sketches

- Can we tolerate insertions and deletions between selected subsequences?



# Fuzzy Seeding

- Can we tolerate *arbitrary* mismatches?



# BLEND: Fuzzy Seed Matching [NARGAB '23]

- **BLEND: a fast, memory-efficient and accurate mechanism to find fuzzy seed matches in genome analysis**
  Can Firtina, Jisung Park, Mohammed Alser, Jeremie S. Kim, Damla Senol Cali, Taha Shahroodi, Nika Mansouri Ghiasi, Gagandeep Singh, Konstantinos Kanellopoulos, Can Alkan, Onur Mutlu
  NAR Genomics and Bioinformatics (**NARGAB**), March 2023
  [PDF] [Code] [DOI]
  Preprint (bioRxiv) [Link] [PDF]
  RECOMB 2023 Talk [Video] [Slides (pptx)] [Slides (pdf)]
  Social media thread [Twitter (X), publication] [Twitter (X), preprint]


Volume 5, Issue 1
March 2023

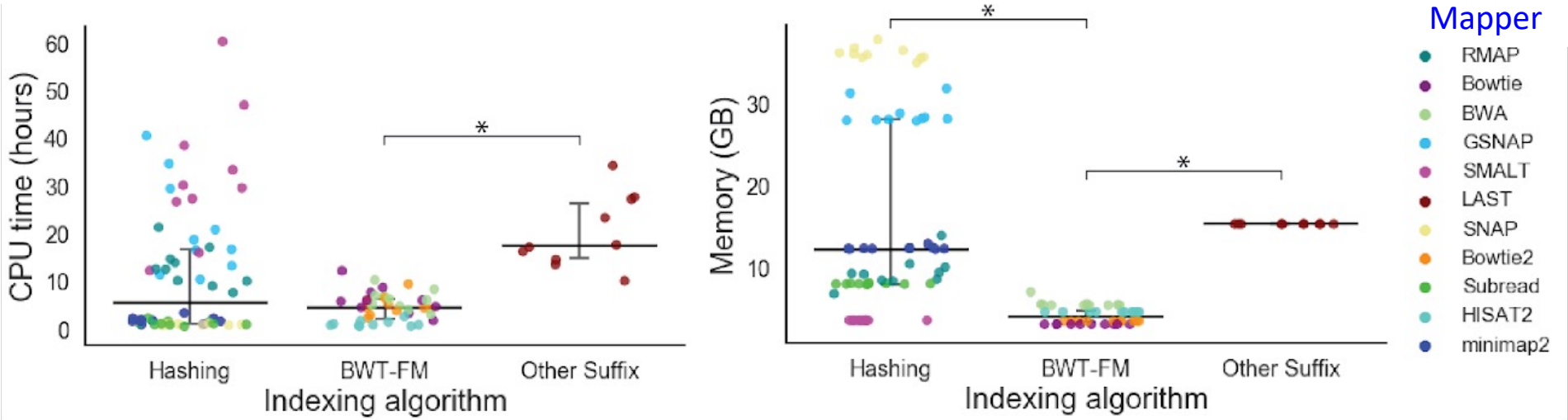
JOURNAL ARTICLE

## BLEND: a fast, memory-efficient and accurate mechanism to find fuzzy seed matches in genome analysis

Can Firtina ✉, Jisung Park, Mohammed Alser, Jeremie S Kim, Damla Senol Cali, Taha Shahroodi, Nika Mansouri Ghiasi, Gagandeep Singh, Konstantinos Kanellopoulos, Can Alkan, Onur Mutlu ✉

*NAR Genomics and Bioinformatics*, Volume 5, Issue 1, March 2023, lqad004,

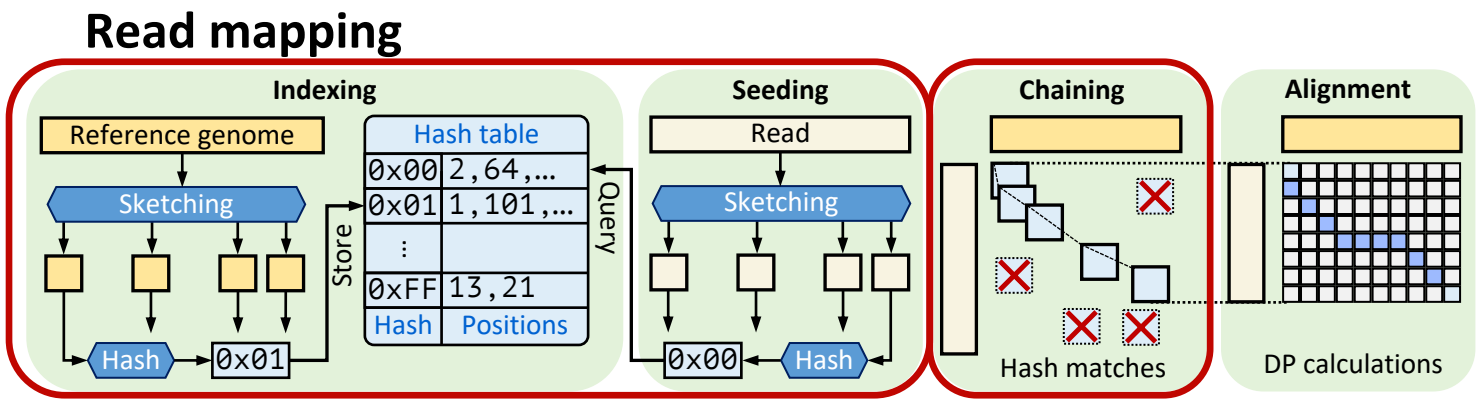
# Indexing Cost: A Historical Comparison



