HHS Public Access

Author manuscript

*SIAM J Math Data Sci*. Author manuscript; available in PMC 2021 June 11.

Published in final edited form as:

*SIAM J Math Data Sci*. 2021 ; 3(1): 113–141. doi:10.1137/20m1365715.

# Spectral neighbor joining for reconstruction of latent tree Models

**Ariel Jaffe**[1], **Noah Amsel**[1], **Yariv Aizenbud**[1], **Boaz Nadler**[2], **Joseph T. Chang**[3], **Yuval Kluger**[1,4,5]

[1]Program in Applied Mathematics, Yale University, New Haven, CT 06511

[2]Department of Computer Science, Weizmann Institute of Science, Rehovot, 76100, Israel

[3]Department of Statistics, Yale University, New Haven, CT 06520, USA

[4]Interdepartmental Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT 06511

[5]Department of Pathology, Yale University New Haven, CT 06511

## Abstract

A common assumption in multiple scientific applications is that the distribution of observed data can be modeled by a latent tree graphical model. An important example is phylogenetics, where the tree models the evolutionary lineages of a set of observed organisms. Given a set of independent realizations of the random variables at the leaves of the tree, a key challenge is to infer the underlying tree topology. In this work we develop Spectral Neighbor Joining (SNJ), a novel method to recover the structure of latent tree graphical models. Given a matrix that contains a measure of similarity between all pairs of observed variables, SNJ computes a spectral measure of cohesion between groups of observed variables. We prove that SNJ is consistent, and derive a sufficient condition for correct tree recovery from an estimated similarity matrix. Combining this condition with a concentration of measure result on the similarity matrix, we bound the number of samples required to recover the tree with high probability. We illustrate via extensive simulations that in comparison to several other reconstruction methods, SNJ requires fewer samples to accurately recover trees with a large number of leaves or long edges.

### Keywords

latent variable models; Markov random fields; evolutionary trees; singular values; spectral methods; neighbor joining; phylogenetics; tree graphical model

### AMS:

62H22; 62M05; 62M15; 15A18

## 1 Introduction

Learning the structure of an unobserved tree graphical model is a fundamental problem in many scientific domains. For example, phylogenetic tree reconstruction methods are used to

infer the evolutionary history of different organisms, see [14, 57] and references therein. In machine learning, applications of latent tree models include human interaction recognition, medical diagnosis and classification of documents [41, 27, 28].

As described in Section 2, in tree based graphical models, each node of the tree has an associated random variable. In many applications one can only observe the values at the terminal nodes of the tree, while the structure of the tree, as well as the values at the internal nodes, are unknown. Given a set of independent realizations of the observed variables, a common task is to infer the tree structure. In phylogeny, the terminal nodes correspond to present-day species, also known as taxa, and the hidden nodes correspond to their common ancestors. Each species is described by an observed string of characters such as a DNA or protein sequence. The task is to infer a tree that models the evolutionary lineages of the observed organisms [19, 13, 57].

Many algorithms have been developed to recover the latent tree structure from observed data. These include distance based methods such as the classic neighbor joining (NJ) [51] and UPGMA [55], maximum parsimony [7, 21], maximum likelihood [18, 25, 56, 49] , quartets and meta trees [3, 59, 45, 54, 50, 33], and Bayesian methods [46]. Other approaches for tree recovery are based on a measure of statistical dependency between pairs of terminal nodes, see [45, 27]. As reviewed in [66, 34], each of these strategies has different strengths and weaknesses.

We elaborate here on two approaches that are of particular relevance to our work. The first is the neighbor joining algorithm, one of the most important methods used in phylogeny. Due to its simplicity and scalability, neighbor joining is widely used in practice, and often serves as a baseline when testing new methods for reconstruction of evolutionary trees [38, 26, 62, 22]. For completeness, this approach is briefly outlined in Section 3.2. Several works investigated the theoretical properties of the neighbor joining algorithm [5, 23, 6, 22, 44, 39]. Atteson [5] studied its consistency and derived a sufficient condition for correct tree recovery. A different guarantee for exact recovery was derived in [39], by exploiting a link between NJ and quartet-based methods. As discussed in [36, 61, 58], to recover certain tree topologies or trees with a large number of terminal nodes, NJ may require a very large number of samples.

A second relevant line of work includes methods based on invariant features [8, 2]. One example is the *Tree SVD* algorithm, derived by Eriksson in [16]. In this algorithm, the tree is constructed using the spectral properties of a matrix called the *flattening matrix*. Every element of this matrix contains the probability of observing one possible assignment of characters in the terminal nodes. Since the number of possible assignments increases exponentially with the number of terminal nodes, applying this method to large trees is intractable. The Tree SVD algorithm was modified in [20] by averaging the row-normalized and the column normalized flattening matrices. For trees with four terminal nodes, it was shown to have similar performance to the maximum likelihood approach. In [1] the flattening matrices were used to detect changes in the tree topology within a DNA sequence. In Section 3 we elaborate on the Tree SVD algorithm and its relation to our approach.

### Our contribution

In this work we derive spectral neighbor joining (SNJ), a novel method to reconstruct tree graphical models. Our approach, described in Section 3 is based on the spectral structure of a similarity matrix between all pairs of observed nodes. The key property we use is the conditional independence of a node from the rest of the tree given the values of its immediate neighboring nodes. As we prove in Lemma 3.1, this implies that certain matrices have a rank one structure, as in our previous works on latent variable models [30, 29, 31, 43]. On the theoretical front, in Section 4.1 we prove the consistency of SNJ given an exact similarity matrix. Furthermore, in Theorem 4.9 we derive a sufficient condition on the difference between the exact and estimated similarity matrices that guarantees perfect recovery of the tree. Next, Lemma 4.11 provides a concentration of measure result on the estimated similarity matrix in the case of the Jukes-Cantor model, a popular model of sequence evolution [35]. Subsequently, in Theorem 4.10 we combine these results and derive an explicit expression of the number of samples that suffice for SNJ to correctly recover the underlying tree under this model, with high probability. In Section 5 we show that our spectral criterion for joining subsets of nodes is closely linked to quartet based approaches for reconstructing trees. Loosely speaking, at each step SNJ merges the two subsets for which the sum of all quartet tests is most consistent with the tree topology. We compare the finite sample guarantee in Theorem 4.10 to guarantees obtained in quartet based methods [15, 3], and discuss the tradeoff between statistical efficiency and computational complexity when recovering trees.

