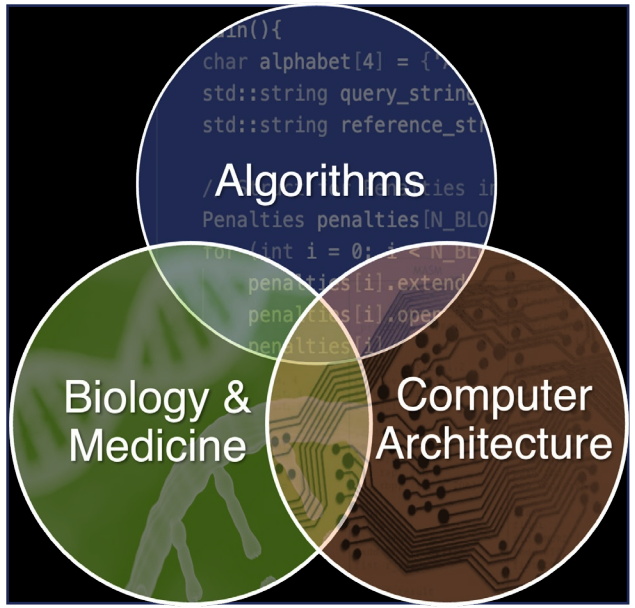
GPU-Accelerated Construction of Ultra-Large Pangenomes via Alignment-Phylogeny Co-Estimation

Prof. Yatish Turakhia

UC San Diego

Turakhia Lab at UCSD



Our research mission is to **develop parallel algorithms, software, and domain-specific hardware accelerators that enable faster and cheaper progress in biology and medicine**.

Parallel Computing Solutions

HPC Clusters

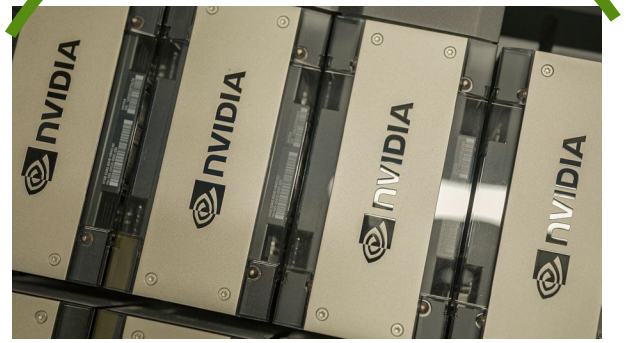
USHER

USHER

ROADIES

WEPP

GPU Processors

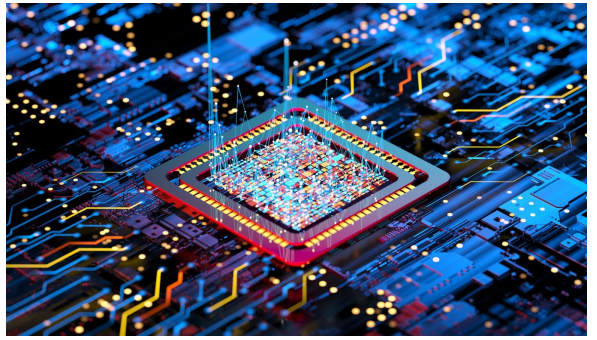


TWILIGHT

DIPPER

Focus on this talk

FPGAs and Custom Chips



TALCO

DP-HLS

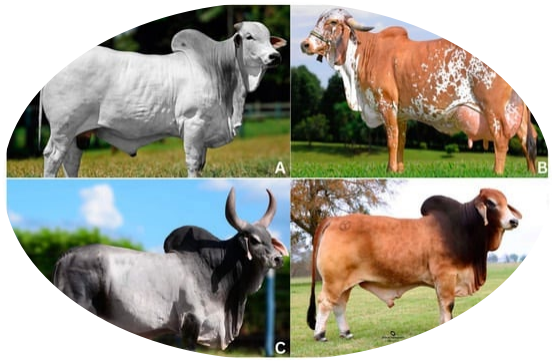
DP-HLS

Introduction to Pangenomics

- **Pangenomics** is an emerging field that deals with a **collection** of **genomes** from a **single species**

- Helps study how the **genomic variation** leads to:
  - **Diseases** in **humans**
  - Desirable **traits** in **plants** and **animals**
  - **Resistance** in **pathogens**

Humans (Liao et al., *Nature* 2023)



Cows (Zhou et al., *Gen. Res.* 2022)

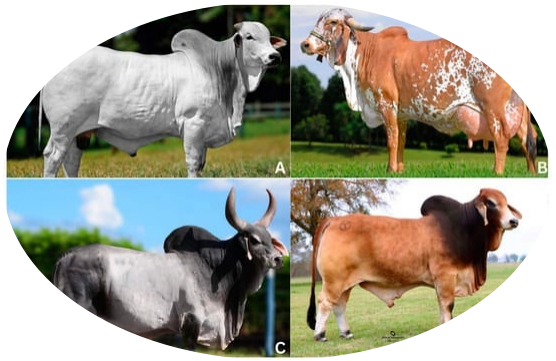
Tomato (Zhou et al., *Nature* 2022)

***E. Coli* (Noll et al., *Microb Genom* 2023)**

Introduction to Pangenomics

- **Pangenomics** is an emerging field that deals with a **collection** of **genomes** from a **single species**
- Helps study how the **genomic variation** leads to:
  - **Diseases** in **humans**
  - Desirable **traits** in **plants** and **animals**
  - **Resistance** in **pathogens**
- Current pangenomic analysis often limited to **tens to hundreds** of genomes

Humans (Liao et al., *Nature* 2023)



Cows (Zhou et al., *Gen. Res.* 2022)

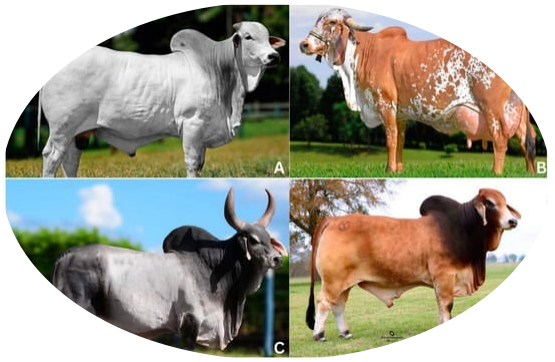
Tomato (Zhou et al., *Nature* 2022)

E. Coli (Noll et al., *Microb Genom* 2023)

Introduction to Pangenomics

- **Pangenomics** is an emerging field that deals with a **collection** of **genomes** from a **single species**
- Helps study how the **genomic variation** leads to:
  - **Diseases** in **humans**
  - Desirable **traits** in **plants** and **animals**
  - **Resistance** in **pathogens**
- Current pangenomic analysis often limited to **tens to hundreds** of genomes
- How do we scale to **millions of genomes?**
  - Focus of **this talk**

47 genomes

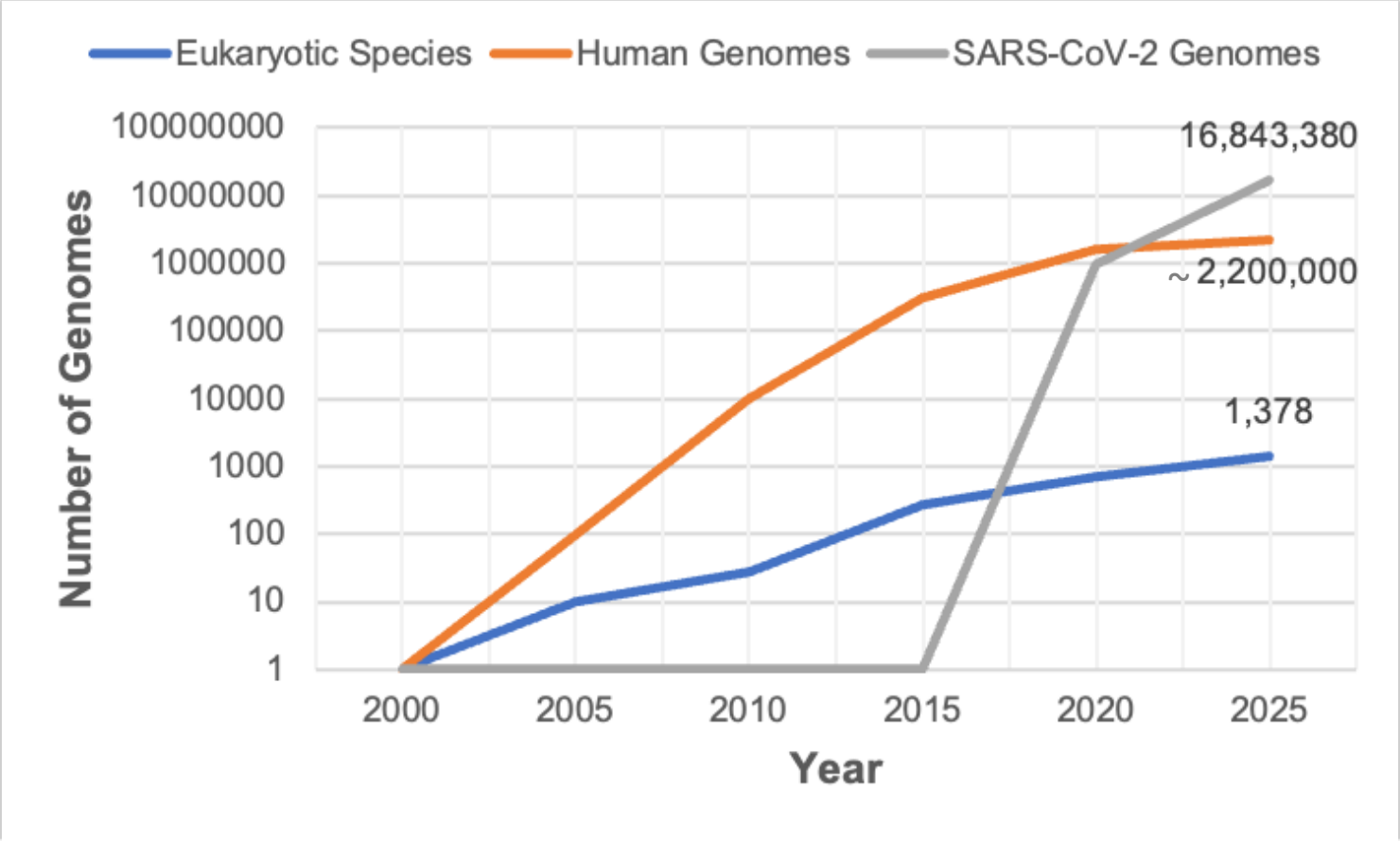


898 genomes

838 genomes

307 genomes

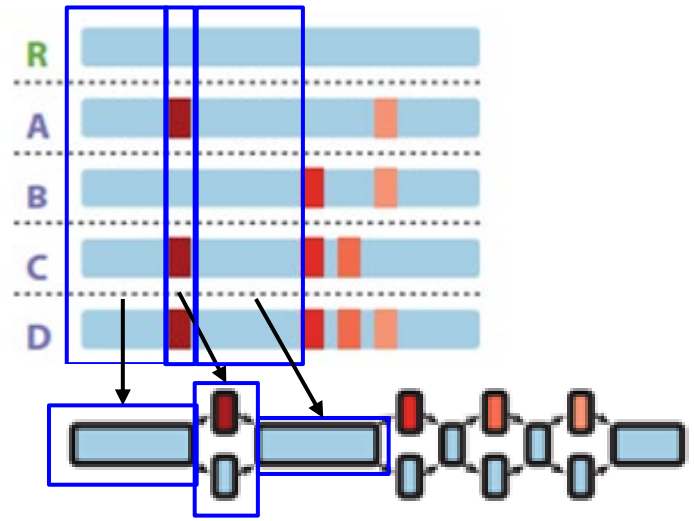
Within-species data dominates *cross-species* data by orders of magnitude



Source: Google Gemini Deep Research

First, we need a data representation

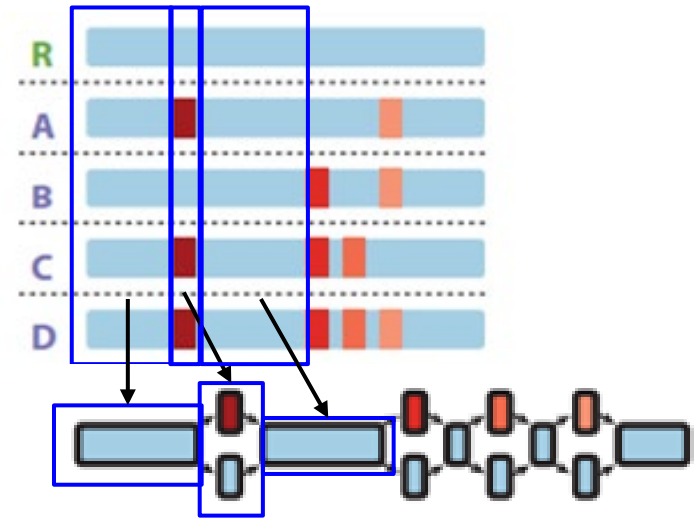
- Currently the most popular representation is using **graph-based** formats



Graph-based Pangenomes

First, we need a data representation

- Currently the most popular representation is using **graph-based** formats
- Excellent for **reducing reference bias** in read alignment



Graph-based Pangenomes

First, we need a data representation

- Currently the most popular representation is using **graph-based** formats
- Excellent for **reducing reference bias** in read alignment
- But there are **limitations**:
  1. Costly to **construct**

PanGraph

Species	No. of sequences	Runtime of PanGraph
SARS-CoV-2	20,000	18.5 hrs
HIV	20,000	17.4 hrs
Escherichia Coli	1,000	27.6 hrs
Klebsiella pneumoniae	1,000	56.2 hrs

PGGB (GFA)

Species	No. of sequences	Runtime of PGGB
SARS-CoV-2	20,000	16.1 hrs
HIV	20,000	14.5 hrs
Escherichia Coli	1,000	32.1 hrs
Klebsiella pneumoniae	1,000	43.5 hrs

First, we need a data representation

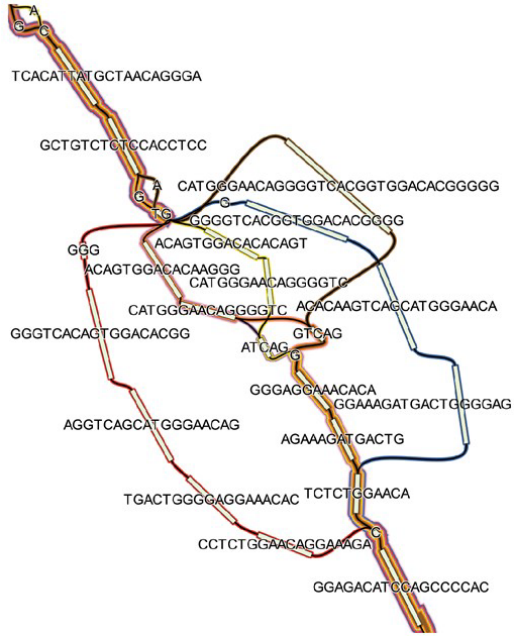
- Currently the most popular representation is using **graph-based** formats
- Excellent for **reducing reference bias** in read alignment
- But there are **limitations**:
  1. Costly to **construct**
  2. Memory-wise **inefficient**

File Size for various pathogens

Species (# of sequences)	GFA	VG	GBZ
SARS-CoV-2 (20,000)	1.1GB	307MB	34MB
Escherichia Coli (1,000)	2.4GB	747MB	900MB
Klebsiella pneumoniae (1,000)	3.6GB	1.1GB	1.4GB

First, we need a data representation

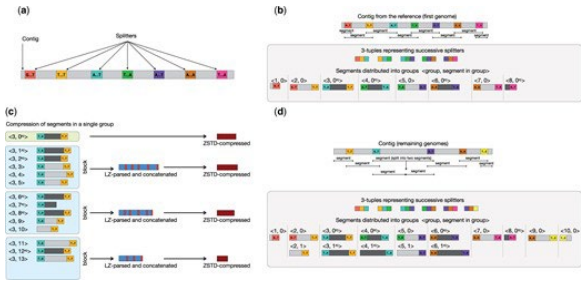
- Currently the most popular representation is using **graph-based** formats
- Excellent for **reducing reference bias** in read alignment
- But there are **limitations**:
  1. Costly to **construct**
  2. Memory-wise **inefficient**
  3. Lack **evolutionary** and **mutational** information (Only variation is stored)



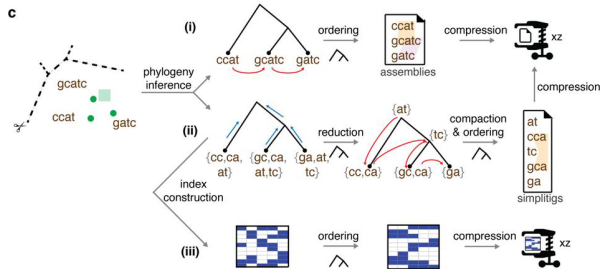
Alternative Pangenomic Representations

- Some formats focus on **compressing** large collections of raw genomic sequences
  - **No variation** is stored
  - Examples: **AGC** and **Miniphy**

Focus on **compression**



AGC (Deorowicz et al., *Bioinformatics* 2023)