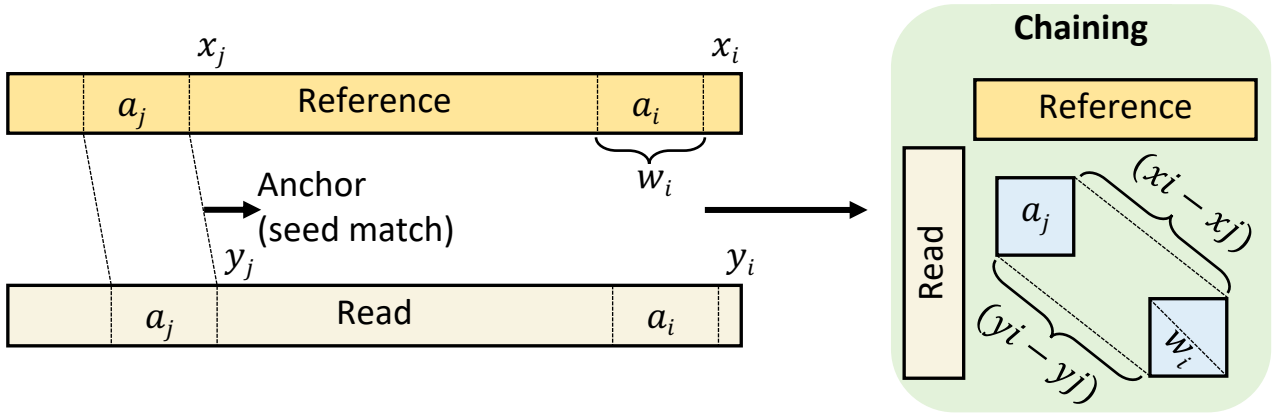



Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021

# Steps in Read Mapping – Chaining

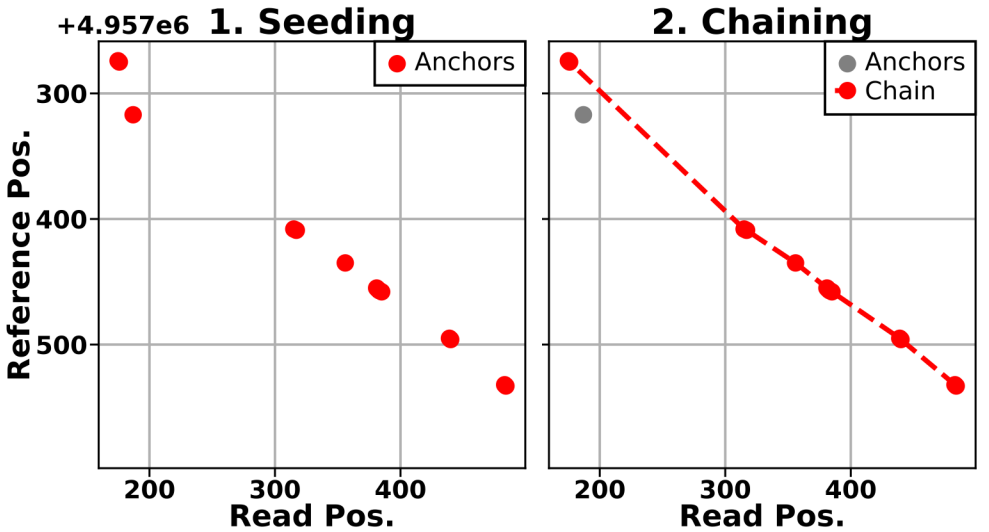


# Chaining (Two Points)



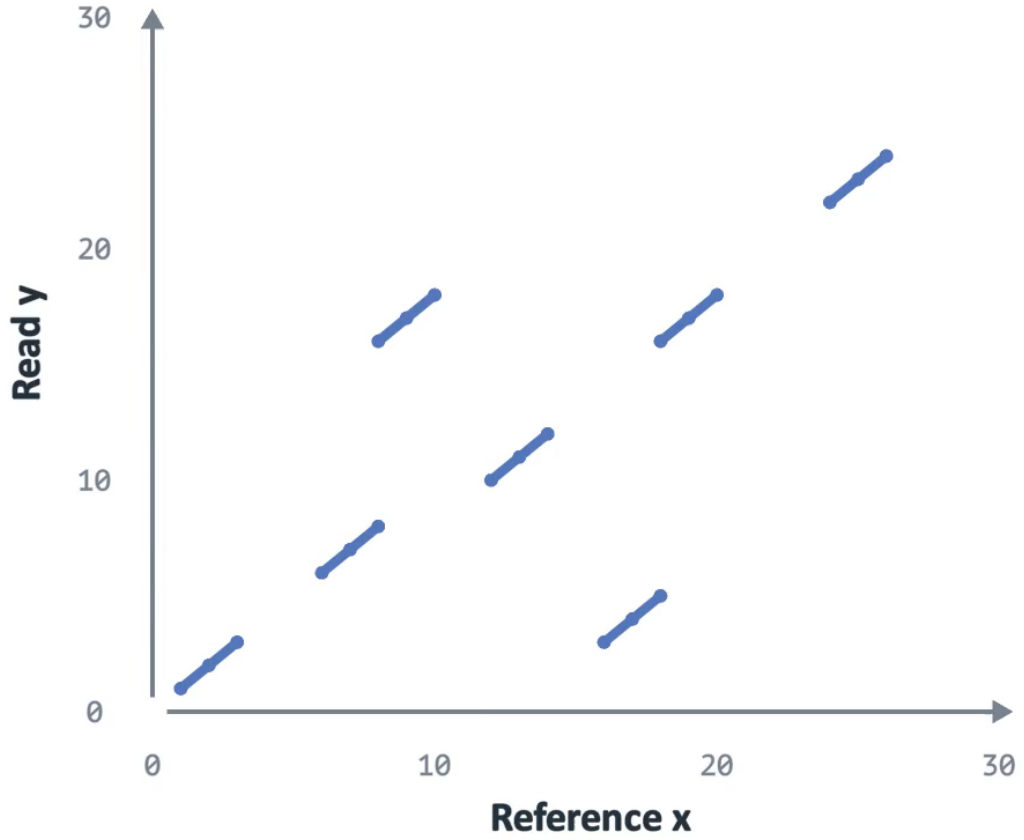
# Chaining (Multiple Points)

- Seed hits provide candidate anchors
- Useful anchors advance in both sequences
- Chaining links consistent anchors across gaps



# Anchors: Coordinates and DP Order

Sort anchors by reference end position (x)