In Section 6 we discuss the analogy between Theorem 4.9 and a classic result obtained by Atteson [5, Theorem 4] for correct tree reconstruction by NJ. We compare the two sufficient conditions under the assumption of equal distances between all adjacent nodes. We show that for trees with a large diameter, our sufficient condition is considerably less strict than the analogous one for classical NJ. Consequently, we anticipate that SNJ will recover the correct tree structure with fewer samples. In Section 7 we illustrate, via extensive simulations, the improved tree reconstruction accuracy of SNJ over NJ [51], Recursive Grouping [11] Tree SVD [16] and Binary forest [27], under a variety of simulated settings.

In summary, the proposed SNJ method shares several desirable properties with NJ, including consistency, scalability to large trees, and simplicity of implementation. Furthermore, as we show both theoretically and via simulations, SNJ outperforms NJ and other methods under various scenarios of relevance to biological applications.

## 2 Problem setup

Let $\mathcal{T}$ be an unrooted bifurcating tree with $m$ terminal nodes. In such a tree, the leaves or terminal nodes each have a single neighbor, while internal nodes have three neighbors. We assume that each node of the tree has an associated discrete random variable attaining values in the set $\{1,\ldots,d\}$. We denote by $\boldsymbol{x} = (x_1,\ldots, x_m)$ the vector of random variables at the $m$ observed terminal nodes of the tree, and by $h_A,h_B,\ldots$ the random variables at the internal nodes. We assume that all of these random variables form a Markov random field on $\mathcal{T}$. This means that the random variable at each node is statistically independent of the rest of the tree

given the value of its neighbors. An edge $e(h_A, h_B)$ connecting a pair of adjacent nodes $(h_A, h_B)$ is equipped with two transition matrices of size $d \times d$,

$$P_{h_A \mid h_B}(a,b) = \Pr[h_A = a \mid h_B = b] \qquad P_{h_B \mid h_A}(b,a) = \Pr[h_B = b \mid h_A = a]\,. \tag{1}$$

The observed data is a matrix $X = [\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(n)}] \in \{1,\ldots,d\}^{m\times n}$, where $\boldsymbol{x}^{(j)}$ are i.i.d. realizations of the random variables at the $m$ terminal nodes of the tree. Each row in the matrix is a sequence of length $n$ that corresponds to one terminal node, see Figure 1. For example, in phylogenetics, each row corresponds to a different species, while each column corresponds to a different site in a DNA or protein sequence. The latent nodes in the tree correspond to the common ancestors of different subsets of the observed organisms, see [14] and references therein.

Given the matrix $X$, the task at hand is to recover the structure of the tree $\mathcal{T}$. For the tree to be identifiable, we assume that for every pair of adjacent nodes $h_A, h_B$, the corresponding $d \times d$ stochastic matrices $P_{h_A|h_B}$ and $P_{h_B|h_A}$ defined in (1) are full rank, with determinants that satisfy

$$0 < \delta < \left| P_{h_A \mid h_B} \right| < \xi < 1 \qquad 0 < \delta < \left| P_{h_B \mid h_A} \right| < \xi < 1\,. \tag{2}$$

Eq. (2) implies that all edge transition matrices are invertible and are not permutation matrices. These are critical conditions for identifiability of the tree topology, see Proposition 3.1 in [9] and [40]. We remark that though our approach can be applied to recover the topology of *rooted trees* as well as unrooted ones, determining the location of the root requires additional assumptions, see [53].

## 3 The spectral neighbor joining algorithm

To introduce our novel spectral approach, in Section 3.1 we first review known measures for similarity and distance between nodes in a latent tree model. For completeness, Section 3.2 briefly describes the standard neighbor joining algorithm. In Section 3.3 we derive a new spectral criterion for neighbor joining and present our algorithm in detail.

### 3.1 The symmetric affinity and distance matrices

We denote by $P_{x_i|x_j}$ the stochastic matrix containing the distribution of $x_i$ given $x_j$. Under the tree model, $P_{x_i|x_j}$ is the product of the stochastic matrices of the edges along the directed path from $x_i$ to $x_j$. For example, in the tree shown in Figure 1, the hidden nodes on the path from $x_1$ to $x_3$ are $h_C$ and $h_A$. Thus,

$$P_{x_1 \mid x_3} = P_{x_1 \mid h_C} P_{h_C \mid h_A} P_{h_A \mid x_3}\,.$$

Several methods to reconstruct trees are based on a measure of similarity or distance between the observed nodes. Accordingly, we denote by $s(x_i, x_j)$ the *symmetric affinity* between a pair of terminal or hidden nodes,

$$r(x_i, x_j) = \sqrt{|P_{x_i|x_j}| \cdot |P_{x_j|x_i}|}, \qquad r(h_A, h_B) = \sqrt{|P_{h_A|h_B}| \cdot |P_{h_B|h_A}|}. \quad (3)$$

Here $|P_{x_i|x_j}|$ denotes the determinant of the matrix $P_{x_i|x_j}$. Let $R \in \mathbb{R}^{m \times m}$ denote the symmetric affinity matrix between all pairs of terminal nodes,

$$R(i, j) = r(x_i, x_j) = \sqrt{|P_{x_i|x_j}| \cdot |P_{x_j|x_i}|}. \quad (4)$$

Note that the symmetric affinity always falls within the range [0, 1]. An important property of $R(i, j)$ is that it is *multiplicative* along the path between $x_j$ and $x_i$. For example, in Figure 1, the affinity between $x_1$ and $x_3$ is equal to

$$\begin{aligned} R(1,3) &= \sqrt{|P_{x_1|h_C}| \cdot |P_{h_C|h_A}| \cdot |P_{h_A|x_3}|}\sqrt{|P_{x_3|h_A}| \cdot |P_{h_A|h_C}| \cdot |P_{h_C|x_1}|} \\ &= \sqrt{|P_{x_1|h_C}| \cdot |P_{h_C|x_1}|}\sqrt{|P_{h_A|h_C}| \cdot |P_{h_C|h_A}|}\sqrt{|P_{x_3|h_A}| \cdot |P_{h_A|x_3}|} \\ &= r(x_1, h_C) r(h_C, h_A) r(h_A, x_3). \end{aligned}$$

This fact follows directly from the multiplicative property of determinants. The following transformation from the similarity measure (4) to a distance function between terminal nodes was proposed in [10] and [37],

$$D(i, j) = -\log r(x_i, x_j). \quad (5)$$

Eq. (5), known as the paralinear distance, was used in several distance based methods for reconstructing trees, see [42, 52] and references therein. Note that the log transformation in (5) yields a distance measure between two observed nodes $x_i$, $x_j$ that is additive along the path connecting them. The additive property is a necessary condition for the consistency of any distance based method [9, 10].

