Global Phylogeny by Local Inferences

Final Project for Phylogeny Inference & Applications

Freie Universität Berlin: Winter Semester 2022-2023
Meghan Kane
Derek Shao

Outline

- Introduction
- Motivation
- Methods
- Materials
- Results
- Discussion
- Contributions

Outline

- **Introduction**
- Motivation
- Methods
- Materials
- Results
- Discussion
- Contributions

What is phylogeny?

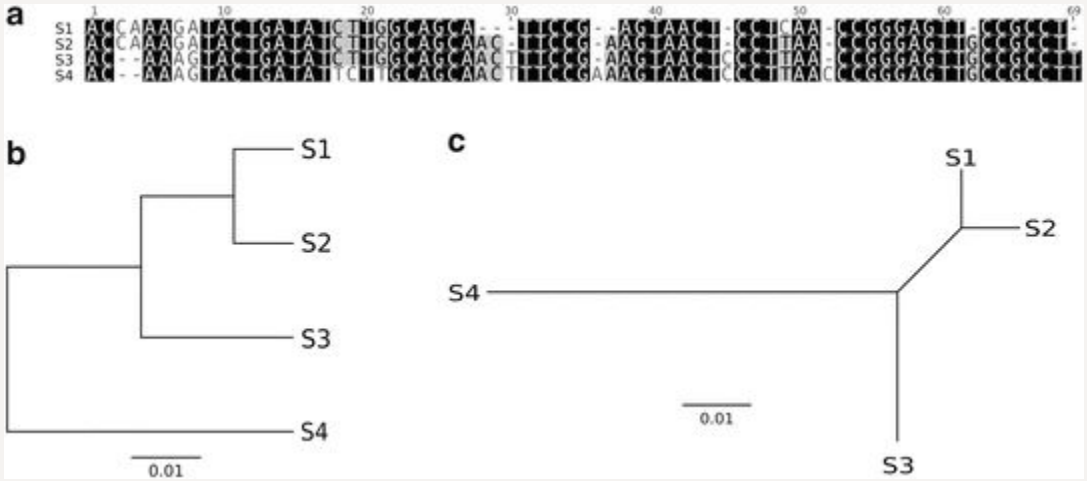
The relationships between entities to reflect their evolutionary histories (De Bruyn et al., 2013)

Key assumption: Measurable similarities between organisms are suggestive of their common evolutionary history.

Word cloud generated from https://en.wikipedia.org/wiki/Phylogenetics

Phylogenetic tree

- Basic components: *vertex (node) & branch (edge)*
- *Rooted* versus *Unrooted* trees
- *Bifurcating* versus *Multifurcating* trees



De Bruyn et al., 2013

Comparison of methods in computational phylogenetics

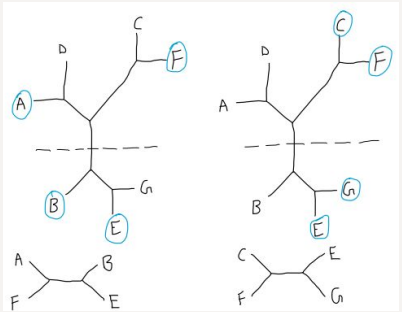
	Attributes (+)	Notes (–)
Distance based (ape, ClustalOmega, EBI, phangorn, FastML)	• Computational speed • Multiple distance metrics possible	• Difficult to calculate distances for sequences with large divergence and alignment gaps • Less sophisticated than probabilistic methods
Maximum parsimony (PHYLIP, Rphylip, GCTree, phangorn, IgTree)	• Intuitive algorithm • Clonal frequency incorporation (GCTree) • Polytomies and internal nodes (IgTree)	• Ignores antibody specific properties (hotspots, transversions, transitions) • Long-branch attraction problem
Maximum likelihood (FastML, MEGA, IQ-TREE, dnaml, IgPhyML)	• Complex substitution models • Hotspot specific codon models (IgPhyML)	• Computationally demanding • Sensitive to model misspecification
Bayesian (BEAST, Mr. Bayes, ImmuniTree)	• Complex substitution models • Can produce rooted trees without explicit outgroup • Possible to incorporate biological knowledge with priors • Mutation rate returned in calendar time (BEAST)	• Sensitive to model misspecification • Highest computational demands due to Markov chain Monte Carlo algorithm

Yermanos et al., 2018

Large-scale phylogenetic inference

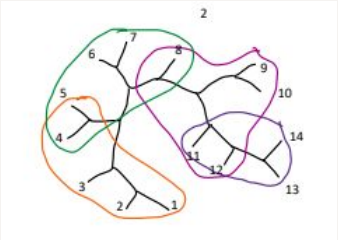
Background:

- Exponentially increasing volume of DNA and protein sequence data
- Identifying the most parsimonious tree is NP-hard (Felsenstein, 2004)
- Finding maximum likelihood (ML) phylogenies is NP-hard (Roch, 2006; Chor & Tuller, 2006)



Common methods:

- Quartet-based method (Ranwez & Gascuel, 2001)



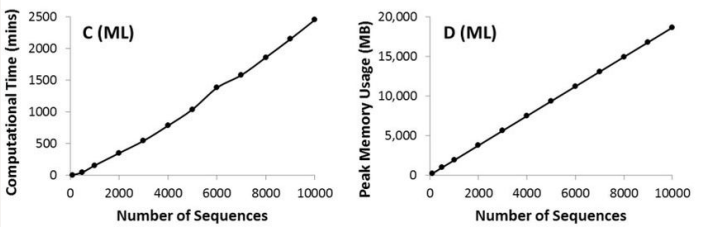
- Disc-covering method (Bayzid et al., 2014)

Outline

- Introduction
- **Motivation**
- Methods
- Materials
- Results
- Discussion
- Contributions

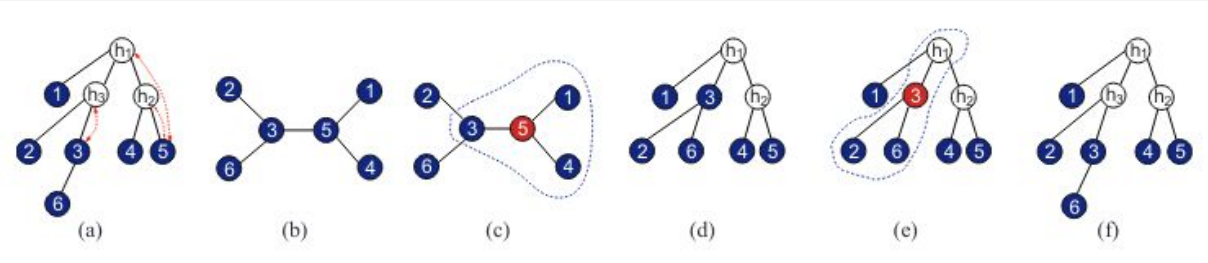
Motivation

- Unacceptable computing time in MP and ML analyses when number of taxa increases



Kumar et al., 2016

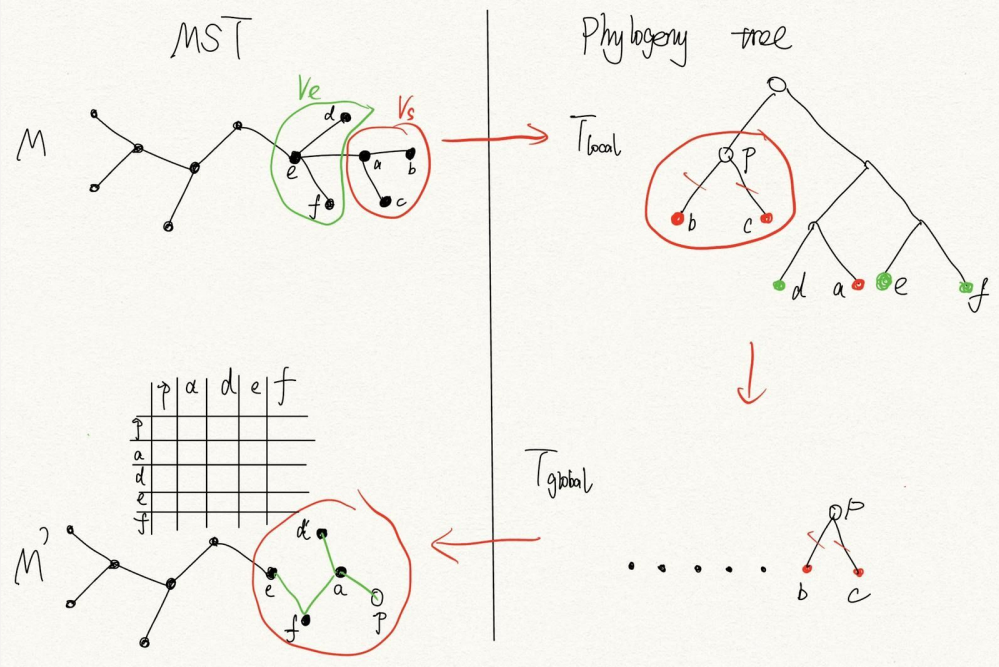
- Limitations of quartet-based and disc-covering methods
- Topological correspondence between unrooted phylogenetic tree and minimum spanning tree constructed using additive distances (Choi et al., 2010; Kalaghatgi & Lengauer, 2017)



Choi et al., 2010

Objective

An efficient and scalable algorithm for phylogenetic inference based on minimum spanning tree!



Outline

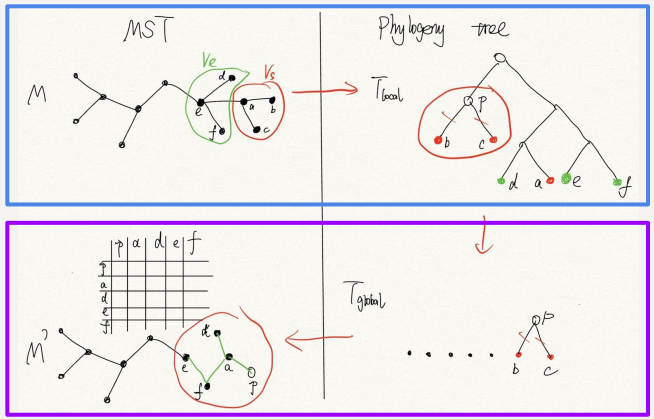
- Introduction
- Motivation
- **Methods**
- Materials
- Results
- Discussion
- Contributions

Workflow

- Initiate global phylogeny tree
- Construct minimum spanning tree

While global tree is not connected:

- Construct a MST-induced MP local tree
- Update global tree & MST



Credit to Kalaghatgi, 2020

Input: Aligned sequences X_L for each taxon in the taxon set L, and a subtree size threshold s

Initialize a global unrooted phylogenetic tree $T = (V_T, E_T)$ as $V_T \leftarrow L$ and $E_T \leftarrow \{\}$;

Compute a minimum spanning tree $M = (V_M, E_M)$ using Hamming distances of each sequence pair in X_L;

while $|E_T| < |V_T| - 1$ **do**

  if *M contains a subtree $\tau_s = (V_s, E_s)$ with more than s vertices, and $|V_M| - |V_s| > s$* **then**

    Select a smallest subtree τ_s of M containing more than s vertices;

    Select s vertices V_e of M that are visited using a BFS starting at the root of τ_s such that $V_e \cap V_s = \{\}$;

    Compute a ML phylogenetic tree t_ρ over vertices $V_s \cup V_e$ by performing tree search using SEM-GM;