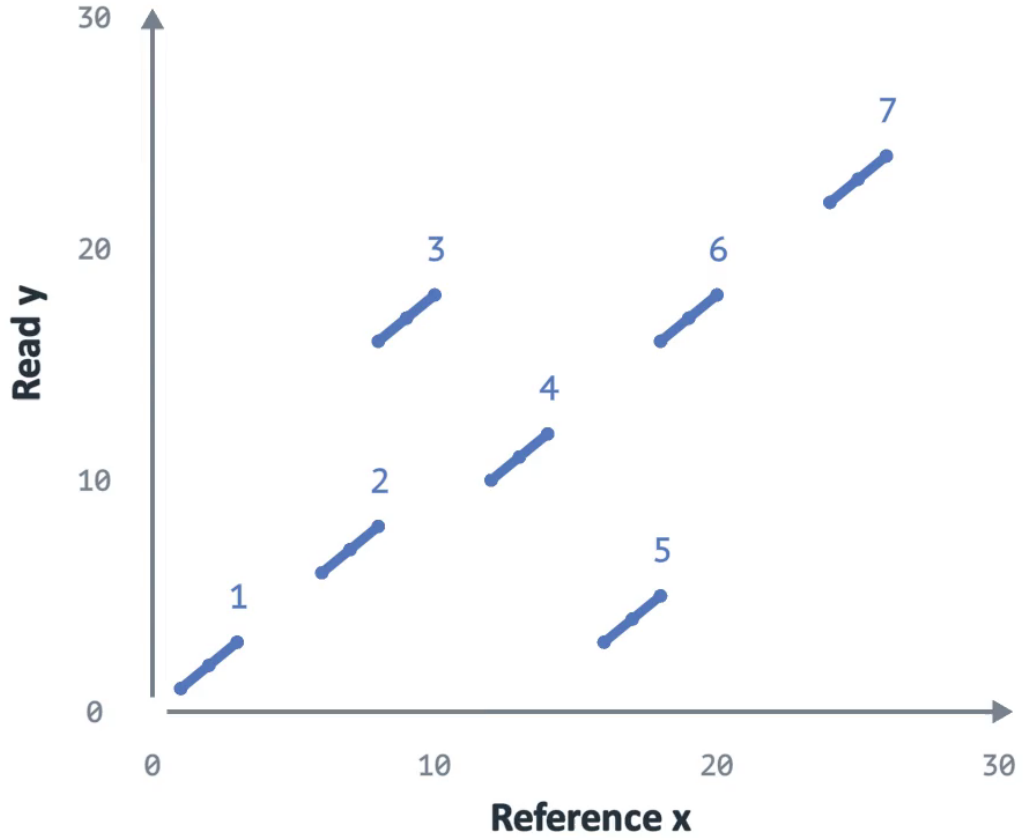



```
(x, y, z)

(20, 18, 3)
( 3,  3, 3)
(26, 24, 3)
(10, 18, 3)
(18,  5, 3)
( 8,  8, 3)
(14, 12, 3)
```