### 3.2 Background: the neighbor joining algorithm

To motivate our approach, we first briefly describe the classical neighbor joining algorithm [51]. The input to NJ is a matrix $\widehat{D} \in \mathbb{R}^{m \times m}$ of estimated distances between observed nodes. NJ iteratively reconstructs the tree via the following procedure:

1. Compute the $Q$ criterion between all pairs,

$$Q(i, j) = (m - 2)\widehat{D}(i, j) - \sum_{k \neq \{i,j\}} \widehat{D}(k, i) - \sum_{k \neq \{i,j\}} \widehat{D}(k, j). \quad (6)$$

2. Reconstruct the tree by repeating the following two steps, until there are three nodes left:

   I. I identify the pair $(\hat{i}, \hat{j})$ that minimizes the $Q$ criterion,

$$(\hat{i}, \hat{j}) = \underset{i,j}{\operatorname{argmin}} Q(i, j).$$

**II.** merge the pair $(\hat{i}, \hat{j})$ into a single node $l$, and update the $Q$ criterion by

$$Q(k,l) = \frac{1}{2}(Q(k,\hat{i}) + Q(k,\hat{j})) \qquad \forall k\,. \tag{7}$$

The neighbor joining method is *consistent*. If the estimated matrix $\widehat{D}$ is sufficiently close to the true distance matrix $D$, the method is guaranteed to reconstruct the correct tree. As proved by [5], a sufficient condition for recovering the tree is

$$\max_{i,j} \mid D(i,j) - \widehat{D}(i,j) \mid \; \leq \frac{d_{min}}{2}, \tag{8}$$

where $d_{min}$ is the distance between the closest pair of adjacent nodes in the tree. The distance between adjacent (not necessarily terminal) nodes is defined identically to the distance between terminal nodes given in Eq. (5).

### 3.3 A spectral criterion for neighbor joining

To describe our approach, we use the terminology of unrooted trees provided by [65]. We define a *clan* of nodes in $\mathcal{T}$ as a subset of nodes that can be separated from the rest of the tree by removing a single edge. For example, in Figure 1 the subset $\{x_1, x_2, h_C\}$ forms a clan. In the paper, we will sometimes refer to the set of *terminal nodes* of a clan such as $\{x_1, x_2\}$, as a clan.

Let $A$ be a subset of $[m] = \{1,\ldots,m\}$. We denote by $x_A$ the set of corresponding terminal nodes $\{x_i\}_{i\in A}$. Let $A$ and $B$ be two disjoint subsets of $[m]$ such that $x_A$ and $x_B$ each form two different clans. We say that $x_A$ and $x_B$ are *adjacent clans* if their union forms another, larger clan. Otherwise, we say that the clans are non-adjacent.

Equipped with these definitions, we describe the spectral neighbor joining approach. In contrast to previous methods that use the symmetric distance (5) or other distance measures, our approach uses the symmetric affinity matrix between terminal nodes $R$ introduced in Section 2. Let $A$ be a subset of $\{1,\ldots,m\}$ with size $|A| \geq 2$. We denote by $R^A$ the submatrix of $R$ of size $|A| \times (m - |A|)$ that contains all the affinities $R(i, j)$ with $i \in A$ and $j \in A^c$, where $A^c$ is the complement of $A$. Lemma 3.1 provides the theoretical foundation for our approach.

**Lemma 3.1.** *The matrix $R^A$ is rank-one if and only if the subset $x_A$ is equal to the terminal nodes of a clan in $\mathcal{T}$.*

By Lemma 3.1, two nodes $x_i$ and $x_j$ are adjacent if and only if their affinities $r(i, k)$ to all other observed nodes are identical up to a multiplicative factor. This will be a crucial property in developing our spectral neighbor joining algorithm. The proof of Lemma 3.1 relies on the following auxiliary lemma which is proven in the supplementary material.

**Lemma 3.2.** *The following two statements are equivalent:*

**1.** The subset $x_A$ is equal to the terminal nodes of a clan in $\mathcal{T}$.

**2.** *All quartets of terminal nodes* $\{x_i, x_k, x_j, x_l\}$ *where* $i, k \in A$ *and* $j, l \in A^c$ *have a topology as in* Figure 2*, in which* $(x_i, x_k)$ *and* $(x_j, x_l)$ *are adjacent.*

*Proof of* Lemma 3.1. Suppose that $x_A$ consists of the terminal nodes of a clan in $\mathcal{T}$, and $x_B$ be the complementary subset. Let $e(h_A, h_B)$ be the edge that separates the clan from the rest of the tree, so that all paths between nodes $x_A$ and $x_B$ pass through $e(h_A, h_B)$. By the multiplicative property of $R(i, j)$, for all $i \in A$, $j \in B$

$$R(i, j) = r(x_i, h_A) r(h_A, h_B) r(h_B, x_j).$$

Let $\boldsymbol{u}_A$ denote a vector of size $|A|$, whose elements are the affinities between $h_A$ and $x_i$ for $i \in A$. Similarly, let $\boldsymbol{u}_B$ be a vector of size $m - |A|$ whose elements are the affinities between $h_B$ and $x_j$ for $j \notin A$. Then $R^A$ is equal to

$$R^A = r(h_A, h_B) \boldsymbol{u}_A \boldsymbol{u}_B^T. \tag{9}$$

Eq. (9) implies that $R^A$ is rank 1.

Now suppose that $x_A$ does not equal the terminal nodes of a clan. By part 2 of Lemma 3.2, this implies that there is at least one quartet of nodes $x_i, x_k, x_j, x_l$ with $(i, j) \in A$ and $(k, l) \in B$ with a structure as in Figure 2, where $x_i$ is closer to $x_k$ than to $x_j$. Let $R_{ij}^{kl}$ be the $2 \times 2$ submatrix of $R^A$ that contains the pairwise affinities between $x_i$, $x_j$ and $x_k$, $x_l$. Then its determinant is

$$\begin{aligned} |R_{ij}^{kl}| &= R(i,k)R(j,l) - R(i,l)R(j,k) \\ &= r(x_i, h_A) r(h_A, x_k) r(x_j, h_B) r(h_B, x_l) \\ &\quad - r(x_i, h_A) r(h_A, h_B) r(h_B, x_l) r(x_j, h_B) r(h_A, h_B) r(h_A, x_k) \\ &= r(x_i, h_A) r(h_A, x_k) r(x_j, h_B) r(h_B, x_l) (1 - r(h_A, h_B))^2. \end{aligned} \tag{10}$$

Combining Eq. (2) with $r(x_i, x_j)$ in Eq. (3) implies that all terms in Eq. (10) are bounded away from zero and from one. Hence, $|R_{ij}^{kl}| \neq 0$ and so $R_{ij}^{kl}$ is full rank. Since $R_{ij}^{kl}$ is a submatrix of $R^A$, it follows that $R^A$ is at least rank two.

