UC San Diego
Electrical and Computer Engineering
JACOBS SCHOOL OF ENGINEERING

ISMB/ECCB 2025 JULY 20-24 LIVERPOOL

TURAKHIA LAB

Ultrafast and Ultralarge Multiple Sequence Alignments using TWILIGHT

Yu-Hsiang Tseng, Sumit Walia and Yatish Turakhia
University of California San Diego

Outline

- Multiple sequence alignment: **applications** and **limitations**
- TWILIGHT: **T**all and **Wi**de A**lig**nments at **H**igh **T**hroughput
- **Key Contributions** and **Results**
- **Conclusion** and **Future Work**
- **Demo**

Outline

- Multiple sequence alignment: **applications** and **limitations**
- TWILIGHT: **T**all and **Wi**de A**lig**nments at **H**igh **T**hroughput
- **Key Contributions** and **Results**
- **Conclusion** and **Future Work**
- **Demo**

Multiple Sequence Alignment (MSA)

- **Multiple sequence alignment:**
  Given a set of sequences, insert gaps (“–”) into each sequence to align ***homologous characters*** across all sequences, maximizing overall similarity and preserving evolutionary or structural relationships.

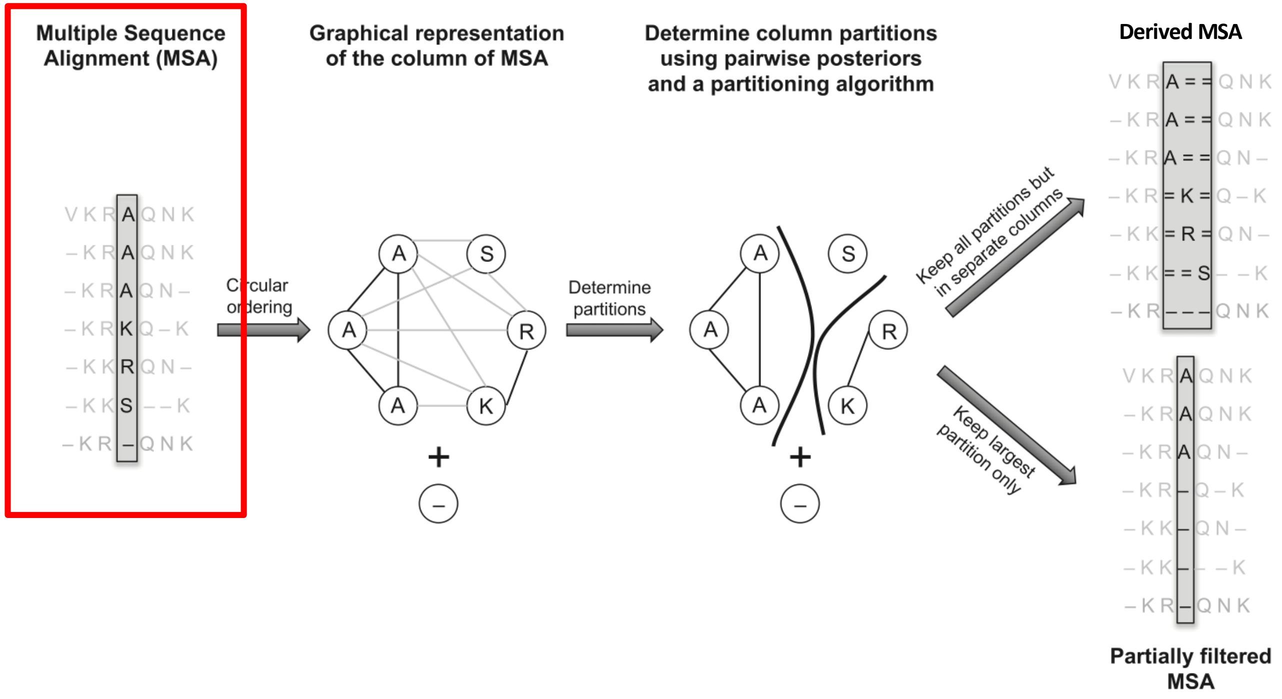
S1: AGCCGTG
S2: ATGCGG
S3: ACGCGG
S4: ATGCCATG
S5: ATGCCGTG

→

S1	A	-	G	C	C	G	T	G
S2	A	T	G	C	-	G	-	G
S3	A	C	G	C	-	G	-	G
S4	A	T	G	C	C	A	T	G
S5	A	T	G	C	C	G	T	G

MSA: Applications

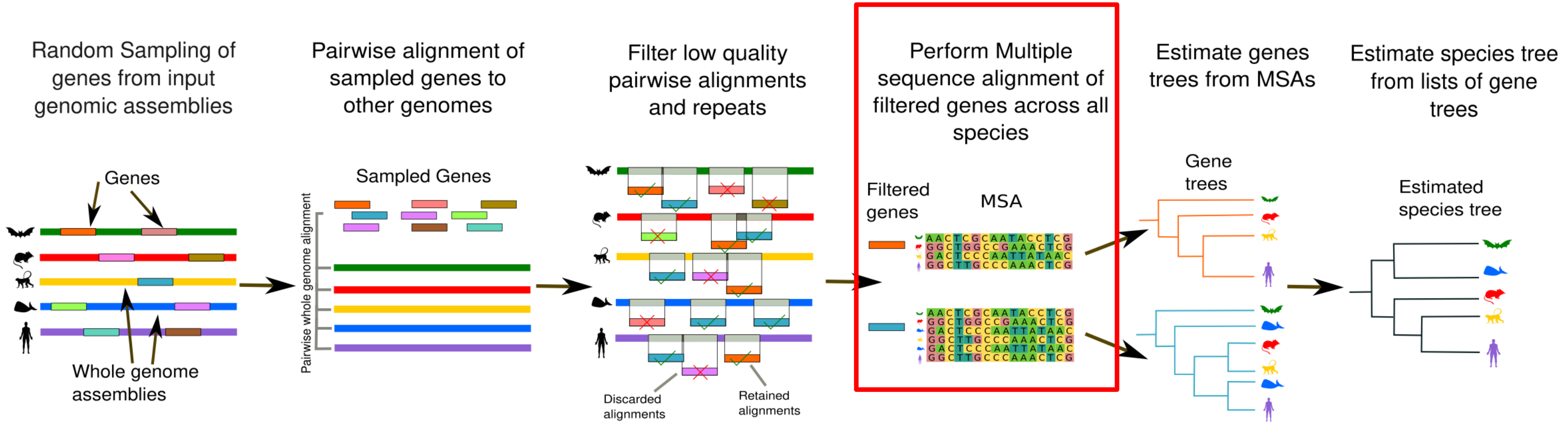
- **Identifying sequence homology and functional regions of the genome**



R. H. Ali, M. Bogusz, and S. Whelan. "Identifying Clusters of High Confidence Homologies in Multiple Sequence Alignments." Mol Biol Evol, 36(10):2340–2351, Oct. 2019.

MSA: Applications

- Identifying sequence homology and functional regions of the genome
- **Inferring evolutionary trees**



Random Sampling of genes from input genomic assemblies

Pairwise alignment of sampled genes to other genomes

Filter low quality pairwise alignments and repeats

Perform Multiple sequence alignment of filtered genes across all species

Estimate genes trees from MSAs

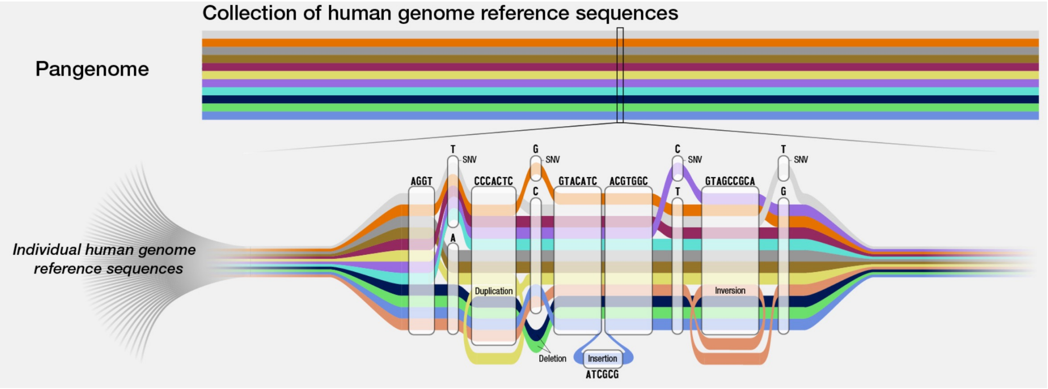
Estimate species tree from lists of gene trees