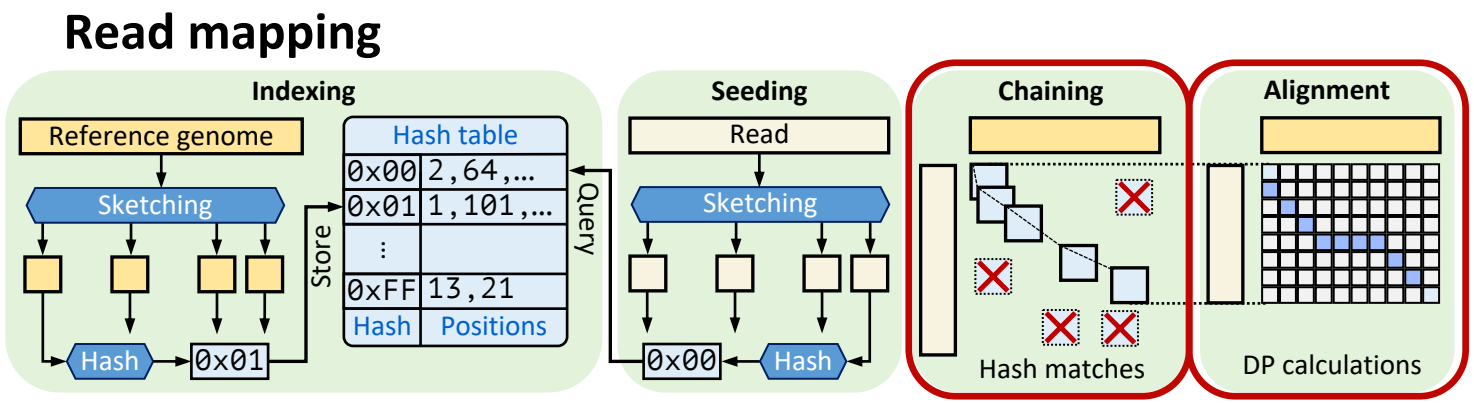
# Chaining: Scoring Consistent Anchors

Example gap penalty: |Δx − Δy|

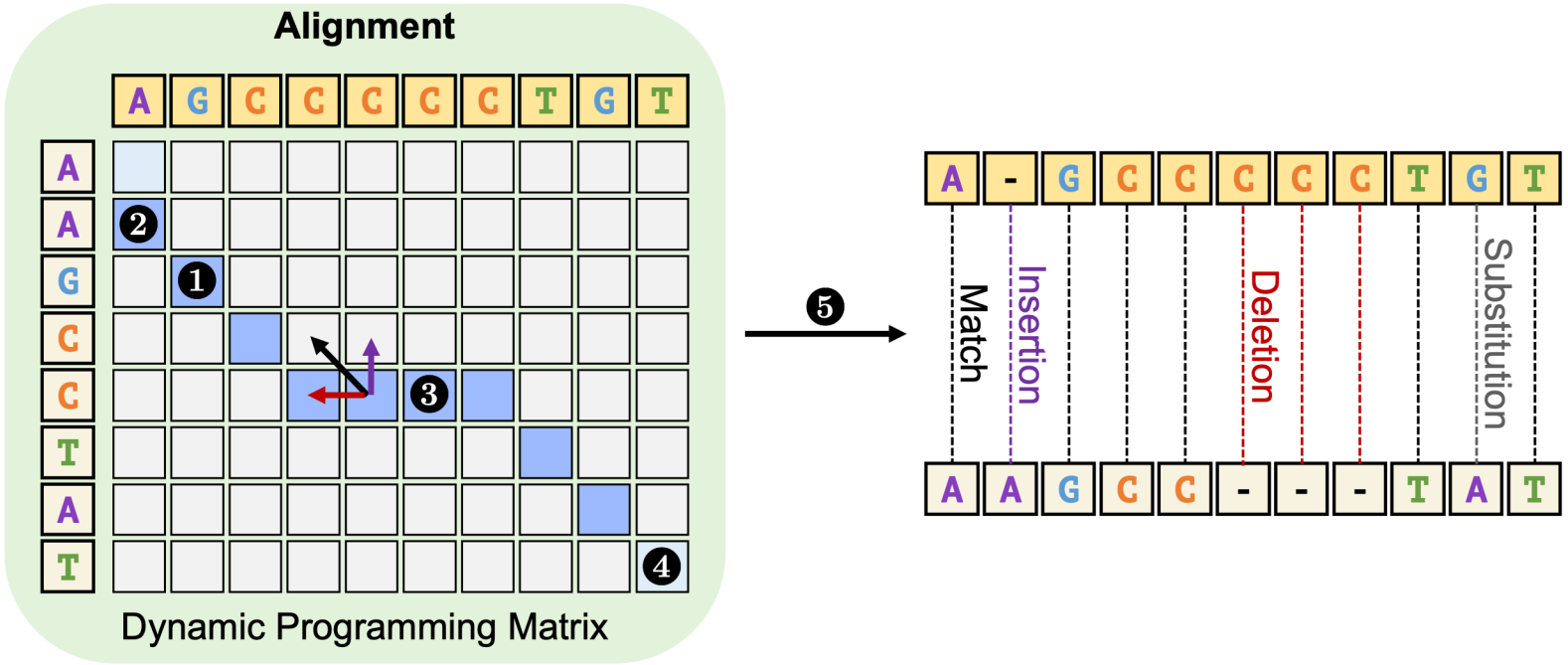



| i | (x, y, z) | f[i] | p[i] |
|---|---|---|---|
| 1 | ( 3, 3, 3) | | |
| 2 | ( 8, 8, 3) | | |
| 3 | (10, 18, 3) | | |
| 4 | (14, 12, 3) | | |
| 5 | (18, 5, 3) | | |
| 6 | (20, 18, 3) | | |
| 7 | (26, 24, 3) | | |

# Steps in Read Mapping – Alignment



# Sequence Alignment



# Alignment Objectives

## Global

Both Sequences, End to End


## Semi-Global

A Full Read Within a Reference


## Local

Matching Substrings of Both

# Alignment: Filling the DP Matrix

Each cell stores the best cost for two prefixes

| | | Reference | | | | | |
|---|---|---|---|---|---|---|---|
| | | - | A | C | G | T | A | C |
| Read | - | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| | A | 1 | | | | | | |
| | C | 2 | | | | | | |
| | T | 3 | | | | | | |
| | T | 4 | | | | | | |
| | C | 5 | | | | | | |

# Alignment: Traceback

Follow a predecessor that explains the current cost

| Read \ Reference | - | A | C | G | T | A | C |
|---|---|---|---|---|---|---|---|
| - | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| A | 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| C | 2 | 1 | 0 | 1 | 2 | 3 | 4 |
| T | 3 | 2 | 1 | 1 | 1 | 2 | 3 |
| T | 4 | 3 | 2 | 2 | 1 | 2 | 3 |
| C | 5 | 4 | 3 | 3 | 2 | 2 | 2 |

Ref

Read

# Affine Gap Costs

**Gap cost = open + length × extend**

Open = 4  extend = 1

**One gap of length 3**

Reference

A C A A A A A A G T

Query

A C A - - - A A G T

**4 + 3 × 1 = 7**

**Three gaps of length 1**

Reference

A C A A A A A A G T

Query

A C - A - A - A G T

**3 × (4 + 1) = 15**

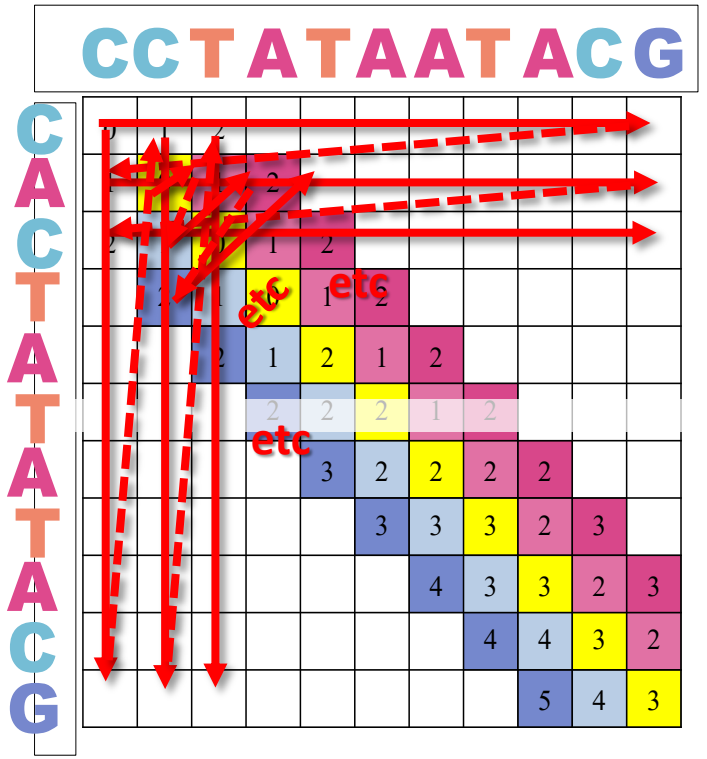
**Opening another gap has an extra cost**