Lemma 3.1 implies that given perfect knowledge of $R$, we can determine whether a given set of terminal nodes $x_A$ is equal to the terminal nodes of a clan by computing the rank of $R^A$. In practice, we typically only have a noisy estimate of the entries of $R$. Then, all submatrices of $R$ are full rank, though for true clans, the corresponding submatrices are approximately rank 1. Accordingly, instead of the rank, our criterion for whether to join two subsets $x_{A_i}$ and $x_{A_j}$ is based on the second largest singular value of $R^{A_i \cup A_j}$, a matrix of dimension $(|A_i| + |A_j|) \times (m - \in A_i| - |A_j|)$ that contains the affinities between terminal nodes in $x_{A_i \cup A_j}$ and the remaining terminal nodes. We denote its second largest singular value by $\sigma_2(R^{A_i \cup A_j})$. Specifically, SNJ recovers the tree by performing the following operations:

- Set $A_j = \{i\}$ for all $i$. Compute a matrix $\Lambda \in \mathbb{R}^{m \times m}$ where

$$\Lambda(i,j) = \sigma_2(R^{A_i \cup A_j}).$$

- Repeat the following two steps until only three subsets remain.

  **I.** Identify the pair $(\hat{i}, \hat{j})$ that minimizes $\Lambda(i, j)$,

$$(\hat{i}, \hat{j}) = \underset{ij}{\operatorname{argmin}} \Lambda(i,j). \tag{11}$$

  **II.** Merge $A_{\hat{i}}$, $A_{\hat{j}}$ into a subset $A_l = A_{\hat{i}} \cup A_{\hat{j}}$. Update the $\Lambda$ criterion via

$$\Lambda(k,l) = \sigma_2(R^{A_k \cup A_l}) \quad \forall k. \tag{12}$$

As one can see, SNJ has a similar algorithmic structure to NJ, with the key difference being the use of the second singular value instead of the Q-criteria. Hence, it is interesting to compare the power of these two test statistics to distinguish between adjacent and non-adjacent terminal nodes. To this end, we generated a random Jukes-Cantor tree model with $m = 512$ terminal nodes, associated with random variables with support of $d = 4$ characters. The topology of the tree was generated by the following process: Given $m$ nodes, we merged a pair of random terminal nodes and replaced them with a single non-terminal node. Next, we merged another pair of random nodes, either terminal or non terminal, and again replaced them with a single non-terminal node. We continued this process until three nodes remained, and we connected them all to a non terminal node. We set the mutation rates between adjacent nodes to be 10% and the number of realizations to be $n = 500$. The left panel of Figure 3 shows the empirical distribution of $\log \sigma_2(R^{A_i \cup A_j})$ at the first SNJ iteration, where $A_i = \{i\}$ for all $i$. The right panel shows the empirical distribution of the NJ Q criterion in Eq. (6). The red and blue lines correspond to pairs of adjacent and non-adjacent terminal nodes, respectively. Comparing the two panels, we clearly see that adjacent pairs can be perfectly separated from non-adjacent pairs by their $\sigma_2$ values, whereas $Q$ values of adjacent and non-adjacent pairs have a significant overlap. As we will illustrate in Section 7, this better separation of $\sigma_2$ vs. the $Q$-criterion allows SNJ to accurately reconstruct trees from fewer number of samples, where NJ fails.

We note that $\sigma_2(R^{A_i \cup A_j})$ is not the only possible measure of how close $R^{A_i \cup A_j}$ is to a rank-1 matrix. An alternative measure is the Euclidean distance to the closest rank-1 matrix, computed by the sum of squares of all but the first singular value. The second singular value criterion is justified by Lemma 4.4 in the following section, where we prove that if $x_{A_i}$ and $x_{A_j}$ are clans, then $R^{A_i \cup A_j}$ is *at most* rank 2. Thus, all non-zero singular values besides $\sigma_1$ and $\sigma_2$ are the result of noise in the similarity estimates, and should not be taken into account.

### 3.4 Heterogeneity of mutation rates

The problem setup presented in Section 2 assumed a fixed rate of mutation across all sites in the sequence. In many applications, this assumption does not hold, and may lead to bias in

estimating the distance between terminal nodes [4]. In biological applications, rate heterogeneity is commonly modeled using a gamma distribution or the related gamma-invariable model [24, 32]. Many substitution models have variants that account for heterogeneity in the mutation rate along a sequence [42, 63]. For example, the classic Jukes-Cantor model has a variant that takes into account heterogeneity in mutation rate, termed Gamma Jukes-Cantor. Similar to the homogeneous rate case, the distances computed by these models are additive along the tree, a key property for reconstruction of trees with distance based methods [10]. Regardless of the assumed model, distance estimates can be transformed into similarity estimates by inverting Equation (5),

$$\boldsymbol{R}(i,j) = e^{-D(i,j)}, \tag{13}$$

If the distance measure $D$ is additive, then the similarity scores obtained from Equation (13) maintain the key property of being multiplicative along the tree, as described in Section 3.1. Thus, SNJ can be combined with any procedure for estimating distances. In particular, SNJ can consistently recover trees with heterogeneity in mutation rates. In Section 7 we show empirically that SNJ outperforms NJ for data generated according to the Gamma model of heterogeneity in mutation rates.

### 3.5 Related spectral methods

A different spectral-based approach to reconstruct trees is the Tree SVD algorithm [16], which similarly to SNJ merges subsets of terminal nodes based on a spectral criterion. The Tree SVD algorithm first estimates the probability of observing all $d^m$ possible patterns in the terminal nodes. For every partition $[m] = A \cup A^c$, these estimates are rearranged into a *flattening matrix* of size $d^{|A|} \times d^{m-|A|}$. Each row of the matrix contains the probabilities of all possible patterns of terminal nodes in $A^c$, with a fixed pattern for the terminal nodes in $A$. The key property of the flatenning matrix is that with the *exact* (rather than estimated) probabilities, its rank is equal to $d$ if and only if $A$ corresponds to a clan in the tree.