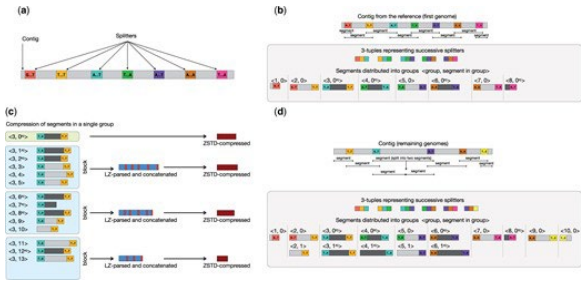
Miniphy (Brinda et al., *Nat. Biotech.* 2025)

Alternative Pangenomic Representations

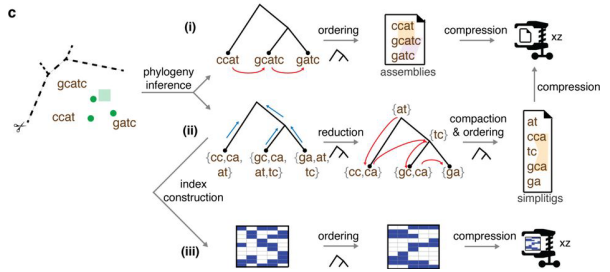
- Some formats focus on **compressing** large collections of raw genomic sequences
  - **No variation** is stored
  - Examples: **AGC** and **Miniphy**

- Others focus on **representing more**
  - Storage-wise **less efficient**
  - Examples: Phylogeny in **PanGraph**, recombination in tree sequences (**tskit**)

Focus on **compression**

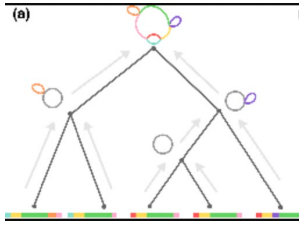


AGC (Deorowicz et al., *Bioinformatics* 2023)

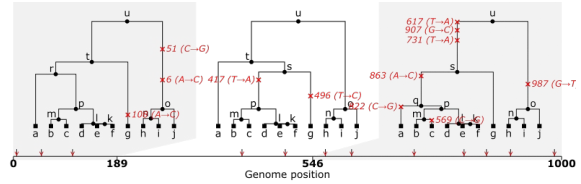


Miniphy (Brinda et al., *Nat. Biotech.* 2025)

Focus on **representing more**

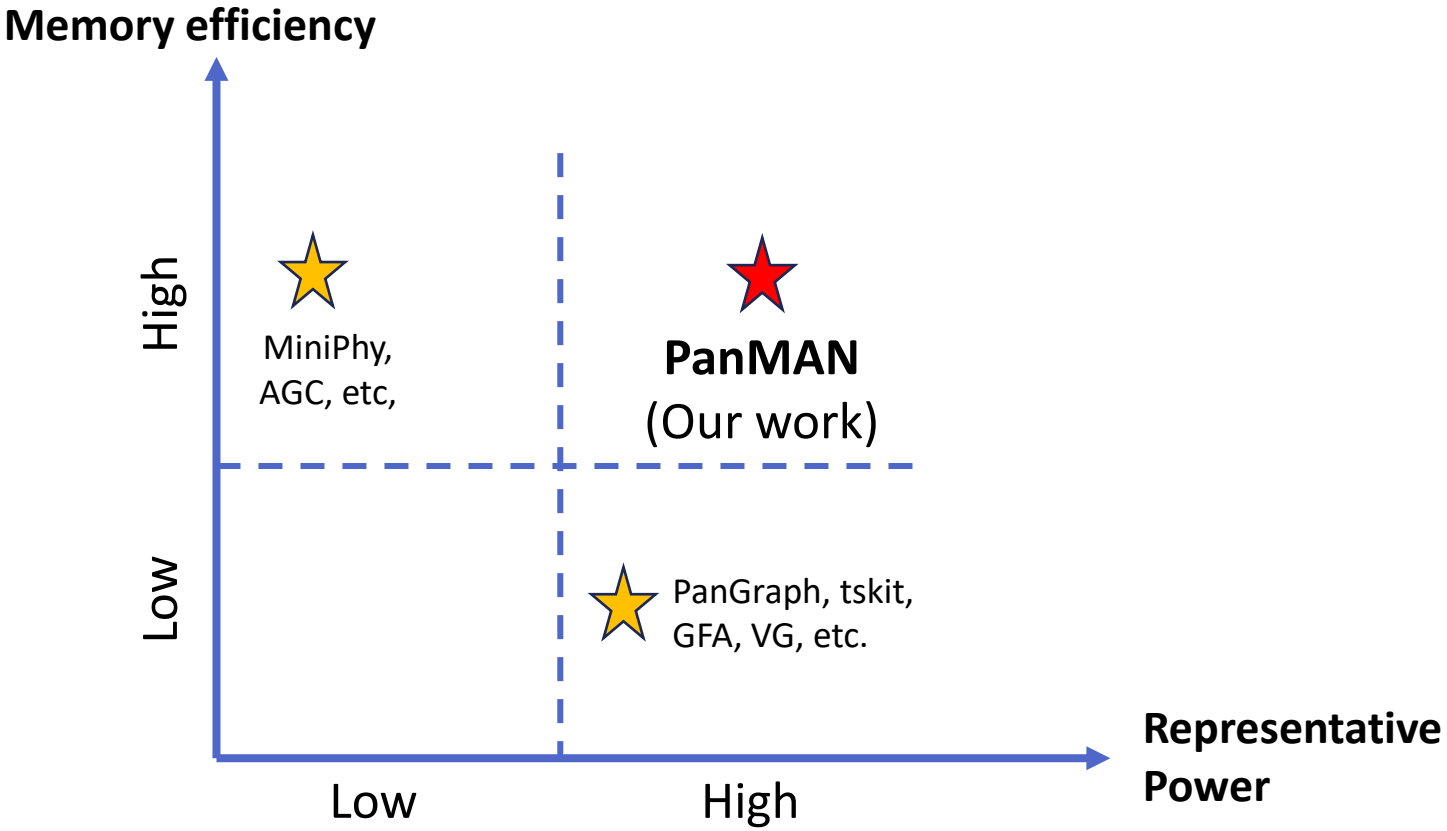


PanGraph (Noll et al., *Microb. Genom.* 2024)



Tskit (Kelleher et al., *Nat. Genet.* 2020)

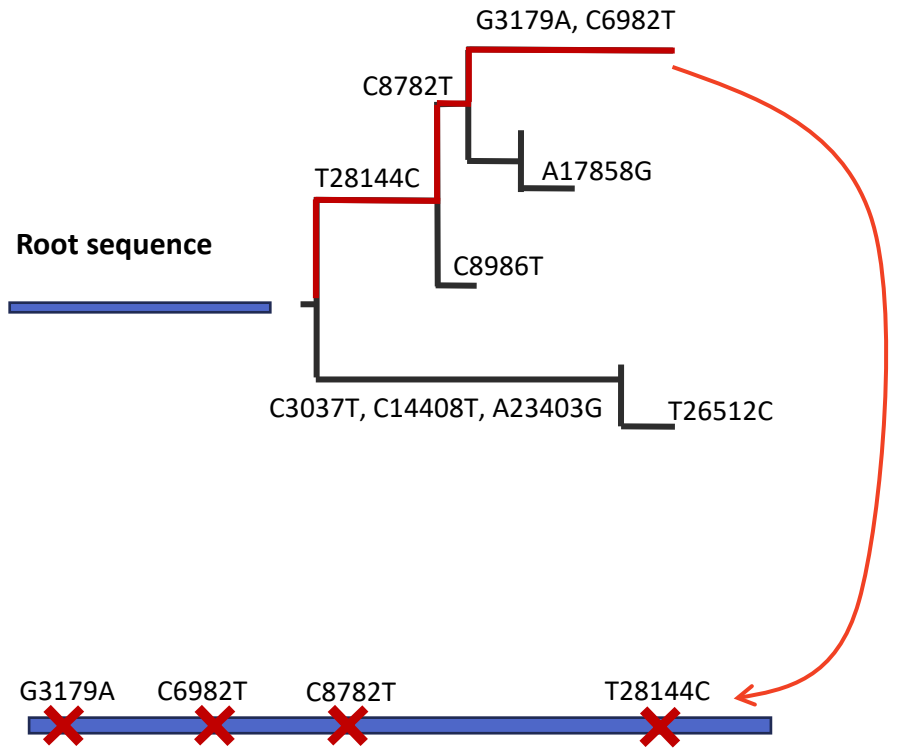
Pangenome Formats: Landscape of Trade-offs



Evolutionary Compression: Key idea

- **Store:**
  1. **Inferred phylogeny** (e.g., inferred tree topology)
  2. **Root sequence**
  3. **Mutations** inferred on each branch

- **Property:** sequence corresponding to every tip or internal node of the phylogeny can be derived from the root sequence and the mutations on its path to the root



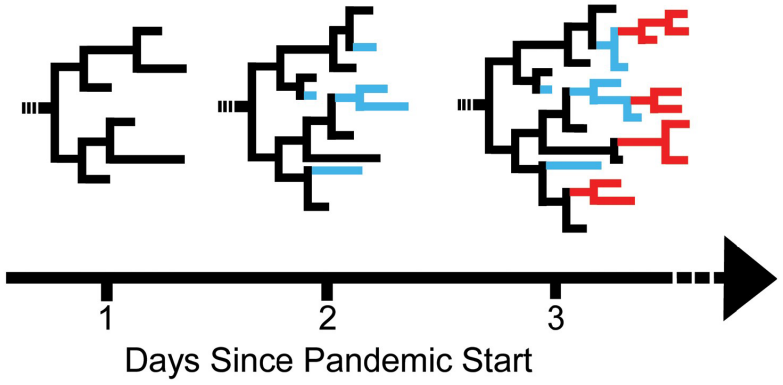
UShER Mutation-Annotated Tree (MAT) used Evolutionary Compression: Applications

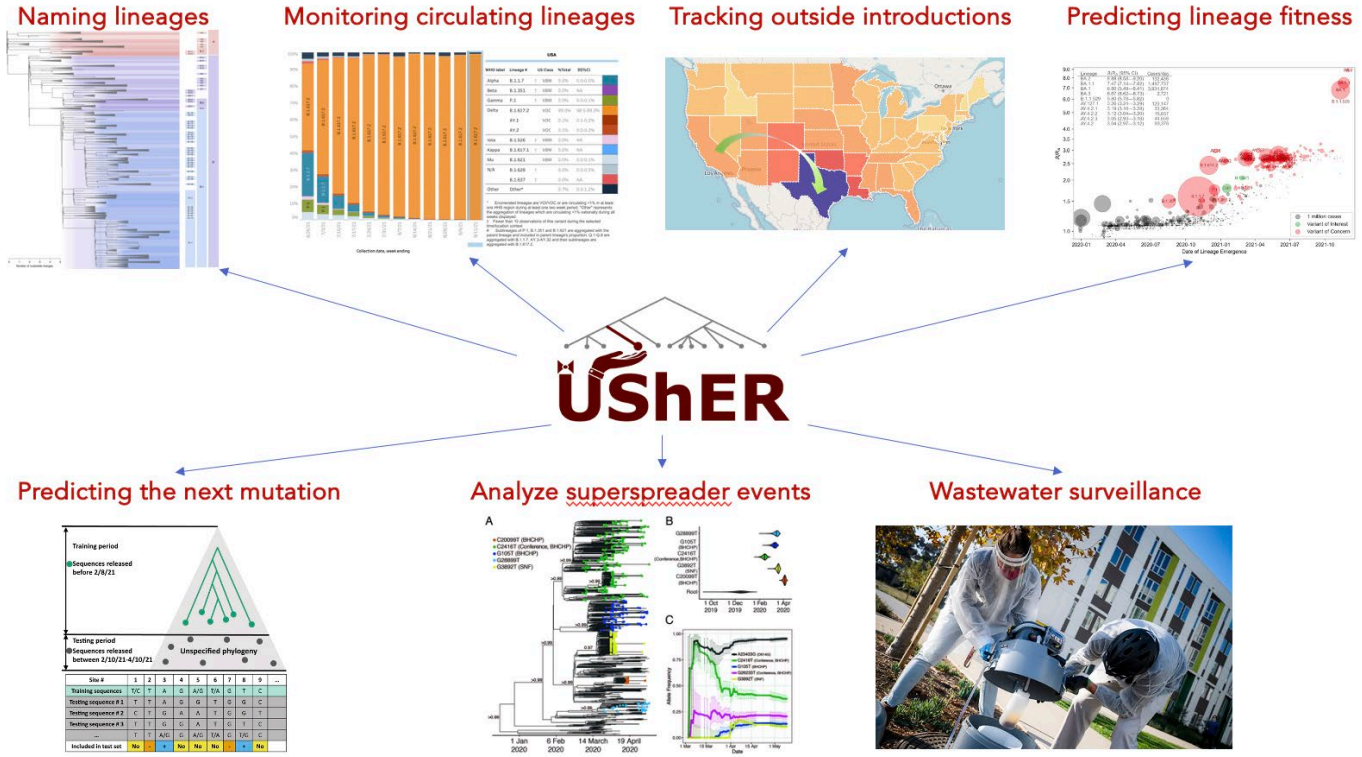
Article | Published: 10 May 2021

Ultrafast Sample placement on Existing tRees (UShER) enables real-time phylogenetics for the SARS-CoV-2 pandemic

Yatish Turakhia, Bryan Thornlow, Angie S. Hinrichs, Nicola De Maio, Landen Gozashti, Robert Lanfear, David Haussler & Russell Corbett-Detig

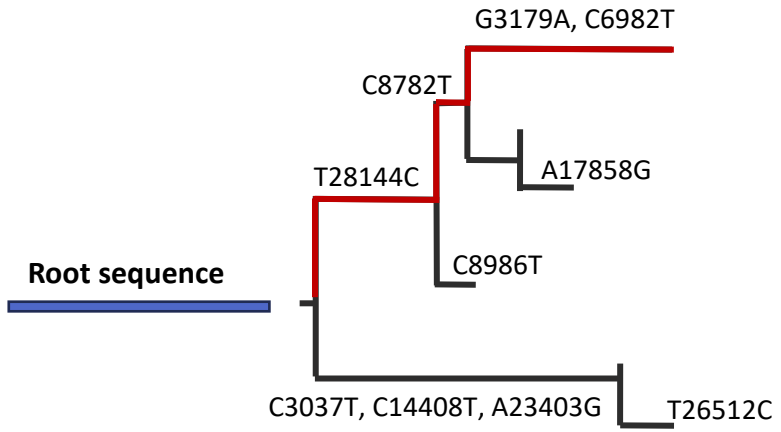
Nature Genetics **53**, 809–816 (2021) | Cite this article





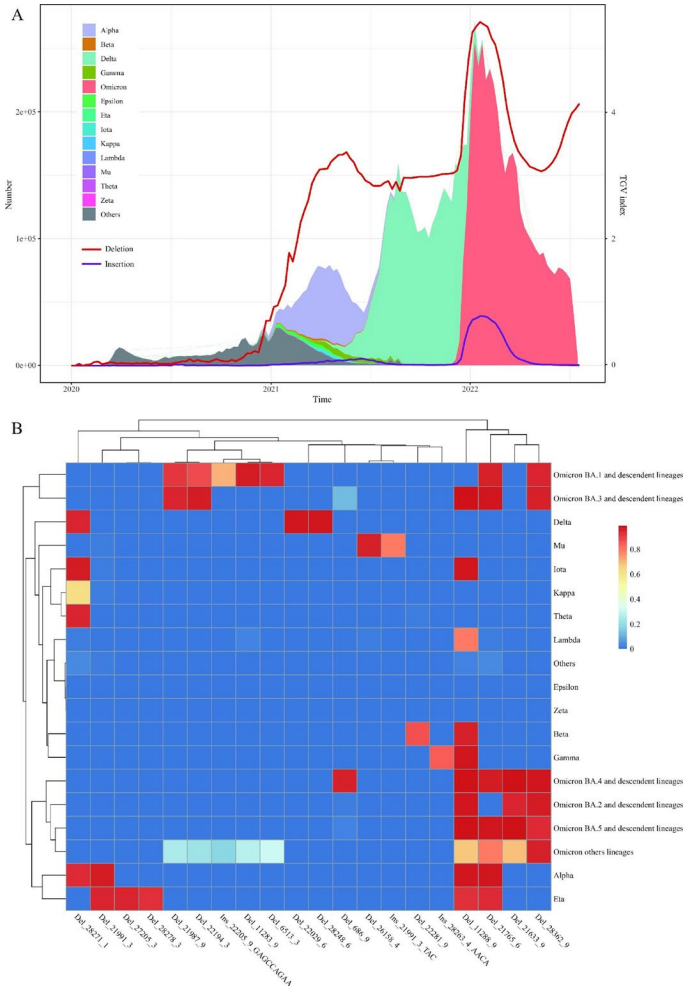
Limitations of UShER-MAT

- **Reference-based**



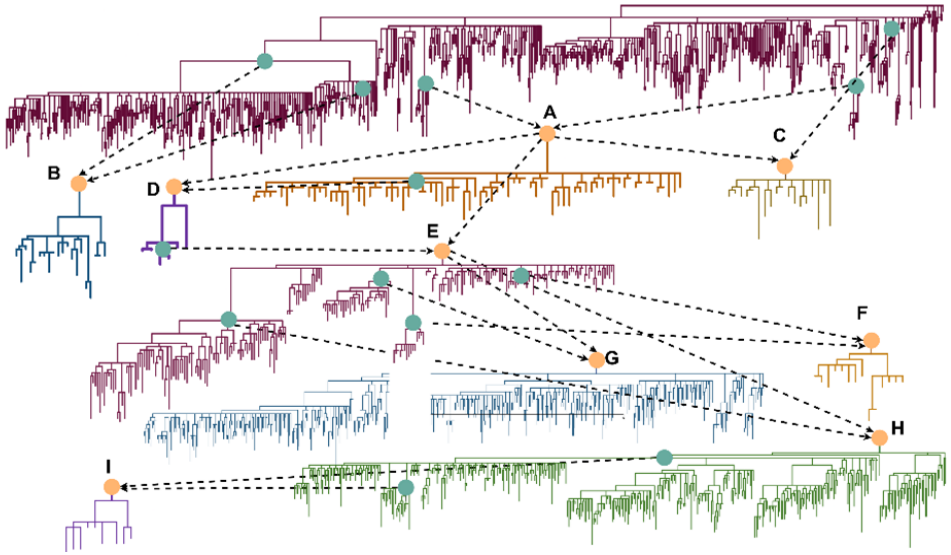
Limitations of UShER-MAT

- **Reference-based**
- Only stores substitutions – **ignores indels**
  - Indels sometimes comprise lineage-defining mutations