# Why Is Read Alignment Slow?

Full DP evaluates O(mn) cells

Dependencies constrain the update order

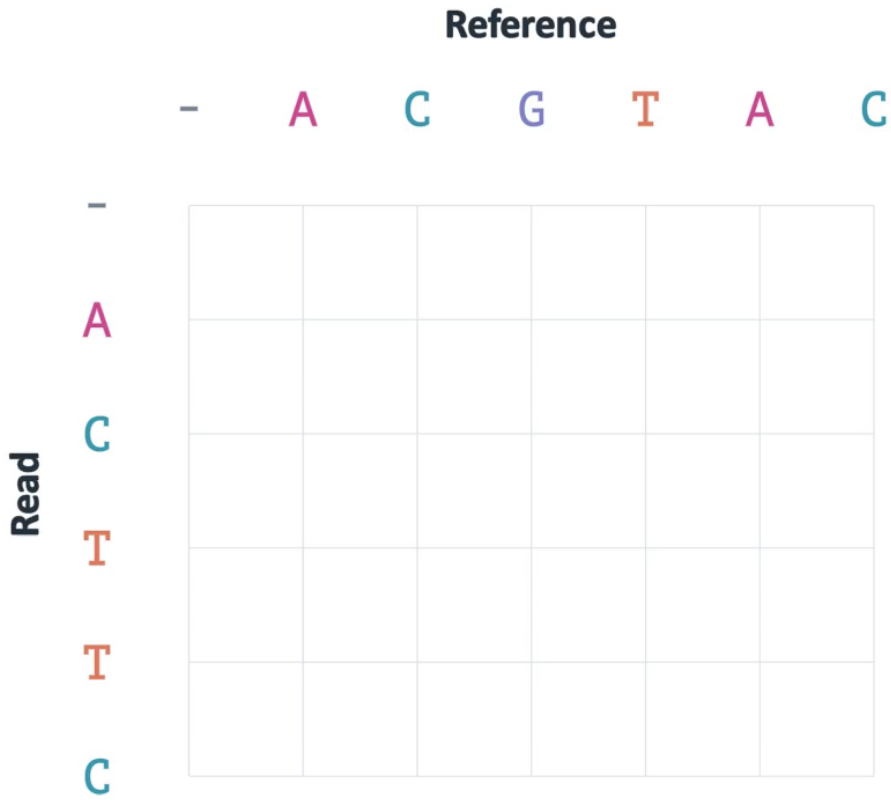
Many cells may be unnecessary for similar sequences




Read Alignment

# Wavefront Alignment

Unit-cost example: furthest reachable point per diagonal




k = x − y

| s | -2 | -1 | 0 | 1 | 2 |
|---|---|---|---|---|---|
| 0 | | | | | |
| 1 | | | | | |
| 2 | | | | | |

F[s,k] = x

# WFA in Practice

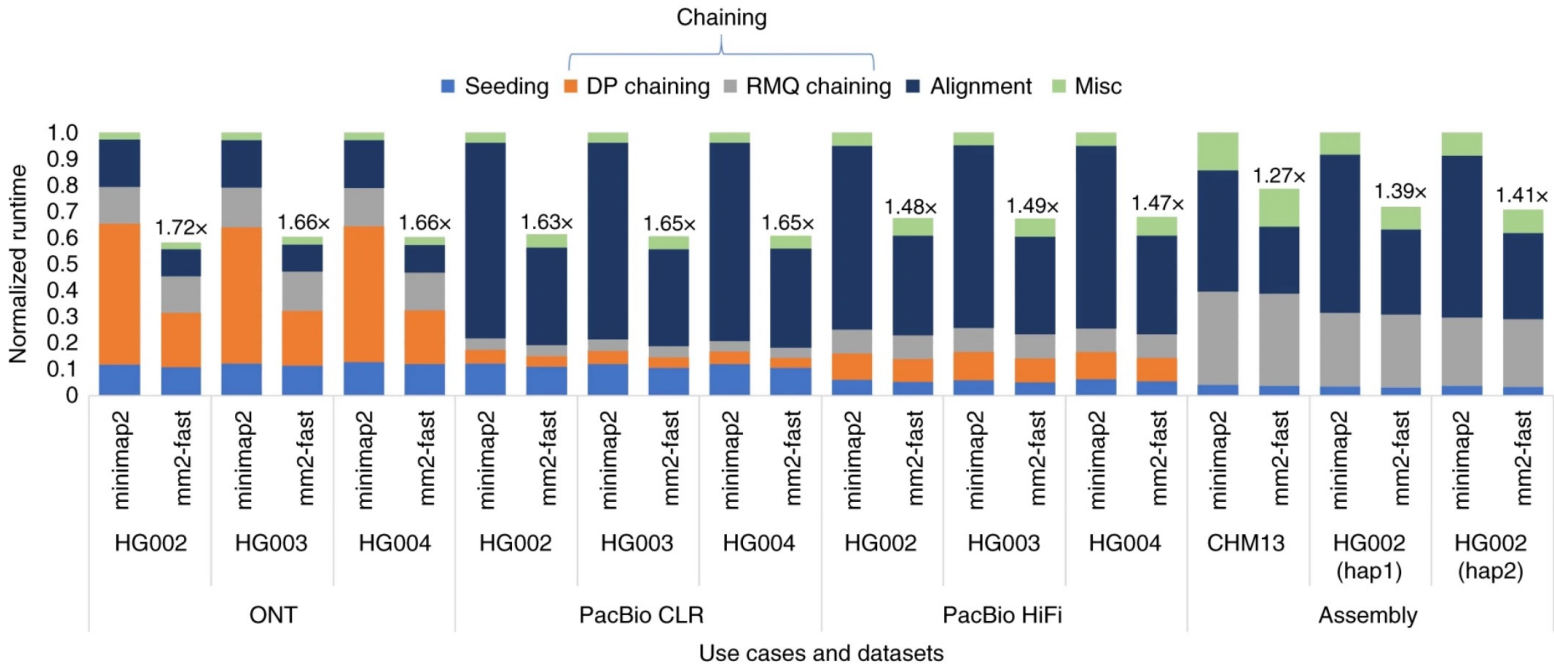
## Similar sequences can have a small alignment penalty

n = sequence length  s = optimal alignment penalty

| | | |
|---|---|---|
| **Time** | $O(ns + s^2)$ | Fewer states when s is small |
| **Memory** | $O(s^2) \rightarrow O(s)$ | BiWFA uses bidirectional divide and conquer |
| **Guarantee** | **Exact** | Heuristic pruning can sacrifice optimality |