Though Tree SVD is consistent, it is impractical for large trees due to the size of the flattening matrix. In contrast, computing the SNJ similarity matrix can be done efficiently, as its dimension is equal to the number of terminal nodes.

A second drawback of Tree SVD, outlined in [2, 1] is that it compares flattening matrices of different sizes. In [2], this fact was shown to cause a bias towards balanced trees. Potentially, this drawback is also relevant to SNJ, as the $\sigma_2$ criterion is compared for matrices of different sizes. One way to measure if an algorithm suffers from such a bias is to count the number of cherries - clans with two terminal nodes - in the reconstructed tree and in the original one. We present such a comparison in Section E of the supplementary material. Figure 21 (right) shows the *bias*, the number of cherries in the trees estimated by SNJ and NJ minus the number of cherries in the ground truth, as a function of $m$. The trees were generated according to the birth death model. Figure 21 (left) shows the RF distance between the estimated trees and the ground truth. Though the recovered tree is not perfect, the results do not indicate any bias towards trees that are more balanced.

# 4 Analysis

In this section we present a theoretical analysis of the SNJ algorithm. First, in section 4.1 we prove consistency of SNJ in the population setting where the similarity matrix $R$ is perfectly known, and assuming Eq. (2) holds. Next, we derive a sufficient condition on the difference between the estimated and exact affinity matrices that guarantees correct tree reconstruction by SNJ. Finally, we derive an explicit expression for the number of samples sufficient to guarantee exact tree reconstruction by SNJ with high probability under the Jukes-Cantor model. Proofs of auxiliary lemmas stated in this section appear in the supplementary material.

## 4.1 Consistency of SNJ in the population setting

For SNJ to correctly recover the tree structure, at each iteration it must merge two adjacent clans of terminal nodes. The following theorem characterizes the second eigenvalue criterion in Eq. (12), depending on whether two subsets are adjacent or not.

**theorem 4.1.** *Let $C = A \cup B$, where $x_A$ and $x_B$ are disjoint subsets of terminal nodes such that each contains exactly the terminal nodes of a clan in $\mathcal{T}$. (i) If $x_A$,$x_B$ are adjacent clans then*

$$\sigma_2(R^C) = 0\,.$$

*(ii) If $x_A$, $x_B$ are non adjacent clans then*

$$\sigma_2(R^C) \geq \begin{cases} \frac{1}{2}(2\delta^2)^{\log_2(m/2)}\delta(1-\xi^2) & \delta^2 \leq 0.5, \\ \delta^3(1-\xi^2) & \delta^2 > 0.5\,. \end{cases} \tag{14}$$

For future use we define

$$f(m,\delta,\xi) = \frac{1}{2}(2\delta^2)^{\log_2(m/2)}\delta(1-\xi^2)\,. \tag{15}$$

Theorem 4.1 has several important implications, which we now discuss. First, as stated in the following corollary, it implies that SNJ is consistent.

**Corollary 4.2.** *Let $\mathcal{T}$ be a tree which satisfies* Eq. (2)*. Then, SNJ with the exact affinity matrix $R$ is consistent and perfectly recovers $\mathcal{T}$.*

To see why the corollary is true, recall that at each iteration, SNJ merges two subsets with the smallest value of $\sigma_2(R^{A_i \cup A_j})$. By Theorem 4.1, adjacent clans have $\sigma_2 = 0$, whereas if clans are non-adjacent, the second singular value corresponding to their union is strictly positive. Hence, given the exact affinity matrix, SNJ merges only adjacent clans until the whole tree has been perfectly reconstructed.

A second important implication of Theorem 4.1 is that the bound in Eq. (14) yields insights into the ability of SNJ to correctly recover trees in the noisy setting, depending on the number of observed nodes $m$ and parameters $\delta, \xi$. Figure 4 shows the lower bound on $\sigma_2(R^C)$ for non-adjacent clans in Eq. (14), as a function of $\delta$ for $m = 4, 8, 64, 128$, with $\xi = 0.95$. Note that for $m = 4$ the formulas for $\delta^2 \leq 0.5$ and $\delta^2 > 0.5$ coincide. For $\delta^2 \leq 0.5$, which in a phylogenetic setting implies a high mutation rate, the bound decreases with a larger number of terminal leaves $m$. This implies that SNJ requires a higher number of samples to learn larger trees.

We remark that in general, the lower bounds in Eq 14. are tight, up to a multiplicative factor of $1/\delta(1 + \xi)$, as described in the following lemma.

**Lemma 4.3.** For $\delta^2 \leq 0.5$, *there exists a tree and two non-adjacent clans* $x_A, x_B$ *such that* $\sigma_2(R^{A\cup B}) = f(m, \delta, \xi)/\delta(1 + \xi)$. *For* $\delta^2 > 0.5$, *there exists a tree with* $m = 4$ *nodes for which* $\sigma_2(R^{A\cup B}) = \delta^3(1 - \xi^2)/\delta(1 + \xi) = \delta^2(1 - \xi)$.

The first part of Theorem 4.1 follows directly from Lemma 3.1. To prove the second part, we first introduce some notations and auxiliary lemmas. Let $x_A, x_B$ be two non adjacent clans in $\mathcal{T}$ and let $h_A, h_B$ be their corresponding root nodes. Since $x_A, x_B$ are not adjacent, there are at least two additional hidden nodes on the path between $h_A$ and $h_B$. Let $h_1,\ldots,h_l$ denote the $l$ hidden nodes on this path, see Fig. 5 for an example with $l = 3$ intermediate nodes. We split the remaining $m - |A| - |B|$ terminal nodes to $l$ subsets as follows: Every terminal node in $(A \cup B)^c$ is assigned to the closest hidden node on the path between $h_A$ and $h_B$ (see Fig. 5). The matrix $R^C$ can be rearranged in the following block structure,

$$R^C = \begin{bmatrix} R_1^A & R_2^A & \ldots & R_l^A \\ R_1^B & R_2^B & \ldots & R_l^B \end{bmatrix} = [R_1 \;\; R_2 \;\; \ldots \;\; R_l], \tag{16}$$

where $R_i^A$ is a matrix of $|A|$ rows with the pairwise affinities between the nodes in $x_A$ and the terminal nodes assigned to $h_i$. The matrix $R_i^B$ with $|B|$ rows is defined similarly. The matrix $R_i$ is the concatenation of $R_i^A$ and $R_i^B$. The following lemma shows that this block structure implies that the matrix $R^C$ has rank at most 2.