Limitations of UShER-MAT

- **Reference-based**
- Only stores substitutions – **ignores indels**
  - Indels sometimes comprise lineage-defining mutations
- Restricted to a single **tree topology** – cannot represent complex mutations (e.g., recombination or horizontal gene transfer) violating the vertical mode of evolution



Pangenome MANs: Approach

- **PanMAN** uses '**evolutionary compression**' to represent pangenomes
- Both a data structure and file format
- Unifies **alignment** and **phylogeny** into a more efficient representation


Sumit Walia


Harsh Motwani


Yu-Hsiang Tseng

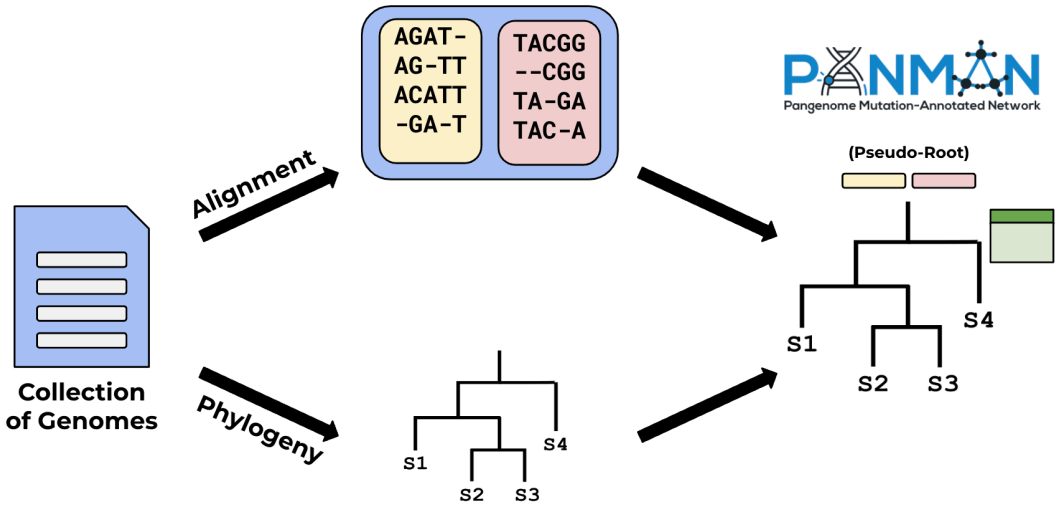
Compressive Pangenomics Using Mutation-Annotated Networks

Sumit Walia, Harsh Motwani, Kyle Smith, Russell Corbett-Detig, Yatish Turakhia

doi: https://doi.org/10.1101/2024.07.02.601807

This article is a preprint and has not been certified by peer review [what does this mean?].

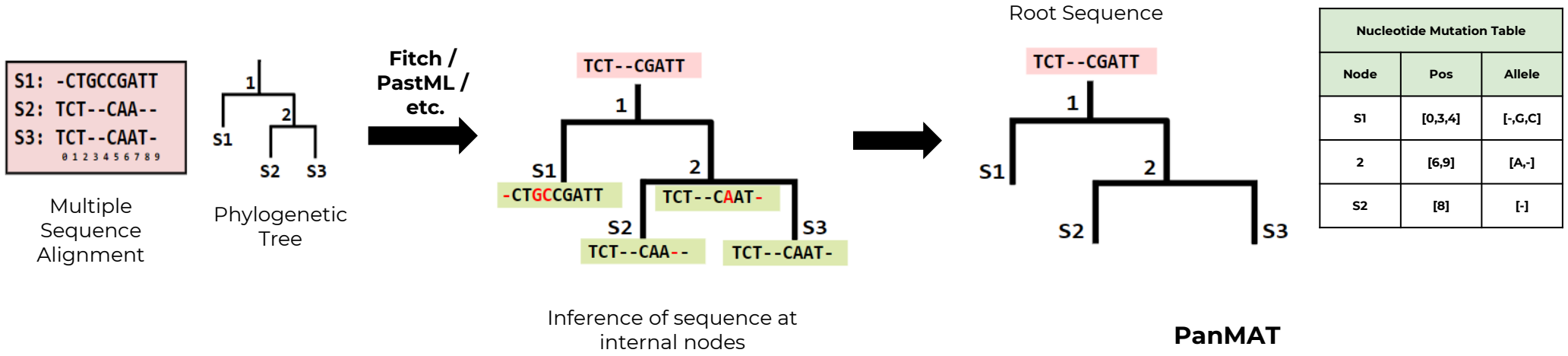
(Under revision in ***Nature Genetics***)



https://github.com/TurakhiaLab/panman

PanMAT: Pangenome Mutation-Annotated Tree

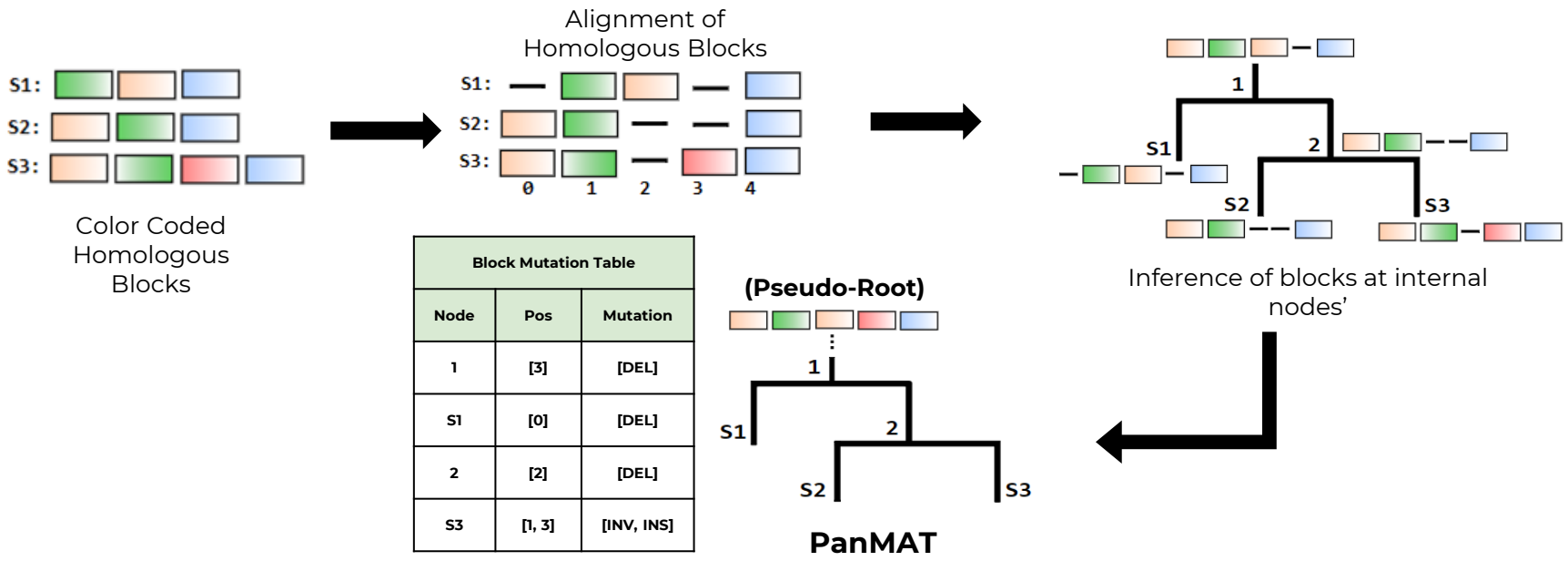
- Incorporating **insertions** and **deletions** (indels) into a MAT
  - MSA defines the coordinate system
  - Gaps (resulting from indels) treated as special characters



Nucleotide Mutation Table		
Node	Pos	Allele
S1	[0,3,4]	[-,G,C]
2	[6,9]	[A,-]
S2	[8]	[-]

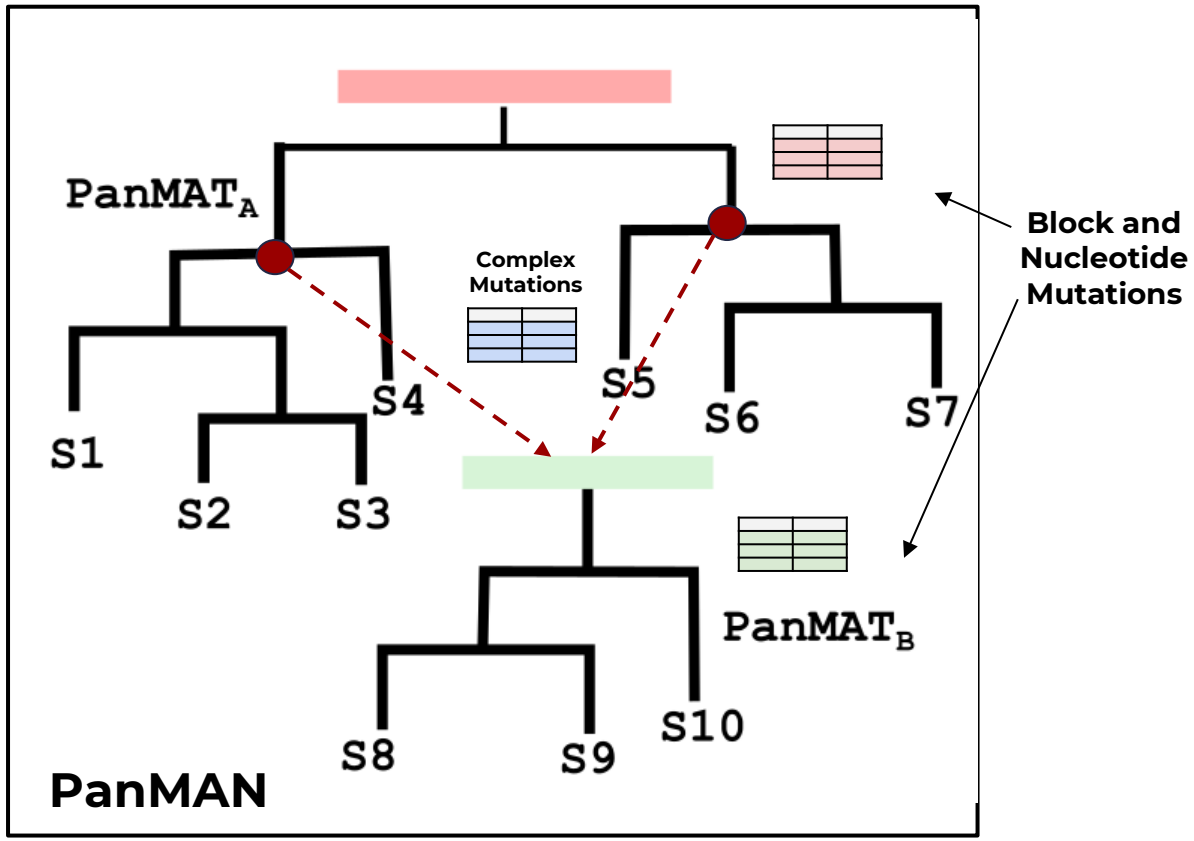
PanMAT: Pangenome Mutation-Annotated Tree

- Incorporating **structural changes** and **rearrangements**
  - Identify homologous blocks
  - MSA of homologous blocks
  - Block mutations are like substitutions to or from gaps

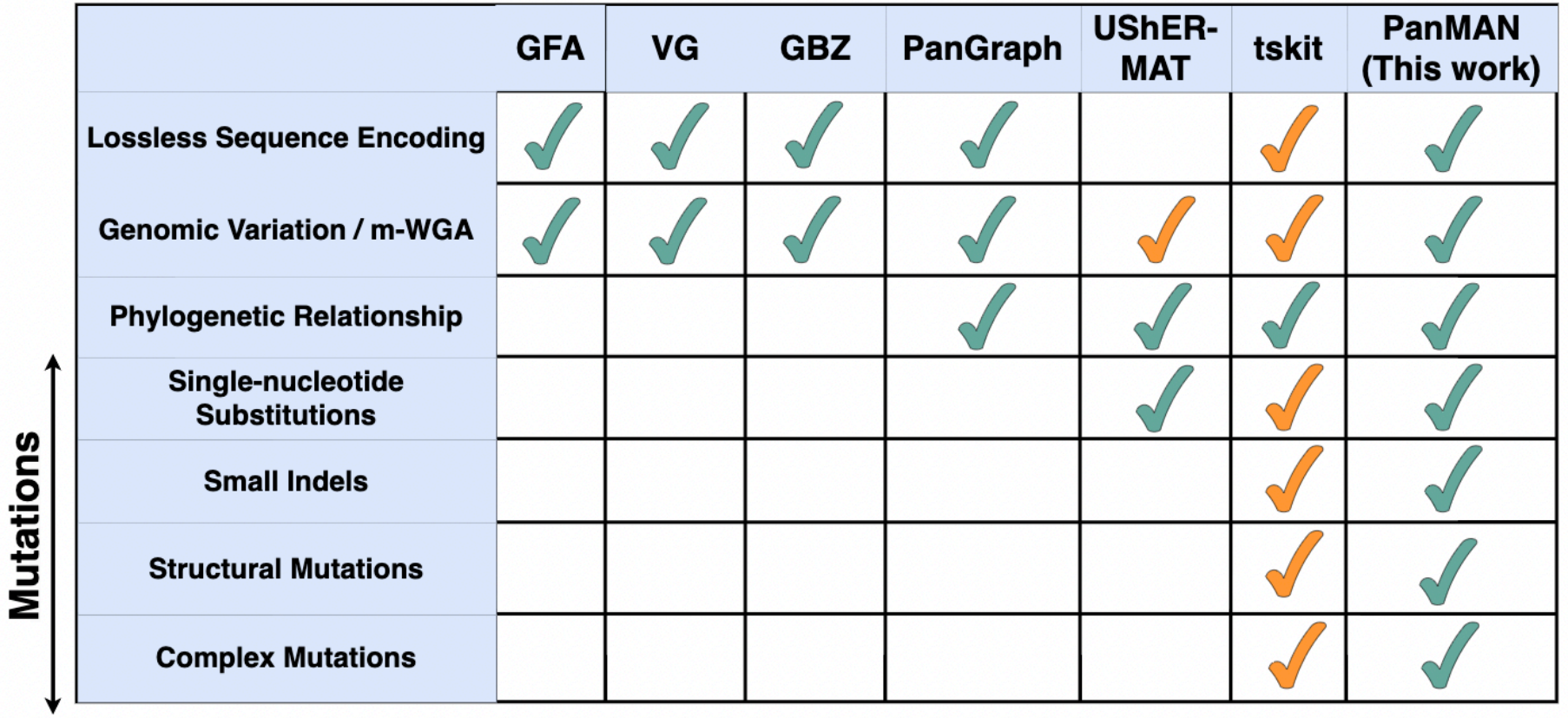


PanMAN: Pangenome Mutation-Annotated Network

- **PanMAN**: Generalization of PanMAT to represent **complex mutations**
- One or more PanMATs are connected with **network edges** (red dotted lines)
- Network edges store breakpoints of complex mutations (blue table), i.e., **Horizontal Gene Transfer** (HGT) and **Recombination**



Representative Power of Pangenome formats



	GFA	VG	GBZ	PanGraph	UShER-MAT	tskit	PanMAN (This work)
Lossless Sequence Encoding	✓	✓	✓	✓		✓	✓
Genomic Variation / m-WGA	✓	✓	✓	✓	✓	✓	✓
Phylogenetic Relationship				✓	✓	✓	✓
Single-nucleotide Substitutions					✓	✓	✓
Small Indels						✓	✓
Structural Mutations						✓	✓
Complex Mutations						✓	✓