    Construct t by suppressing the root of t_ρ, and select the largest non-singleton subforest f_s of t that is induced by V_s;

    Add edges in f_s to the global phylogenetic tree T;

    Update M by removing leaves of subtrees, and adding the root of each subtree;

  else

    Compute a ML phylogenetic tree t_ρ over all vertices in V_M using SEM-GM;

    Construct an unrooted phylogenetic tree t by suppressing the root in t_ρ, and add all edges in t to T;

  end

end

Output: A global unrooted phylogenetic tree T

Approach

1. Construct minimum-spanning tree (MST)
2. Define global phylogeny tree
3. Construct MST-induced local subtrees
4. Update MST

Initiate global phylogeny tree

- Apply custom `Tree` class developed during semester
- Read sequence data from influenza_98.fasta file
- **Assign all observed sequences to leaf nodes by setting out_degree = 0**

So, the starting global tree has 98 vertices and 0 edges

Construct MST

I. **Compute pairwise Jukes-Cantor distances between sequences**

Estimate $d = d_a + d_b$ given sequences X_a & X_b with Hamming distance $\hat{p}$

$$\hat{d} = -\frac{3}{4}\ln\left(1 - \frac{4\hat{p}}{3}\right)$$

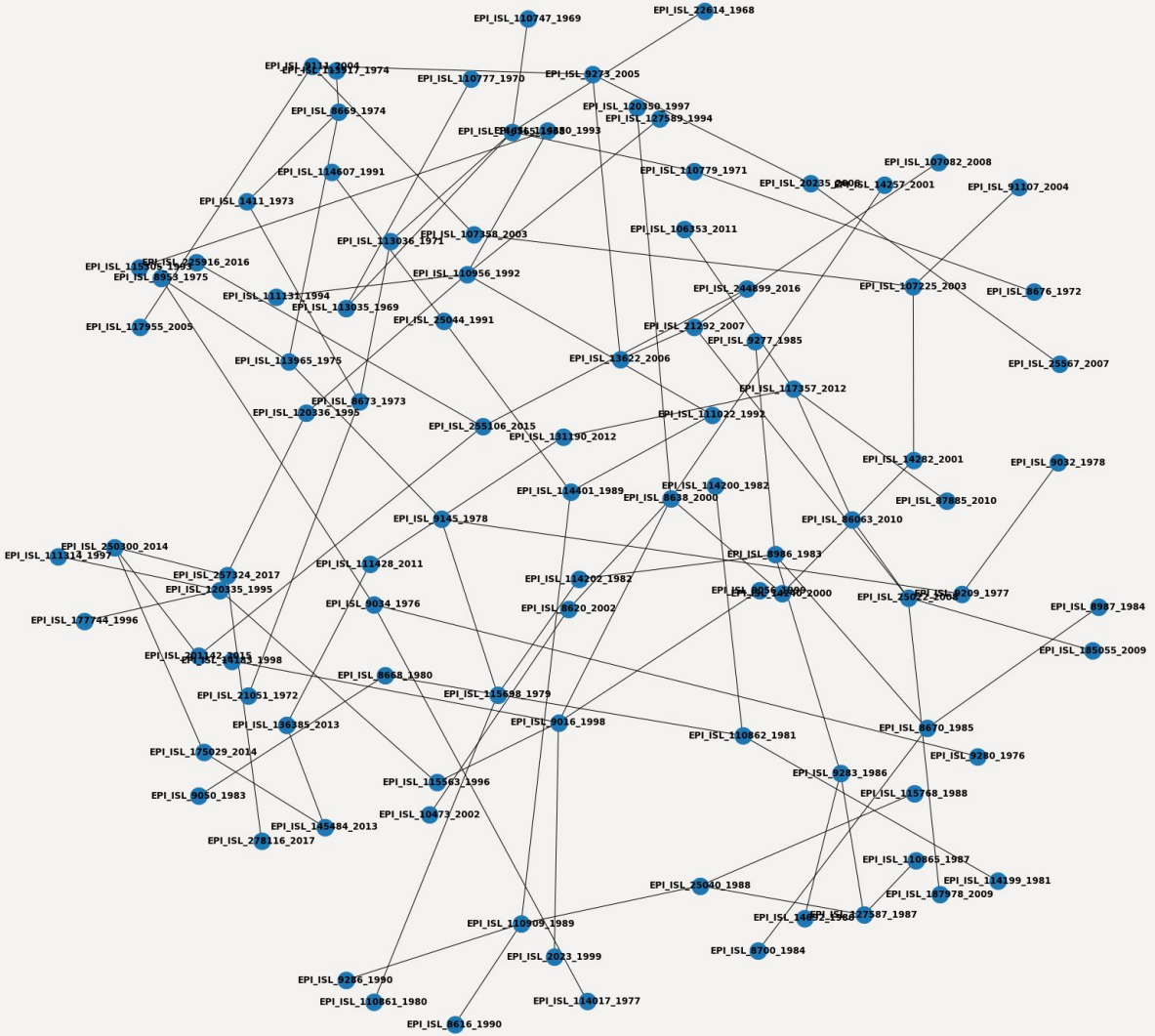
II. Create fully connected graph with distance matrix

III. **Compute MST in the connected graph**

- `networkx` (Hagberg et al., 2008)

Construct MST

`influenza_98.fasta`

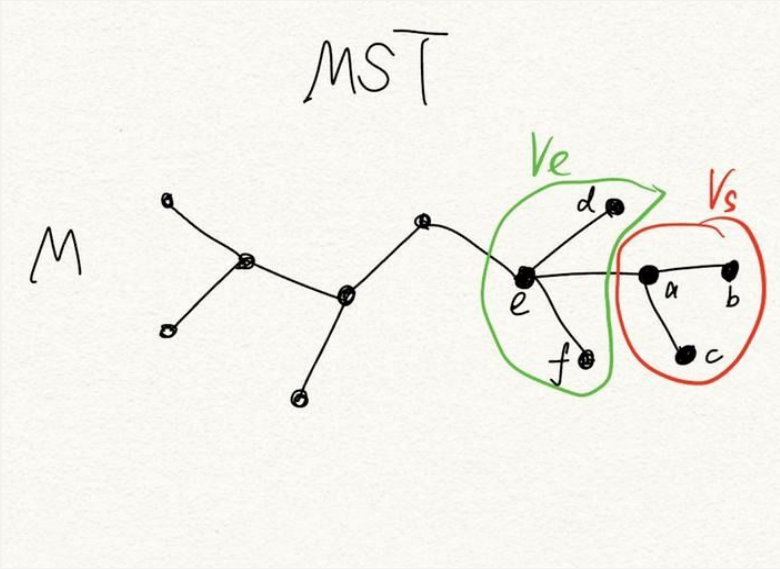


Construct MST-induced MP local trees

I. Convert `networkx` object to our custom `Tree` class

II. Select V_s from MST s.t. a subtree is induced

- *Threshold*: minimum number of vertices in Vs (subtree)



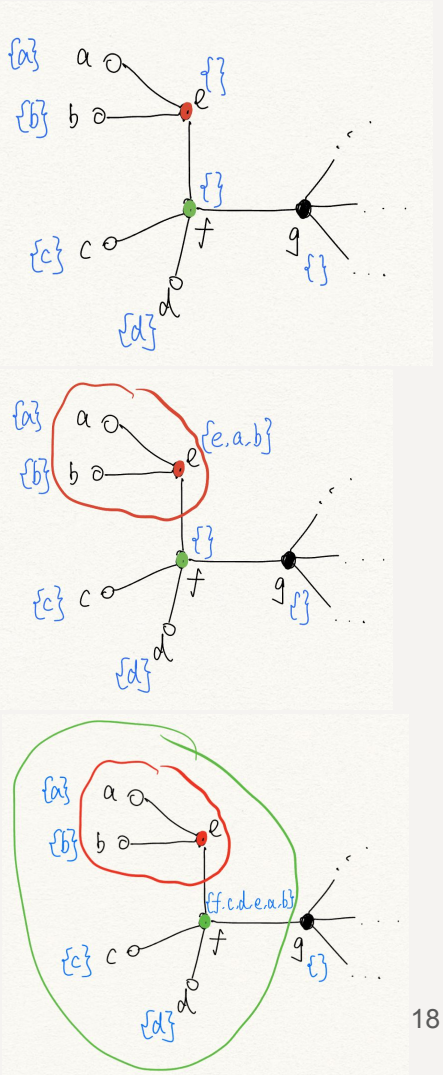
Vs = {a, b, c} at *Threshold* = 3

II. Select V_s from MST s.t. a subtree is induced

Algorithm 1 Select V_s from MST such that it induces a subtree

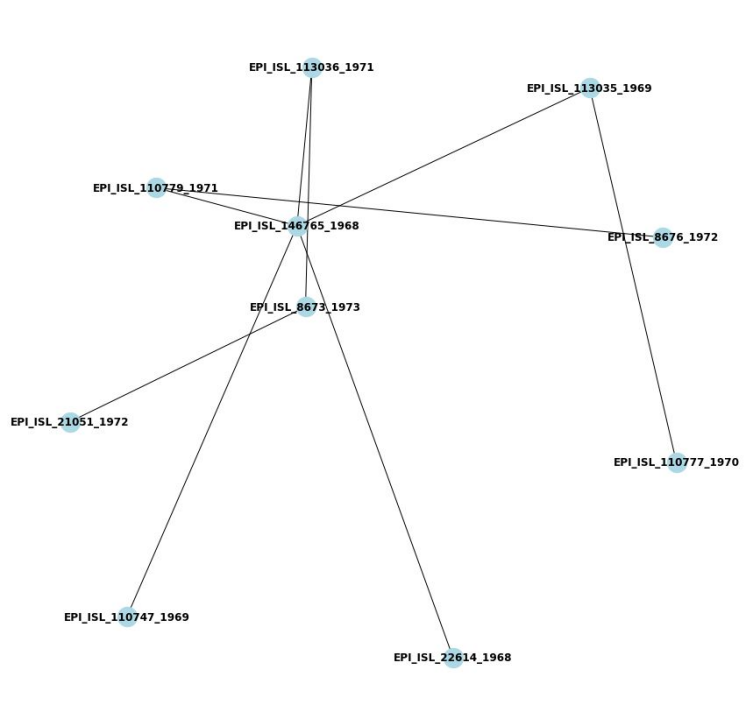
Input: MST nodes, *Threshold*
Output: Subset V_s of MST nodes

```
1:  vertices_to_visit ← Assign all leaf nodes of MST
2:  v ← Pop one random element from vertices_to_visit
3:  while vertices_to_visit is not empty do
4:      if v is a leaf node then
5:          v.subtree ← Assign v itself
6:          vertices_to_visit ← Append v.neighbor to the tail
7:      else if v is an internal node then
8:          Check the number of neighbors with empty subtree
9:          if Only one of its neighbors has an empty(unassigned) subtree then
10:             v.subtree ← Assign the union of its neighbors' subtrees
11:         else
12:             vertices_to_visit ← Queue v for revisiting after updating its neighbors
13:         end if
14:     end if
15:     if Size of v.subtree is larger than Threshold then
16:         Return V_s = v.subtree
17:     end if
18:     v ← Pop the first element from vertices_to_visit
```



II. Select V_s from MST s.t. a subtree is induced

Example: Influenza_98.fasta, threshold = 10



Construct MST-induced MP local trees

I. Convert `networkx` object to our custom `Tree` class

II. Select V_s from MST s.t. a subtree is induced

- Threshold: minimum number of vertices in Vs (subtree)

