Scalable Reconstruction of SARS-CoV-2 Phylogeny With Recurrent Mutations

Daniel Novikov,[1] Sergey Knyazev,[1] Mark Grinshpon,[2]
Pelin Icer Baykal,[4,5] Pavel Skums,[1] Alex Zelikovsky[1,3*]

[1]Department of Computer Science,
Georgia State University, Atlanta, GA 30303, USA

[2]Department of Mathematics and Statistics,
Georgia State University, Atlanta, GA 30303, USA

[3]World-Class Research Center
"Digital Biodesign and Personalized Healthcare", I.M. Sechenov
First Moscow State Medical University, Moscow, Russia

[4]Department of Biosystems Science and Engineering
ETH Zurich, CH-4058 Basel, Switzerland

[5]SIB Swiss Institute of Bioinformatics
CH-1015 Lausanne, Switzerland

[*]To whom correspondence should be addressed;
E-mail: alexz@gsu.edu.

February 11, 2022

Keywords: phylogenetic tree inference, recurrent mutations, SARS-CoV-2 se-

quences

Abstract: This paper presents a novel, scalable, character-based phylogeny algorithm for dense viral sequencing data called SPHERE (**S**calable **PH**ylog**E**ny with **RE**current mutations). The algorithm is based on an evolutionary model where recurrent mutations are allowed, but backward mutations are prohibited. The algorithm creates rooted character-based phylogeny trees, wherein all leaves and internal nodes are labeled by observed taxa. We show that SPHERE phylogeny is more stable than Nextstrain's, and that it accurately infers known transmission links from the early pandemic. SPHERE is a fast algorithm that can process more than 200,000 sequences in less then 2 hours, which offers a compact phylogenetic visualization of GISAID data.

Introduction

Equipped with the Next Generation Sequencing tools which are much more productive than ever before, the scientific community have collected an unprecedented amount of SARS-CoV-2 genomic data, enabling tracking the entire history of SARS-CoV-2 evolution with high precision [2]. This tracking requires advanced phylogeny reconstruction software. However, the current state-of-the-art phylogeny algorithms were created to handle significantly sparser genomic data than what is available for SARS-CoV-2. The majority of the popular phylogenetic tools assume that only the final product of evolution is available, while all intermediate evolutionary taxa are unknown. Also, these tools usually require significant computational resources, taking hours and sometimes days to reconstruct the SARS-CoV-2 phylogeny even for a small subset of the available genomes.

The SARS-CoV-2 sequencing data are similar to single cell sequencing data in cancer studies, where sequenced mutations from thousands of cancer cells offer a much closer look at cancer evolution. For such densely sequenced samples, perfect phylogeny models are more insightful than maximum likelihood models [5]. The perfect phylogeny model requires each mutation to occur only once and never disappear. More realistic cancer evolution models allow widespread loss and recurrence of mutations [8, 11, 1].

In contrast to cancer evolution, in viral evolution backward mutations are rarer than recurrent mutations [12]. In the evolution of the SARS-CoV-2 virus, recurrent mutations are mostly induced by the host's non-specific immune response. As they tend to be selectivity neutral, these mutations appear with higher frequency [13].

These properties of the SARS-CoV-2 genomic data, its density and a relatively high frequency of recurrent mutations, motivate the need for a parsimony-based phylogeny algorithm that is scalable to the entire collection of SARS-CoV-2 sequences available on GISAID (which numbers about 2.3 million sequences at the time of writing).

In this work, we follow the approach proposed in [7], which uses mutation trees [5] associated with character-based phylogenies that keep track of the accumulation of mutations in viral populations. We choose to employ parsimony-based phylogenetic analysis, because it explains evolutionary history with the fewest number of mutations to reproduce the variations in the genomic data. Targeting both accuracy and resolution as aspects of information contained in a phylogenetic tree, the maximum parsimony approach yields the results that are at least as good or better than probabilistic approaches [9].

We propose SPHERE, **S**calable **PH**ylog**E**ny with **RE**current mutations, as an efficient phylogeny reconstruction method that incorporates this model. Us-

ing this tool, we analyzed the available GISAID data with over 300,000 genome sequences. We compared the trees produced by SPHERE with those produced by Nextstrain, and demonstrated that SPHERE trees are more stable with respect to extending datasets. Finally, we validated that the phylogeny trees produced by SPHERE more reliably discover valid transmission links than other state-of-the-art algorithms.