**Lemma 4.4.** *Let* $R^C$ *be the matrix of* Eq. (16). *Then* $1 \leq rank(R^C) \leq 2$.

*Proof of* Lemma 4.4. Recall that $R^A$ denotes the affinity matrix between $x_A$ and $x_{A^c}$. Under the assumption that $x_A$ contains the terminal nodes of a clan, by Lemma 3.1, $R^A$ has rank one. The upper part of $R^C$ which includes $\{R_i^A\}_{i=1}^l$ is a submatrix of $R^A$ and hence has rank one as well. Similarly, the lower part of $R^C$, which includes $\{R_i^B\}_{i=1}^l$ is a submatrix of $R^B$ and also has rank one. The concatenation of two rank one matrices is at most rank two. □

Next, we present two auxiliary lemmas. The first concerns rank-2 matrices.

**Lemma 4.5.** *Let* $M$ *be a rectangular matrix with* $1 \leq rank(M) \leq 2$, *and let* $\sigma_2(M)$ *be its second singular value. Then*

$$\sigma_2(M)^2 \geq \frac{1}{2}\frac{\|M\|_F^4 - \|M^T M\|_F^2}{\|M\|_F^2}. \tag{17}$$

The next auxiliary lemma expresses $\|R^C\|_F^4 - \|(R^C)^T R^C\|_F^2$ in terms of the norms of the individual blocks $R_i^A$, $R_i^B$ of the matrix $R^C$.

**Lemma 4.6.** *Let $R^C$ be the matrix of* Eg. (16) *with blocks $R_i^A$ and $R_i^B$. Then*

$$\|R^C\|_F^4 - \|(R^C)^T R^C\|_F^2 = \sum_{j=1}^{l}\sum_{k=1}^{l}(\|R_j^A\|_F\|R_k^B\|_F - \|R_j^B\|_F\|R_k^A\|_F)^2. \tag{18}$$

*Proof of* Theorem 4.1*, part (ii).* Let $\boldsymbol{u}_A$ be the vector of affinities between $h_A$ and nodes in $x_A$, and $\boldsymbol{u}_b$ be the vector of affinities between $h_B$ and $x_B$,

$$\boldsymbol{u}_A = \{r(x_i, h_A)\}_{i \in A}, \qquad \boldsymbol{u}_B = \{r(x_j, h_B)\}_{j \in B}. \tag{19}$$

Similarly, let $\boldsymbol{v}_j$ be a vector of affinities between $h_j$ and the terminal nodes associated with it. By the multiplicative property of the affinity $r(x_i, x_j)$, the blocks $R_i^A$ and $R_j^B$ that are part of the matrix $R^C$ in (16) have the following form,

$$R_i^A = \boldsymbol{u}_A r(h_A, h_i)\boldsymbol{v}_i^T \qquad R_i^B = \boldsymbol{u}_B r(h_B, h_i)\boldsymbol{v}_i^T, \tag{20}$$

where $r(h_A, h_i)$ is the affinity between the hidden nodes $h_A$ and $h_i$. The proof of the theorem is composed of the following three steps:

1. Lower bound $\sigma_2(R^{A\cup B})$ in terms of $\|R_i^A\|_F$ and $\|R_i^B\|_F$.
2. Expand $\|R_i^A\|_F$ and $\|R_i^B\|_F$ in terms of $\|\boldsymbol{u}_A\|$, $\|\boldsymbol{u}_B\|$ and $\|\boldsymbol{v}_i\|$.
3. Lower bound $\|\boldsymbol{u}_A\|$, $\|\boldsymbol{u}_b\|$ and $\|\boldsymbol{v}_i\|$ as a function of $m$, $\xi$, $\delta$.

**Step 1:** Combining Lemmas 4.4, 4.5 and 4.6 gives that

$$\sigma_2(R^C) \geq \frac{\sum_{j=1}^{l}\sum_{k=1}^{l}(\|R_j^A\|_F\|R_k^B\|_F - \|R_j^B\|_F\|R_k^A\|_F)^2}{\|R^C\|_F^2}.$$

**Step 2:** We express $\|R_i^A\|_F$ and $\|R_i^B\|_F$ in terms of $\|\boldsymbol{u}_A\|$, $\|\boldsymbol{u}_B\|$ and $\|\boldsymbol{v}_i\|$. This step follows directly from Eq. (20),

$$\|R_i^A\|_F = r(h_A, h_i)\|\boldsymbol{u}_A\|\|\boldsymbol{v}_i\| \qquad \|R_i^B\|_F = r(h_B, h_i)\|\boldsymbol{u}_B\|\|\boldsymbol{v}_i\|. \tag{21}$$

**Step 3:** The following auxiliary lemma provides a bound on $\|\boldsymbol{u}_A\|$ in terms of $|A|$ and the affinity lower bound $\delta$.

**Lemma 4.7.** *Let $x_A$ be equal to the terminal nodes of a clan in $\mathcal{T}$ and let $\mathbf{u}_A$ be the vector of Eq. (19). Then,*

$$\|\boldsymbol{u}_A\|^2 \geq \begin{cases} (2\delta^2)^{\log |A|} & \delta^2 \leq 0.5, \\ 2\delta^2 & \delta^2 > 0.5\,. \end{cases} \tag{22}$$

Similar bounds hold for $\|\boldsymbol{u}_B\|^2$ and $\|\boldsymbol{v}_k\|^2$. Having described steps 1-3, we are now ready to conclude the proof of Theorem 4.1. To this end, we use the following auxiliary lemma, which follows from steps 1 and 2.

**Lemma 4.8.** *Let $C = A \cup B$, where $x_A$ and $x_B$ are non-adjacent clans in $\mathcal{T}$. Then*

$$\sigma_2(R_C)^2 \geq \frac{1}{4}\min\{\|\boldsymbol{u}_A\|, \|\boldsymbol{u}_B\|\}^2 \times \min_j \min_{k \neq j} \|\boldsymbol{v}_k\|^2 (1 - r(h_j, h_k)^2)^2 \min\{\max_k r(h_A, h_k), \max_k r(h_B, h_k)\}^2\,. \tag{23}$$

Next, we insert the lower bounds in Eqs. (2) and (22) into Eq. (23). For $\delta^2 > 0.5$,

$$\sigma_2^2(R^C) \geq \tfrac{1}{4}(2\delta^2)(2\delta^2)\delta^2(1-\xi^2)^2 = \delta^6(1-\xi^2)^2\,.$$

For $\delta^2 \leq 0.5$ we obtain,

$$\sigma^2(R^C) \geq \frac{1}{4}(2\delta^2)^{\log|A| + \log|B|}\delta^2(1-\xi^2)^2 = \frac{1}{4}(2\delta^2)^{\log|A||B|}\delta^2(1-\xi^2)^2\,. \tag{24}$$