III. Select V_e via BFS from root of V_s-induced subtree

IV. Create one random staring tree topology from $V_s \cup V_e$

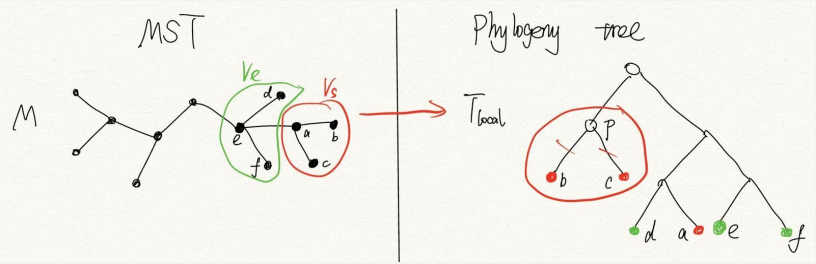
- `ete3.populate` - generates a random tree topology (Huerta-Cepas et al., 2016)

V. Find local tree with maximum parsimony score

- `Bio.Phylo.TreeConstruction.ParsimonyTreeConstructor`

VI. Find minimum subtree unit induced by only V_s in local tree

- Select V_s vertices with same parent (e.g., vertex *b* & *c*, and local parent vertex *p*)



Construct MST-induced local subtrees (M)

I. Convert `networkx` object to our custom `Tree` class

II. Select V_s from MST s.t. a subtree is induced

III. Select V_e via BFS from root of induced subtree Vs

IV. Create random initial unrooted tree from $V_s \cup V_e$

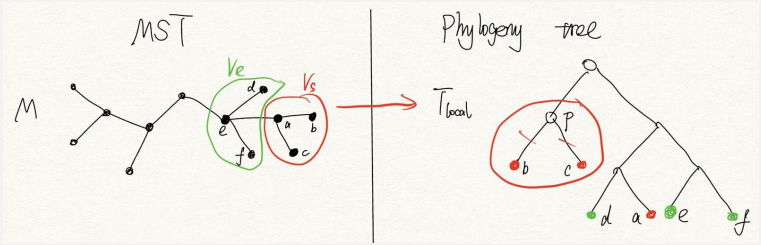
- `ete3.populate` - generates a random tree topology (Huerta-Cepas et al., 2016)

V. Find local tree with maximum parsimony score

- `Bio.Phylo.TreeConstruction.ParsimonyTreeConstructor`

VI. Find minimum subtree unit induced by only V_s in local tree

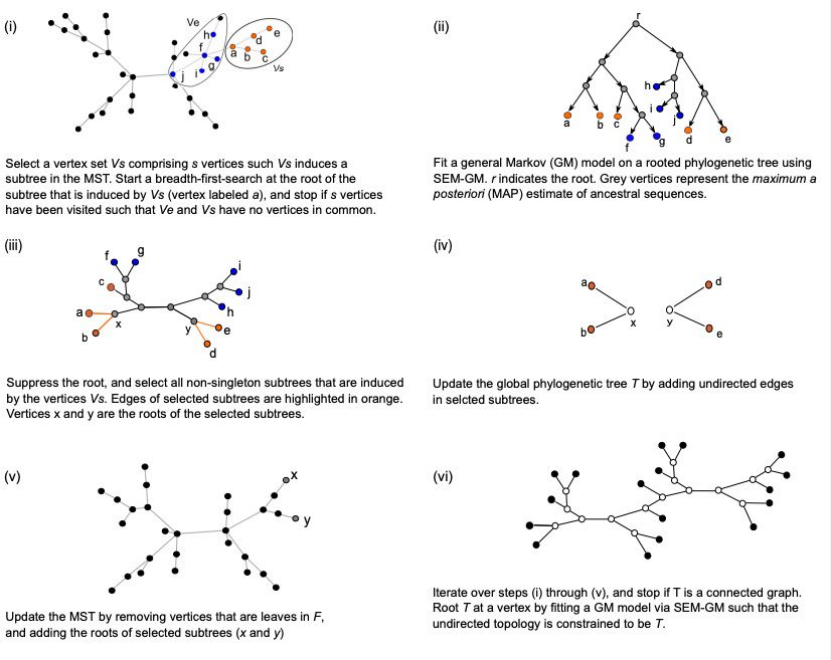
- Select V_s vertices with same parent (e.g., vertices *b* & *c* and local parent *p*)

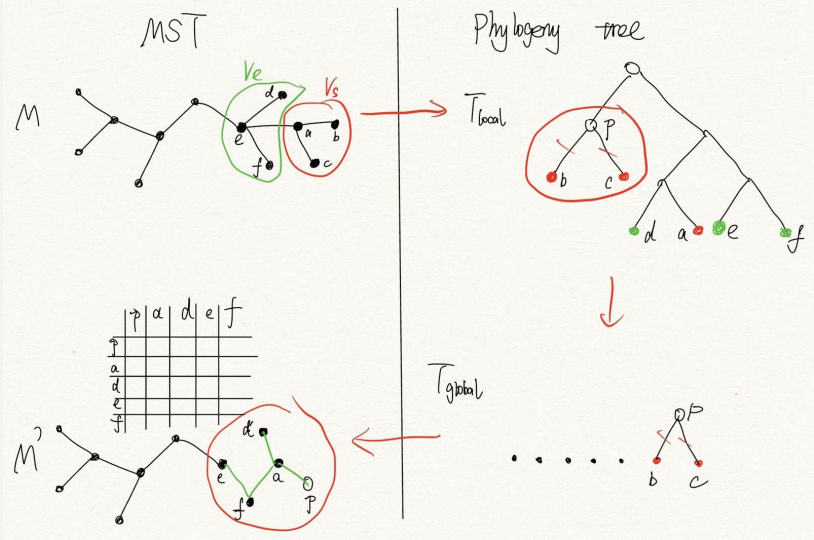


Construct MST-induced local subtrees (M)

TODO: create & insert graph similar to Prabhav's (Fig 2) adjusted to our project

https://www.biorxiv.org/content/10.1101/2020.06.30.180315v1.full.pdf





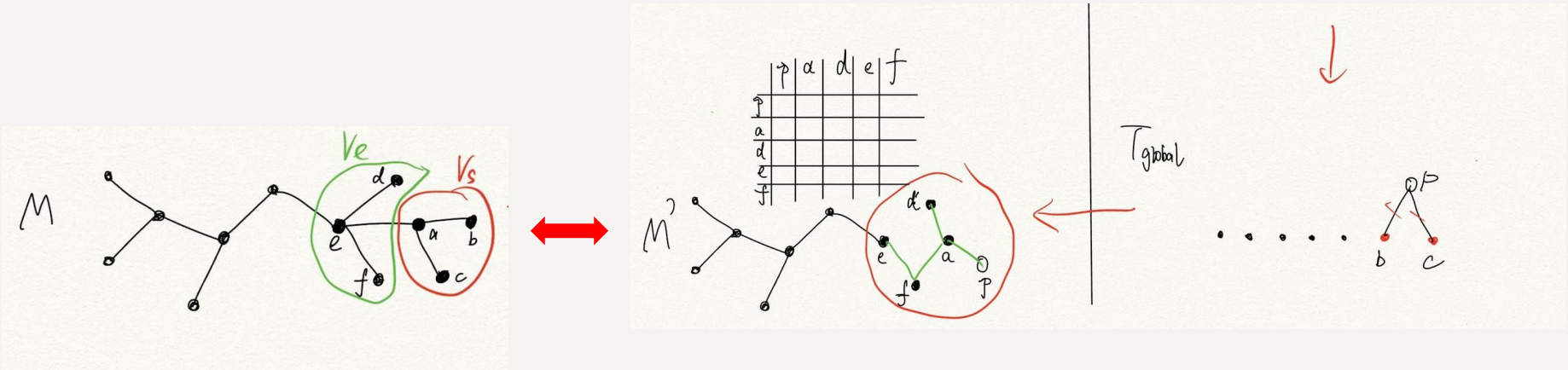
Update global tree & MST

I. **Update global phylogeny tree**
   - Add new local parent vertex (p) & corresponding edges to global tree

II. Delete selected V_s vertices & add new local parent vertices in MST

III. Update sequence distance matrix with $V_s \cup V_e$

IV. **Update the subgraph of $V_s \cup V_e$ in MST**

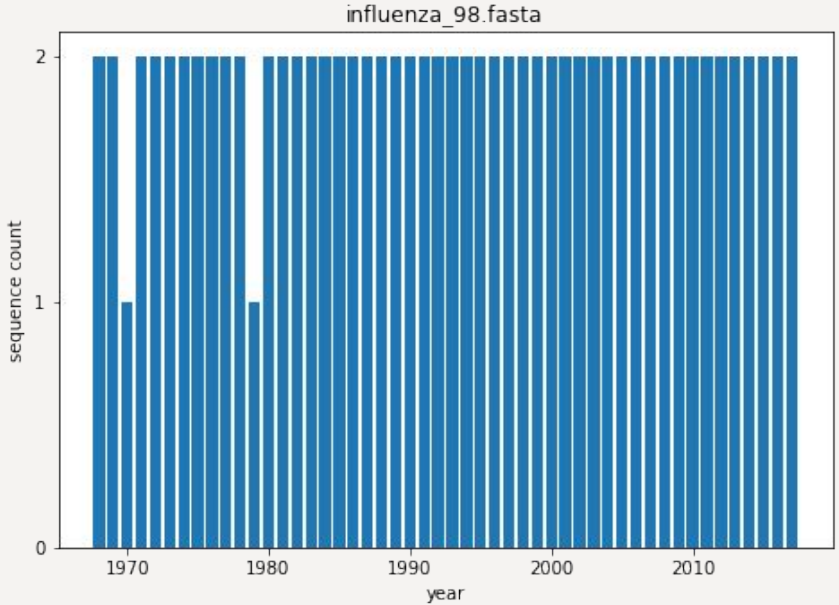


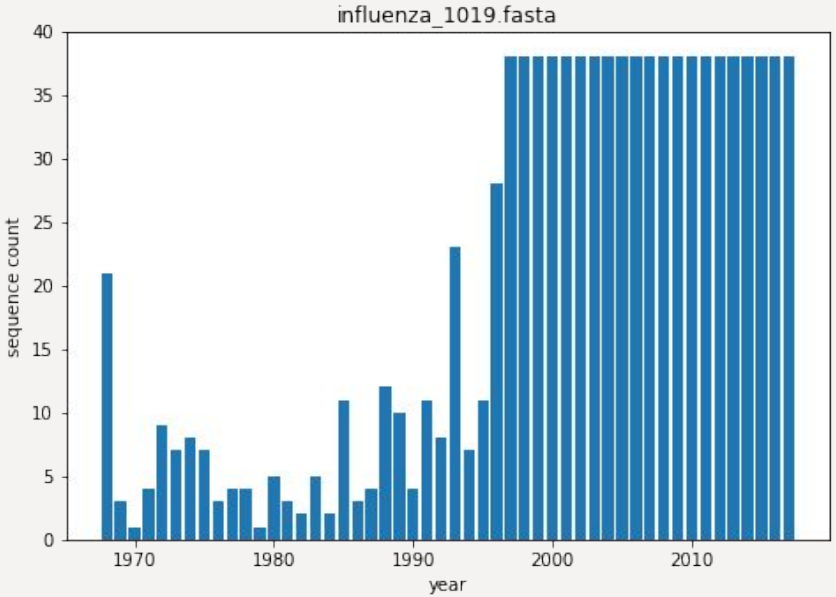
Outline

- Introduction
- Motivation
- Methods
- **Materials**
- Results
- Discussion
- Contributions