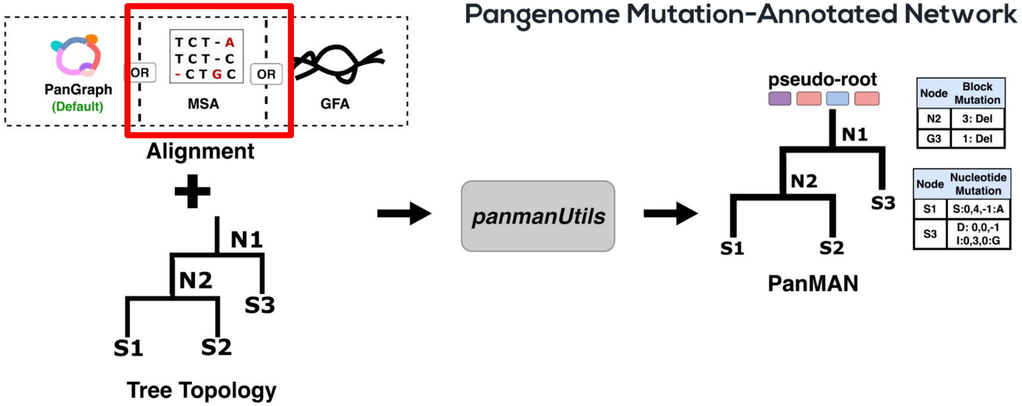
A. Gupta, S. Mirarab, & Y. Turakhia, “Accurate, scalable, and fully automated inference of species trees from raw genome assemblies using ROADIES”, Proc. Natl. Acad. Sci. U.S.A. 122 (19) (2025).

MSA: Applications

- Identifying sequence homology and functional regions of the genome
- Inferring evolutionary trees
- **Constructing and analyzing pangenomes**

PANMAN
Pangenome Mutation-Annotated Network





Node	Block Mutation
N2	3: Del
G3	1: Del

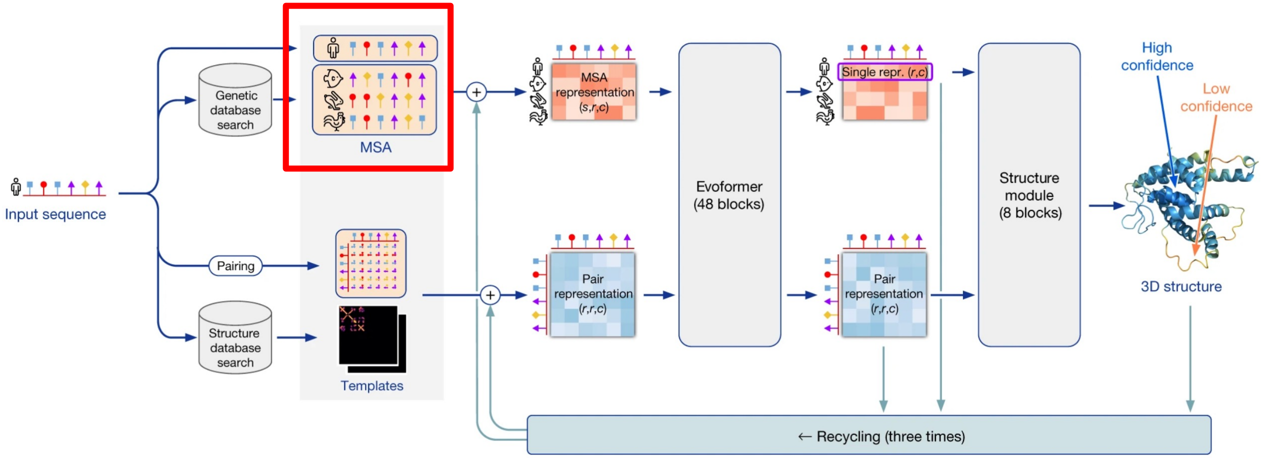
Node	Nucleotide Mutation
S1	S:0,4,-1:A
S3	D: 0,0,-1 I:0,3,0:G

Pangenome figure source: https://www.genome.gov/genetics-glossary/Pangenome

S. Walia, H. Motwani, K. Smith, R. Corbett-Detig, Y. Turakhia, "Compressive Pangenomics Using Mutation-Annotated Networks", bioRxiv 2024.07.02.601807

MSA: Applications

- Identifying sequence homology and functional regions of the genome
- Inferring evolutionary trees
- Constructing and analyzing pangenomes
- **Serving as training data for bioinformatics-related machine learning models**


Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

Jumper, J., Evans, R., Pritzel, A. et al. “Highly accurate protein structure prediction with AlphaFold”. Nature 596, 583–589 (2021).

State-of-the-Art MSA Tools

We have many powerful and well-established MSA tools:

- **MAGUS (Smirnov and Warnow, 2021)**
- **PASTA (Mirarab et al., 2015)**
- **MAFFT (Katoh and Standley, 2013)**
- **UPP2 (Park et al. 2023)**
- **Muscle5 (Edgar, 2022)**
- **T-Coffee Regressive mode (Garriga et al., 2019)**

	RNASim Dataset (S. Guo, et al. 2009)	**Simulated sequences using AliSim (N. Ly-Trong et al. 2023)**
Seq. Count (Length)	100,000 (1,500)	10,000 (10,000)
MAFFT	44 min	16 min
PASTA	2.6 hr	Mem. Error
T-Coffee	5.8 hr	3.9 hr
MAGUS	9.9hr	> 24 hr

Despite these advances, we still face some **key limitations**:

- **Insufficient speed** to keep up with high-throughput genome sequencing
- Struggles with **long and very large-scale** sequences

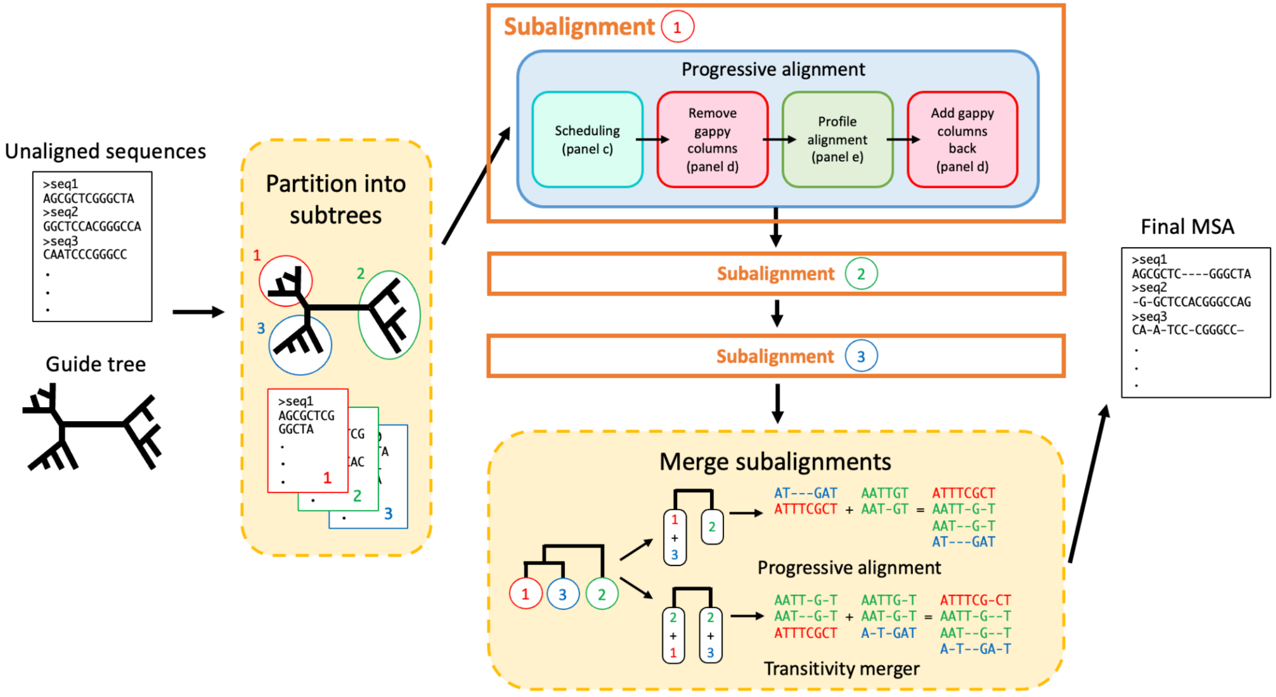
Outline

- Multiple sequence alignment: **applications** and **limitations**
- TWILIGHT: **T**all and **Wi**de A**lig**nments at **H**igh **T**hroughput
- **Key Contributions** and **Results**
- **Conclusion** and **Future Work**
- **Demo**