Methods

Most Parsimonious Phylogeny Problem

Given a set of aligned sequences, possibly containing missing positions, and a reference sequence with no missing positions, find a character-based phylogenetic tree that is rooted at the reference sequence, that has the minimum total edge length, and that does not admit backward mutations.

An algorithm to meet these criteria should infer the phylogeny tree from a set S of size n of aligned SARS-CoV-2 genome sequences. All of these sequences are built on the nucleotide alphabet (A,C,T,G) and may contain missing positions (N). The reference sequence is required to have no missing positions, i.e., no occurrences of N. Under the assumption of allowing recurrent mutations but not allowing backward mutations, our algorithm creates a maximum parsimony phylogeny tree for the given dataset rooted at the reference sequence.

Nodes in the phylogeny tree represent sequence haplotypes that match one or more sequences in the data. Edges in the tree are directed, and represent the ancestor/descendant relationship between two sequences. The length of an edge is the Hamming distance between the sequences that label its endpoints.

Algorithm Overview

For a given set S of aligned sequences and a reference sequence, the proposed Algorithm 1 finds a maximum parsimony phylogenetic tree on the haplotypes in S, rooted at the reference sequence, with no backward mutations allowed.

A tree rooted at the reference sequence is initialized, and all sequences in S are inserted into a queue. Each sequence is then assigned a set of positions at which the sequence contains a different nucleotide (i.e., a mutation) from the reference sequence. Each sequence's priority in the queue is determined by the size of its set of reference mutations, i.e., by its Hamming distance from the root. Finally, each sequence's parent is initialized to be the root by default.

Sequences are removed from the queue and added to the tree in increasing order of Hamming distance from the root. To achieve the minimal total edge length, the parent of a node must be on the shortest path from the root, and it must be the lowest such parent in the tree. By default, the parent of a sequence is the root, and we look for a better parent as we add the sequence to the tree. Once the parent is determined and the sequence is inserted into the tree, we fill any missing positions in the sequence from its parent. If a sequence's Hamming distance to its parent is 0, the sequence is collapsed to the parent node and a new node is not created.

Parent Selection

When adding a sequence to the tree, the process of choosing its parent looks similar to a Dijkstra's shortest-path algorithm comparison. A sequence $\mathbf{u}$ is a parent of a sequence $\mathbf{v}$ if and only if (see Figure 1):

- $\text{distance}(\mathbf{root}, \mathbf{u}) + \text{distance}(\mathbf{u}, \mathbf{v}) = \text{distance}(\mathbf{root}, \mathbf{v})$;
- $\mathbf{u}$ is the lowest such node in the tree.

Algorithm 1 Character-Based Phylogeny

Input: Set S of aligned sequences (with possible missing positions), reference sequence (with no missing positions);

Output: Character-based phylogenetic tree on aligned sequences, rooted at the reference sequence. Backward mutations are not allowed, recurrent mutations are allowed.

```
Initialize a tree rooted at reference, and a queue of all sequences
For each sequence, assign set of positions where the sequence differs from reference
Set root as initial parent of all sequences
while the queue is not empty do
    Dequeue minimum priority sequence x
    for each node v in reversed order of vertex set do
        Check if v is a parent of x
        Break when a parent is found.
    end for
    if Hamming distance to x's parent is 0 then
        collapse x to its parent
    else add x to the tree:
        Connect x to its parent
        Fill missing positions in x from the parent
    end if
end while
```

Together, these two conditions imply that **u** is on the shortest path from **root** to **v**, that **u** immediately precedes **v**, and that the total length of the tree after inserting **v** is the minimal possible.

In the original implementation, see Algorithm 2, we updated the parents of all nodes in the queue on each insertion, as shown in Figure 2. Whenever we would pop a new sequence from the queue to add to the tree, we iterate through all nodes in the queue and check if the popped sequence is a better parent.

Algorithm 2 Parent Selection

```
Pop node x from queue and add it to the tree
Add edge from x's parent to x
for each node v in the queue do
    Update parent of v with x, if necessary
end for
```

With this approach, the parent selection procedure checks every possible

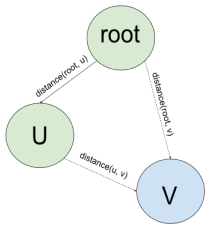

Figure 1: A visual representation of the parent selection constraint.