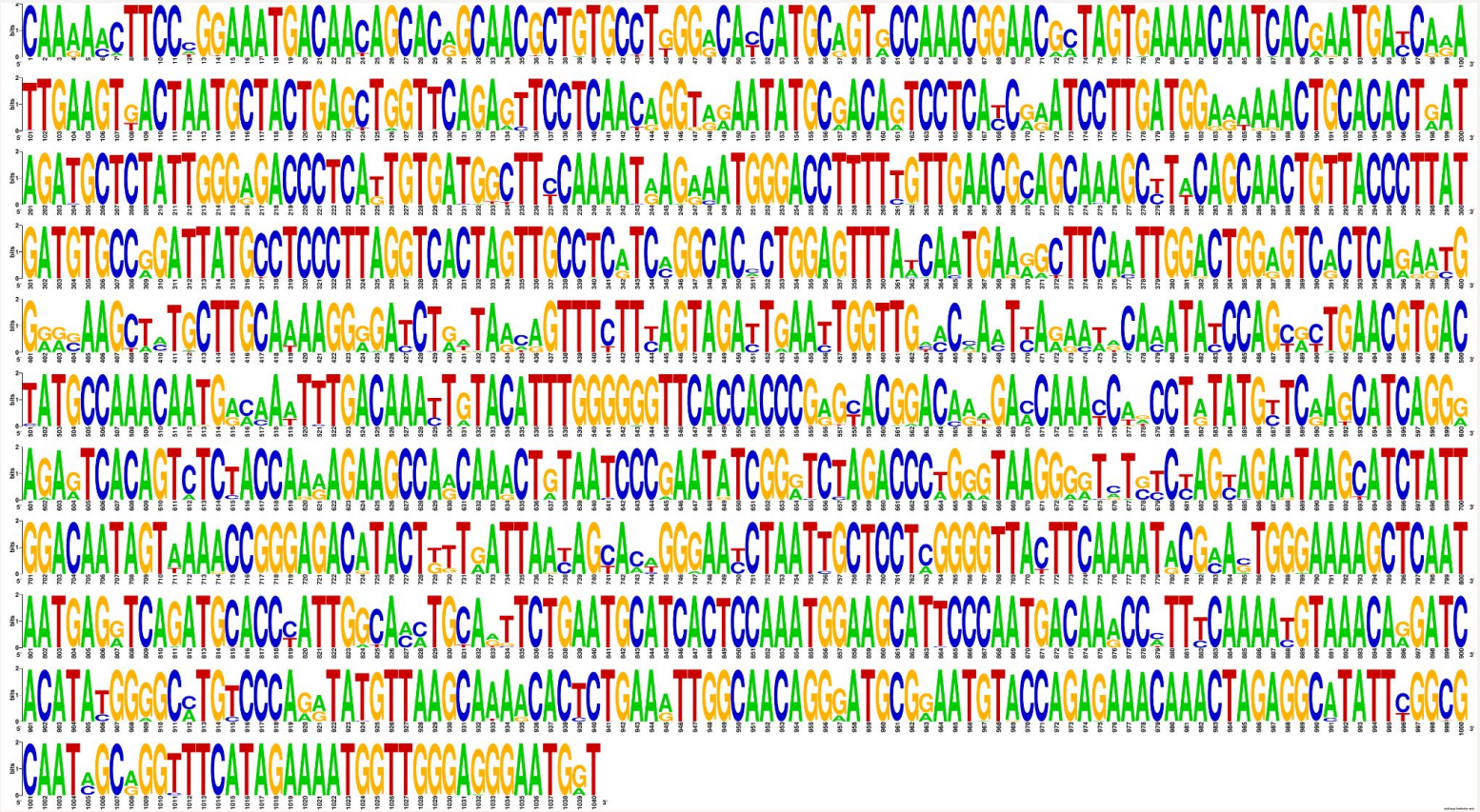
Materials

- **2 fasta files:** `influenza_98.fasta` & `influenza_1019.fasta`
- **Sequence length**: 1040 nucleotides
- **Time period**: 1968 - 2017, distribution of data different between datasets





Nucleotide distribution in Influenza_98.fasta

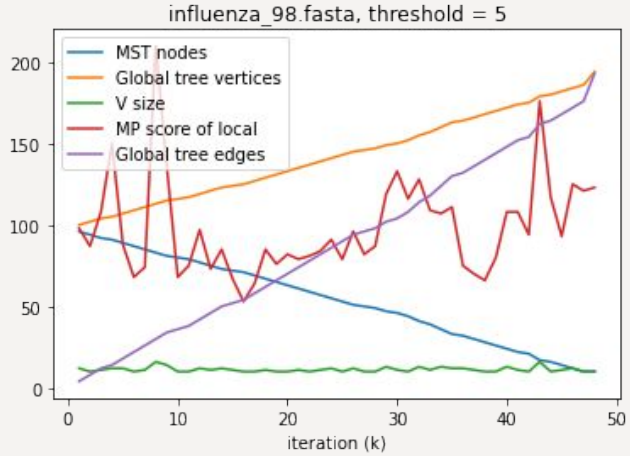


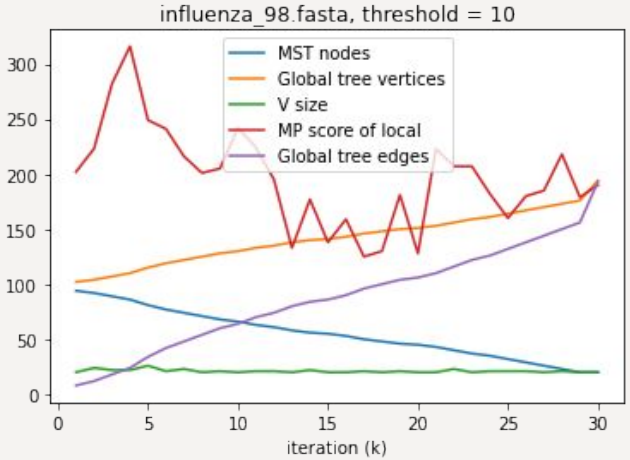
Crooks et al., 2004

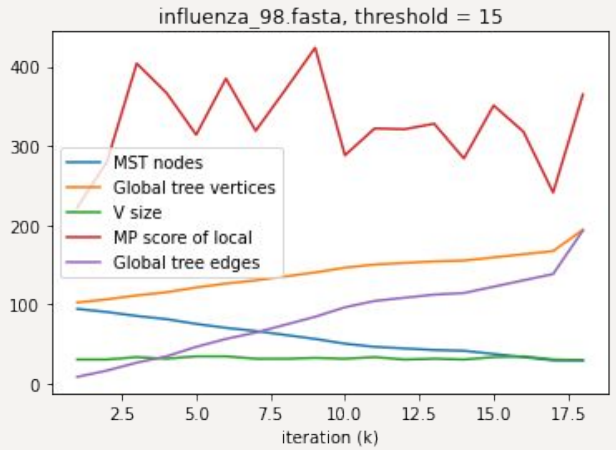
Outline

- Introduction
- Motivation
- Methods
- Materials
- **Results**
- Discussion
- Contributions

Method Performance: `influenza_98`







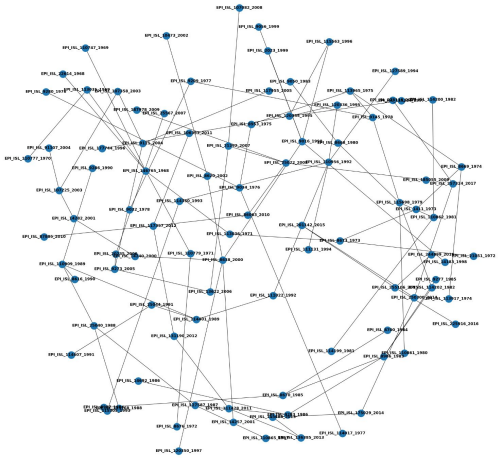
Runtime: larger tree space to search when threshold is increased

96 s	**637 s**	**1571 s**
(1 min 36 s)	(10 min 37 s)	(26 min 11 s)

influenza_98
threshold = 10

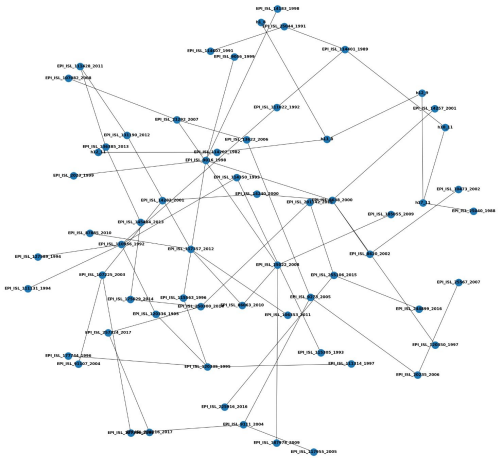
Method Performance: MST plots

start



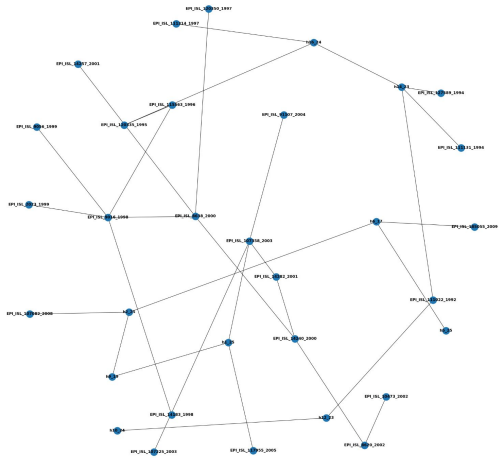
 # nodes: 98

checkpoint #1



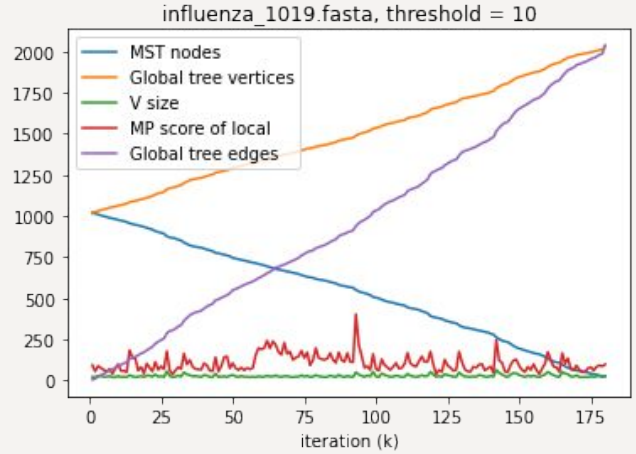
63

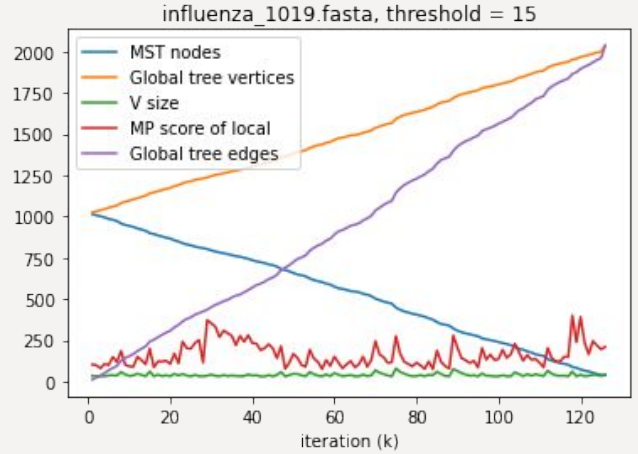
checkpoint #2

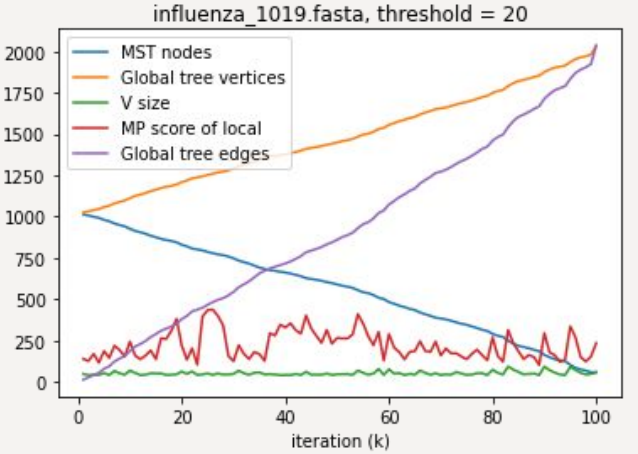


32

Method Performance: `influenza_1019`







Runtime:

3048 s
(50 min 48 s)

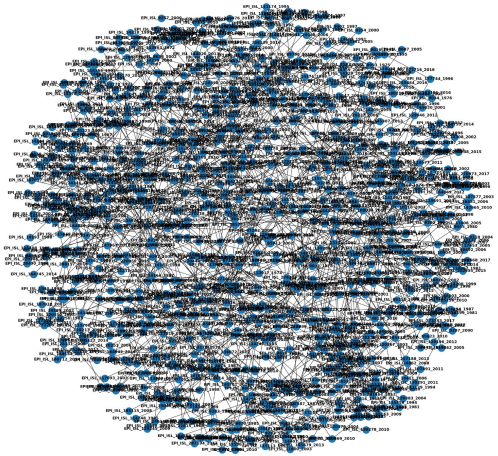
5212 s
(86 min 52 s)

11958 s
(199 min 18 s)

Method Performance: MST plots

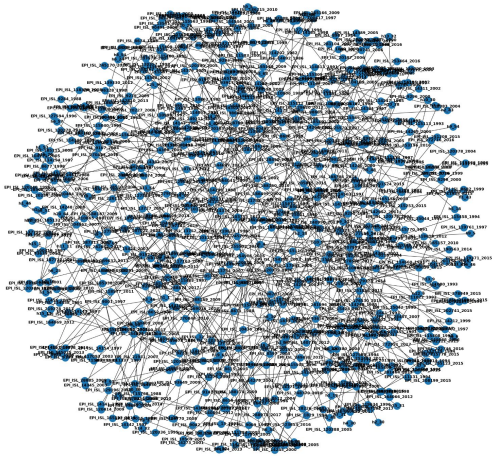
influenza_1019
threshold = 10

start



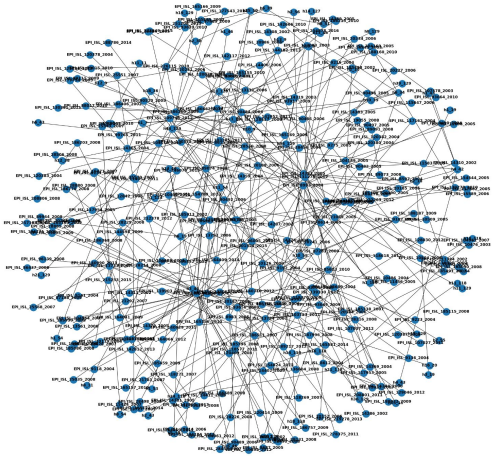
 # nodes: 1019

checkpoint #1



676

checkpoint #2



335

Runtime

Dataset	Threshold	Runtime (our method)	Runtime (RAxML benchmark)
`influenza_98`	5	96 s (1 min 36 s)	66 s (1 min 6 s)
	10	637 s (6 min 37 s)	
	15	1571 s (26 min 11 s)	
`influenza_1019`	10	3048 s (50 min 48 s)	6152 s (102 min 32 s)
	15	5212 s (86 min 52 s)	
	20	11958 s (199 min 18 s)	

OLD (skip)

Results vs Benchmark: Robinson-Foulds distance (RF)

$$normalized\ RF = \frac{|S_1 \cup S_2| - |S_1 \cap S_2|}{|S_1 \cup S_2|}$$

Dataset	Threshold	$\|S_1 \cup S_2\|$	$\|S_1 \cap S_2\|$	RF distance	RF max	Normalized RF distance
influenza_98	5	246	142	104	190	0.423
	10	249	139	110	190	0.442
	15	259	129	130	190	0.502
influenza_1019	10	2875	1197	1678	2032	0.584
	15	2881	1991	1690	2032	0.587
	20	2891	1181	1710	2032	0.591

Results vs Benchmark: Robinson-Foulds distance (RF)

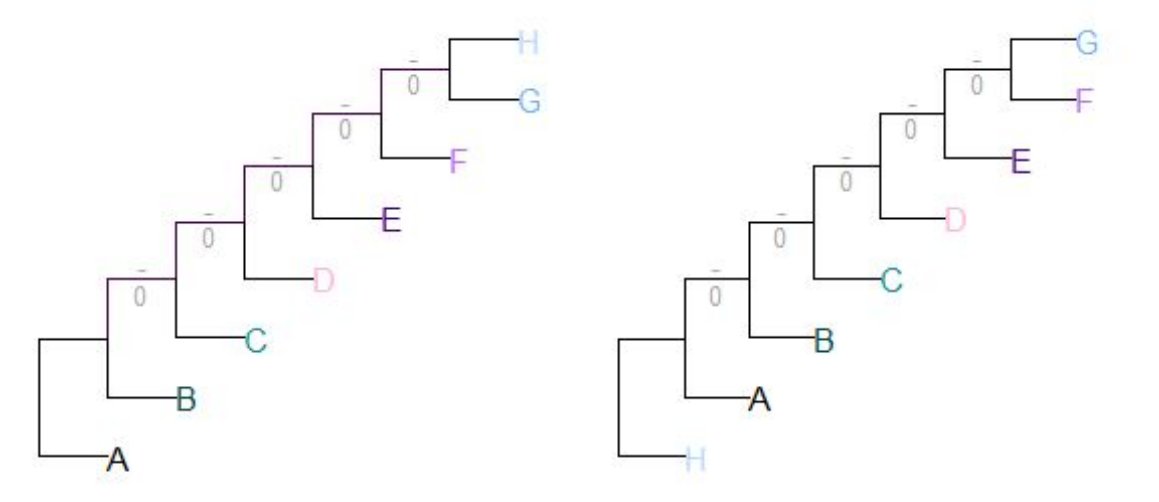
$$normalized\ RF = \frac{|S_1 \cup S_2| - |S_1 \cap S_2|}{max\ RF}$$

Dataset	Threshold	$\|S_1 \cup S_2\|$	$\|S_1 \cap S_2\|$	RF distance	RF max	Normalized RF distance
influenza_98	5	246	142	104	190	0.548
	10	249	139	110	190	0.579
	15	259	129	130	190	0.684
influenza_1019	10	2875	1197	1678	2032	0.826
	15	2881	1991	1690	2032	0.832
	20	2891	1181	1710	2032	0.842

Drawbacks of using RF distance

- Maximum RF distance occurs after moving a single leaf (H)

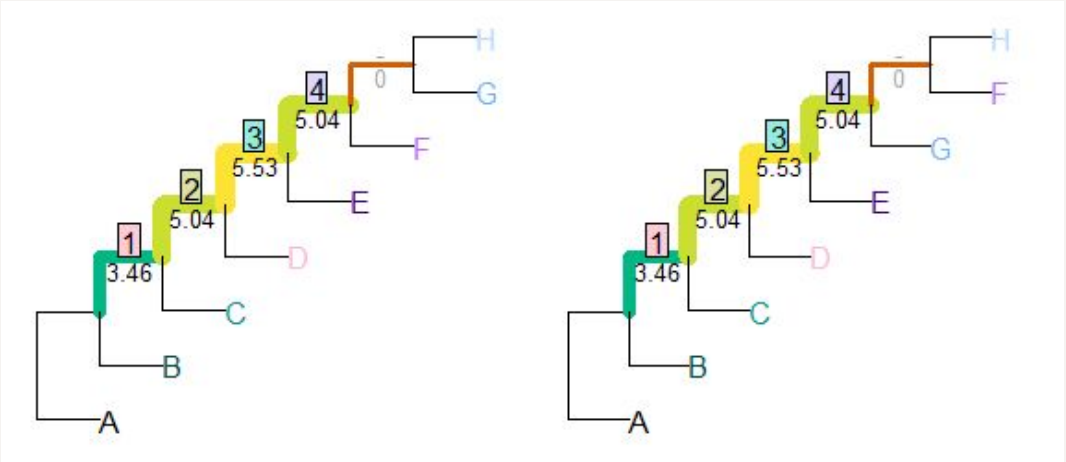
$$normalized\ RF = \frac{\left|S_1 \cup S_2\right| - \left|S_1 \cap S_2\right|}{max\ RF} = \frac{19 - 9}{10} = 1$$



Quantifying Information in Split

- Inspired by Claude Shannon's information theory
- Information contained in some splits is more instructive than others

$$inf\,ormation\ content\ (\,bits) = -\log_2(\,probability)$$



Normalized RF = 1
Normalized info RF = 0.153

Information corrected RF distance (info RF) = 22.54 − (3.46 + 5.04 + 5.53 + 5.04) = **3.46 bits**
Normalized info RF = 3.46/22.54 = **0.153**

OLD (skip)

Beyond Classic RF distance: generalized RF distance

$$inf ormation\ content\ (\ bits) = -\log_2(\ probability)$$

Dataset	Threshold	RF	Info RF	Shared Phylogenetic Info	Different Phylogenetic Info	Mutual Clustering Info
`influenza_98`	5	0.547	0.554	0.723	0.277	0.727
	10	0.579	0.622	0.700	0.304	0.701
	15	0.684	0.764	0.614	0.386	0.634