TWILIGHT: Overview

TWILIGHT: Tall and Wide Alignments at High Throughput

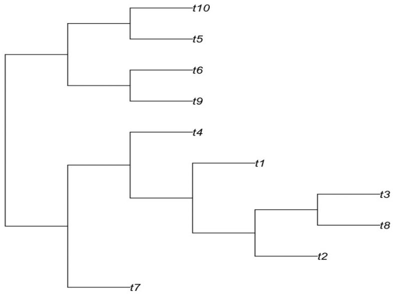
- **Progressive alignment**
- **Divide-and-Conquer strategy**
- **TALCO algorithm**
- **Remove gappy column heuristic**
- **Highly parallelized**



TWILIGHT: Progressive Alignment Scoring Scheme

- **Apply branch-proportional sequence weighting** (Julie D. Thompson, 1994)
- **Compute profiles**
- **Calculate pairwise column scores**
- **Affine-gap penalty with position-specific gap penalty** (Julie D. Thompson, 1994)

$$W = \sum_{i=1}^{n} \frac{l_i}{o_i}$$



w(t3)=1+½+⅓+¼+⅕+⅙

t3	G
t8	G
t2	A
t1	G
t4	-

⟹

A	0.2
C	0
G	0.6
T	0
-	0.2

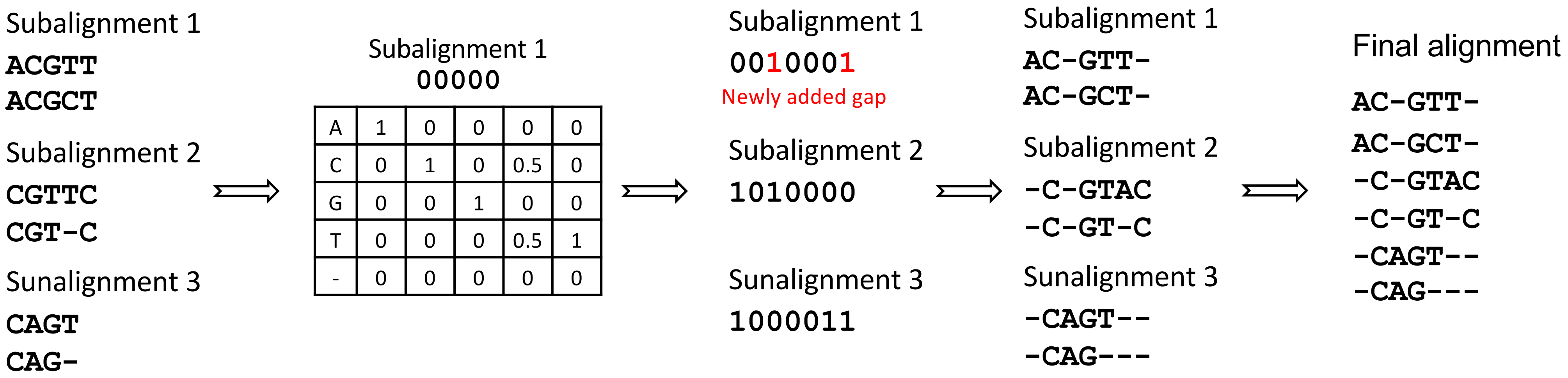
$$H(i,j) = \max \begin{cases} H(i-1,j-1) + ps(i,j) \\ I(i-1,j-1) + ps(i,j) \\ D(i-1,j-1) + ps(i,j) \end{cases}$$

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

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

TWILIGHT: Divide-and-Conquer Strategy

- For large datasets, memory usage is dominated by sequence storage
- Divide the dataset and process them sequentially
- Represent the subalignment using a binary string and a profile
- Update subalignments sequentially and merge subalignments using the Unix `cat` utility



Subalignment 1
ACGTT
ACGCT

Subalignment 2
CGTTC
CGT-C

Sunalignment 3
CAGT
CAG-

Subalignment 1
00000

A	1	0	0	0	0
C	0	1	0	0.5	0
G	0	0	1	0	0
T	0	0	0	0.5	1
-	0	0	0	0	0

Subalignment 1
0010001
Newly added gap

Subalignment 2
1010000

Sunalignment 3
1000011

Subalignment 1
AC-GTT-
AC-GCT-

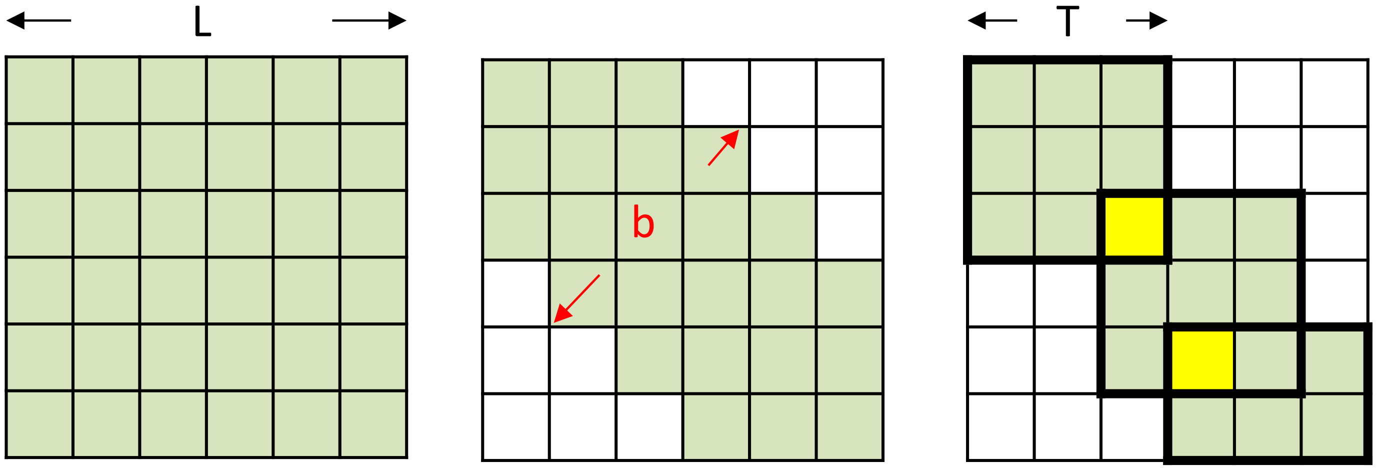
Subalignment 2
-C-GTAC
-C-GT-C

Sunalignment 3
-CAGT--
-CAG---

Final alignment

AC-GTT-
AC-GCT-
-C-GTAC
-C-GT-C
-CAGT--
-CAG---

TWILIGHT: TALCO algorithm (background)

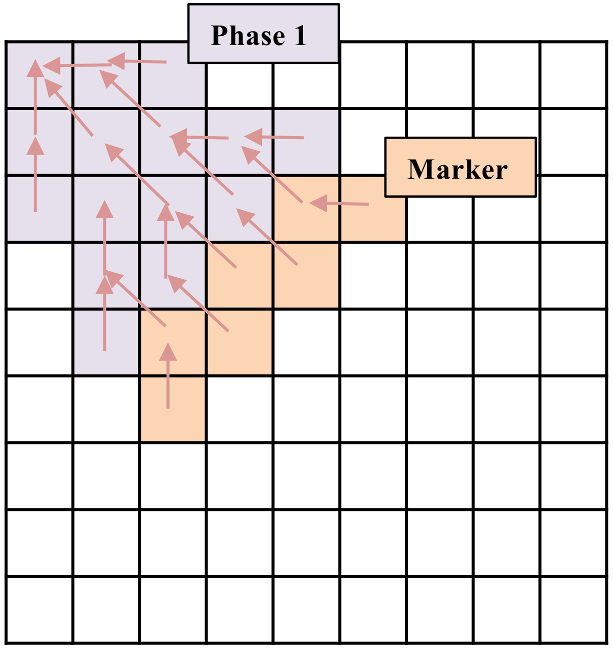


	Full-Matrix	Banded	Tiling
Time Complexity	O(L^2), Quadratic	O(L), Linear	O(L), Linear
Space Complexity	O(L^2), Quadratic	O(bL), Linear	O(T^2), Constant
Accuracy	Optimal	Near-optimal	Low