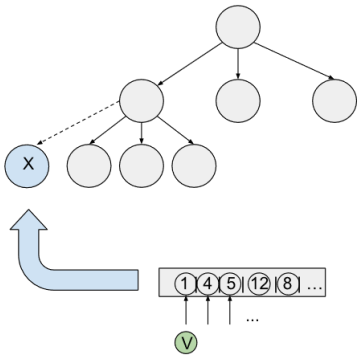

Figure 2: Choosing parents, original implementation. On each node insertion, for each sequence **v** remaining in the queue, check if the most recently inserted node **x** is a better parent of **v**.

edge between nodes throughout the execution of the algorithm, and thus gives us a quadratic run time of O(Size of queue) comparisons for each of the $O(n)$ node insertions, where n is the number of sequences.

With this implementation, the performance of our algorithm exceeded quadratic runtime (Figure 3). This is too slow for our goal of designing a scalable tool capable of processing available SARS-CoV-2 genomic data in a reasonable amount of time.

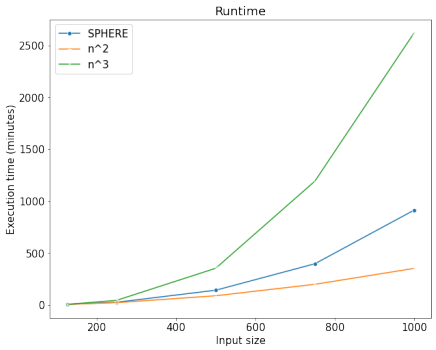

Figure 3: Runtime of our phylogeny algorithm in comparison to $O(n^2)$ and $O(n^3)$ runtimes. The blue curve represents the runtime of our algorithm, the orange curve represents $O(n^2)$ complexity, and the green curve $O(n^3)$ complexity. This version of parent selection admits a runtime that is slightly greater than $O(n^2)$.

Performance Improvements

Speeding Up Parent Selection Originally, after each node insertion, we iterated through the queue updating parents as needed. This mode of parent selection results in a quadratic runtime complexity, as each node is compared to each other node throughout execution of the algorithm. Instead, as shown in Figure 4, we decided to iterate through the tree vertices when looking for parents, rather than updating parents in the queue. In the worst case, this still has a quadratic runtime; however, this mode of operation allows us to escape the parent selection procedure early when a parent is found.

On each node insertion, our algorithm now iterates through the tree vertices in reverse order of insertion, looking for the parent that satisfies the triangle equality parent selection constraint shown in Figure 1. We can break this iteration through the tree early as soon as a parent better than the root is found, saving on the number of comparisons we need to make, see Algorithm 3.

We reduced the average complexity of our algorithm to below $O(n^2)$. Furthermore, the hidden complexity coefficient also dropped. In Figure 3, which illustrates the original runtime, we see that processing 1,000 sequences required

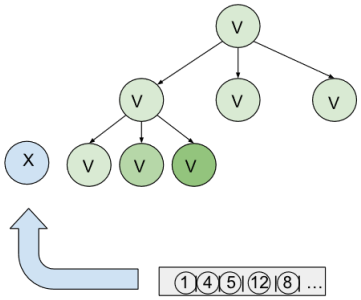

Figure 4: When adding node **x** to the tree, we search the nodes in the tree in reverse order of insertion (starting with the most recently inserted node) looking for parents that satisfy the triangle equality condition. We can stop iteration early every time we find a parent.

Algorithm 3 Faster Parent Selection

```
Pop node x from queue and add it to the tree
for each node v in reversed order of vertex set do
    Check if v is a parent of x
    Break when a parent is found
end for
Add edge from v to x
```

almost 1,000 minutes of runtime. As is evident in Figure 5, this change to the parent selection algorithm increased our speed, so that we could now process 1,000 sequences in just a couple of minutes. However, 8,000 sequences still required almost 6 hours of runtime.

Speeding up Hamming distance The length of SARS-CoV-2 genome is 30,000 nucleotides, and mutations have already been observed in more than 20,000 of them. However, any two available SARS-CoV-2 genome sequences differ by no more than 300 mutations.