*`TreeDist` tool *wouldn't return results for* `influenza_1019` *dataset*

Metrics Beyond Classic RF distance

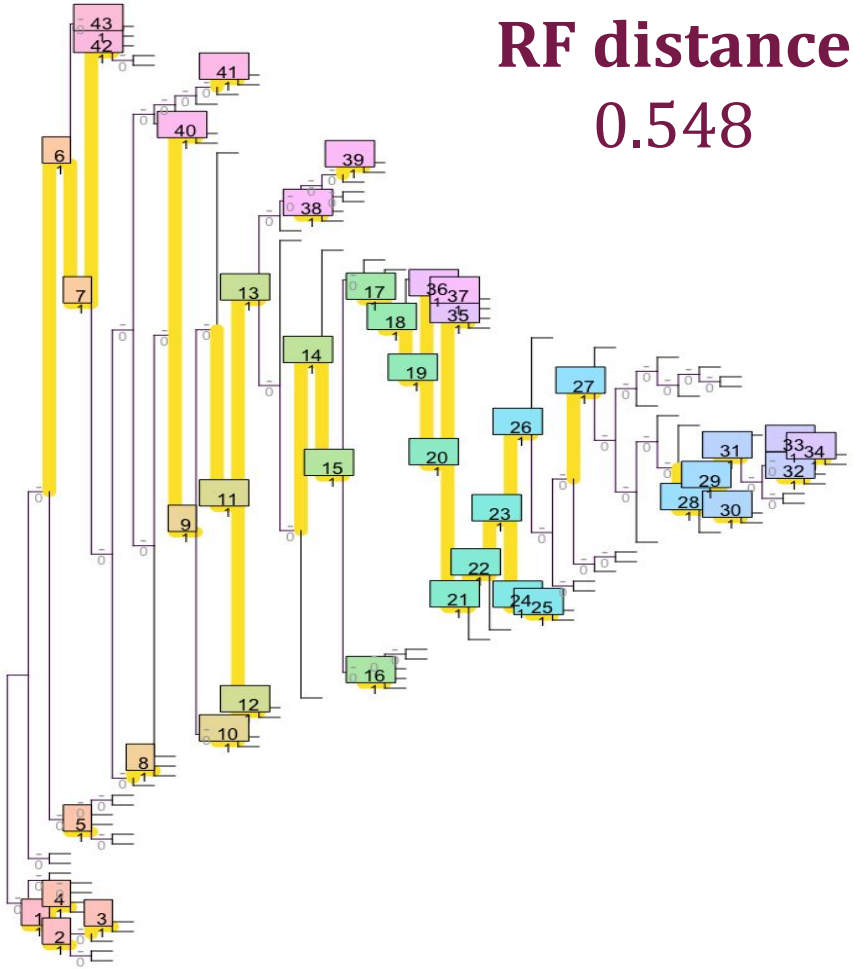
$$inf\,ormation\ content\ (\,bits) \;=\; -\log_2(\,probability)$$

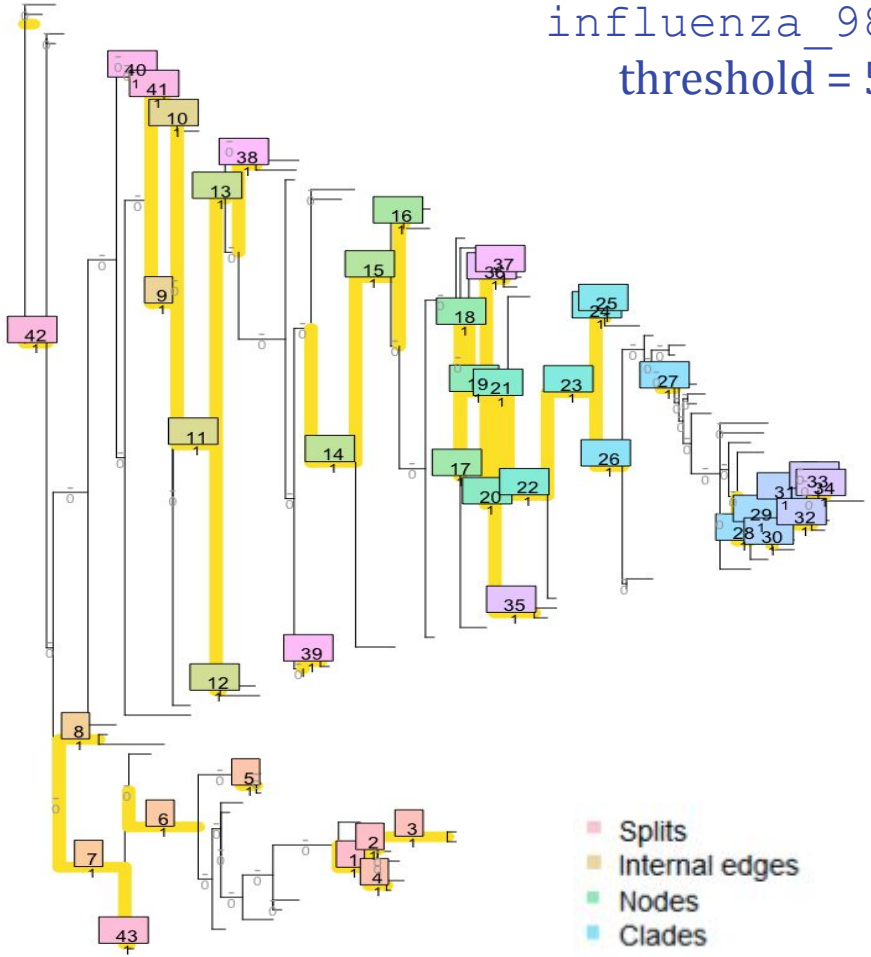
Dataset	Threshold	RF	Info RF	Shared Phylogenetic Info	Different Phylogenetic Info
`influenza_98`	5	0.547	0.554	0.723	0.277
	10	0.579	0.622	0.700	0.304
	15	0.684	0.764	0.614	0.386