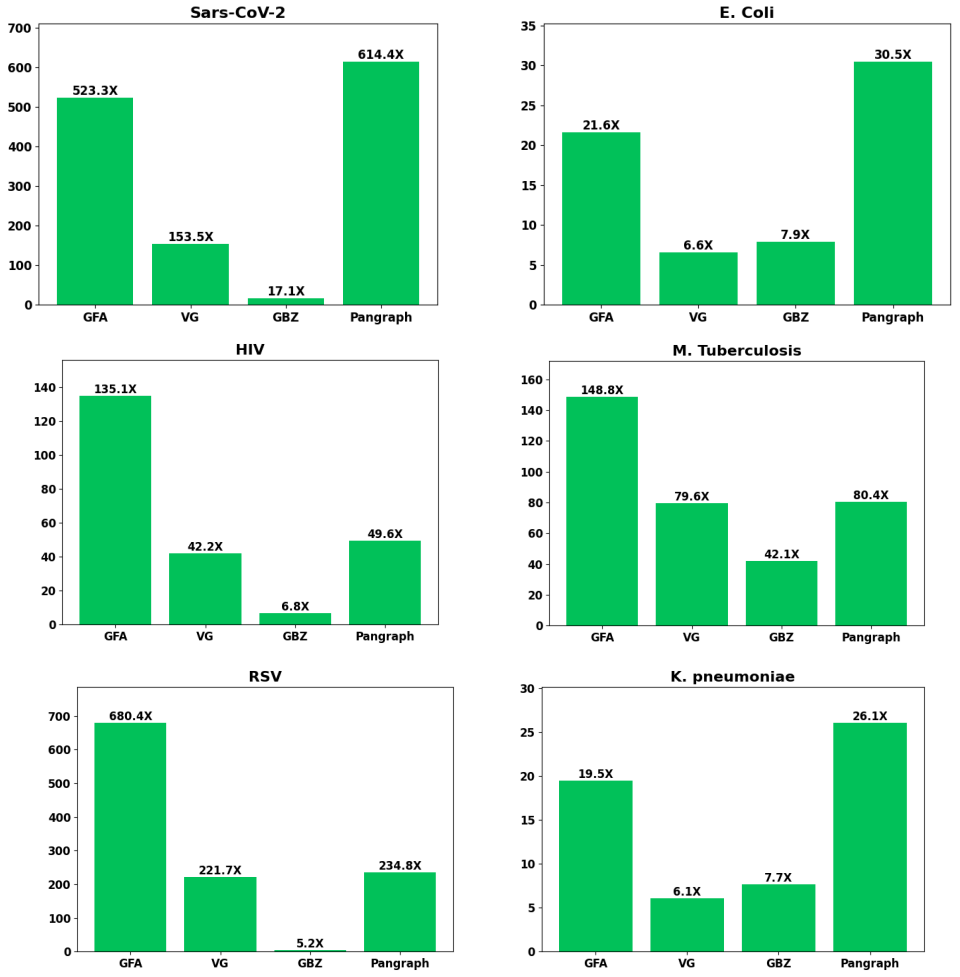
Mutations

Inferred MSA, Phylogeny, and **mutations** all in **one format!**
PanMAN is not just **information-rich** but also more **compact** and **scalable**

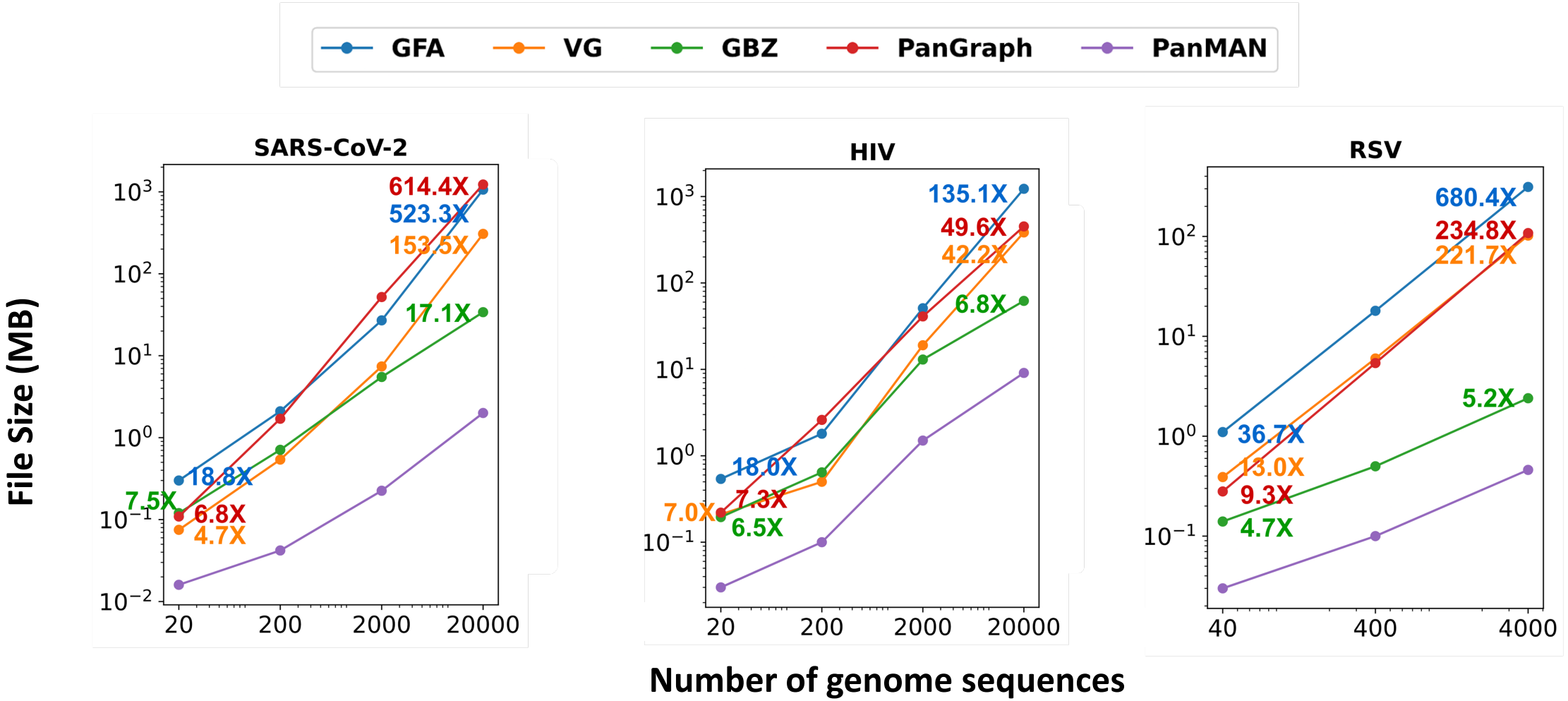
Compression of PanMAN versus Other File Formats

- Datasets:
  - 20K SARS-CoV-2
  - 1K E. Coli
  - 20K HIV
  - 400 M. Tuberculosis
  - 4K RSV
  - 1K Klebs

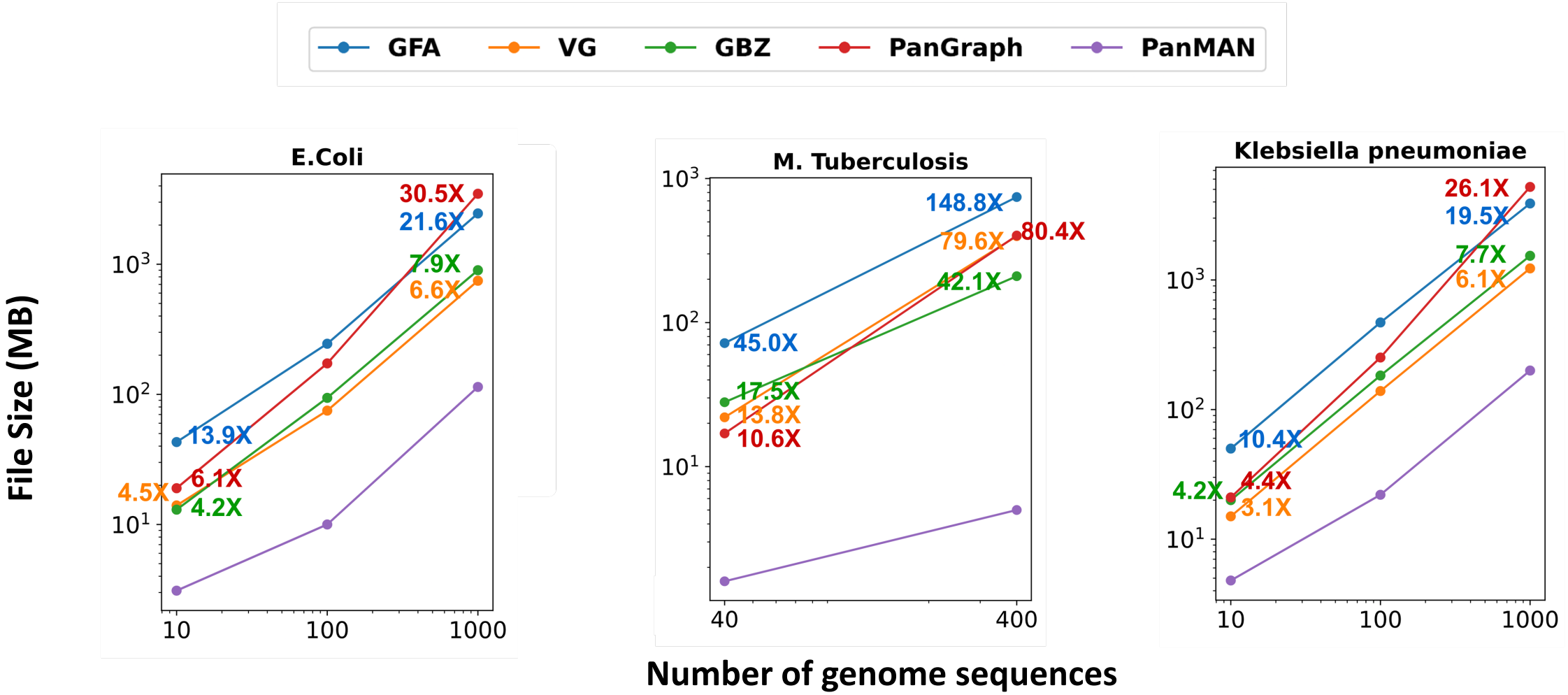
- Compression ratios:
  - **19–680x** over **GFA**
  - **6–152x** over **VG**
  - **5–42x** over **GBZ**
  - **26–614x** over **PanGraph**



PanMAN scales well relative to other formats



PanMAN scales well relative to other formats



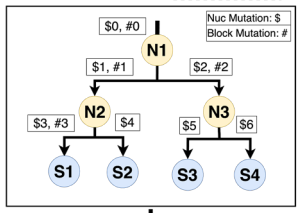
More Compression than BAM, CRAM, AGC, Miniphy, GZIP, XZ, etc.

File sizes (in MB)

Species	No. of sequences	FASTA	ZIP	GZIP	XZ	SAM	BAM	CRAM	AGC	MiniPhy	GFA (GZIP)	PanGraph (GZIP)	VCF (GZIP)	spVCF (GZIP)	PanMAN
SARS-CoV-2	20,000	581	138	138	2.6	573	169	5.3	57	1.6	305	195	19.2	21.2	2.27
	8,112,719	245,780	42125.4	42125.4	892.4	236227.1	62485.3	1224.4	1942.6	N/A	N/A	N/A	N/A	N/A	366
HIV	20,000	174	26	26	6.5	181	35	16	35	5.6	61	62	36.7	35.1	12.54
RSV	4,000	58	7.8	7.8	0.34	61	15	5.1	12	0.39	107	9.9	2.1	1.9	0.69
Escherichia Coli	1,000	4785	1434	1434	557	8678	2450	1119	1228.8	649	699	388	234.7	222.9	101.8
Mycobacterium Tuberculosis	400	1741	508	508	102	2458	646	154	377	102	166	42	8.9	9.2	5.15
Klebsiella pneumoniae	1,000	5427	1638	1638	877	13312	3584	1536	1332	870	1092	676	318.4	310.6	200.8

PanMAN Command-line Utilities

PanMAN `input.panman`



panmanUtils

Command: `./panmanUtils -I input.panman {opt}`

1) Raw Sequences

```
>S1
TACACGCA
>S2
TGCGCT
>S3
TGCGA
>S4
TGCGCA
```

`opt:--fasta`

2) MSA

```
>S1
TACACGCA
>S2
TGC--GCT
>S3
TGC--G-A
>S4
TGC--GCA
```

`opt:--fasta-aligned`

3) Sub-MSA

```
>S1
CACG
>S2
C--G
>S3
C--G
>S4
C--G
```

`opt:--index --reference S1 --start 2 --end 5`

4) m-WGA (MAF)

```
##maf

a
s S1 0 8 TACACGCA
s S2 0 6 TGC--GCT
s S3 0 5 TGC--G-A
s S4 0 6 TGC--GCA
…
```

`opt:--maf`

5) GFAs

```
S N1 T
S N2 G
S N3 A
S N4 C
S N5 AC
S N6 GC
P1 S1 N1 N3 N4 N5 N6 N3
P2 S2 N1 N2 N4 N6 N1
P3 S3 N1 N2 N4 N2 N3
P4 S4 N1 N2 N4 N6 N3
```

`opt:--gfa`

6) VCFs

```
#VCF
#reference=S2
```

POS	REF	ALT	S1	S2	S3
1	G	T	1	0	0
2	C	CAC	1	0	0
4	C	-	0	1	0
5	T	A	1	1	1

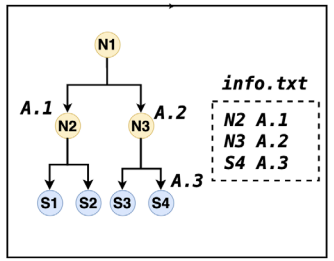
`opt:--vcf -n ref`

7) Summary

```
Average Height
Max Height
# Nodes
# Substitutions
# Deletions
# Insertions
...
```

`opt:--summary`

8) Annotate

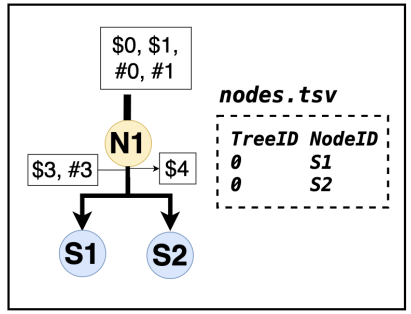


info.txt

```
N2 A.1
N3 A.2
S4 A.3
```

`opt:--annotate -i info.txt`

9) sub-PanMAN



nodes.tsv

TreeID	NodeID
0	S1
0	S2

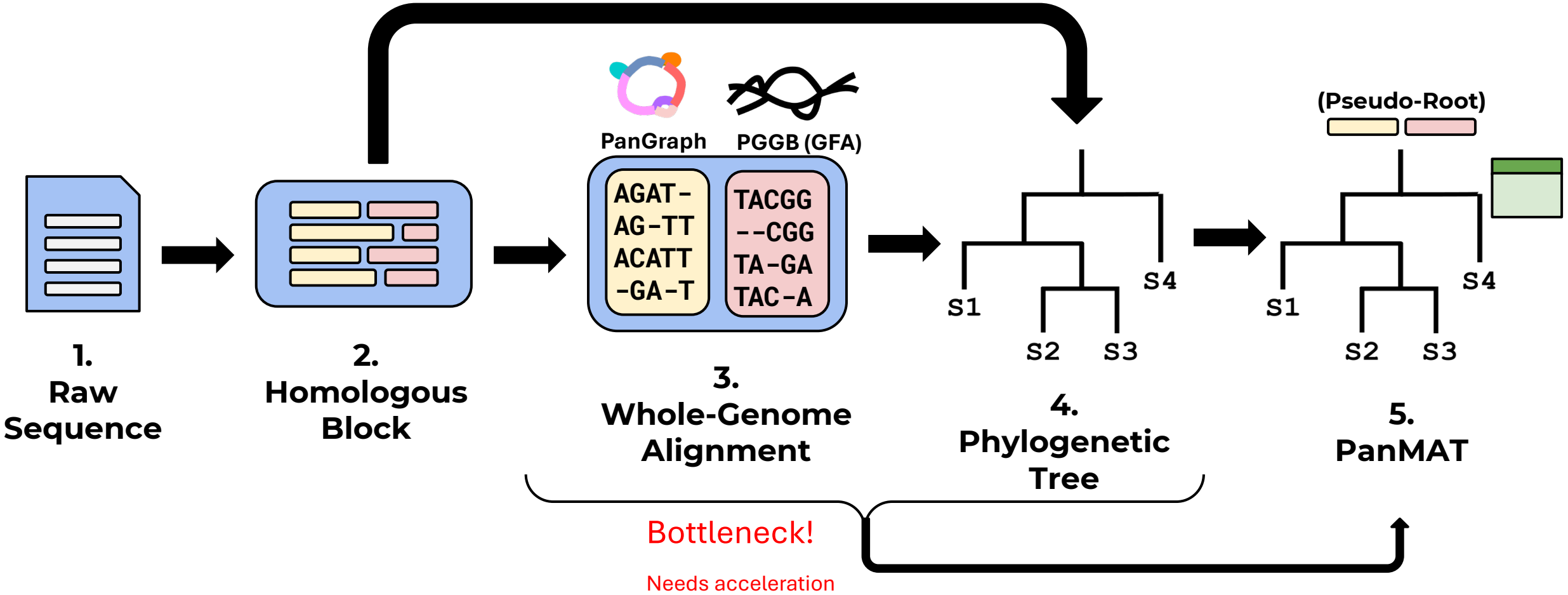
`opt:--subnet -i nodes.tsv`

10) AA translation

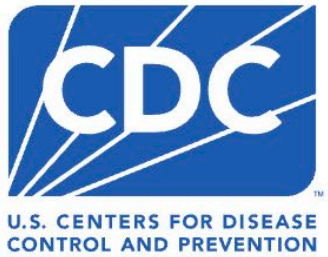
Node-ID	aa-mutations	codon-change
S3	S:0:Y;	TGC>TAC;