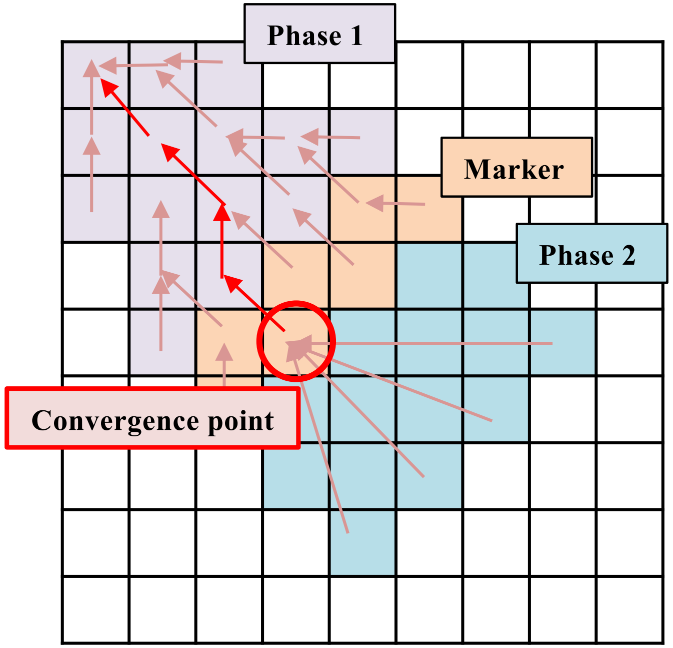
Turakhia Y., Bejerano G., Dally W.J. "Darwin: A Genomic Co-processor Provides 15,000× Speedup on long read assembly", ASPLOS (2018)

TWILIGHT: TALCO algorithm

- The traceback pointers require only constant space, allowing them to be stored in the GPU's shared memory
- Guarantees optimality under banding constraints



Phase 1:
Stores traceback pointers till the Marker

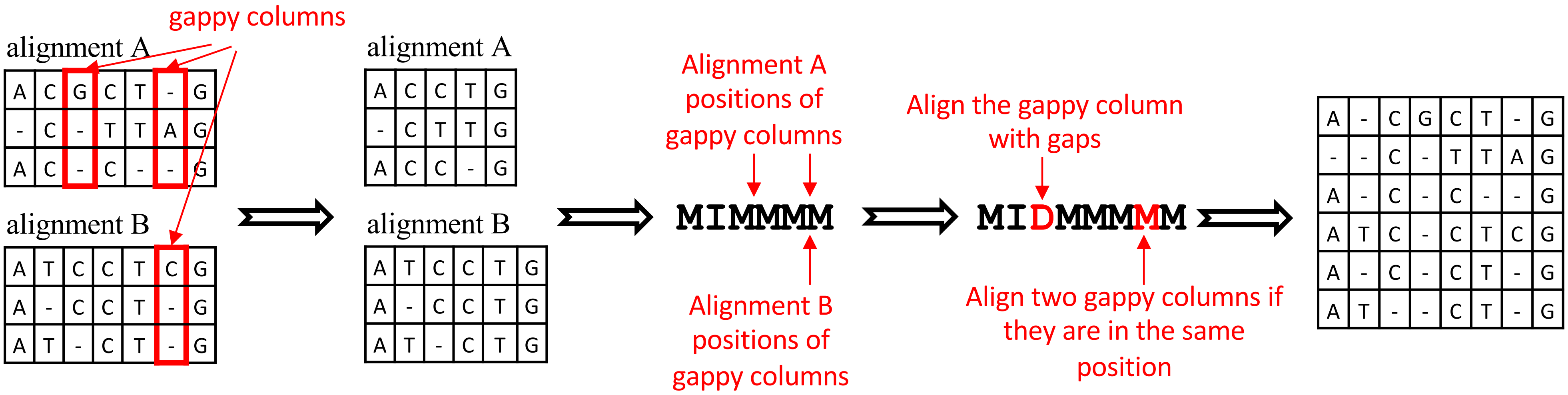


Phase 2:
Find the point of convergence and start the traceback from it

S. Walia et al. TALCO: "Tiling Genome Sequence Alignment Using Convergence of Traceback Pointers." HPCA, pages 91–107, Mar. 2024.

TWILIGHT: Remove gappy column heuristic

- **Minimizes length expansion, which in turn prevents substantial slowdowns during alignment, given the O(L) time complexity**
- **Avoids generating excessively large alignment files**



gappy columns

alignment A

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

alignment B

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

alignment A

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

alignment B

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

Alignment A positions of gappy columns

MIMMMM

Alignment B positions of gappy columns

Align the gappy column with gaps

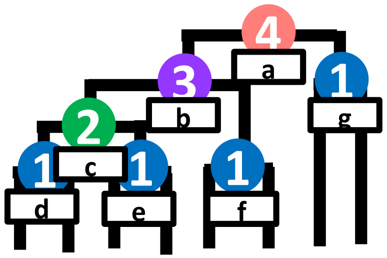
MIDMMMMM

Align two gappy columns if they are in the same position

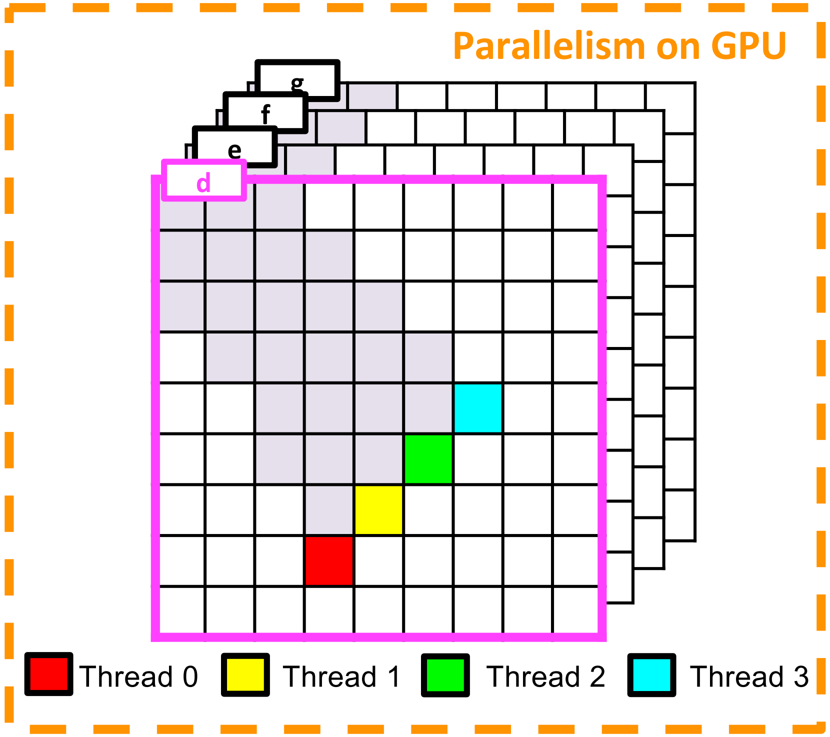
A	-	C	G	C	T	-	G
-	-	C	-	T	T	A	G
A	-	C	-	C	-	-	G
A	T	C	-	C	T	C	G
A	-	C	-	C	T	-	G
A	T	-	-	C	T	-	G

TWILIGHT: Parallelization

- Inter-alignment parallelism: One block handles one node
- Intra-alignment parallelism: One thread calculates the score of a cell in the same wavefront
- Multi-GPU parallelism