Since $x_A, x_B$ are non adjacent clans, there are at least two additional observed nodes that are not in $x_{A \cup B}$. It follows that $|A| + |B| \leq m - 2$ and hence $|A||B| < m^2/4$. Replacing $|A||B|$ with $m^2/4$ in (24) gives

$$\sigma^2(R^C) \geq \tfrac{1}{4}(2\delta^2)^{\log(m^2/4)}\delta^2(1-\xi^2)^2 = \tfrac{1}{4}(2\delta^2)^{2\log(m/2)}\delta^2(1-\xi^2)^2,$$

which completes the proof of Theorem 4.1. □

## 4.2 Required number of samples for exact reconstruction

We now focus on the finite sample setting, where we can only compute an approximate affinity matrix $\widehat{R}$. For NJ, the finite sample setting was addressed in [5], where NJ was proved to reconstruct the correct tree if the estimated distance matrix $\widehat{D}$ satisfies Eq. (8). In the following theorem we derive an analogous result for SNJ.

**theorem 4.9.** *Assume that* Eq. (2) *holds. Then a sufficient condition for spectral neighbor joining to recover the correct tree from* $\widehat{R}$ *is that*

$$\|R - \widehat{R}\| \leq \begin{cases} \frac{f(m,\delta,\xi)}{2} & \delta^2 \leq 0.5 \\ \frac{1}{2}\delta^3(1-\xi^2) & \delta^2 > 0.5\,. \end{cases} \tag{25}$$

Next, we derive a concentration bound on the similarity matrix. This yields an upper bound on the number of samples required to obtain an estimated similarity matrix that satisfies Eq. (25). For simplicity, the finite sample bound is derived for the Jukes-Cantor (JC) model, a popular model in phylogenetic inference, see [19]. Under the JC model, the probability over the $d$ states in all the nodes is uniform, and that the stochastic matrix between adjacent nodes $h_i$, $h_j$ is equal to

$$\Pr(h_i \mid h_j)_{kl} = \begin{cases} 1-\theta(i,j) & k = l \\ \theta(i,j)\,/\,(d-1) & \text{otherwise,} \end{cases}$$

where $\theta(i, j)$ is the mutation rate between nodes $h_i$ and $h_j$. Under these assumptions, the affinity between terminal nodes in Eq. (4) simplifies to

$$R(i,j) = \left(1 - \frac{d}{d-1}\theta(i,j)\right)^{d-1}. \tag{26}$$

By assumption (2) $R(i, j)$ is strictly positive, and hence $\theta(i, j) < (d–1)/d$. Given $n$ i.i.d. realizations $\{x^l\}_{l=1}^n$ from the Jukes-Cantor model, we estimate $\hat{\theta}$ and $\widehat{R}$ via

$$\hat{\theta}(i,j) = \min\left\{\frac{1}{n}\sum_{l=1}^{n} \mathbf{1}_{x_i^l \neq x_j^l}, \frac{d-1}{d}\right\} \quad \widehat{R}(i,j) = \left(1 - \frac{d}{d-1}\hat{\theta}(i,j)\right)^{d-1}. \tag{27}$$

Applying SNJ to $\widehat{R}$ estimated via Eq. (27), we have the following guarantee.

**theorem 4.10.** *Assume the data was generated according to the Jukes-Cantor model. If the number of samples n satisfies*

$$n \geq \begin{cases} \frac{2d^2m^2}{f(m,\delta,\xi)^2}\log\left(\frac{2m^2}{\epsilon}\right) & \delta^2 \leq 0.5 \\ \frac{2d^2m^2}{\delta^6(1-\xi^2)^2}\log\left(\frac{2m^2}{\epsilon}\right) & \delta^2 > 0.5, \end{cases}$$

*where* $f(m, \delta, \xi)$ *was defined in* (15), *then SNJ will recover the correct tree topology with probability at least* $1 - \epsilon$.

To understand the dependency of $n$ on the number of terminal nodes $m$, we replace $f(m, \delta, \xi)$ with its definition (15), and treat $\delta$, $\xi$ and $d$ as constants. For $\delta^2 \leq 0.5$,

$$n = \Omega\left(m^{4\log_2(1/\delta)}\log(m/\epsilon)\right).$$

If $\delta^2 > 0.5$,

$$n = \Omega\left(m^2\log(m/\epsilon)\right).$$

Thus, up to a logarithmic factor, the number of samples required for an exact recovery of the tree is quadratic in $m$ for $\delta^2 > 0.5$, but can reach $\Omega(m^\beta)$ with exponent $\beta \to \infty$ for very low values of $\delta$. Next, considering the dependence on $\xi$, Theorem 4.10 implies that $n$ scales as $U(1/(1-\xi^2))$. A high value of $\xi$ corresponds to a tree that has at least one very short edge, and is thus hard to reconstruct. A similar result appears in the guarantee derived by Atteson in Eq. (8), which depends on the minimal distance between adjacent nodes. In Section 7 we simulate trees with equal simiarity between all adjacent nodes such that $\delta = \xi$. The dependency of SNJ's performance on $\xi$ for these simulations is in accordance with this theoretical analysis.

The proof of Theorem 4.10 is based on the following auxiliary lemma, which states a concentration result on the estimated matrix $\hat{R}$.

**Lemma 4.11.** *Let* $\hat{R} \in \mathbb{R}^{m\times m}$ *be the matrix given by* Eq. (27). *Then*

$$\Pr\left(\|\hat{R} - R\| \leq t\right) \geq 1 - 2m^2\exp\left(-\frac{2nt^2}{d^2m^2}\right).$$

*Proof of* Theorem 4.10. We prove the finite sample theorem by combining Theorem 4.9 with the concentration bound on $\hat{R}$ in Lemma 4.11. For $\delta^2 \leq 0.5$, we replace $t$ with $f(m, \delta, \xi)/2$ in Lemma 4.11,

$$\Pr\left(\|\hat{R} - R\| \leq \frac{f(m,\delta,\xi)}{2}\right) \geq 1 - 2m^2\exp\left(-\frac{2n(f(m,\delta,\xi)/2)^2}{d^2m^2}\right).$$

Let $1-\xi$ be a lower bound on this probability, such that

$$1 - 2m^2\exp\left(-\frac{2n(f(m,\delta,\xi)/2)^2}{d^2m^2}\right) \geq 1 - \epsilon.$$

Rearranging the above equation yields the following lower bound on $n$ in terms of $m$, $d$ and $\epsilon$,

$$n \geq \frac{2d^2m^2}{f(m,\delta,\xi)^2}\log\left(\frac{2m^2}{\epsilon}\right),$$

which concludes the proof for $\delta^2 \leq 0.5$. For $\delta^2 > 0.5$, we replace $f(m, \delta, \xi)$ with $\delta^3(1 - \xi^2)$.