We assigned each sequence a set of positions where it has mutated from the reference sequence. Then we compute the Hamming distance between two sequences as follows:

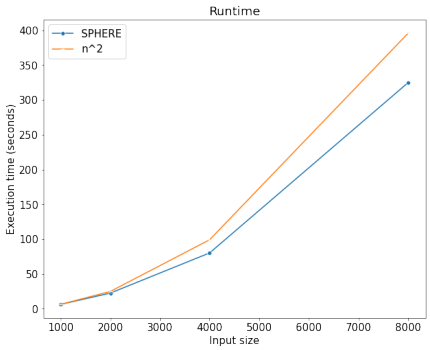

Figure 5: Performance after implementing the change to parent selection. This change has brought our algorithm's runtime down to below $O(n^2)$ in the average case. The hidden complexity coefficient has also decreased slightly, allowing us to process notably larger datasets in the same amount of time.

- The size of the symmetric difference between the two sets is added to the Hamming distance immediately.
- For each position in the intersection of the two sets, check if the sequences differ at those positions.

Results

Datasets

For comparison and evaluation purposes, we use the following five datasets:

- `C2C`: The Coast–to–Coast dataset consists of 168 global SARS-CoV-2 sequences, including 9 sequences from COVID-19 patients identified in Connecticut [3].
- `F22C`: This dataset consists of 1,293 global SARS-CoV-2 sequences, which are all GISAID sequences recorded up until February 22$^{\text{th}}$, 2020, as well as the sequences in the `C2C` dataset. all GISAID sequences recorded up until February 22th, 2020, as well as the sequences in the C2C dataset.

- `M14`: This dataset consists of 9,286 global SARS-CoV-2 sequences, which are all GISAID sequences recorded up until March 14^{th}, 2020.

- `M22`: This dataset consists of 21,473 global SARS-CoV-2 sequences, which are all GISAID sequences recorded up until March 22^{th}, 2020.

- `ETL`: The Early Transmission Links dataset consists of 294 global SARS-CoV-2 sequences collected before March 9^{th}, 2020. This dataset was constructed to match the 25 known transmission links. These transmission links were collected from news articles detailing transmissions prior to the pandemic declaration, in the MIDAS 2019 Novel Coronavirus Repository.

Since all sequences in the `C2C` and `ETL` datasets were recorded before March 14^{th}, 2020, both are entirely contained in the `M14` and `M22` datasets.

Validation Metrics

Comparing Phylogenetic Trees One of the standard tools for comparing phylogenetic trees is the Robinson-Foulds (RF) distance, which is the size of the symmetric difference of the sets of bipartitions in two trees on the same set of taxa. Since the number of bipartitions in a SPHERE tree is significantly less than in the Nextstrain tree for the same taxa, we separately report two differences, each representing the number of bipartitions in one tree that are not present in the other tree.

However, the RF metric suffers from the several drawbacks including small range, over-sensitivity to minor differences, and assigning higher distances to more balanced trees [9]. Therefore, we also report the triplet and quartet distances that provide more precise measures of dissimilarity that don't suffer from the same shortcomings as bipartions [9].

We use `Dendropy` [10] and `tqDist` [6] to calculate the RF distance and the

triplet and quartet distances, respectively. Both tools require input trees in the Newick format with only leaves labeled by taxa. We convert a SPHERE tree to the Newick format as follows: each internal node labeled by a taxon is replaced by an unlabeled node with a child labeled by the same taxon; if a node is labeled by several taxa, we replace it with a new internal node, which is the parent of the new leaf nodes, each labeled by a single taxon (Figure 6).