*`TreeDist` tool *wouldn't return results for* `influenza_1019` *dataset*

RF distance
0.548

influenza_98
threshold = 5



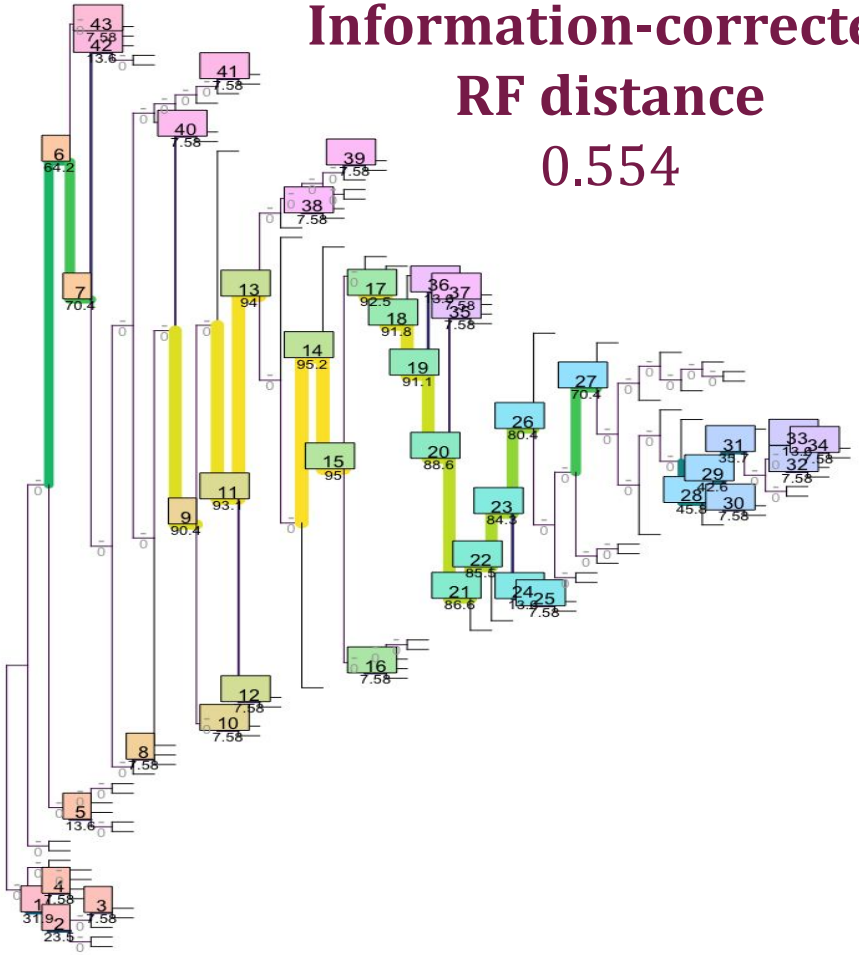


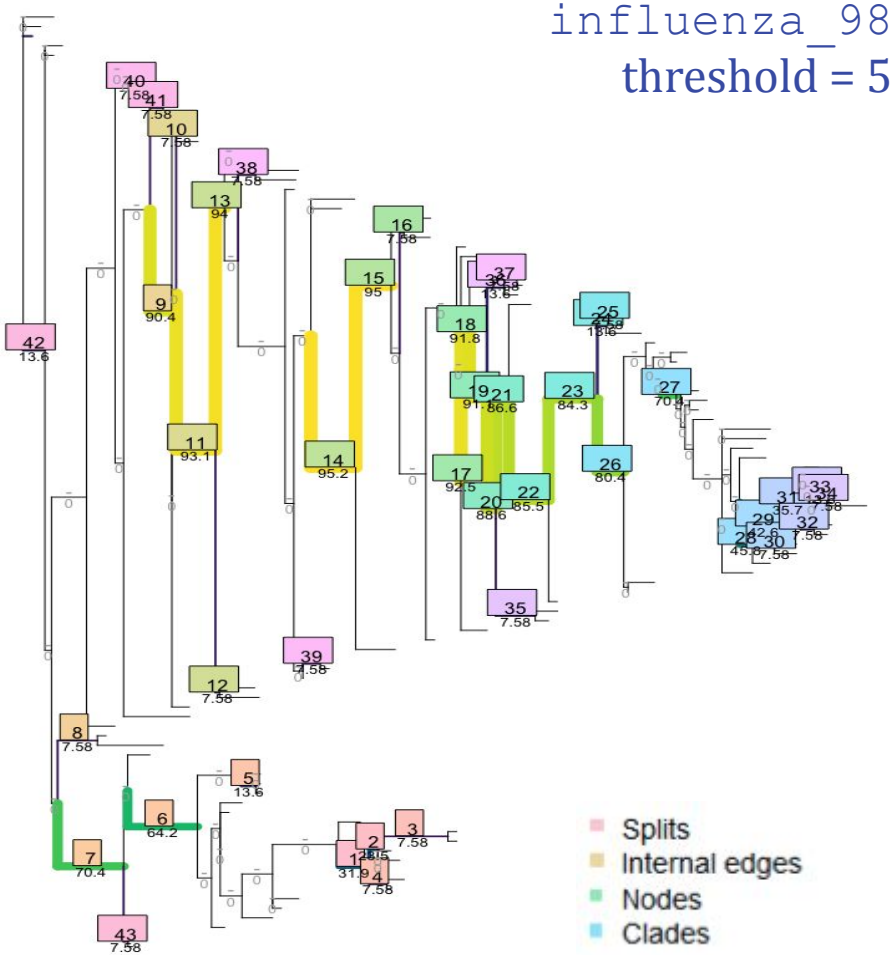
Information-corrected RF distance

0.554

influenza_98

threshold = 5



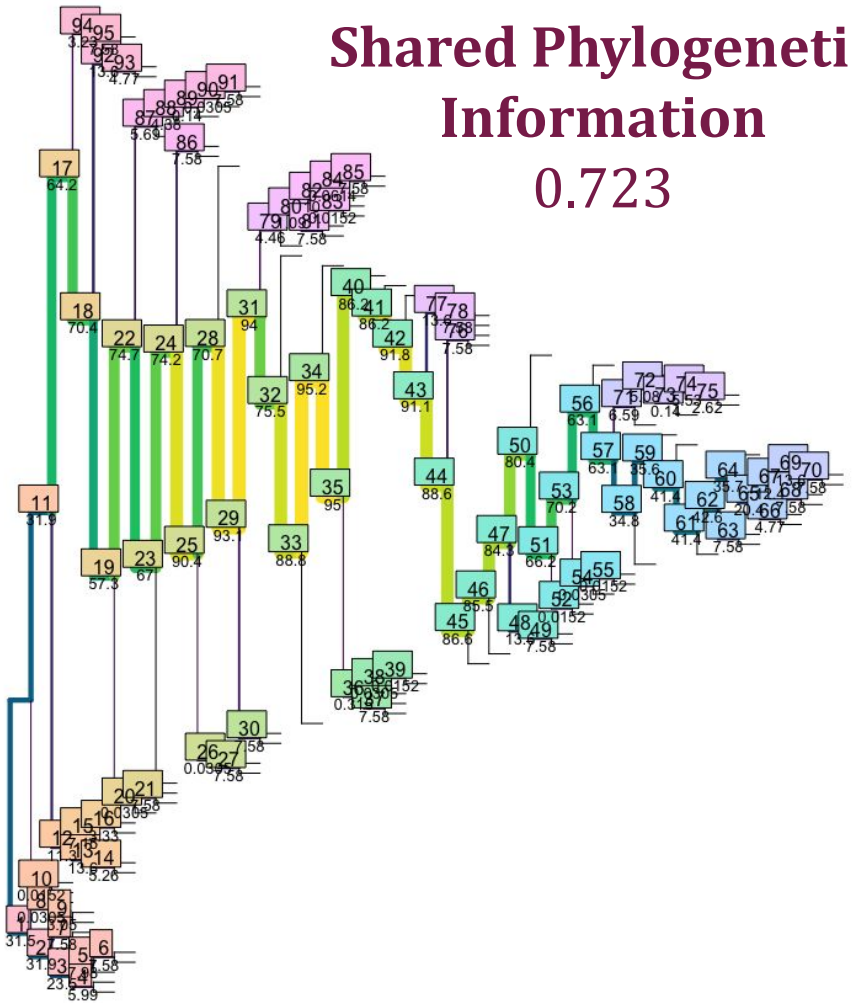


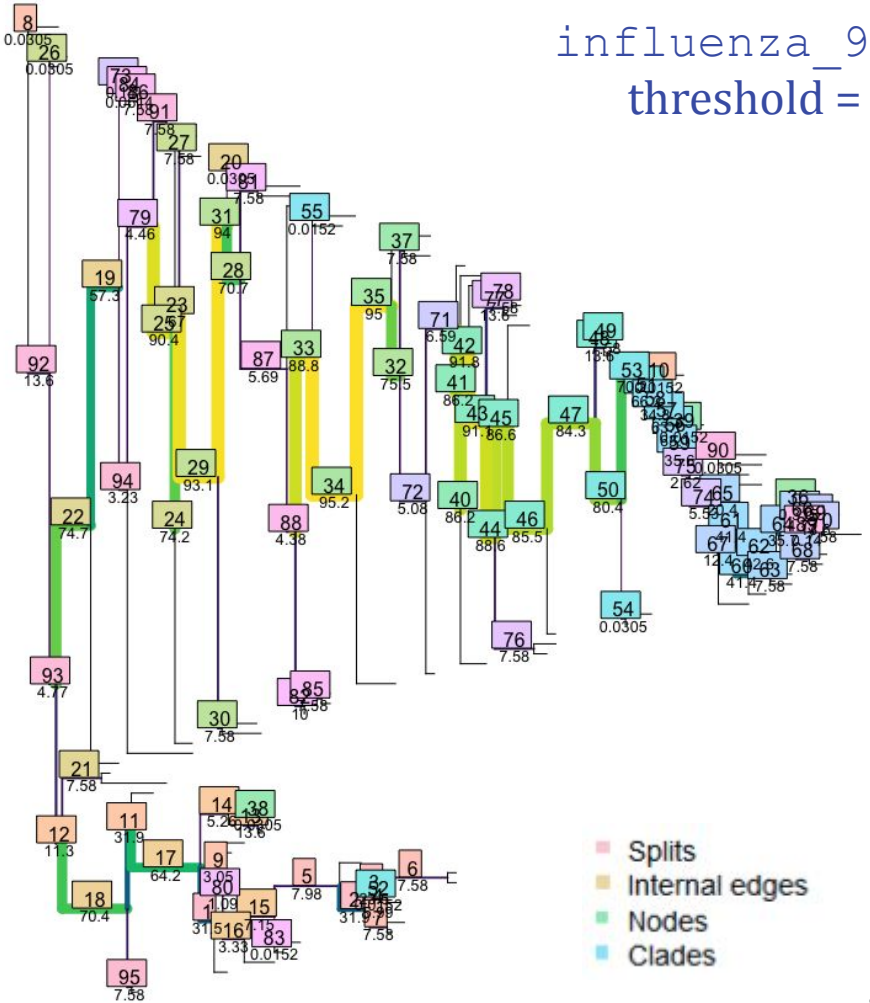
Shared Phylogenetic Information

0.723

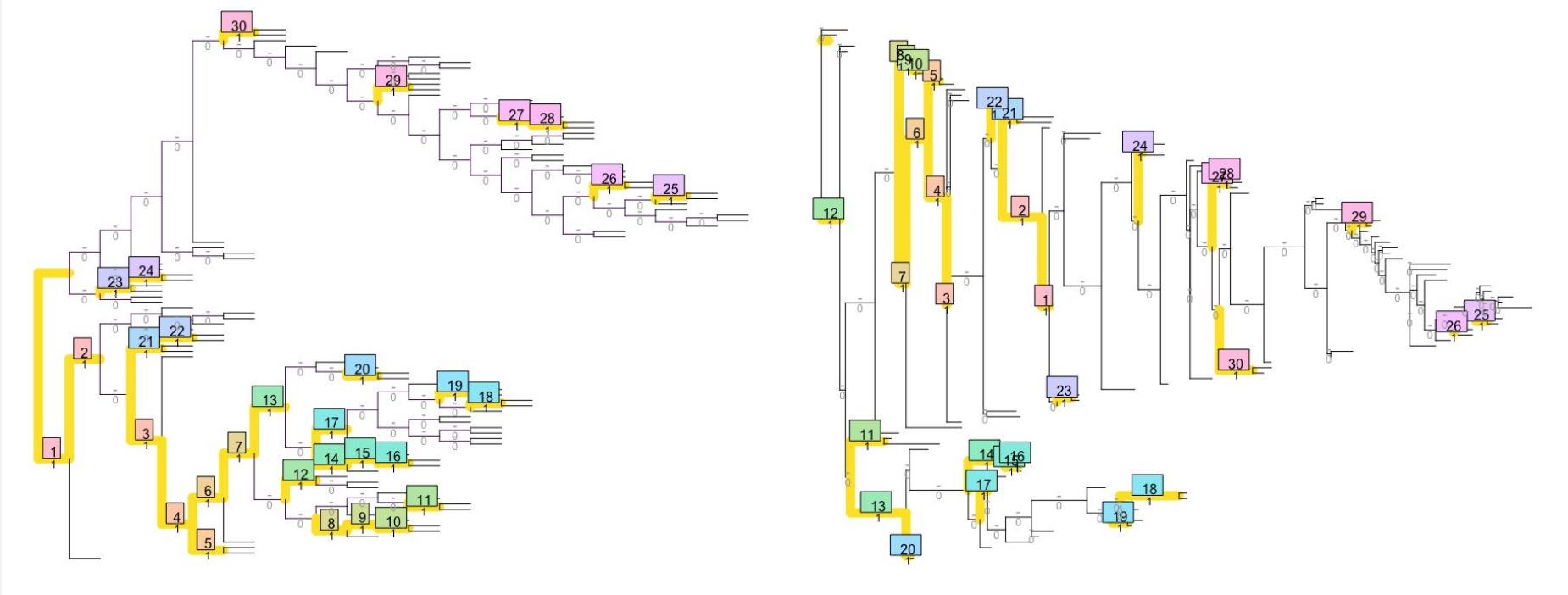
influenza_98

threshold = 5



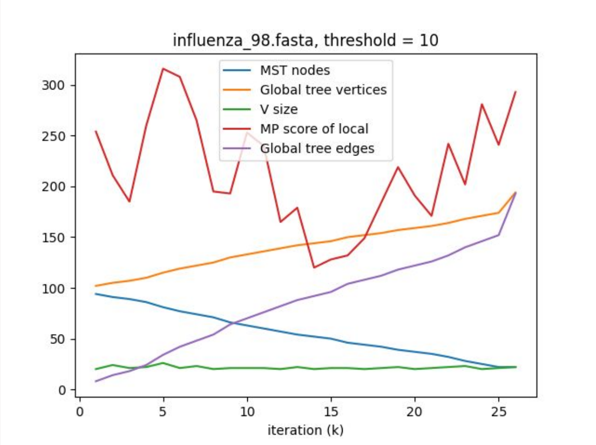


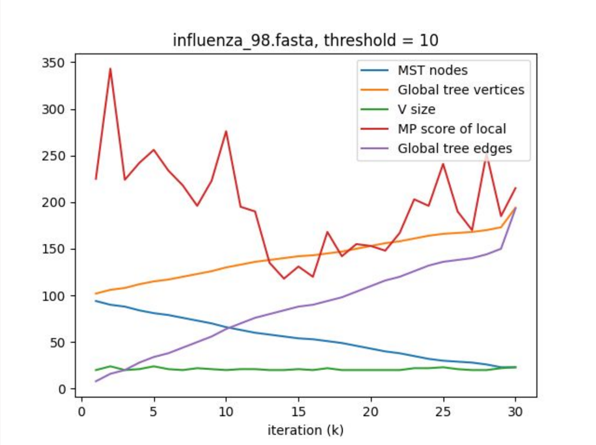
RF distance: `influenza_98`, threshold = 15

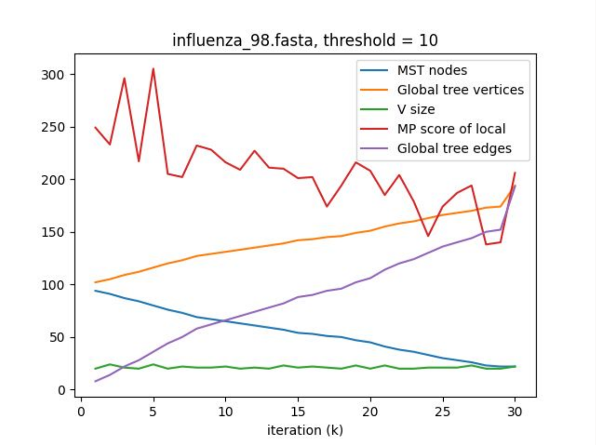


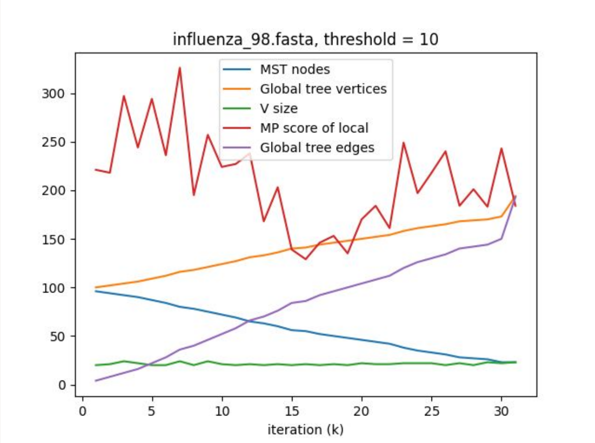
Method Performance: `influenza_98`, threshold = 10 (5 runs)