`opt:--aa-mutations`

PanMAT *De Novo* Construction Process

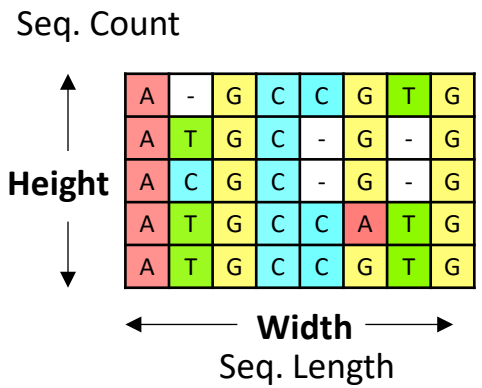


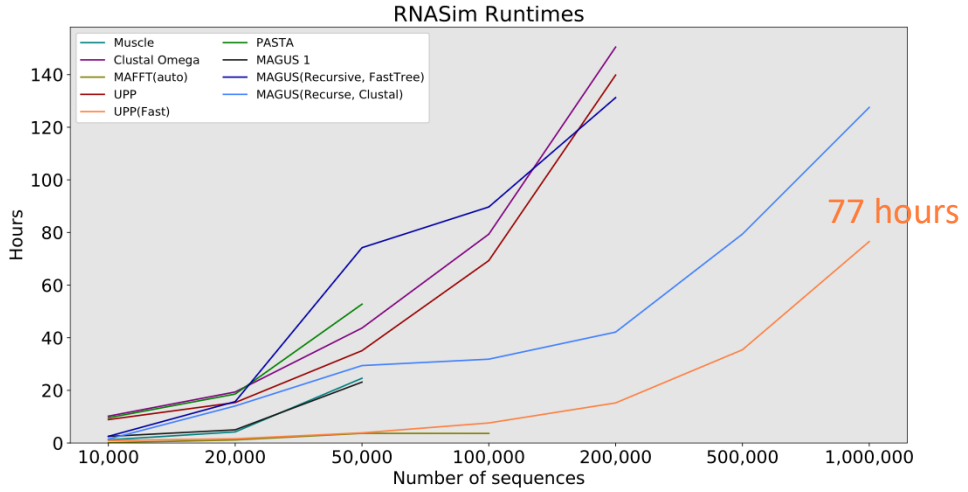
Acknowledgments

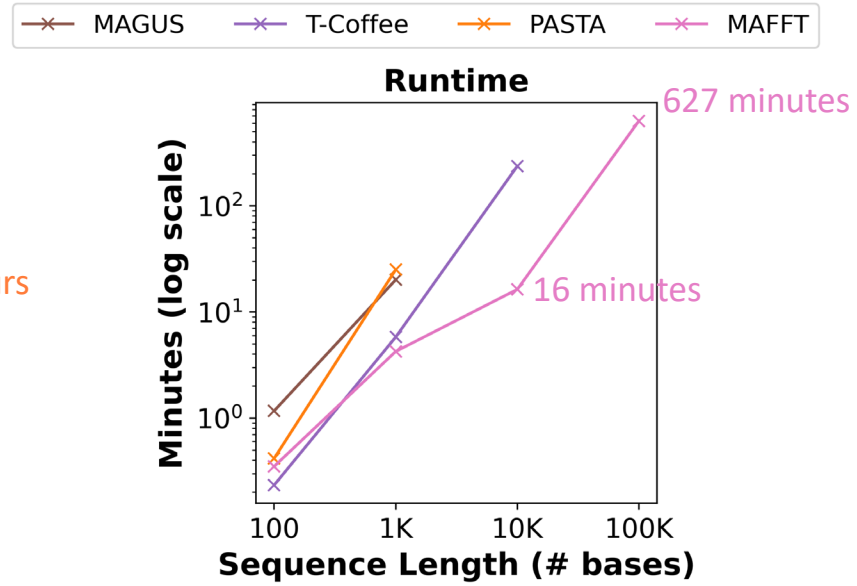


Accelerating MSAs

Current MSA tools struggle with handling tall and wide alignment







Accelerating MSAs on GPUs

Bioinformatics, 2025, **41**, i332–i341
https://doi.org/10.1093/bioinformatics/btaf212
ISMB/ECCB 2025 Supplement

OXFORD

Ultrafast and ultralarge multiple sequence alignments using TWILIGHT

Yu-Hsiang Tseng[1,*], Sumit Walia[1], Yatish Turakhia[1,*]

[1]Department of Electrical and Computer Engineering, University of California San Diego, San Diego, CA 92093, United States

*Corresponding authors. Yu-Hsiang Tseng, Department of Electrical and Computer Engineering, University of California San Diego, 9500 Gilman Drive, San Diego, CA 92093, United States. E-mail: y3tseng@ucsd.edu; Yatish Turakhia, Department of Electrical and Computer Engineering, University of California San Diego, 9500 Gilman Drive, San Diego, CA 92093, United States. E-mail: yturakhia@ucsd.edu.

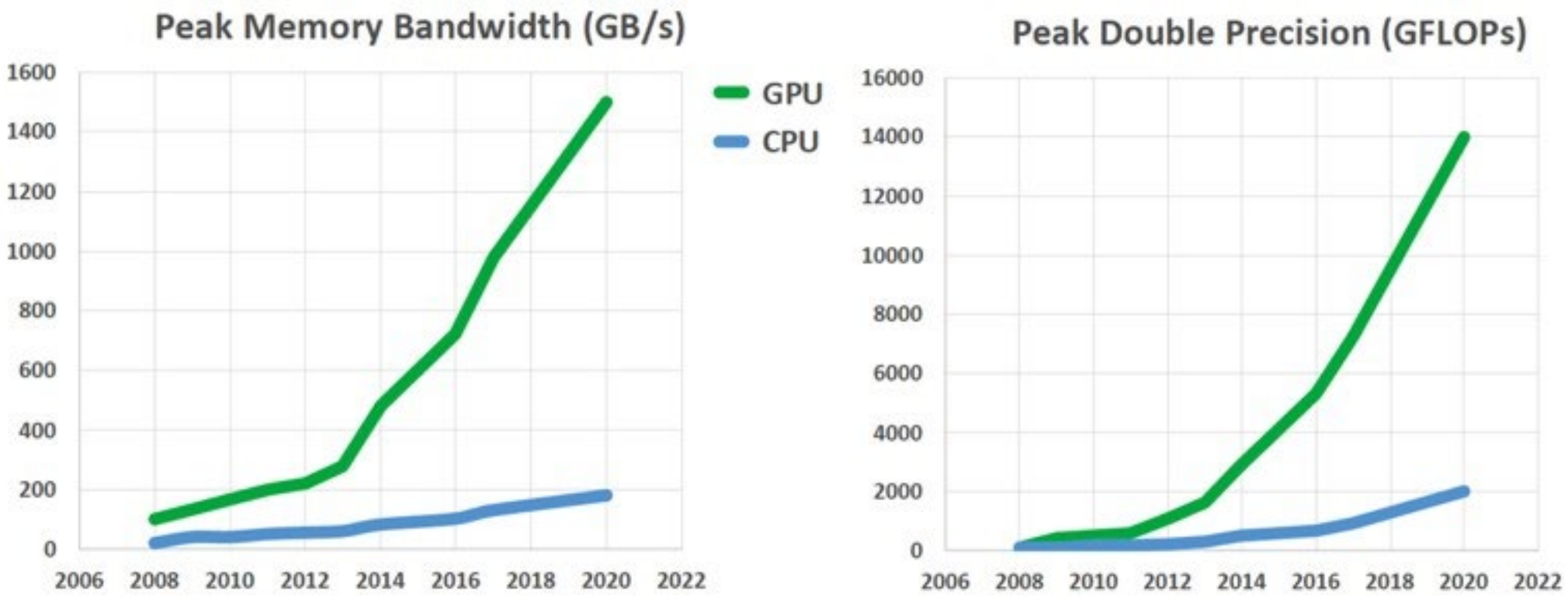
TWILIGHT

https://github.com/TurakhiaLab/TWILIGHT

Yu-Hsiang Tseng

Sumit Walia

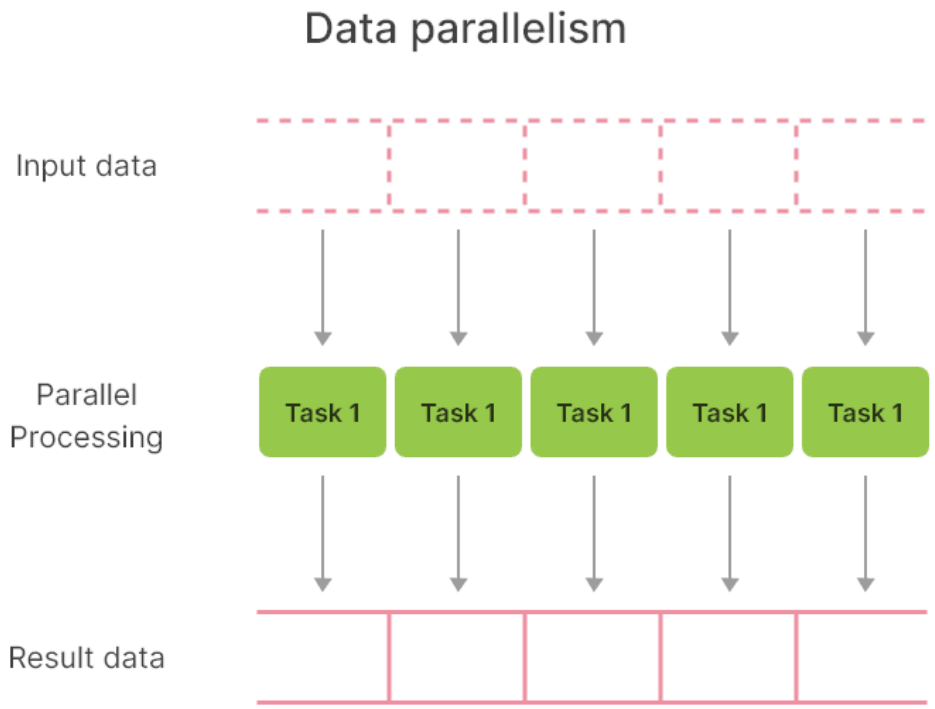
Why GPUs?



Source: The Next Platform

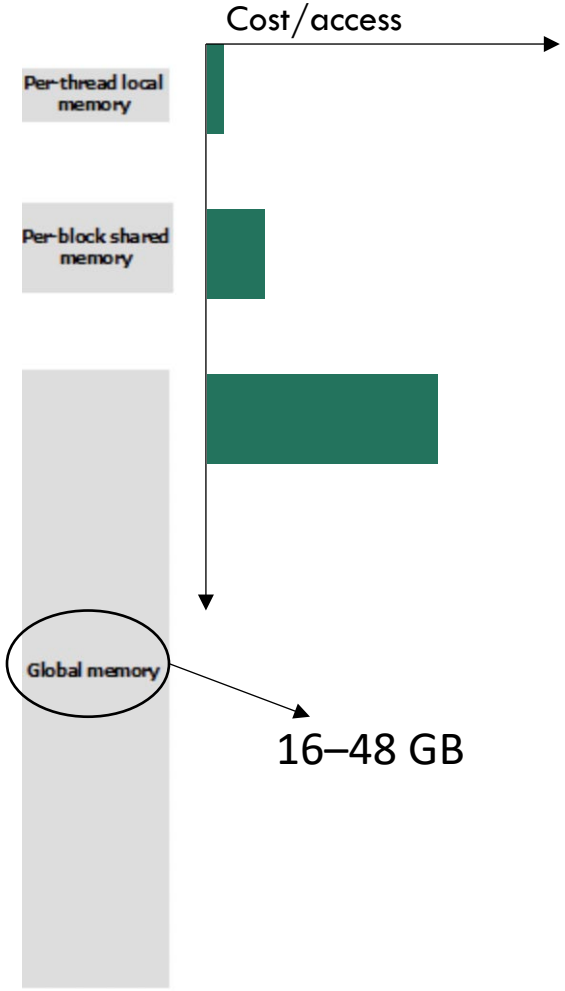
Challenges with GPU Acceleration

- Requires high **data parallelism**
  - same operation on large datasets



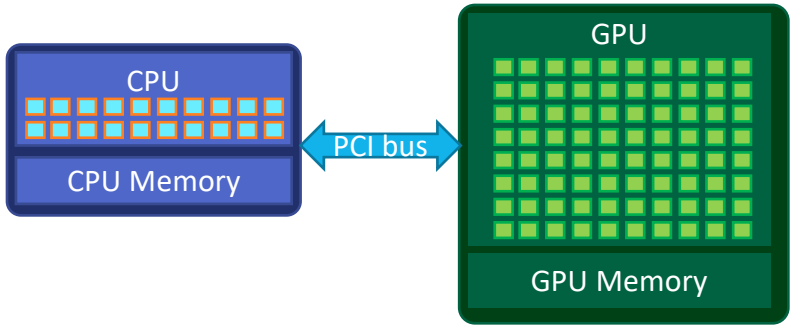
Challenges with GPU Acceleration

- Requires high **data parallelism**
  - same operation on large datasets
- Programmers **manage memory** (explicit control of *global*, *shared*, and *local* memory)
  - Global memory capacity is more limited than a CPU
  - Global memory has high bandwidth but is latency prone
  - Shared and local memories are fast (low latency) but have small capacity