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

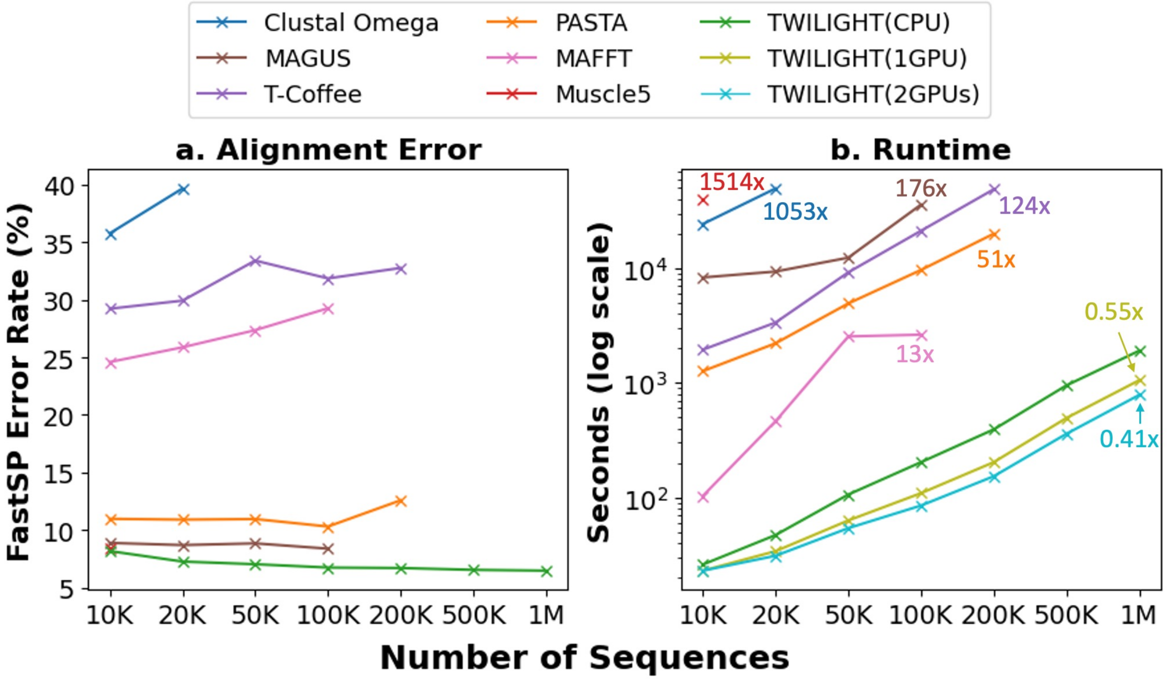


Outline

- Multiple sequence alignment: **applications** and **limitations**
- TWILIGHT: **T**all and **Wi**de A**lig**nments at **H**igh **T**hroughput
- **Key Contributions** and **Results**
- **Conclusion** and **Future Work**
- **Demo**

TWILIGHT: Contributions and Results

- **Demonstrates strong performance in both speed and accuracy**



32 min (CPU)

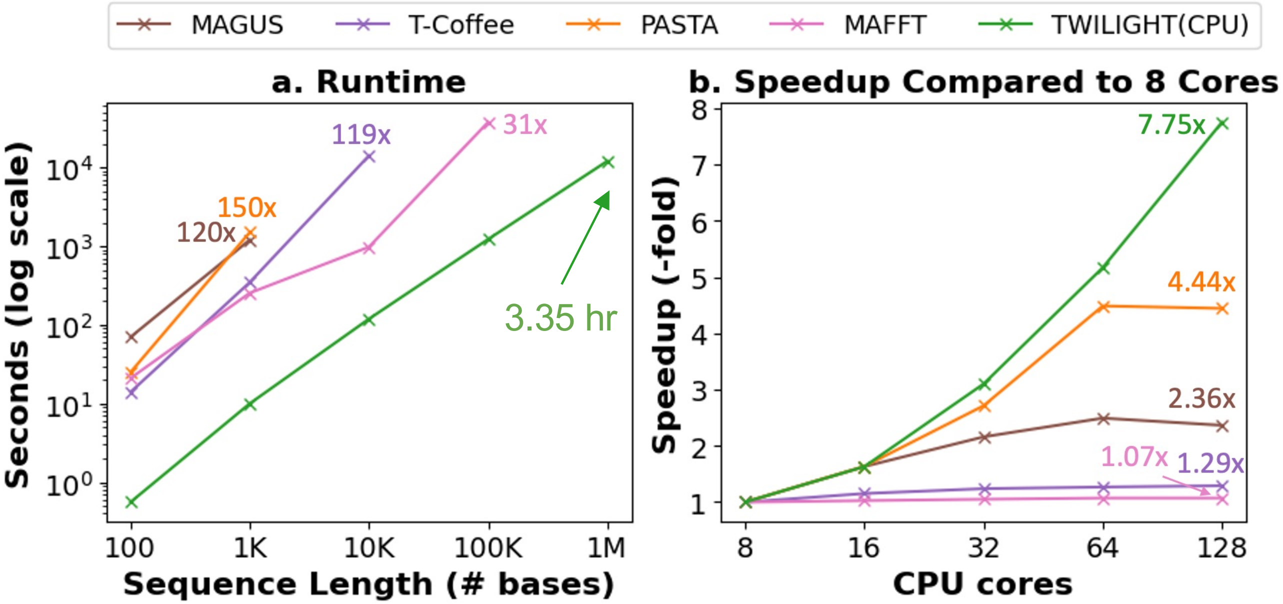
13 min (2 GPUs)

RNASim Dataset (S. Guo, et al. 2009)

S. Mirarab and T. Warnow. FASTSP: linear time calculation of alignment accuracy. Bioinformatics, 27(23):3250–3258, Dec. 2011.

TWILIGHT: Contributions and Results

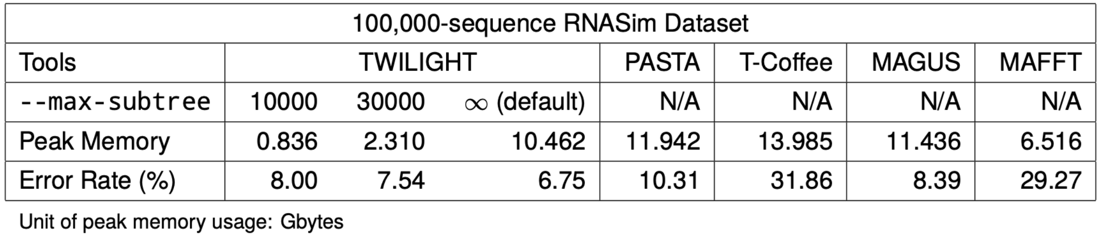
- Demonstrates strong performance in both speed and accuracy
- **Scales linearly with sequence length and effectively leverages available parallelism**



Simulated sequences using AliSim (N. Ly-Trong et al. 2023)
sequence count: 10k

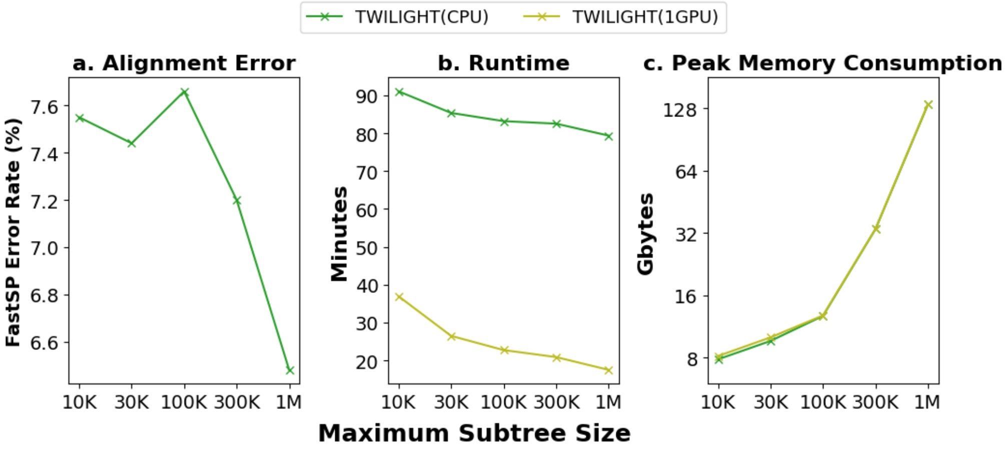
TWILIGHT: Contributions and Results

- Demonstrates strong performance in both speed and accuracy
- Scales linearly with sequence length and effectively leverages available parallelism
- **Adapts to platforms with limited memory constraints**



100,000-sequence RNASim Dataset							
Tools	TWILIGHT			PASTA	T-Coffee	MAGUS	MAFFT
`--max-subtree`	10000	30000	∞ (default)	N/A	N/A	N/A	N/A
Peak Memory	0.836	2.310	10.462	11.942	13.985	11.436	6.516
Error Rate (%)	8.00	7.54	6.75	10.31	31.86	8.39	29.27

Unit of peak memory usage: Gbytes



RNASim Dataset (S. Guo, et al. 2009)

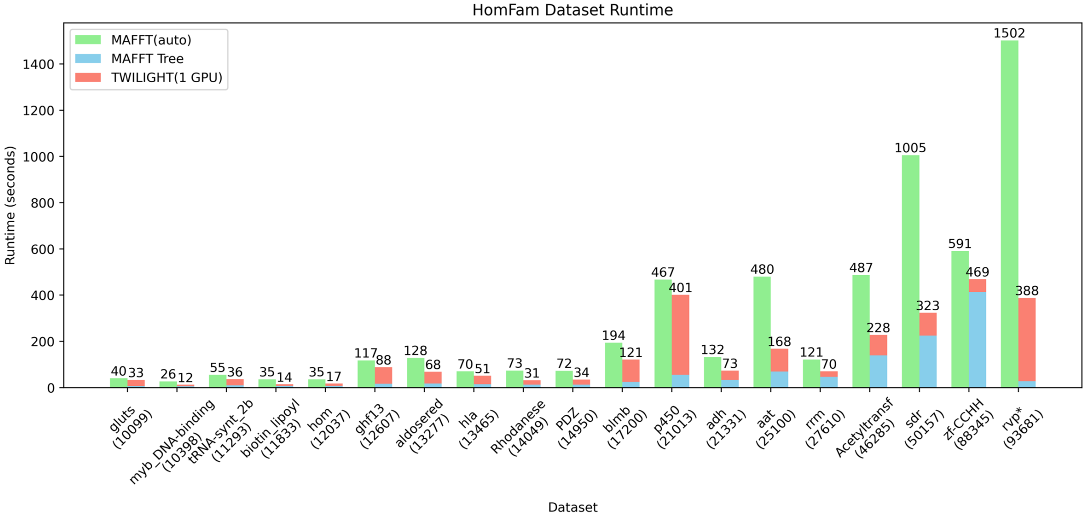
Trade-offs:

Runtime – Smaller subtrees reduce parallelism and introduce overhead from repeated file access.