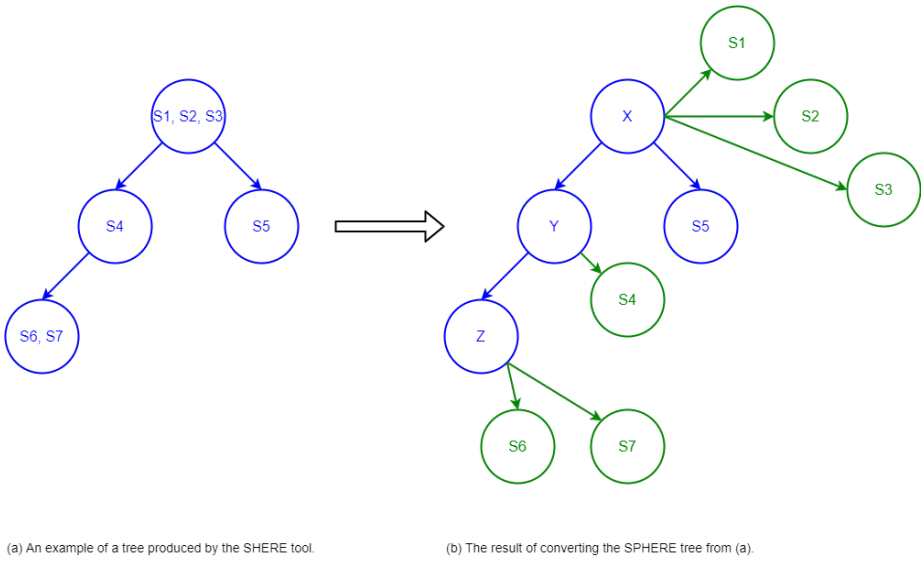

Figure 6: Example of converting a SPHERE tree into the Newick format. Three new internal nodes X, Y, and Z are introduced. The node X becomes the parent of S1, S2, and S3. The node Y becomes the parent of S4. The node Z becomes the parent of S6 and S7.

Transmission Network Comparison When geographical metadata for SARS-CoV-2 sequences is available, the phylogeny trees produced by our method imply a SARS-CoV-2 transmission network. We analyze the predictive value of the transmission network by computing a phylogeny tree on the `ETL` dataset, extracting its implied network, and comparing it to the known transmission links that accompany the dataset.

In a SPHERE phylogeny, a directed transmission link between two locations

is defined by a parent/child relationship in the tree between two sequences sampled at those locations. For matching sequences collapsed into a single node in the tree, we resolve the direction of their transmission link as earlier date $\rightarrow$ later date. We calculate precision and recall of the transmission network as follows:

$$\text{Precision} = \frac{\text{Number of true links predicted by the tool}}{\text{Total number of predicted links}}$$
$$\text{Recall} = \frac{\text{Number of true links predicted by the tool}}{\text{Total number of given true links}}$$

Phylogenetic trees for C2C data

The SPHERE phylogeny tree has all internal nodes annotated (Figure 7) in comparison to the Nextstrain tree (Figure 8). Nodes in both trees are colored by the locations they represent, where multi-color nodes in the SPHERE tree have assigned sequences from different locations. The sizes of the nodes in the SPHERE tree are proportional to the number of sequences they represent. Edges in the SPHERE tree are labeled by the number of mutations from parent to child haplotype. Some edges in the Nextstrain tree are labeled by codes of mutations.

Comparing Phylogenetic Trees

We compare eight trees created by applying the two phylogeny tools, SPHERE and Nextstrain, to the four datasets: `C2C`, `F22C`, `M14`, and `M22` (see Table 1). Nextstrain prunes highly divergent sequences, leading to a slight reduction of the number of sequences for `F22C` and `M14`. The number of edges in SPHERE trees is much smaller than in Nextstrain trees since SPHERE does not introduce internal nodes and collapses taxa that agree with each other in the sequenced

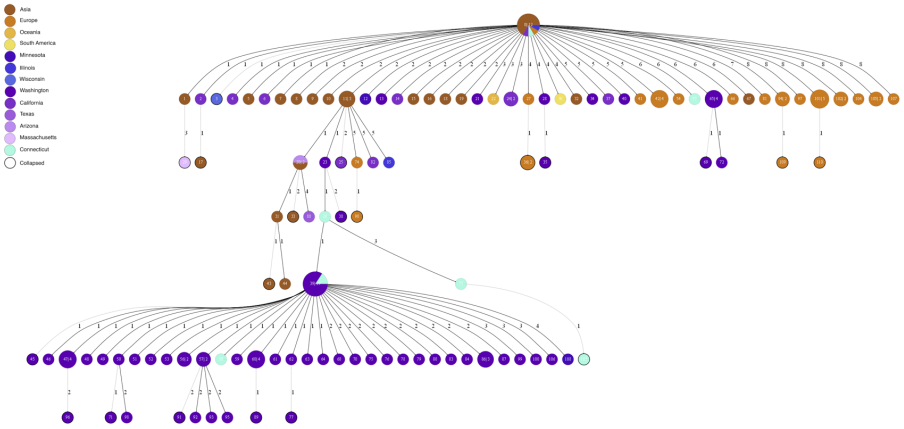

Figure 7: The phylogeny tree on the C2C dataset produced by SPHERE. Each edge is annotated by the number of mutations between the parent and the child. The sizes of the nodes represent the number of sequences assigned to the node. Multi-color nodes have assigned sequences from different locations.

positions.