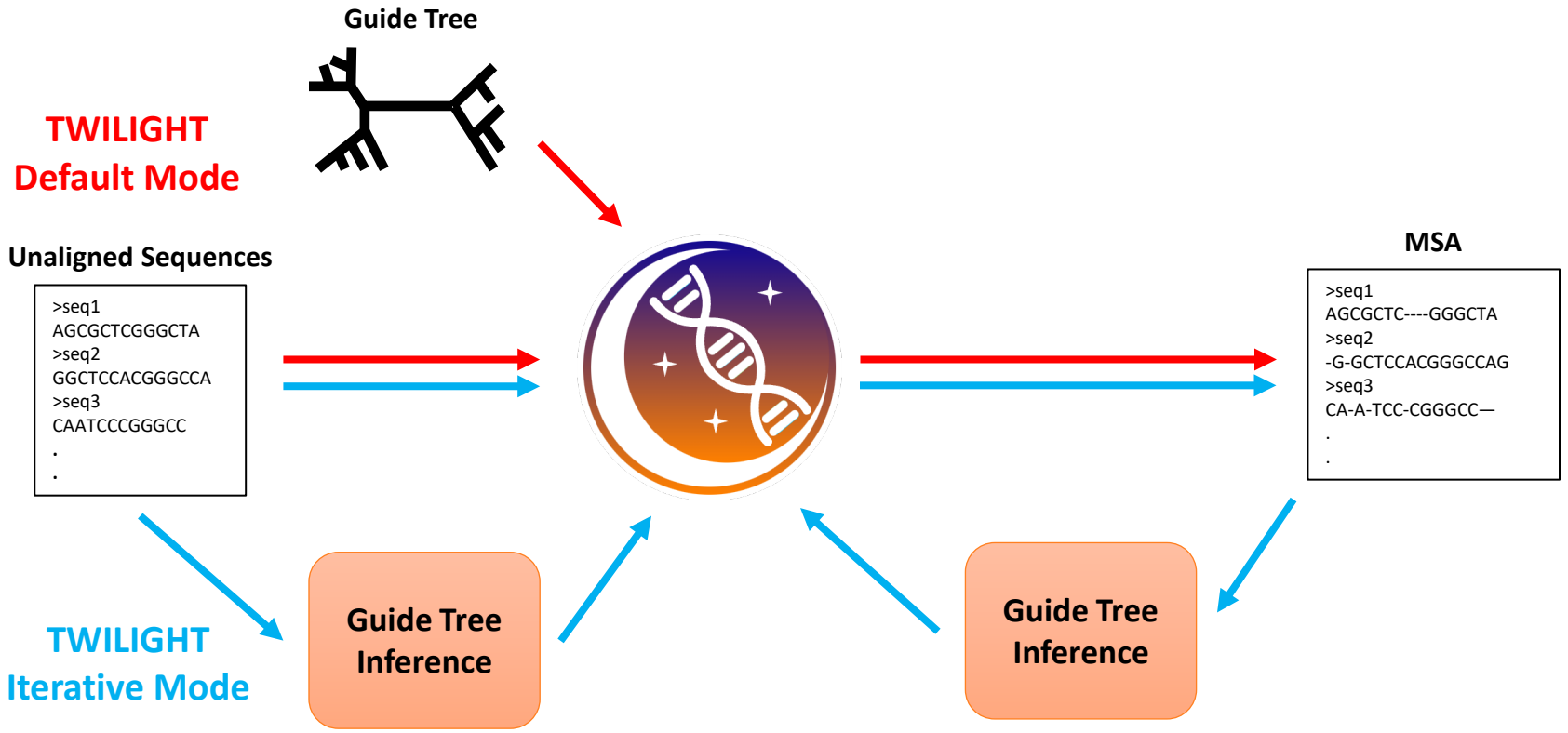


Challenges with GPU Acceleration

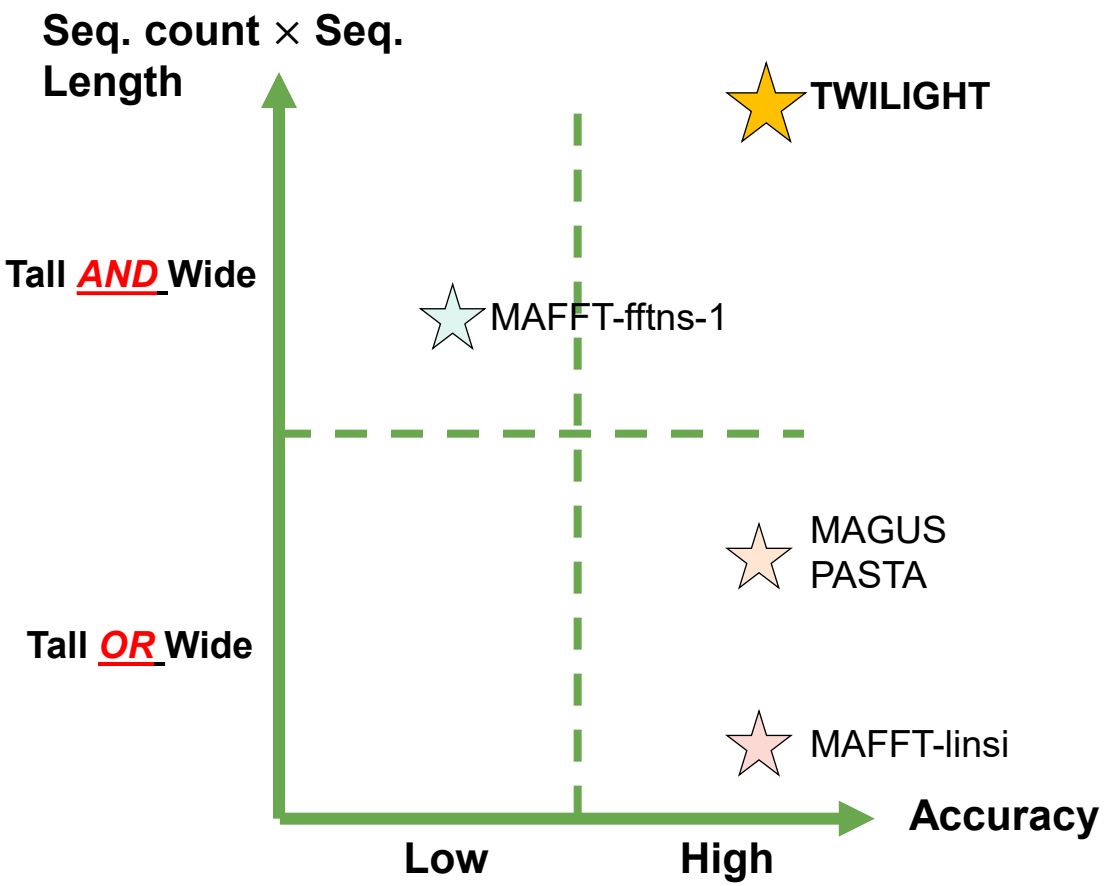
- Requires high **data parallelism**
  - same operation on large datasets
- Programmers **manage memory** (explicit control of *global*, *shared*, and *local* memory)
  - Global memory capacity is more limited than a CPU
  - Global memory has high bandwidth but is latency prone
  - Shared and local memories are fast (low latency) but have small capacity
- **High communication cost** with CPU
- **Algorithms** have to be adapted to this cost model
  - increase parallelism
  - exploit local/shared memories
  - optimize memory access patterns
  - minimize data transfers with CPU

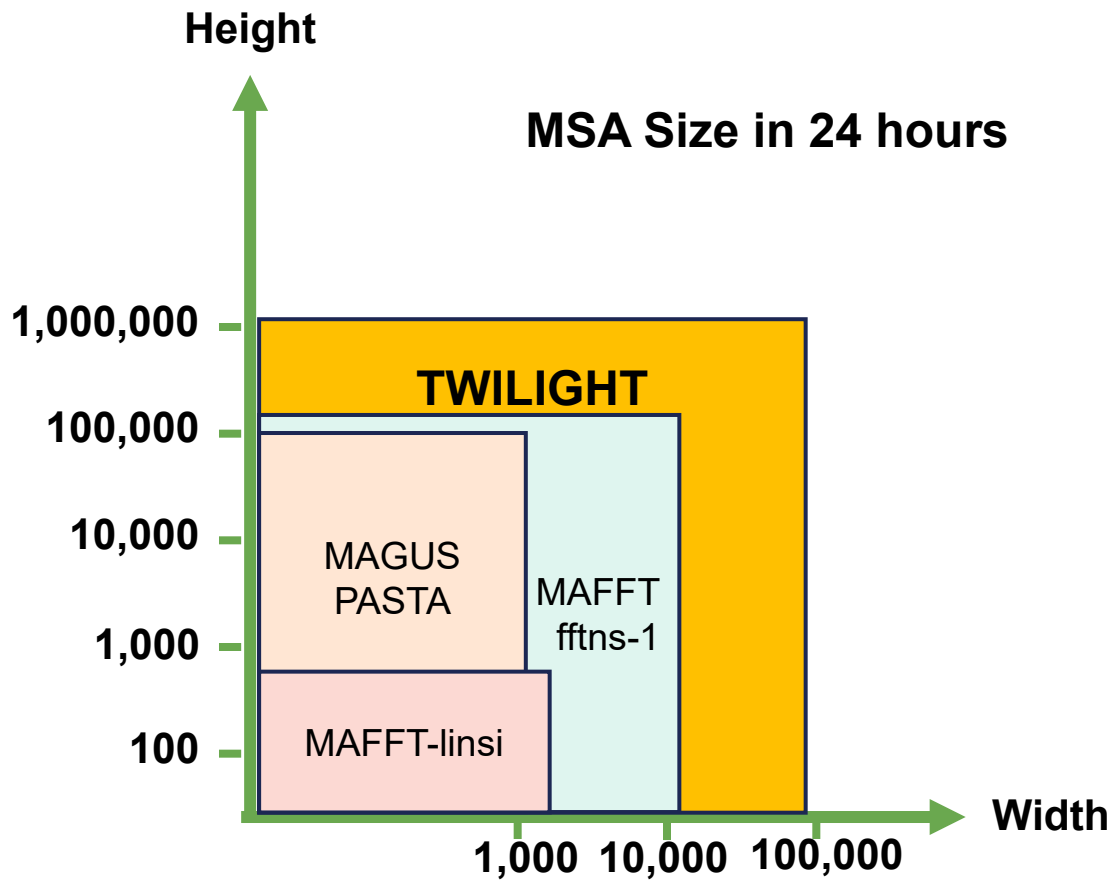


TWILIGHT: Tall and WIde ALIGnment at High Throughput



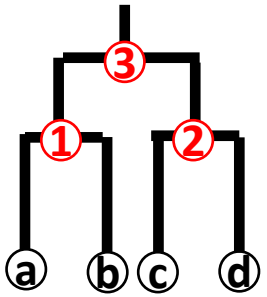
TWILIGHT Overview





TWILIGHT: High-Level Approach

- **Progressive alignment** is used in TWILIGHT to build MSAs




```
a:ACGT
b:AGT
c:ACT
d:ACTT
```

①

```
a:ACGT
b:A-GT
```

②

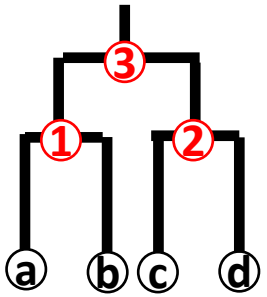
```
c:ACT-
d:ACTT
```

③

```
a:ACGT-
b:A-GT-
c:AC-T-
d:AC-TT
```

TWILIGHT: High-Level Approach

- **Progressive alignment** is used in TWILIGHT to build MSAs
- Use **Profiles** to represent alignments




```
a:ACGT
b:AGT
c:ACT
d:ACTT
```

①

```
a:ACGT
b:A-GT
```

②

```
c:ACT-
d:ACTT
```

③

```
a:ACGT-
b:A-GT-
c:AC-T-
d:AC-TT
```

Alignment

A	C	G	C	T
-	C	-	T	T
A	C	-	C	-
A	C	-	C	T
A	C	-	C	T

⟹

Profile

A	0.8	0	0	0	0
C	0	1.0	0	0.8	0
G	0	0	0.2	0	0
T	0	0	0	0.2	0.8
-	0.2	0	0.8	0	0.2

Needleman-Wunsch algorithm
Affine Gap Penalty

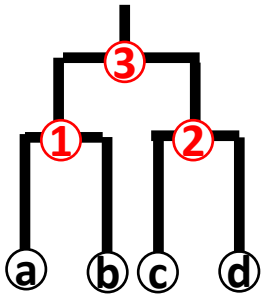
$$H(i,j) = \max \begin{cases} H(i-1,j-1) + ps(i,j) \\ I(i-1,j-1) + ps(i,j) \\ D(i-1,j-1) + ps(i,j) \end{cases}$$

$$I(i,j) = \max \begin{cases} H(i-1,j) + gop_{A_i} \\ I(i-1,j) + gep_{A_i} \end{cases}$$

$$D(i,j) = \max \begin{cases} H(i,j-1) + gop_{B_j} \\ D(i,j-1) + gep_{B_j} \end{cases}$$

TWILIGHT: High-Level Approach

- **Progressive alignment** is used in TWILIGHT to build MSAs
- Use **Profiles** to represent alignments
- Affine-gap penalty with **position-specific gap penalty** (Julie D. Thompson, 1994)




```
a:ACGT
b:AGT
c:ACT
d:ACTT
```

①	②	③
a:ACGT	c:ACT-	a:ACGT-
b:A-GT	d:ACTT	b:A-GT-
		c:AC-T-
		d:AC-TT

Alignment

A	C	G	C	T
-	C	-	T	T
A	C	-	C	-
A	C	-	C	T
A	C	-	C	T

Profile

A	0.8	0	0	0	0
C	0	1.0	0	0.8	0
G	0	0	0.2	0	0
T	0	0	0	0.2	0.8
-	0.2	0	0.8	0	0.2

GOP: -50 GOP: -20

Position-specific gap penalty

Needleman-Wunsch algorithm
Affine Gap Penalty

$$H(i,j) = \max \begin{cases} H(i-1,j-1) + ps(i,j) \\ I(i-1,j-1) + ps(i,j) \\ D(i-1,j-1) + ps(i,j) \end{cases}$$

$$I(i,j) = \max \begin{cases} H(i-1,j) + gop_{A_i} \\ I(i-1,j) + gep_{A_i} \end{cases}$$

$$D(i,j) = \max \begin{cases} H(i,j-1) + gop_{B_j} \\ D(i,j-1) + gep_{B_j} \end{cases}$$

TWILIGHT: Challenges

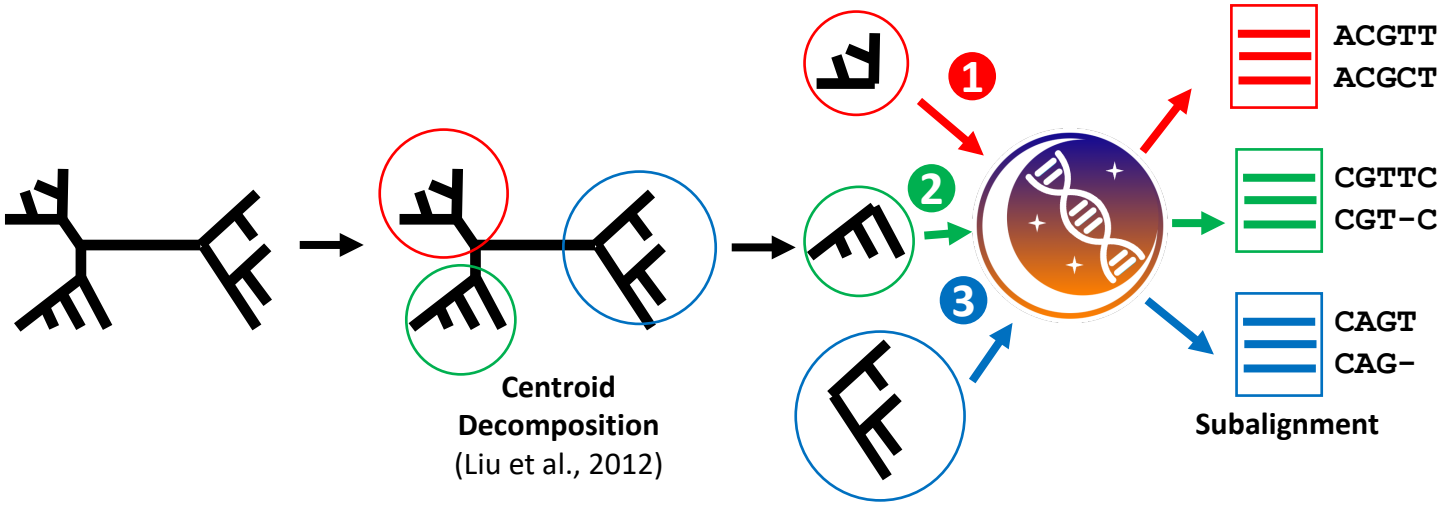
1. Prohibitive **memory usage** for ultralarge MSAs

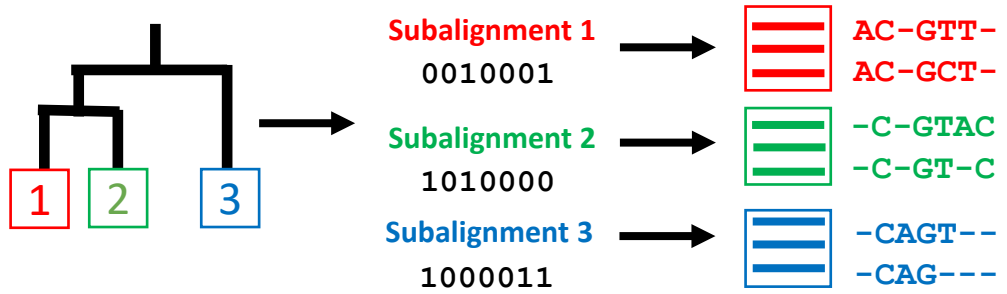
Requires 240 GB just to store raw sequences!
Alignment expansion increases this further (>1TB)

GPU memory: 16-48 GB

TWILIGHT: Challenges

1. Prohibitive **memory usage** for ultralarge MSAs
   - **Solution**: Divide and conquer strategy

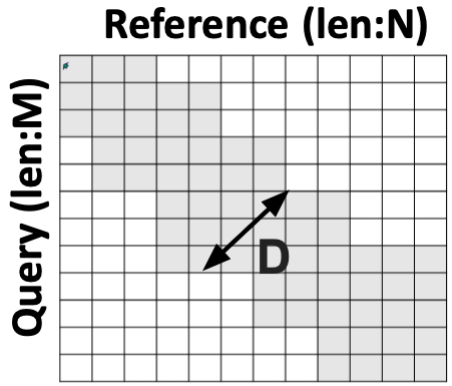




TWILIGHT: Challenges

1. Prohibitive **memory usage** for ultralarge MSAs
   - **Solution**: Divide and conquer strategy

2. **Traceback memory** requirements scale with alignment length



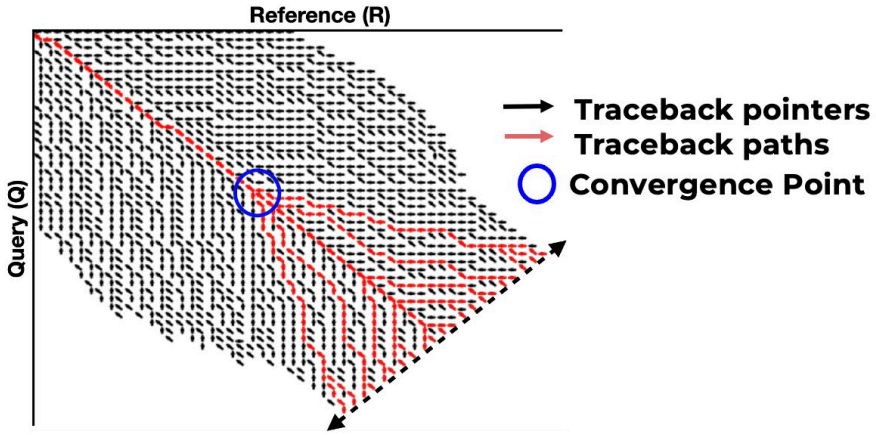
Space Complexity: O(ND)