Worst-case quadratic cost still remains possible

# Read Mapping Execution Time




## Chaining + alignment

## Dominate these minimap2 benchmarks

Kalikar et al., Nature Computational Science, 2022

# Why do we rely on a single reference genome?

# Recall: Solving the Puzzle

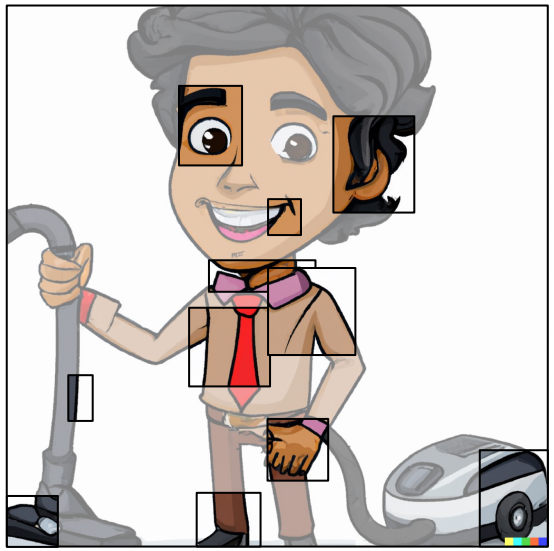
**Reference genome**

**A human reference**

**About 3.2 billion bases**

**One representation of genomic sequence**

**Reads**



**Short and long reads**

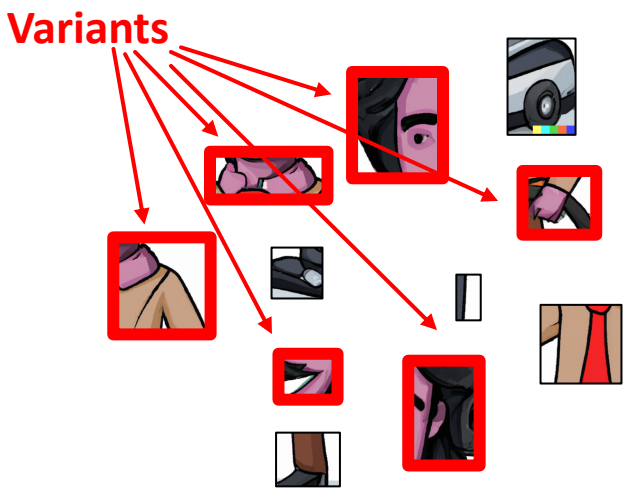
**Candidate locations are found**

**Relative to a reference**

# The Problem: Variants in the Genome

Reference genome

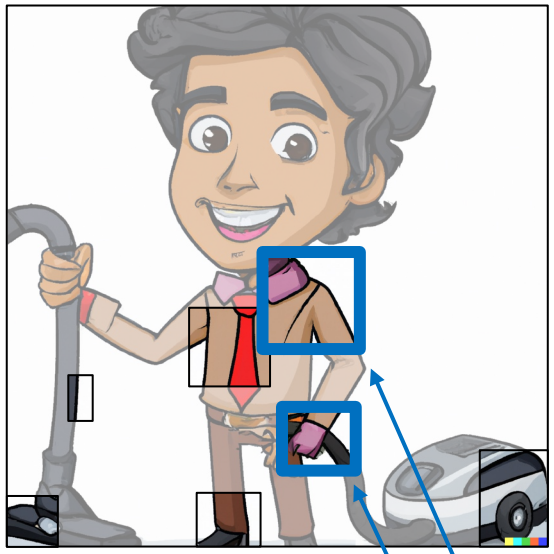
Reads



# The Problem: Variants in the Genome

Reference genome

Reads



Some reads can be mapped due to sufficient context

Some reads fail to be mapped because they are too different from the single reference

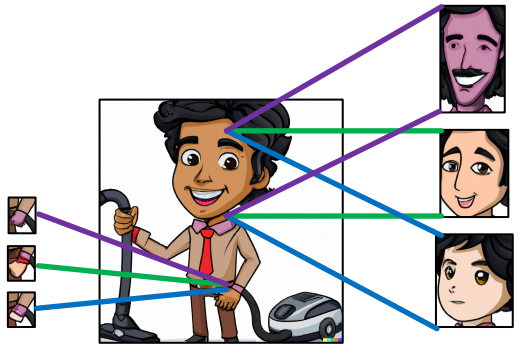
Reference bias!

# Mapping to Multiple References

- Map against multiple reference genomes
- Repeating the search duplicates work
- The sample may contain new combinations of variants

# Mapping to a Graph Reference

- Share sequence across reference genomes
- Represent alternative alleles as graph paths
- Search these alternatives within one reference structure



# Genome Graphs

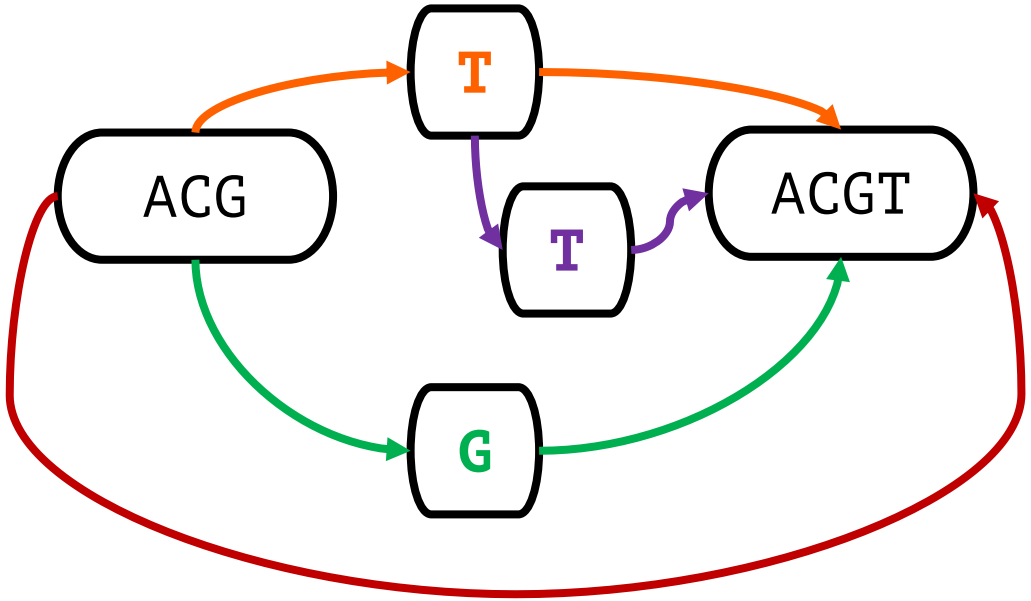
- Nodes store sequence; edges connect alternatives
- Paths represent different genomic sequences
- A graph can reduce single-reference bias