Accuracy – Merging may slightly deviate from the original guide tree topology, resulting in a minor loss of accuracy.

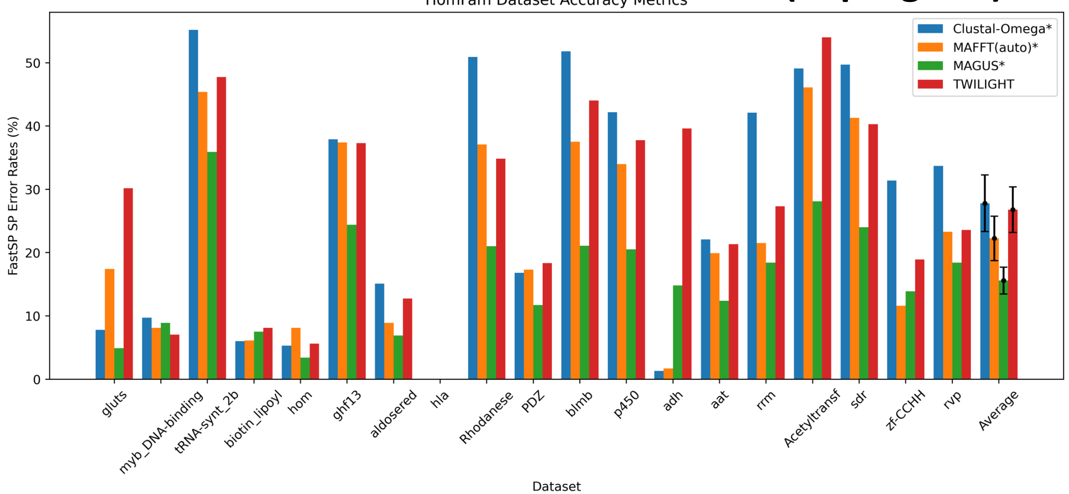
TWILIGHT: Contributions and Results

- Demonstrates strong performance in both speed and accuracy
- Scales linearly with sequence length and effectively leverages available parallelism
- Adapts to platforms with limited memory constraints
- **Provides great speed on large-scale protein datasets**



*: PartTree is used instead of MAFFT Tree

(In progress)

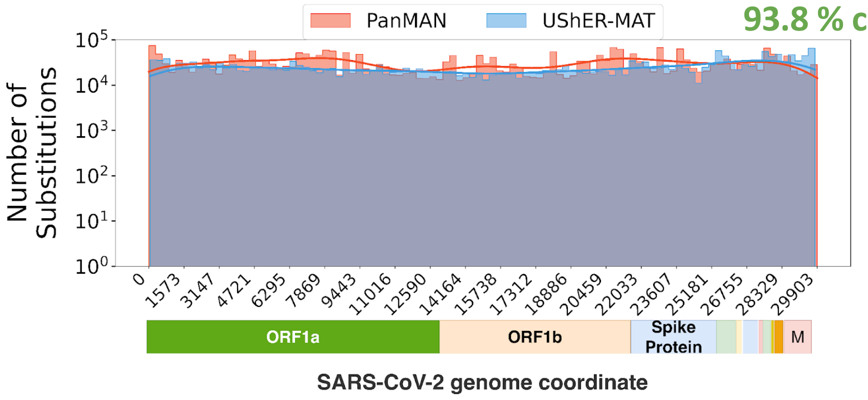


*: The results are taken from the supplementary material of the publication by V. Smirnov (2021).

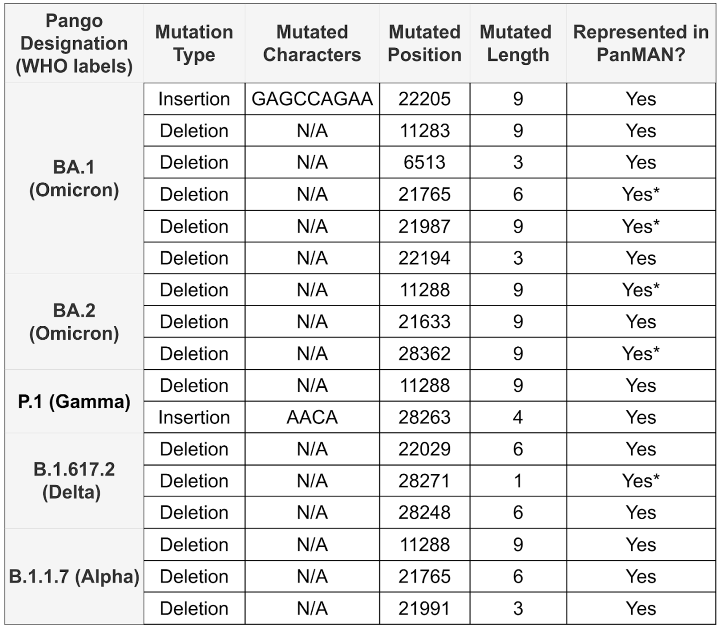
V. Smirnov. Recursive MAGUS: Scalable and accurate multiple sequence alignment. PLoS Comput Biol, 17(10):e1008950, Oct. 2021.
Smirnov, V. (Creator) (Mar 31 2021). Datasets used in "Recursive MAGUS: scalable and accurate multiple sequence alignment". University of Illinois Urbana-Champaign. 10.13012/B2IDB-1048258_V1

TWILIGHT: Contributions and Results

- Demonstrates strong performance in both speed and accuracy
- Scales linearly with sequence length and effectively leverages available parallelism
- Adapts to platforms with limited memory constraints
- Provides great speed on large-scale protein datasets
- **Facilitates the construction of an ultra-large pangenome of 8 million SARS-CoV-2 sequences**



93.8 % concordance



Pango Designation (WHO labels)	Mutation Type	Mutated Characters	Mutated Position	Mutated Length	Represented in PanMAN?
BA.1 (Omicron)	Insertion	GAGCCAGAA	22205	9	Yes
	Deletion	N/A	11283	9	Yes
	Deletion	N/A	6513	3	Yes
	Deletion	N/A	21765	6	Yes*
	Deletion	N/A	21987	9	Yes*
	Deletion	N/A	22194	3	Yes
BA.2 (Omicron)	Deletion	N/A	11288	9	Yes*
	Deletion	N/A	21633	9	Yes
	Deletion	N/A	28362	9	Yes*
P.1 (Gamma)	Deletion	N/A	11288	9	Yes
	Insertion	AACA	28263	4	Yes
B.1.617.2 (Delta)	Deletion	N/A	22029	6	Yes
	Deletion	N/A	28271	1	Yes*
	Deletion	N/A	28248	6	Yes
B.1.1.7 (Alpha)	Deletion	N/A	11288	9	Yes
	Deletion	N/A	21765	6	Yes
	Deletion	N/A	21991	3	Yes

S. Walia, H. Motwani, K. Smith, R. Corbett-Detig, Y. Turakhia, "Compressive Pangenomics Using Mutation-Annotated Networks", bioRxiv 2024.07.02.601807
Y. Turakhia et al. Ultrafast Sample placement on Existing tRees (UShER) enables real-time phylogenetics for the SARS-CoV-2 pandemic. Nat Genet, 53(6):809–816, June 2021.