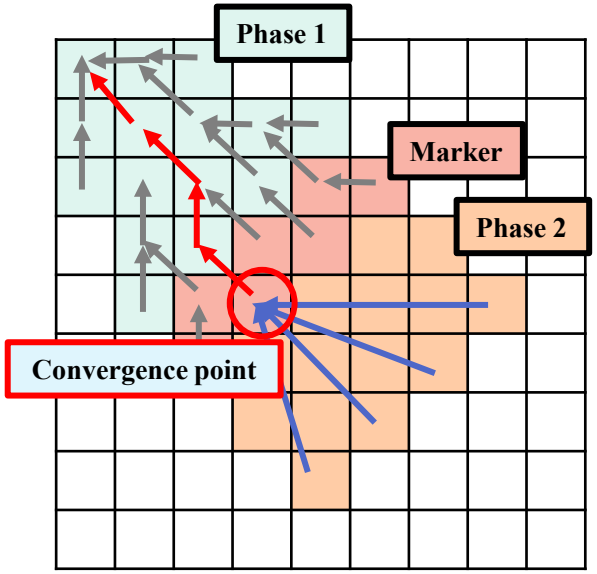
Example: X-Drop algorithm

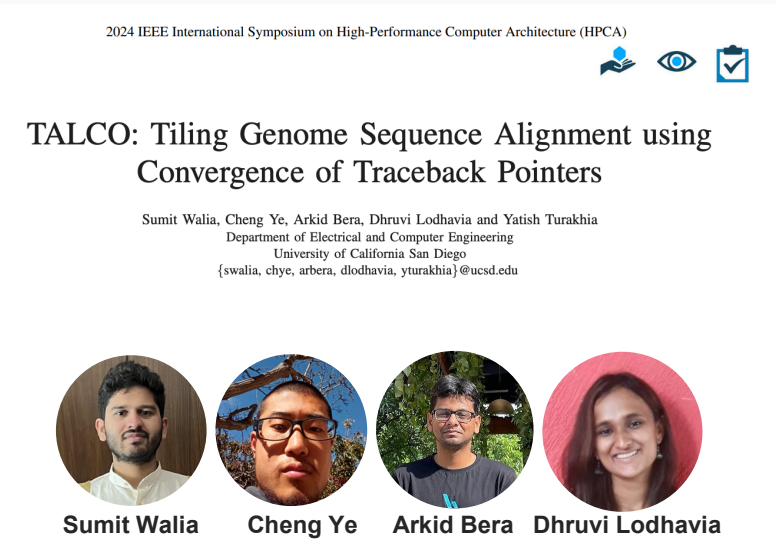
TWILIGHT: Challenges

1. Prohibitive **memory usage** for ultralarge MSAs
   - **Solution**: Divide and conquer strategy

2. **Traceback memory** requirements scale with alignment length
   - **Solution**: TALCO tiling strategy



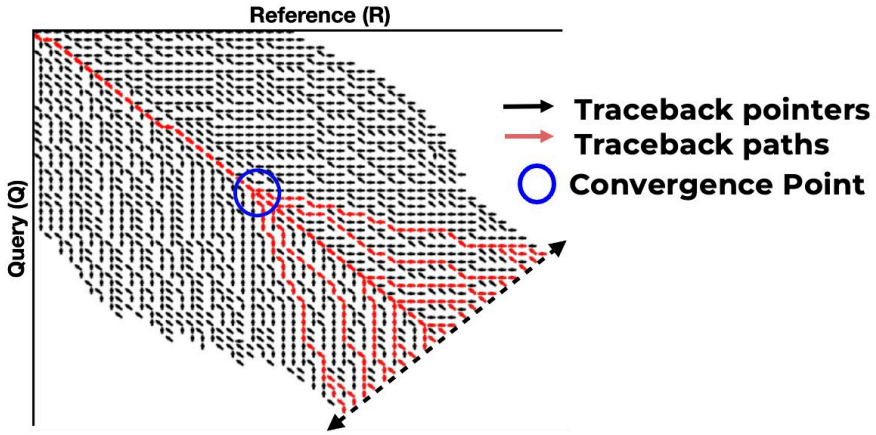


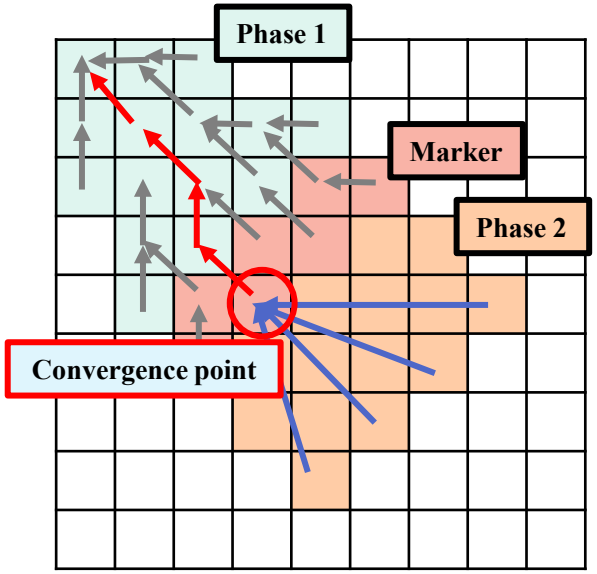


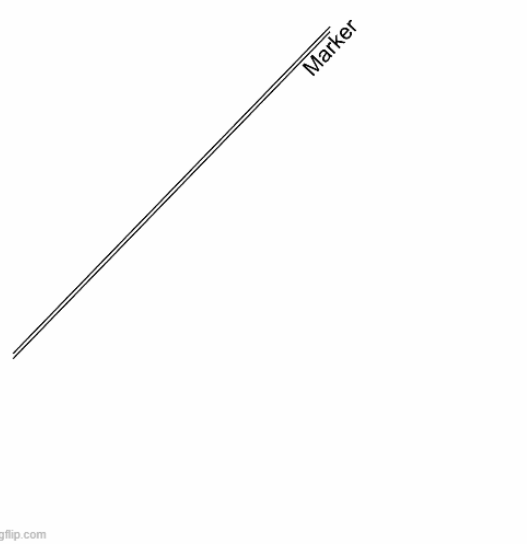
TWILIGHT: Challenges

1. Prohibitive **memory usage** for ultralarge MSAs
   - **Solution**: Divide and conquer strategy

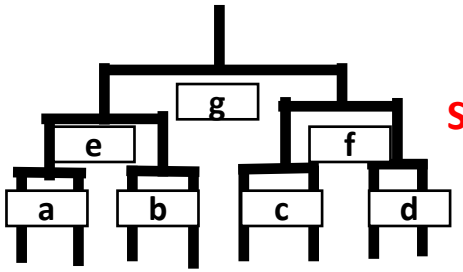
2. **Traceback memory** requirements scale with alignment length
   - **Solution**: TALCO tiling strategy





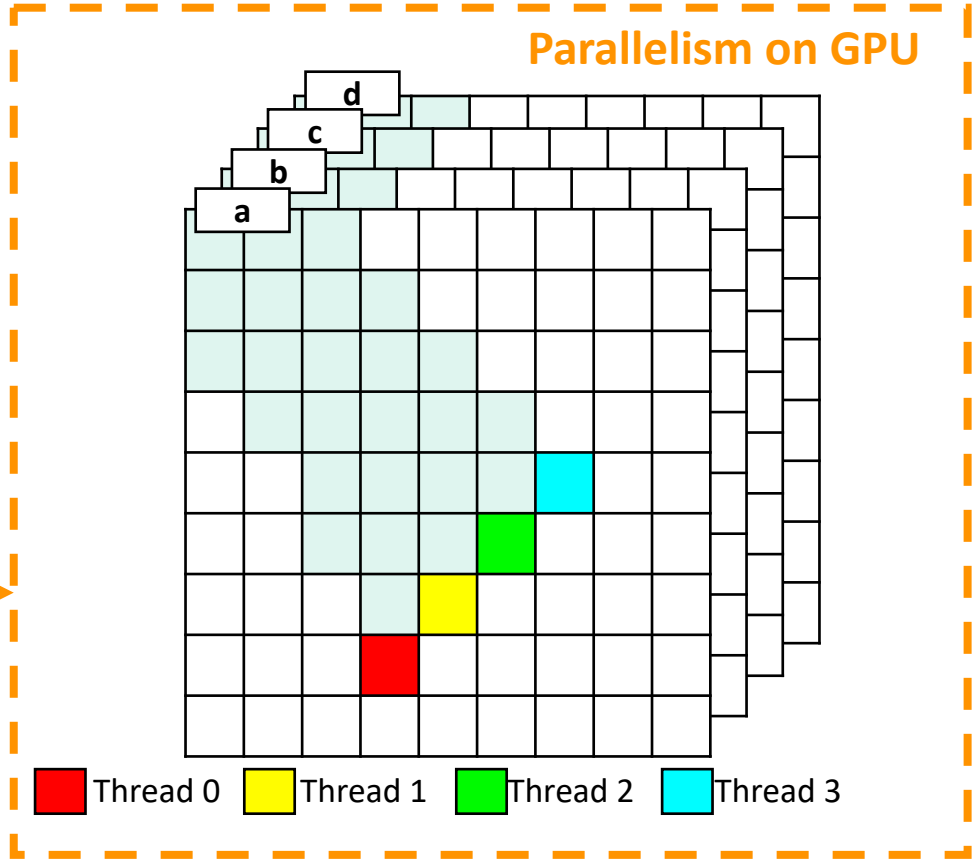


TWILIGHT: GPU Parallelization



Scheduling

Order	Nodes
1	a, b, c, d
2	e, f
3	g



TWILIGHT: Results

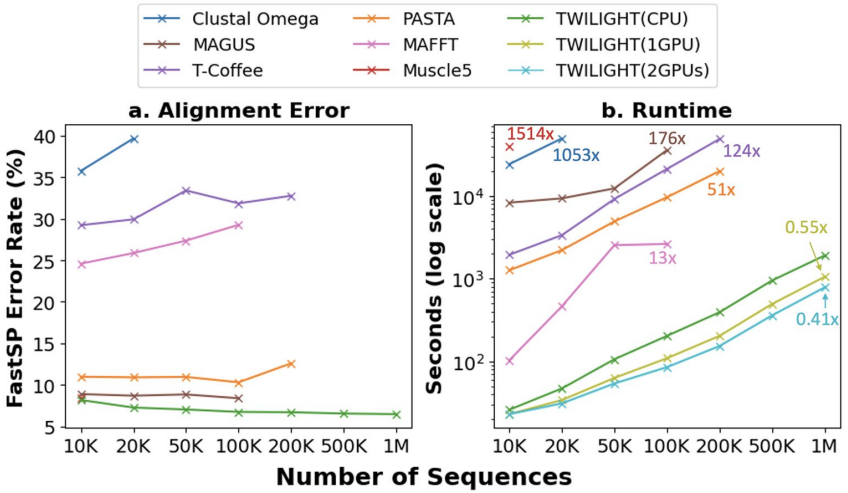
1. Handles much larger number of sequences (**tall** alignments) and sequence lengths (**wide** alignments)

2. Offers **high speedup** and **accuracy** compared to existing tools

3. *Only* tool to complete an MSA of **8-million SARS-CoV-2 genomes**
   - **28 hours** on 2 GPUs using the UShER guide tree

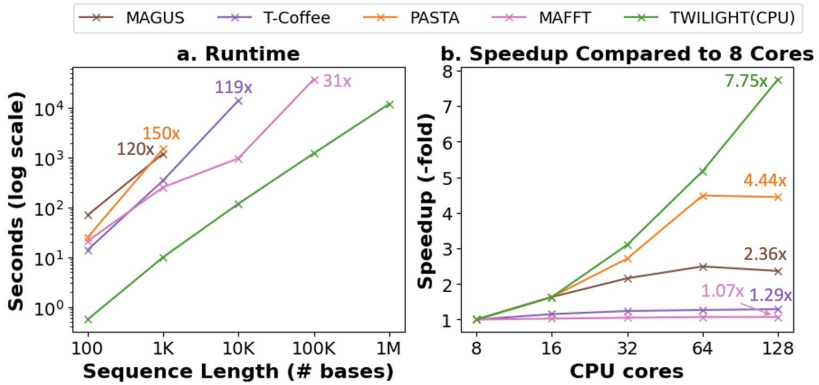
4. Supports **multi-GPU acceleration** on **NVIDIA** and **AMD** GPUs

5. Supports **protein alignments**



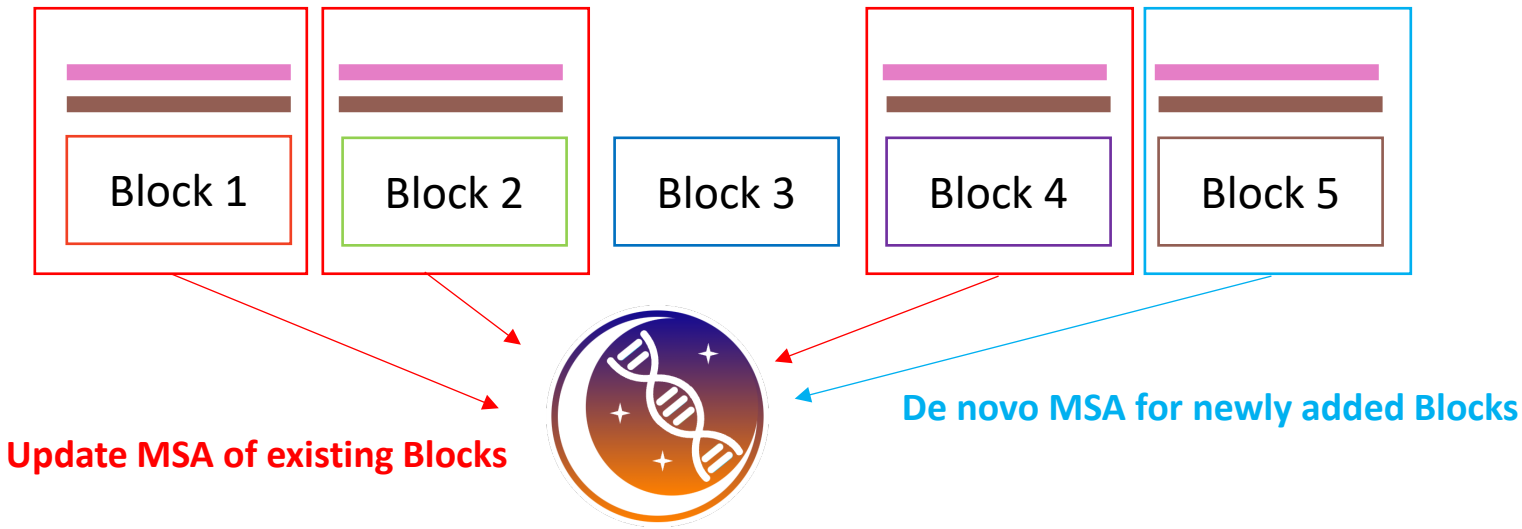
32 min (CPU)

13 min (2GPUs)



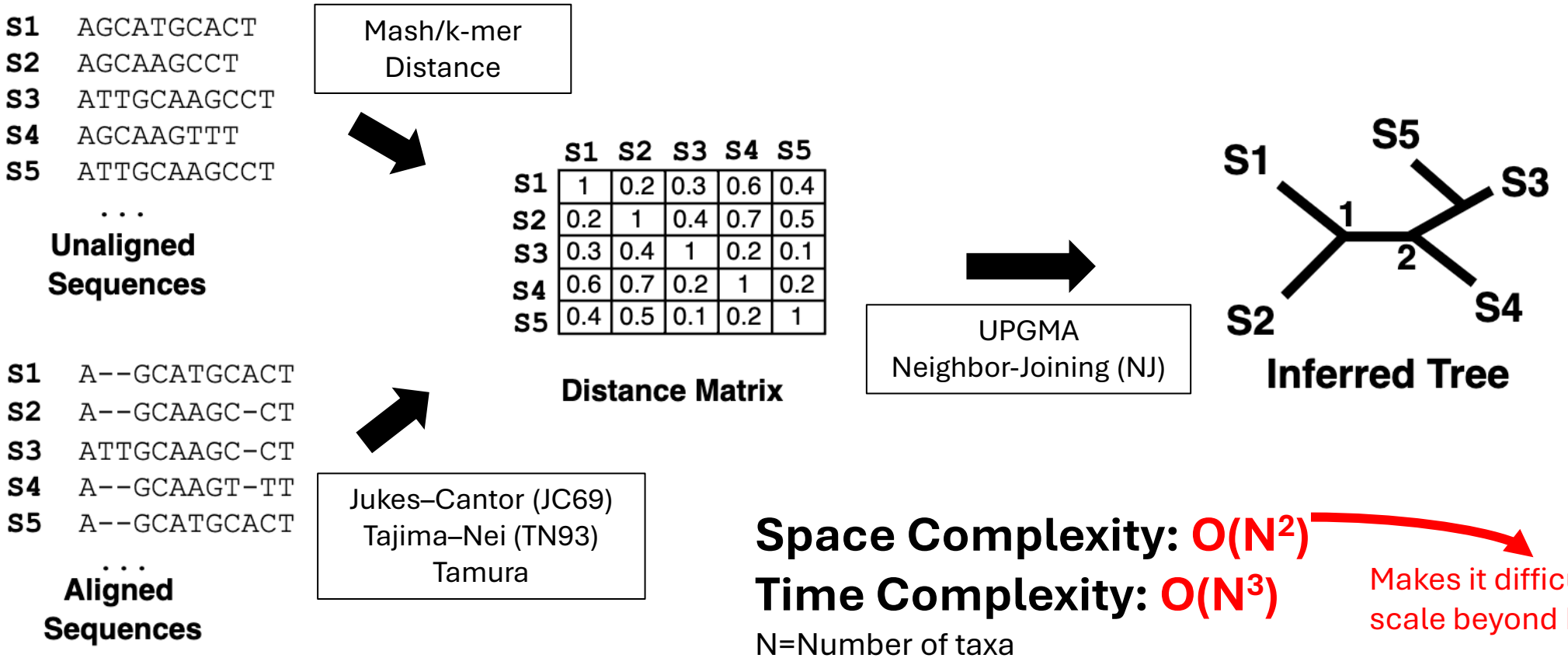
Future Work

- **Multiple Whole-Genome Alignment (m-WGA)** with TWILIGHT
  - Account for non-linear rearrangements (translocations, inversions, duplications, large indels, etc.)