**Sequence #1:** ACG**T**ACGT

**Sequence #2:** ACG**G**ACGT

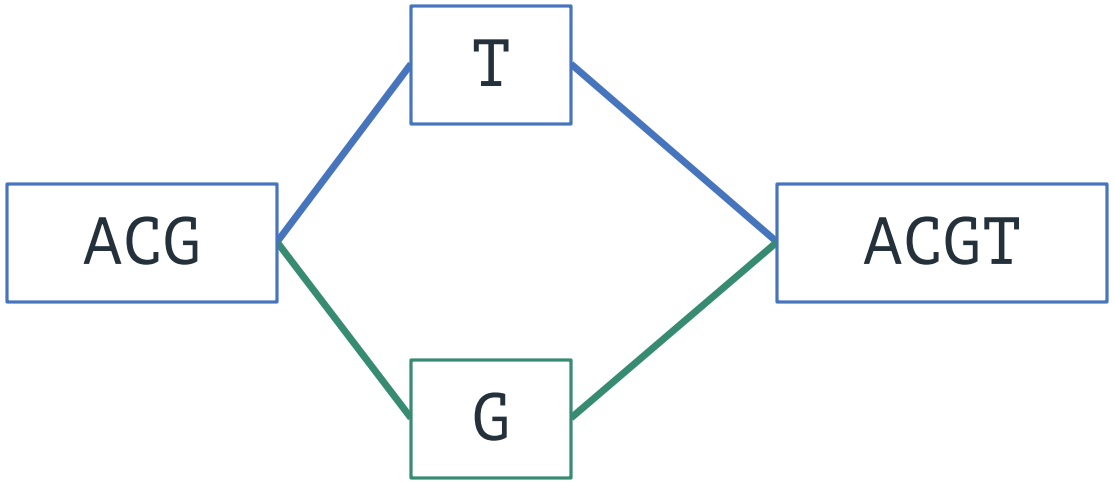
**Sequence #3:** ACG**TT**ACGT

**Sequence #4:** ACGACGT



# Read Mapping to a Pangenome Graph

**Shared sequence, alternative paths**




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

## Seeds

Matches on graph paths

## Chains

Compatible order and graph distances

## Extension

Base-level alignment in a candidate region

**vg Giraffe: short reads and long-read chaining**

Minigraph-Cactus and PGGB build graphs; mappers search them

# Haplotype-Aware References

**Graph topology alone can allow unobserved combinations**

# Sequence Analysis: Key Ideas

| | |
|---|---|
| **Seeds** | Find candidate matches |
| **Chains** | Link consistent positional evidence |
| **Alignment** | Resolve base-level differences |
| **Pangenomes** | Represent alternatives across genomes |

# Recommended: Genome Analysis Steps [CSBJ '22]

- **From molecules to genomic variations: Accelerating genome analysis via intelligent algorithms and architectures**
  Mohammed Alser, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu
  Computational and Structural Biotechnology Journal (**CSBJ**), August 2022.
  [PDF] [Code] [DOI]