Outline

- Multiple sequence alignment: **applications** and **limitations**
- TWILIGHT: **T**all and **Wi**de A**lig**nments at **H**igh **T**hroughput
- **Key Contributions** and **Results**
- **Conclusion** and **Future Work**
- **Demo**

Conclusion and Future Work

- We present **TWILIGHT**, an MSA tool to overcome the scalability limitations of existing solutions
  - Maintains a **constant memory footprint** using **TALCO** algorithm
  - Prevents slowdown by **removing gappy columns** before the alignment step
  - Effectively leverages available **parallelisms** of modern HPC platforms (CPU, GPU)
  - Significantly **reduces memory usage** by the divide-and-conquer techniques

Conclusion and Future Work

- We present **TWILIGHT**, an MSA tool to overcome the scalability limitations of existing solutions
  - Maintains a **constant memory footprint** using **TALCO** algorithm
  - Prevents slowdown by **removing gappy columns** before the alignment step
  - Effectively leverages available **parallelisms** of modern HPC platforms (CPU, GPU)
  - Significantly **reduces memory usage** by the divide-and-conquer techniques
- TWILIGHT aligns **1 million RNASim sequences in 32 minutes** and **10,000 sequences of 1 million bases each in just 3.35 hours**
- To the best of our knowledge, TWILIGHT is **the first** to perform **non-reference-based MSA on 8 million SARS-CoV-2 sequences**

Conclusion and Future Work

- We present **TWILIGHT**, an MSA tool to overcome the scalability limitations of existing solutions
  - Maintains a **constant memory footprint** using **TALCO** algorithm
  - Prevents slowdown by **removing gappy columns** before the alignment step
  - Effectively leverages available **parallelisms** of modern HPC platforms (CPU, GPU)
  - Significantly **reduces memory usage** by the divide-and-conquer techniques
- TWILIGHT aligns **1 million** RNASim sequences in **32 minutes** and 10,000 sequences of **1 million bases** each in just **3.35 hours**
- To the best of our knowledge, TWILIGHT is **the first** to perform **non-reference-based MSA** on **8 million SARS-CoV-2 sequences**
- **Future Work**
  - Incorporates more sensitive methods for **highly divergent alignments**
  - Expands to a **multiple whole-genome aligner** capable of handling nonlinear genomic rearrangements

Acknowledgments

Thank you for your attention!

Co-authors

Sumit Walia
Ph.D. student

Yatish Turakhia
Assistant Professor, UCSD

UC San Diego
Electrical and Computer Engineering
JACOBS SCHOOL OF ENGINEERING

TURAKHIA LAB

Special Thanks to

Prof. Siavash Mirarab, UCSD
Jade Wang, UCSD
Anshu Gupta, UCSD
for their valuable feedback

Prof. Tandy Warnow and her group, UIUC
Contributors of SARS-CoV-2 data to NCBI GenBank and COG-UK databases
for sharing their datasets, which greatly facilitated the evaluation of our tool

Kyle Smith, UCSD
for collecting and preprocessing 8M SARS-CoV-2 datasets

Funding

CDC
U.S. CENTERS FOR DISEASE CONTROL AND PREVENTION

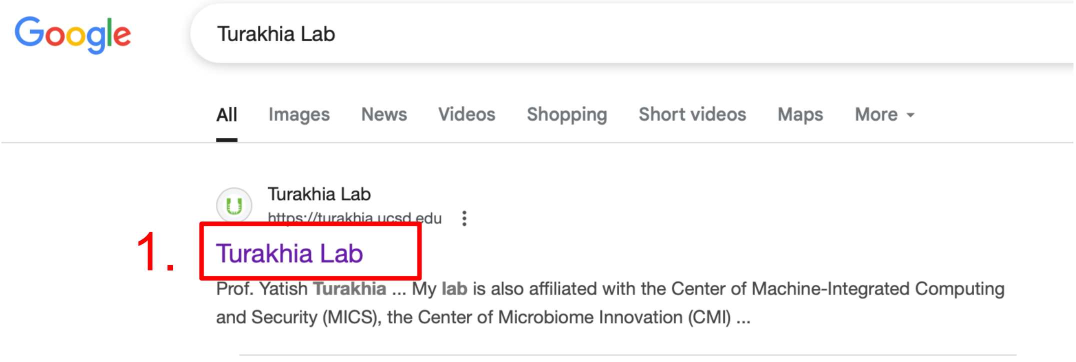
AMD
AI & HPC Fund

Hellman Fellows Fund

Outline

- Multiple sequence alignment: **applications** and **limitations**
- TWILIGHT: **T**all and **Wi**de A**lig**nments at **H**igh **T**hroughput
- **Key Contributions** and **Results**
- **Conclusion** and **Future Work**
- **Demo**

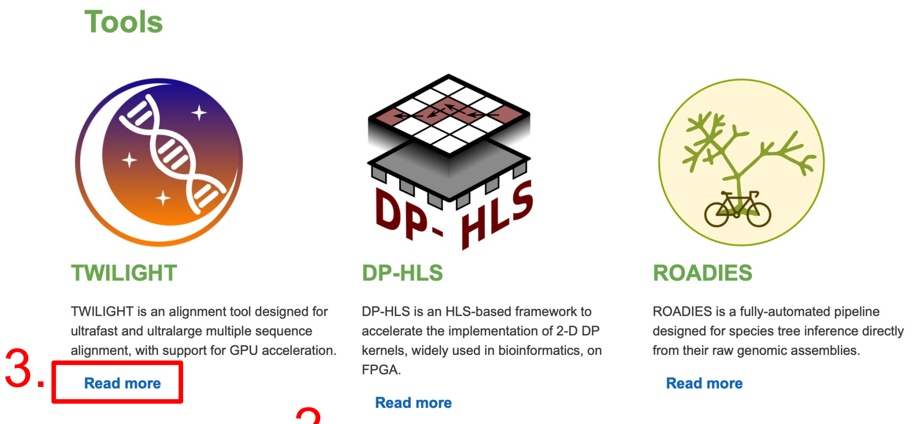
TWILIGHT: GitHub Page



Turakhia Lab
https://turakhia.ucsd.edu

1. Turakhia Lab

Prof. Yatish **Turakhia** ... My **lab** is also affiliated with the Center of Machine-Integrated Computing and Security (MICS), the Center of Microbiome Innovation (CMI) ...



Tools

TWILIGHT

TWILIGHT is an alignment tool designed for ultrafast and ultralarge multiple sequence alignment, with support for GPU acceleration.

3. Read more

DP-HLS

DP-HLS is an HLS-based framework to accelerate the implementation of 2-D DP kernels, widely used in bioinformatics, on FPGA.

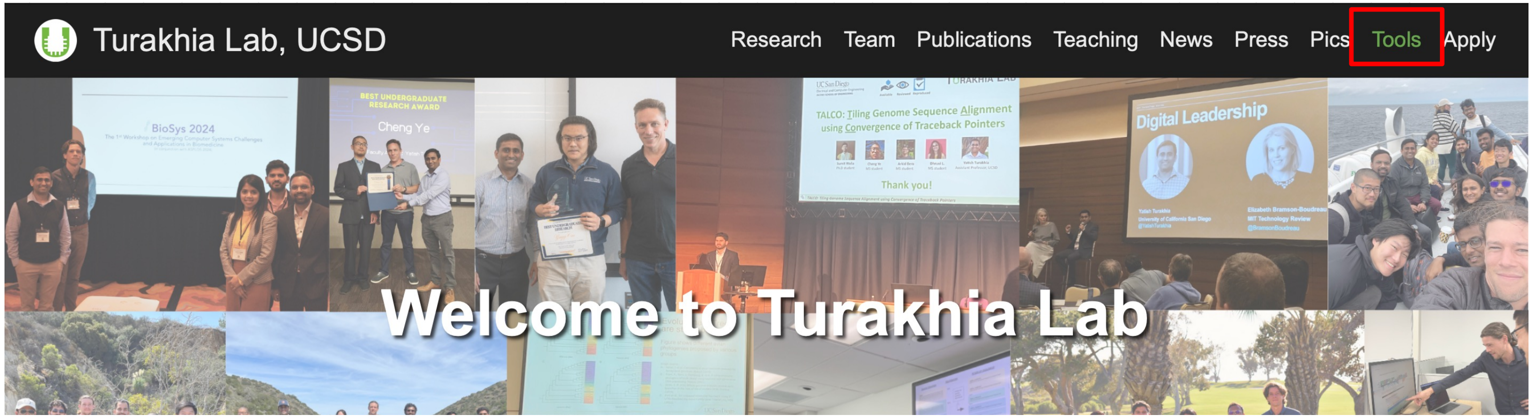
Read more

ROADIES

ROADIES is a fully-automated pipeline designed for species tree inference directly from their raw genomic assemblies.