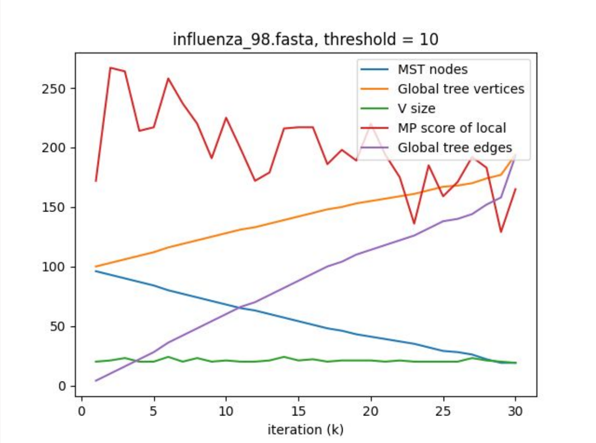
- Run 5 times with 5 random seeds for more robust statistics











Runtime: `influenza_98`, threshold = 10 (5 runs)

Runtime (average) = 475 s (7 min 55 s)

Dataset	Run	Runtime
`influenza_98` threshold = 10	1	348 s (5 min 48 s)
	2	497 s (8 min 17 s)
	3	425 s (7 min 5 s)
	4	624 s (10 min 24 s)
	5	482 s (8 min 2 s)
	average	475 s (7 min 55 s)

RF distance: `influenza_98`, threshold = 10 (5 runs)

normalized RF distance (average) = 0.473

$$normalized\ RF = \frac{\left|S_1 \cup S_2\right| - \left|S_1 \cap S_2\right|}{max\ RF}$$

Dataset	Run #	$\left\|S_1 \cup S_2\right\|$	$\left\|S_1 \cap S_2\right\|$	RF distance	RF max	Normalized RF distance
`influenza_98` threshold = 10	1	263	125	138	190	0.525
	2	256	132	124	190	0.484
	3	251	137	114	190	0.454
	4	247	141	106	190	0.429
	5	254	134	120	190	0.472

Normalized RF distance: `influenza_98`, threshold = 10

Pairwise: 5 runs with different random seeds

	Seed 1	**Seed 2**	**Seed 3**	**Seed 4**	**Seed 5**
Seed 1	0	0.552	0.574	0.558	0.579
Seed 2		0	0.536	0.574	0.547
Seed 3			0	0.541	0.579
Seed 4				0	0.530
Seed 5					0

Info RF distance: `influenza_98`, threshold = 10

Pairwise: 5 runs with different random seeds

	Seed 1	**Seed 2**	**Seed 3**	**Seed 4**	**Seed 5**
Seed 1	0	0.903	0.909	0.859	0.880
Seed 2		0	0.846	0.927	0.858
Seed 3			0	0.808	0.830
Seed 4				0	0.837
Seed 5					0

Shared Phylogenetic Info: `influenza_98`, threshold = 10

Pairwise: 5 runs with different random seeds

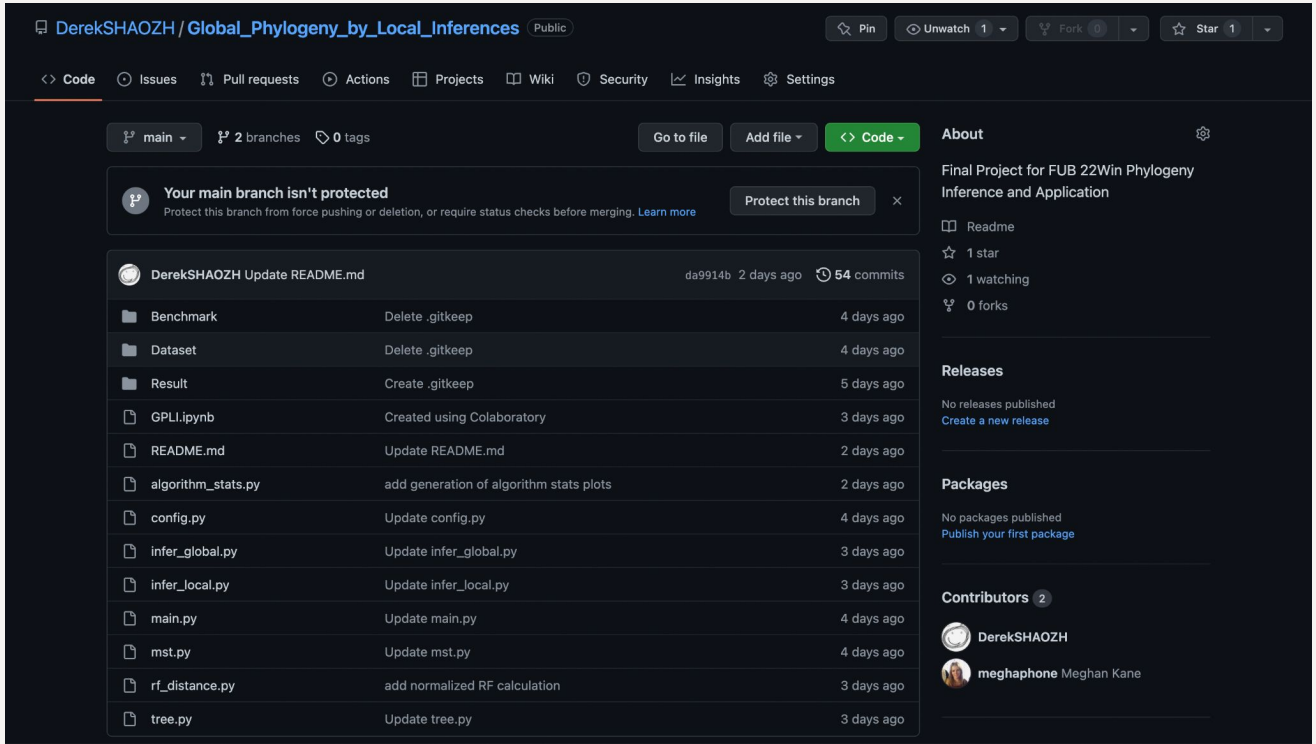
	Seed 1	**Seed 2**	**Seed 3**	**Seed 4**	**Seed 5**
Seed 1	1	0.567	0.503	0.541	0.516
Seed 2		1	0.544	0.558	0.583
Seed 3			1	0.645	0.604
Seed 4				1	0.635
Seed 5					1

Beyond Classic RF distance

Dataset	Threshold	RF	Info RF	Shared Phylogenetic Info	Different Phylogenetic Info	Mutual Clustering Info
influenza_98	5	0.547	0.547	0.723	0.277	0.727
	10	0.579	0.622	0.700	0.304	0.701
	15	0.684	0.764	0.614	0.386	0.634
influenza_1019	10					
	15					
	20					

Method implementation on GitHub

https://github.com/DerekSHAOZH/Global_Phylogeny_by_Local_Inferences



Outline

- Introduction
- Motivation
- Methods
- Materials
- Results
- **Discussion**
- Acknowledgement

Discussion

Summary of results interpretation

- Lower thresholds gave shorter runtime and better recall (accuracy)
- Inconsistency between global trees was possibly caused by local optima of MP
- Global phylogenetic analysis was prone to erroneous local inferences
- Other metrics beyond RF distance gave us other insights into tree similarity

Further Research

- Improve accuracy
- Multiple starting tree topologies to avoid local optima (20 in RAxML by default)

Outline

- Introduction
- Motivation
- Methods
- Materials
- Results
- Discussion
- **Contributions**

Contributions

Method Design - Prabhav, Derek & Meghan

Method Implementation:

- Tree.py - Derek & Meghan
- Mst.py - Meghan
- Infer_local.py - Derek
- Infer_global.py - Derek & Meghan

Program Execution - Derek

- Our program: req. 50GB RAM
- RAxML benchmark: Apple M1, 8 cores, 16 GB RAM, GTR model, ML

Data Analyses - Meghan

References

Bayzid, M. S., Hunt, T., & Warnow, T. (2014). Disk covering methods improve phylogenomic analyses. *BMC Genomics*, *15*(S6). https://doi.org/10.1186/1471-2164-15-s6-s7

Böcker, Sebastian, et al. “The Generalized Robinson-Foulds Metric.” *Lecture Notes in Computer Science*, 2013, pp. 156–169., https://doi.org/10.1007/978-3-642-40453-5_13.

Choi, M. J., Tan, V. Y. F., Anandkumar, A., & Willsky, A. S. (2010, September 14). *Learning latent tree graphical models*. arXiv.org. Retrieved March 7, 2023, from https://doi.org/10.48550/arXiv.1009.2722

Chor, B., & Tuller, T. (2006). Finding a maximum likelihood tree is hard. *Journal of the ACM*, *53*(5), 722–744. https://doi.org/10.1145/1183907.1183909

Crooks, G. E., Hon, G., Chandonia, J.-M., & Brenner, S. E. (2004). WebLogo: A sequence logo generator. *Genome Research*, *14*(6), 1188–1190. https://doi.org/10.1101/gr.849004

De Bruyn, A., Martin, D. P., & Lefeuvre, P. (2013). Phylogenetic Reconstruction Methods: An overview. *Methods in Molecular Biology*, 257–277. https://doi.org/10.1007/978-1-62703-767-9_13

Hagberg , A. A., Schult , D. A., & Swart , P. J. (2008). Exploring Network Structure, Dynamics, and Function using NetworkX. *Proceedings of the 7th Python in Science Conference (SciPy 2008)*.

Huerta-Cepas, J., Serra, F., & Bork, P. (2016). ETE 3: Reconstruction, analysis, and visualization of Phylogenomic Data. *Molecular Biology and Evolution*, *33*(6), 1635–1638. https://doi.org/10.1093/molbev/msw046v

References

Kalaghatgi, P. (2020). Phylogeny inference under the general markov model using MST-backbone. https://doi.org/10.1101/2020.06.30.180315

Kalaghatgi, P., & Lengauer, T. (2017). Computing phylogenetic trees using topologically related minimumspanning trees. *Journal of Graph Algorithms and Applications*, *21*(6), 1003–1025. https://doi.org/10.7155/jgaa.00447

Kumar, S., Stecher, G., & Tamura, K. (2016). Mega7: Molecular evolutionary genetics analysis version 7.0 for bigger datasets. *Molecular Biology and Evolution*, *33*(7), 1870–1874. https://doi.org/10.1093/molbev/msw054

Ranwez, V., & Gascuel, O. (2001). Quartet-based phylogenetic inference: Improvements and limits. *Molecular Biology and Evolution*, *18*(6), 1103–1116. https://doi.org/10.1093/oxfordjournals.molbev.a003881

Roch, S. (2006). A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *3*(1), 92–94. https://doi.org/10.1109/tcbb.2006.4

Smith, Martin R. *Comparing Splits Using Information Theory*, https://cran.r-project.org/web/packages/TreeDist/vignettes/information.html.

Yermanos, A. D., Dounas, A. K., Stadler, T., Oxenius, A., & Reddy, S. T. (2018). Tracing antibody repertoire evolution by systems phylogeny. *Frontiers in Immunology*, *9*. https://doi.org/10.3389/fimmu.2018.02149