Tree	C2C_S	C2C_N	F22C_S	F22C_N	M14_S	M14_N	M22_S	M22_N
# Taxa	168	168	1,293	1,283	9,286	9,265	21,473	21,473
# Edges	110	277	694	2,265	3,843	17,108	9,010	39,722

Table 1: Eight phylogeny trees are created by applying SPHERE ("_S") and Nextstrain ("_N") to the four datasets C2C, F22C, M14, and M22.

For each pair of trees, we report the directional Robinson-Foulds distance (see Table 2), the triplet distance (see Table 3), and the quartet distance (see Table 4). All distances are with respect to the common taxa between the trees being compared, normalized by the total number of bipartitions, triplets, or quartets, respectively.

Our results show that SPHERE is more stable than Nextstrain. Indeed, consider the chain of datasets C2C $\subset$ F22C $\subset$ M14 $\subset$ M22. A more stable phylogeny reconstruction method has lesser distances between trees for consecutive

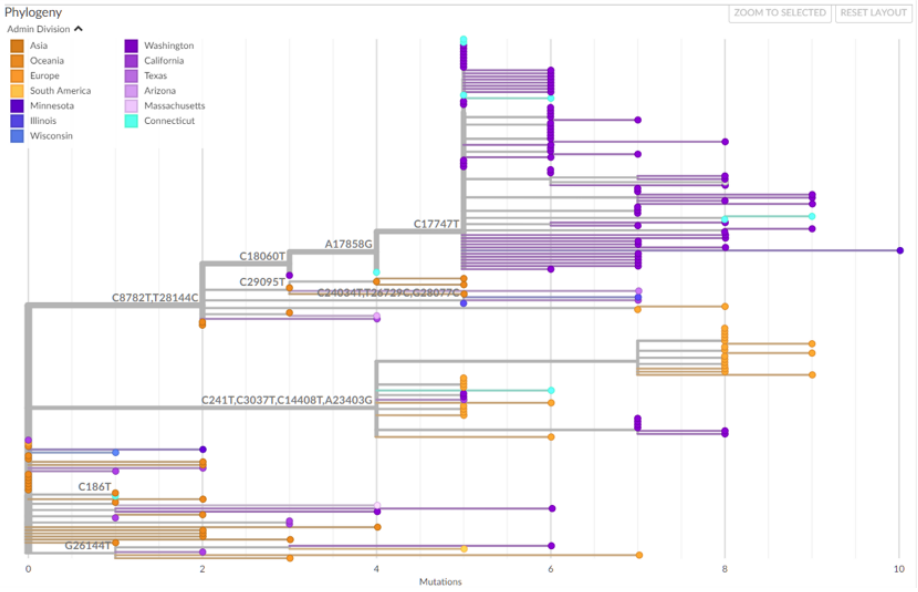

Figure 8: The phylogeny tree on the `C2C` dataset produced by Nextstrain.

datasets. The corresponding normalized directed RF distances for SPHERE are 6.45%, 12.89%, and 18.28%, respectively; while for the trees produced by Nextstrain the distances are much larger 57.41%,63.37%, and 69.17%, respectively. Similarly, the normalized triplet distances for SPHERE are 9.41%, 0.26%, and 16.12%, respectively; while for Nextstrain, they are 8.36%, 25.33%, and 22.48%, respectively. Finally, the normalized quartet distances for SPHERE are 10.18%, 0.55%, and 23.55%, respectively; while for Nextstrain, the quartet distances are 17.76%, 26.83%, and 12.05%, respectively. We can see that in most cases SPHERE method is more stable than Nextstrain.

Inferring Transmission Links

We have compared precision and recall of transmission networks inferred with SPHERE, the ILP-based character state phylogeny (CS-phylogeny) [7] and the

	C2C_S	C2C_N	F22C_S	F22C_N	M14_S	M14_N	M22_S	M22_N
C2C_S	0	48.39	6.45	45.16	16.13	45.16	16.13	45.16
C2C_N	85.19	0	83.33	57.41	80.56	56.48	79.63	55.56
F22C_S	21.62	51.35	0	50.26	12.89	52.11	18.04	52.58
F22C_N	87.12	65.15	90.32	0	89.49	63.37	89.29	63.88
M14_S	38.1	50.0	19.14	49.76	0	49.68	18.28	51.43
M14_N	85.22	59.13	91.03	64.63	93.08	0	92.02	69.17
M22_S	43.48	52.17	25.7	50.0	26.19	47.48	0	50.68
M22_N	86.29	61.29	91.25	66.16	93.28	69.14	93.0	0

Table 2: The normalized directional RF distances between trees, given as percentages. Each entry represents the number of bipartitions in the row tree that are not present in the column tree, normalized by the total number of bipartitions in the row tree.