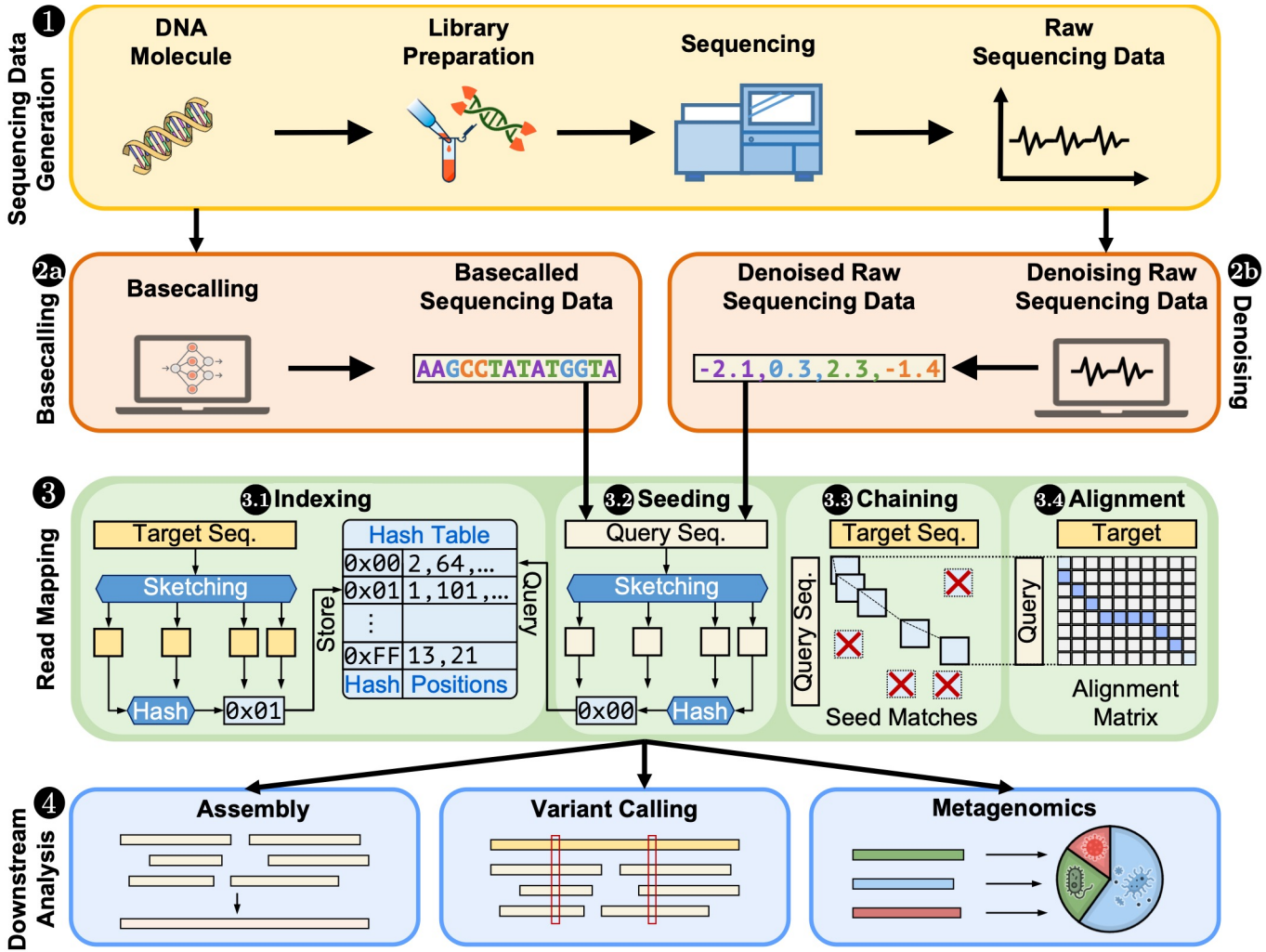
Review

## From molecules to genomic variations: Accelerating genome analysis via intelligent algorithms and architectures

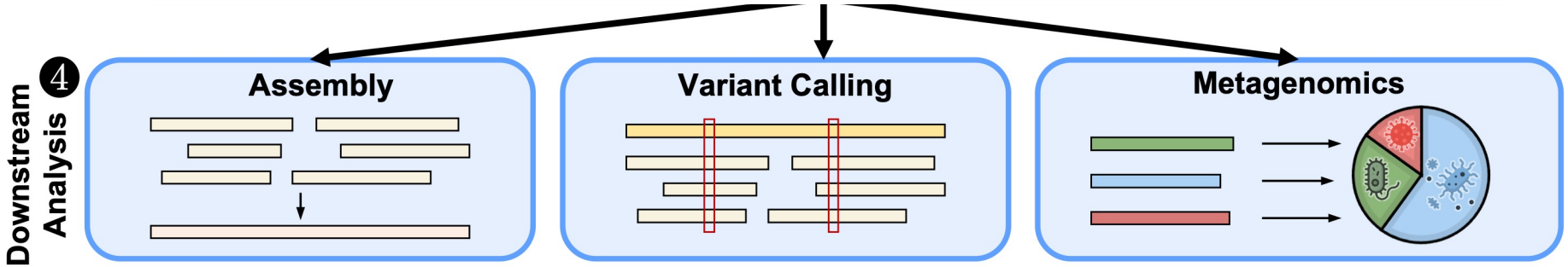

Mohammed Alser *, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu *

*ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland*

# Genome Analysis Pipeline

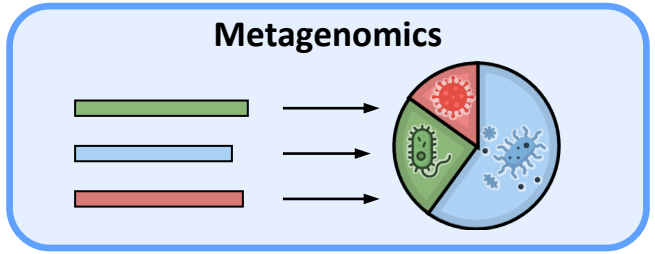


# Further Steps in Genome Analysis

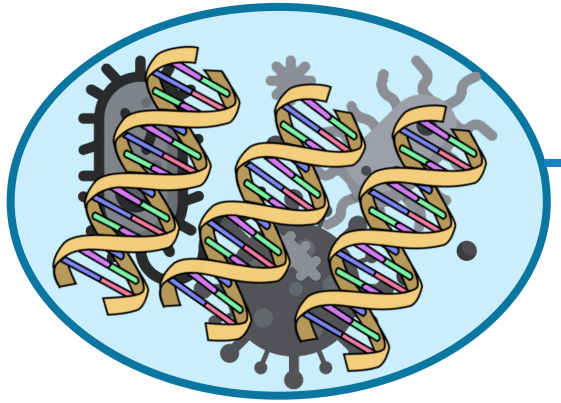


# Metagenomics

**How to analyze many *unknown* organisms**
in the same environment?



# Metagenomic Analysis



Metagenomic sample with species that are **not known** in advance

A large database containing information on **many species**

Preparation of input queries

Query k-mers

GCTCA

CTCAT

TCATG

...

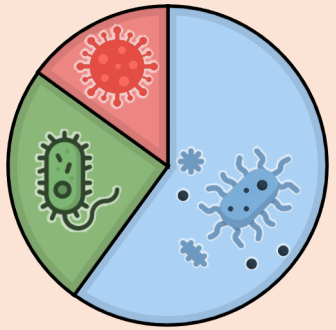
Presence/absence identification

V. cholerae

SARS-CoV-2

E. coli

Abundance estimation



**Database size depends on its content and representation**

# Questions?

# Next Week

- Finalizing paper assignments

- What happens when there is no reference genome available?
  - De novo assembly

- AI/ML in genomics

# CMSC 829C: Algorithms, AI, and Hardware for Biological Data

## Lecture 4: Sequence Analysis in Genomics

Fall 2026

September 10, 2026

**Can Firtina** | canfirtina@gmail.com | cs.umd.edu/~firtina

Assistant Professor of Computer Science at UMD

UNIVERSITY OF MARYLAND