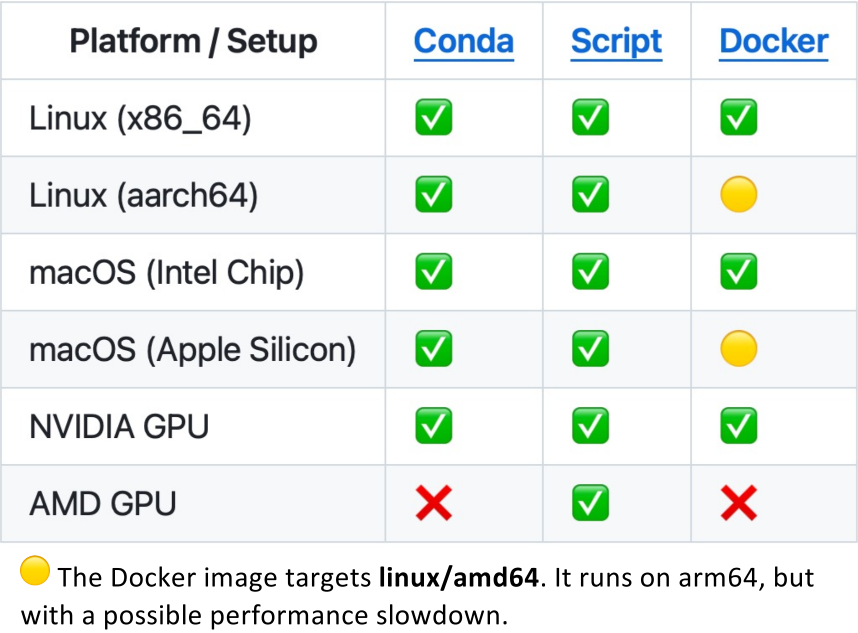
Read more

2.



Or visit directly at: https://github.com/TurakhiaLab/TWILIGHT

TWILIGHT: Installation



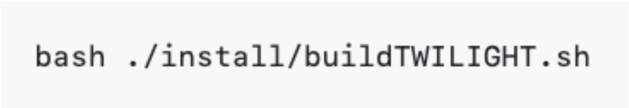
Platform / Setup	Conda	Script	Docker
Linux (x86_64)	✅	✅	✅
Linux (aarch64)	✅	✅	🟡
macOS (Intel Chip)	✅	✅	✅
macOS (Apple Silicon)	✅	✅	🟡
NVIDIA GPU	✅	✅	✅
AMD GPU	❌	✅	❌

🟡 The Docker image targets **linux/amd64**. It runs on arm64, but with a possible performance slowdown.

Supports **Apple M-series, NVIDIA,** and **AMD GPUs**

Install through **Bioconda** and **Docker**

Installation scripts are also provided



```
bash ./install/buildTWILIGHT.sh
```

TWILIGHT: Default Mode

See `--help` or visit http://turakhia.ucsd.edu/TWILIGHT/ for detailed command-line options

Run with default settings

```
./twilight -t ../dataset/RNASim.nwk -i ../dataset/RNASim.fa -o RNASim.aln -C 8
```

Tree file, required
(Newick format)

Sequence file, required
(FASTA format)

Output file, required

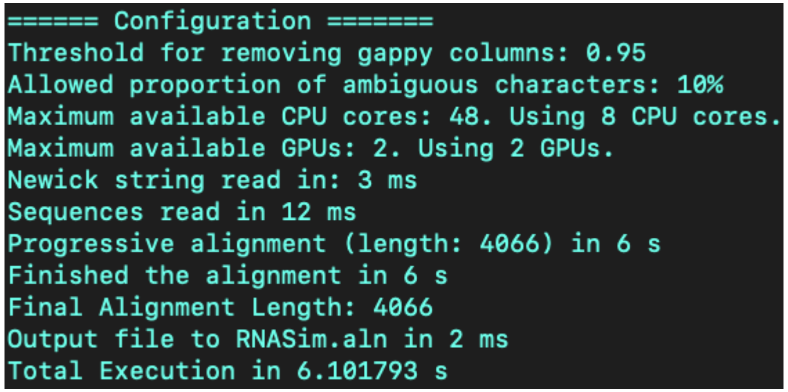
Number of CPUs,
default: all available CPUs

Run with divide-and-conquer method

```
./twilight -t ../dataset/RNASim.nwk -i ../dataset/RNASim.fa -o RNASim.aln -m 200
```

Maximum subtree size,
default: ∞

Expected output log message



```
====== Configuration =======
Threshold for removing gappy columns: 0.95
Allowed proportion of ambiguous characters: 10%
Maximum available CPU cores: 48. Using 8 CPU cores.
Maximum available GPUs: 2. Using 2 GPUs.
Newick string read in: 3 ms
Sequences read in 12 ms
Progressive alignment (length: 4066) in 6 s
Finished the alignment in 6 s
Final Alignment Length: 4066
Output file to RNASim.aln in 2 ms
Total Execution in 6.101793 s
```

TWILIGHT: Iterative Mode

Visit http://turakhia.ucsd.edu/TWILIGHT/ for details

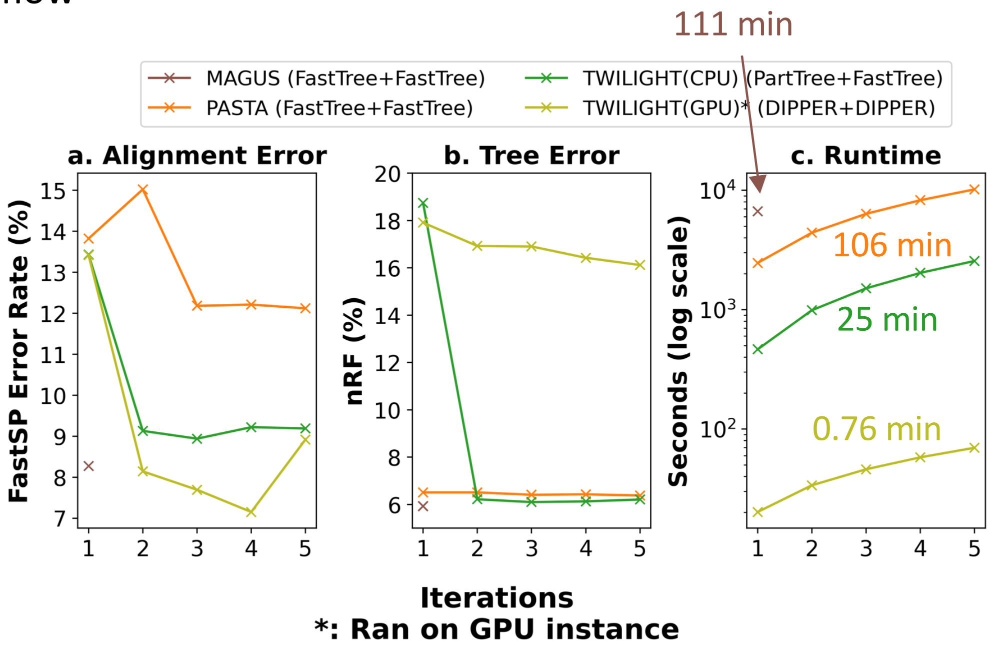
Install Snakemake and the tree inference tool via Conda (packaged in the installation script).

```
bash ./install/installIterative.sh
```

DIPPER

Enter the `workflow` directory and run the Snakemake workflow

`Snakemake \`	
`--cores all \`	Number of CPU cores
`--config \`	
`TYPE=n \`	Sequence type, **required**
`SEQ=../dataset/RNASim.fa \`	Input sequences file, **required**
`OUT=RNASim.aln \`	Output alignment file name
`DIR=tempDir \`	Directory for storing temporary files
`ITER=2 \`	Number of iterations
`INITTREE=maffttree \`	Tree tool for initial guide tree
`ITERTREE=raxml \`	Tree tool for subsequent iterations
`GETTREE=yes \`	Estimate tree after final alignment
`OUTTREE=RNASim.tree`	Output tree file name



***: Ran on GPU instance**

Options

Sequence type (**n:** nucleotide or **p**: protein)

Initial guide tree (MashTree, PartTree, MAFFTTree)

Tree for subsequent iterations (FastTree, RAxML, IQTree)

MashTree (Katz et al., 2019), PartTree (Katoh and Toh, 2007), MAFFTTree (Katoh and Standley, 2013), FastTree (Price et al., 2010), RAxML (Stamatakis, 2006), IQTree (Minh et al., 2020)