	C2C_S	C2C_N	F22C_S	F22C_N	M14_S	M14_N	M22_S	M22_N
C2C_S	0	30.44	9.41	31.19	11.13	36.23	10.18	31.4
C2C_N	30.44	0	22.53	8.36	21.38	18.0	22.38	7.59
F22C_S	9.41	22.53	0	49.98	0.26	57.37	0.6	52.07
F22C_N	31.19	8.36	49.98	0	49.92	25.33	50.18	18.98
M14_S	11.13	21.38	0.26	49.92	0	37.95	16.12	23.95
M14_N	36.23	18.0	57.37	25.33	37.95	0	44.59	22.48
M22_S	10.18	22.38	0.6	50.18	16.12	44.59	0	29.54
M22_N	31.4	7.59	52.07	18.98	23.95	22.48	29.54	0

Table 3: Triplets comparisons. Values represent the normalized triplet distance between each pair of trees, given as percentages.

	C2C_S	C2C_N	F22C_S	F22C_N	M14_S	M14_N	M22_S	M22_N
C2C_S	0	30.96	10.18	33.84	13.28	35.89	12.58	32.82
C2C_N	30.96	0	23.93	17.76	22.48	15.48	23.38	15.15
F22C_S	10.18	23.93	0	45.97	0.55	50.31	1.22	47.25
F22C_N	33.84	17.76	45.97	0	45.83	26.83	46.28	24.43
M14_S	13.28	22.48	0.55	45.83	0	33.84	23.55	31.68
M14_N	35.89	15.48	50.31	26.83	33.84	0	38.84	12.05
M22_S	12.58	23.38	1.22	46.28	23.55	38.84	0	37.4
M22_N	32.82	15.15	47.25	24.43	31.68	12.05	37.4	0

Table 4: Quartets comparisons. Values represent the normalized quartet distance between each pair of trees, given as percentages.

character-based phylogeny NETWORK5011CS [4]. SPHERE has the best recall over existing methods (Table 5). Note that all methods have small precision because the ETL dataset contains only verified transmission links. The number or such links is only 25 for 294 nodes. There should be other transmission links, however they are not validated. SPHERE has comparable precision to other methods that indicates that all methods output similar number of predicted transmission links.

Tool	Recall %	Precision %
SPHERE	**88**	4.3
CS-phylogeny	80	4.76
NETWORK5011CS	72	4.99
SPHERE-directed	84	4.3

Table 5: Comparison of SPHERE with CS-phylogeny and NETWORK5011CS tools without taking in account the transmission direction. SPHERE-directed also takes in account the transmission direction.

Runtime

We ran SPHERE on the cluster hardware consisting of 128 cores Intel(R) Xeon(R) CPU E7-4850 v4 CPU @ 2.10GHz, with 3 TB of RAM, running Ubuntu 16.04.7 LTS.

Figure 9 shows that SPHERE is indeed a scalable method with a sub-quadratic runtime. For example, it is able to process 200,000 sequences in two hours, while Nextstrain requires 2 days to process 21,000 sequences on the same hardware, with 32 cores dedicated to the process.

Conclusion and Future Work

It is shown that the development of a character-based shortest-path phylogenetic tree is viable. First, a shortest-path phylogeny is fast and scalable. Second, the resulting maximum parsimony trees produced by our method are more stable

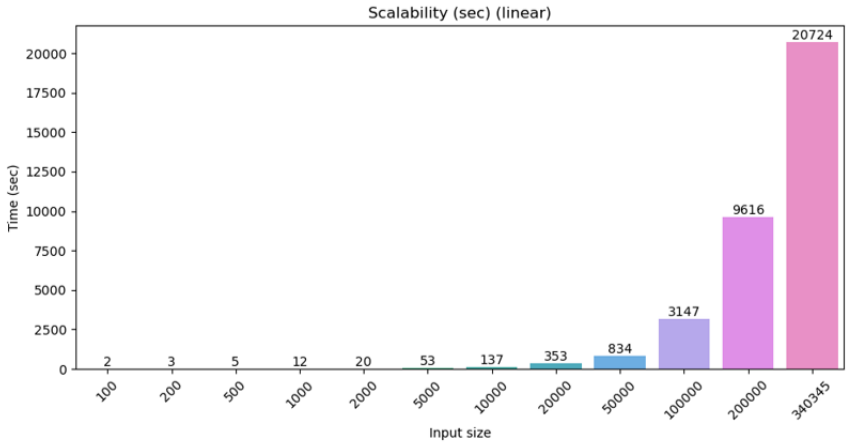

Figure 9: A graph of input size vs runtime. Units are in seconds. This figure highlights the significant performance improvements observed after optimizing the parent selection and Hamming distance methods.