Accelerating Phylogenetics

Distance-based Phylogenetics



Space Complexity: $O(N^2)$

Time Complexity: $O(N^3)$

N=Number of taxa

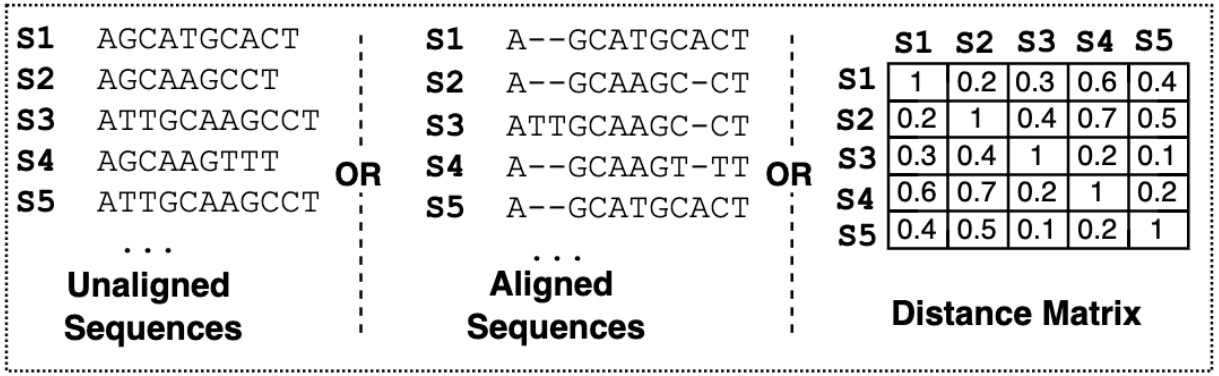
Makes it difficult to scale beyond $N \sim 10^5$

DIPPER: Distance-based Phylogenetic Placer

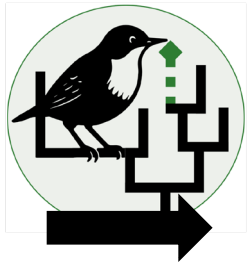

Sumit Walia

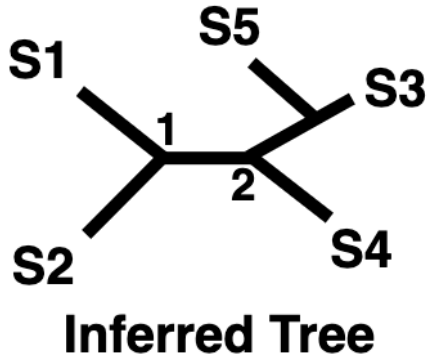

Zexing Chen


Yu-Hsiang Tseng



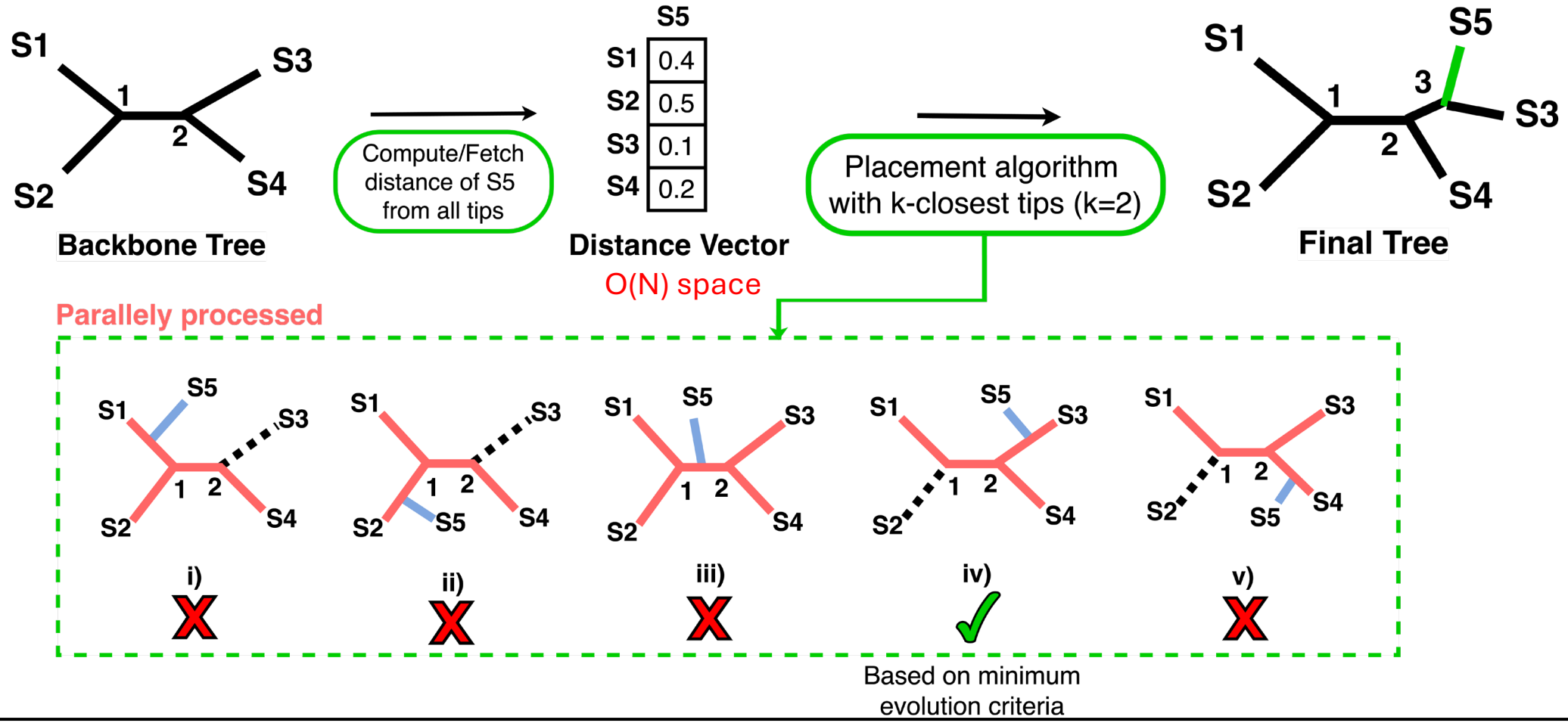
	S1	S2	S3	S4	S5
S1	1	0.2	0.3	0.6	0.4
S2	0.2	1	0.4	0.7	0.5
S3	0.3	0.4	1	0.2	0.1
S4	0.6	0.7	0.2	1	0.2
S5	0.4	0.5	0.1	0.2	1



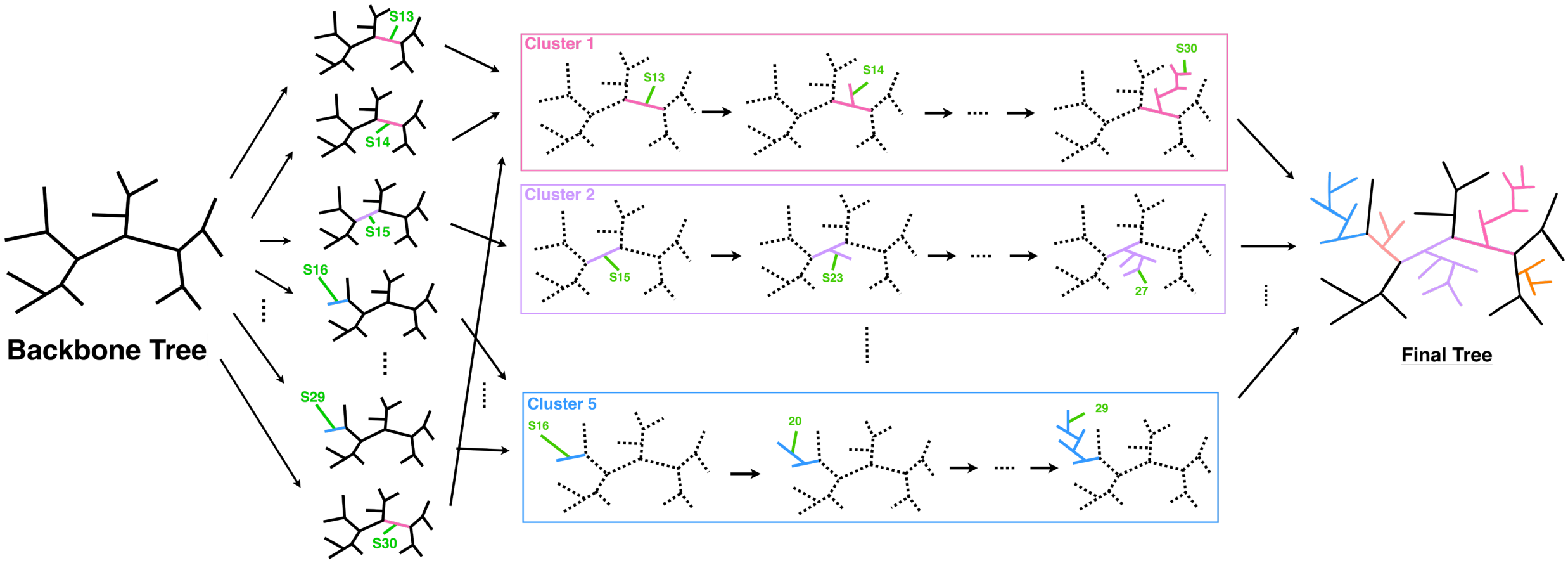


(Preprint releasing soon!)

DIPPER: Overview



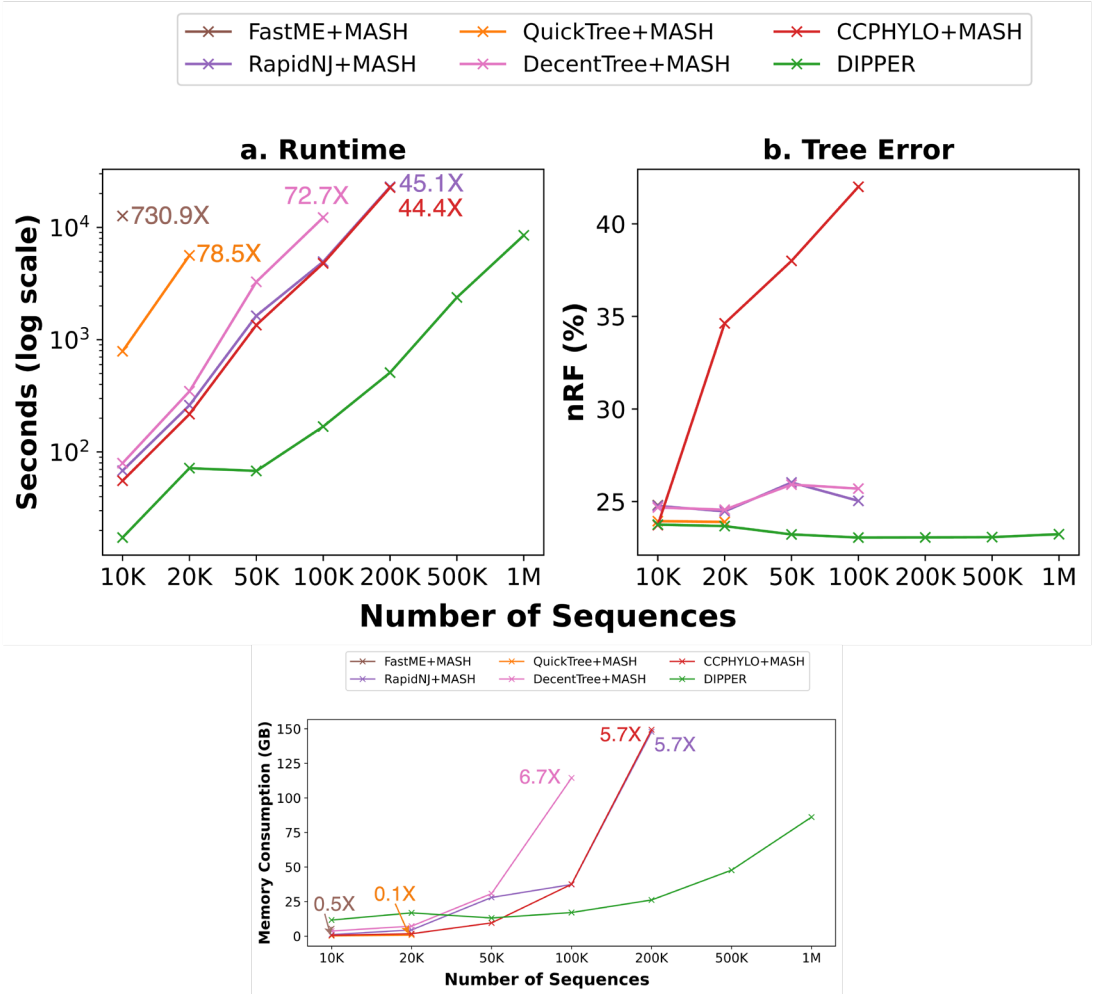
DIPPER: Overview



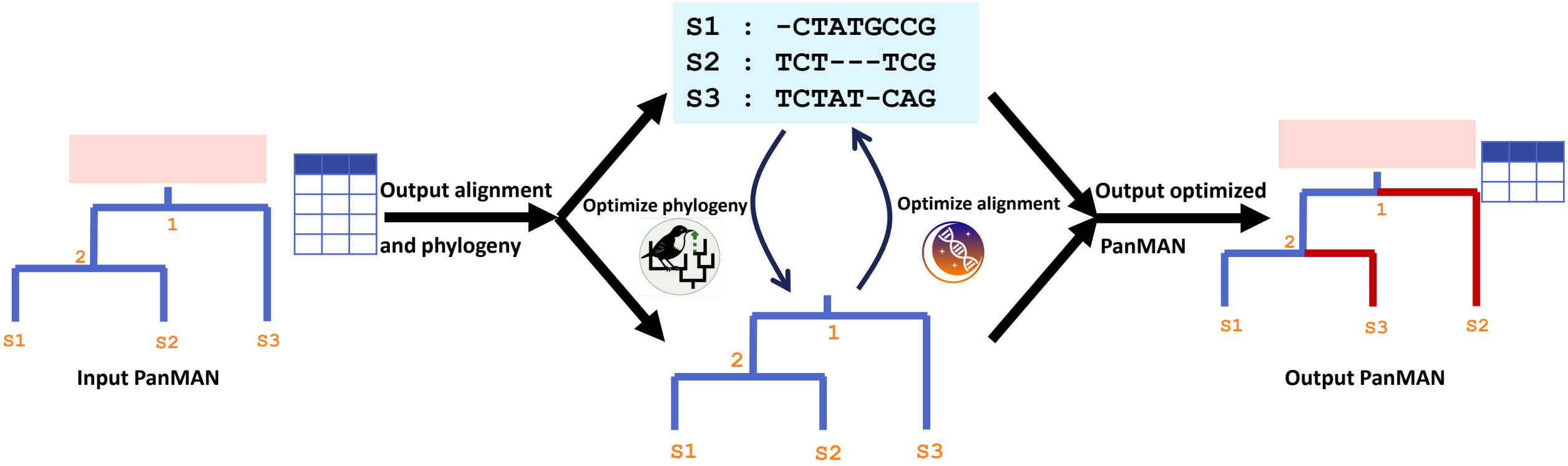
https://github.com/TurakhiaLab/DIPPER

DIPPER: Contributions

- **GPU-accelerated** distance-based phylogenetic **placement** algorithm
  - O(N) memory and O(N.log N) runtime
- **Accuracy** comparable or better than conventional Neighbor-Joining
- **Faster** and more **scalable**
  - **1M sequences** analyzed in **2.4 hours; 10M sequences** analyzed in under **7 hours**
- **Lower memory** requirement



Future Vision: Alignment-Phylogeny Co-optimization Framework Under PanMAN



Summary

- **Significant results:**
  1. **Largest SARS-CoV-2 pangenome** with ~8M genomes stored in only **366MB** with **PanMAN** (the most compressible pangenome format with increased representative power)
  2. Largest **MSA of ~8M SARS-CoV-2 genomes** analyzed in under **30 hours** using **TWILIGHT**
  3. **Distance-based phylogeny** of **~10M sequences** estimated in under **7 hours** with only 1 GPU using **DIPPER**

- **Coming Soon**:
  1. Ultralarge **pangenome construction** and **alignment-phylogeny co-estimation** of diverse species under the PanMAN framework

All tools available at: https://github.com/TurakhiaLab

Thank you for listening!