than the Nextstrain's maximum likelihood tree. Third, the inferred transmission network quality is higher or comparable with existing tools. We plan to incorporate sparse backward mutations into the algorithm and add Steiner points corresponding to internal vertices.

Acknowledgements

DN, SK, and AZ were partially supported by NSF grants 1564899 and 16119110 and by NIH grant 1R01EB025022-01. PS was partially supported by NIH grant 1R01EB025022-01 and NSF grant 2047828. SK was partially supported by the GSU Molecular Basis of Disease Fellowship.

References

[1] Simone Ciccolella, Camir Ricketts, Mauricio Soto Gomez, Murray Patterson, Dana Silverbush, Paola Bonizzoni, Iman Hajirasouliha, and Gianluca Della Vedova. Inferring cancer progression from single-cell sequencing while allowing mutation losses. *Bioinformatics*, 37(3):326–333, February 2021.

[2] S. Elbe and G. Buckland-Merrett. Data, disease and diplomacy: GISAID's innovative contribution to global health. *Global Challenges*, 1:33–46, 2017.

[3] Joseph R Fauver, Mary E Petrone, Emma B Hodcroft, Kayoko Shioda, Hanna Y Ehrlich, Alexander G Watts, Chantal BF Vogels, Anderson F Brito, Tara Alpert, Anthony Muyombwe, et al. Coast-to-coast spread of sars-cov-2 during the early epidemic in the united states. *Cell*, 181(5):990–996, 2020.

[4] Peter Forster, Lucy Forster, Colin Renfrew, and Michael Forster. Phylogenetic network analysis of sars-cov-2 genomes. *Proceedings of the National Academy of Sciences*, 117(17):9241–9243, 2020.

[5] Katharina Jahn, Jack Kuipers, and Niko Beerenwinkel. Tree inference for single-cell data. *Genome biology*, 17(1):86, 2016.

[6] Andreas Sand, Morten K. Holt, Jens Johansen, Gerth Stølting Brodal, Thomas Mailund, and Christian N. S. Pedersen. tqDist: a library for computing the quartet and triplet distances between binary or general trees. *Bioinformatics*, 30(14):2079–2080, 03 2014.

[7] Pavel Skums, Alexander Kirpich, Pelin Icer Baykal, Alex Zelikovsky, and Gerardo Chowell. Global transmission network of sars-cov-2: from outbreak to pandemic. *medRxiv*, 2020.

[8] Pavel Skums, Viachaslau Tsyvina, and Alex Zelikovsky. Inference of clonal selection in cancer populations using single-cell sequencing data. *Bioinformatics*, 35(14):i398–i407, 2019.

[9] M. R. Smith. Bayesian and parsimony approaches reconstruct informative trees from simulated morphological datasets. *Biology Letters*, 15, 2019.

[10] Jeet Sukumaran and Mark T. Holder. Dendropy phylogenetic computing library, `https://dendropy.org/`, 2009-2021.

[11] Viachaslau Tsyvina, Alex Zelikovsky, Sagi Snir, and Pavel Skums. Inference of mutability landscapes of tumors from single cell sequencing data. *PLoS Computational Biology*, 16(11):e1008454, 2020.

[12] Lucy van Dorp, Mislav Acman, Damien Richard, Liam P. Shaw, Charlotte E. Ford, Louise Ormond, Christopher J. Owen, Juanita Pang, Cedric C.S. Tan, Florencia A.T. Boshier, Arturo Torres Ortiz, and François Balloux. Emergence of genomic diversity and recurrent mutations in sars-cov-2. *Infection, Genetics and Evolution*, 83:104351, 2020.

[13] Lucy van Dorp, Damien Richard, Cedric C S Tan, Liam P Shaw, Mislav Acman, and François Balloux. No evidence for increased transmissibility from recurrent mutations in sars-cov-2. *Nature communications*, 11(1):5986, November 2020.