### 4.3 Finite sample guarantees for alternative models of mutation

The proof of Theorem 4.10 consists of two steps, corresponding to Theorem 4.9 and Lemma 4.11: (i) Given a sufficiently accurate similarity matrix, SNJ gives the correct tree, and (ii) An expression for the number of samples required for such an accurate estimate.

The first step does not depend on any specific substitution model or any distribution of states at some node of the tree. The derivation of the second step, however, holds only for the Jukes-Cantor model, where a transition matrix $P_{x_i|x_j}$ is completely determined by a single mutation rate $\theta(i, j)$. For this model, the affinity between terminal nodes simplifies to a polynomial in $\theta(i, j)$, see Eq. (26).

With no assumptions on the structure of the transition matrices $P_{x_i|x_j}$, such a simplification is not possible. Here, we derive a bound that generalizes Lemma 4.11, for unstructured transition matrices $P_{x_i|x_j}$. We make one simplifying assumption, that the transition matrices are symmetric with $P_{x_i|x_j} = P_{x_j|x_i}$. Thus, the similarity between terminal nodes $x_i$, $x_j$ in Eq. (3) simplifies to $R(i, j) = \det(P_{x_i|x_j})$ where det() denotes the matrix determinant.

Let $n_k(x_i)$ be the number of samples equal to state $k$ in terminal node $x_i$ and let $\gamma$ be equal to

$$\gamma = \frac{1}{n}\min_{i \in [m],\, k \in [d]} n_k(x_i).$$

In words, $\gamma$ is the minimum proportion of one of the states $[d]$ in all terminal nodes $\{x_i\}_{i=1}^m$. The following lemma gives the number of samples required for an accurate estimate of the similarity matrix, for general transition matrices.

**Lemma 4.12.** *Let* $\hat{R} \in \mathbb{R}^{m \times m}$ *be the matrix given by* Eq. (27). *Then*

$$\Pr\left(\|\hat{R} - R\| \leq t\right) \geq 1 - 2d^2m^2\exp\left(-\frac{2\gamma n t^2}{d^4m^2}\right).$$

There are two important differences between the bounds in Lemmas 4.11 and 4.12. First, the number of samples required is of order $O(d^4)$, rather than $O(d^2)$ in the JC model. This is expected due to lack of structure in the transition matrices. Second, if one or more of the states $\{1,\ldots,d\}$ appears with low frequency, than the required number of samples is increased, due to the dependency on $\gamma$.

## 5 The spectral criterion and a quartet based approach

In this section we show that the spectral criterion for merging subsets of terminal nodes is closely related to quartet based inference, a popular approach to recover latent tree models, see [34, 47, 3, 45, 48, 54] and references therein. Quartet based inference is often a two step procedure. (i) estimate the topology for a large number of quartets of terminal nodes. (ii) Based on the individual quartets, estimate the topology of the full tree.

There are several approaches for the recovery of the full tree in step (ii). One approach is to find a tree that is consistent with the topology of the largest number of quartets, as estimated in step (i). The drawback of this approach is that in general it is a computationally hard problem, see [12]. The quartet puzzling approach applies a greedy algorithm that first estimates the topology of a single quartet, and successively adds a single node at a time [54, 59]. An alternative method [48] computes a pairwise distance matrix between all taxa based on the collection of quartets. The tree is then constructed via a distance based method.

Mihaescu et. al. [39] derived a link between quartet methods and NJ by proving a new guarantee for NJ. Let $ik; jl$ denote a quartet of terminal nodes $x_i, x_k, x_j, x_l$, with a topology as in Figure 2, where the pairs $(x_i, x_k)$ and $(x_j, x_l)$ are siblings. Informally, [39] showed that NJ recovers the correct tree if the estimated distance matrix $D$ satisfies, for all quartets $ik; jl$, the following four point condition,

$$D(i,k) + D(j,l) \leq \min\{ D(i,j) + D(k,l), D(i,l) + D(j,k)\}\,. \tag{28}$$

Here, we derive a similar connection between quartet based inference and SNJ. To this end, in Section 5.1 we define *the quartet determinant* criterion and establish its relation to the four point condition in Eq. (28). Next, in Section 5.2 we prove that SNJ's spectral criterion is proportional to the normalized sum of squared quartet determinants. In Section 5.3 we compare the finite sample guarantee in Theorem 4.10 to the guarantees obtained for quartet based methods in [15, 3]. Based on the results of Section 5.2, we derive a quartet based approach by replacing SNJ's *sum of squared quartets* merging criterion with a *max quartet* criterion. With the new criterion, we prove that under the Jukes-Cantor model, the required number of samples for accurate reconstruction is similar to [15, 3]. Comparing SNJ to the max-quartet approach, we discuss the trade off between statistical efficiency and computational complexity.

### 5.1 The quartet determinant and the four point condition

Let $w(ik; jl)$ denote the following $2 \times 2$ determinant,

$$w(ik;jl) = \begin{vmatrix} R(i,j) & R(i,l) \\ R(k,j) & R(k,l) \end{vmatrix}\,.$$

By Lemma 3.1, $w(ik; jl) = 0$ if and only if the pairs $(x_i, x_k)$ and $(x_j, x_l)$ are siblings. Thus, one can use the value of $w(ik; jl)$ to determine the topology of a quartet. Several works derived algorithms that recover latent tree models based on the quartet values $w(ik; jl)$. Anandkumar et. al. [3] developed spectral recursive grouping, which determines if $x_i, x_k$ are

siblings by computing $w(ik; jl)$ for all $j, l$. To reconstruct a three layer tree, [29] applied spectral clustering to the following score matrix,

$$S(i,k) = \sum_{k,l} |\, w(ik; jl)\,| \ .$$

Applying the spectral properties established in Lemma 3.1 to a tree of four nodes translates directly into the four point condition. If $x_i, x_k$ are siblings then $w(ik; jl) = 0$ and hence

$$R(i,j)R(k,l) = R(i,l)R(k,j)\,.$$

Recall that by Eq. (13) $D(i, j) = \log R(i, j)$. Taking logs on both sides yields

$$D(i,j) + D(k,l) = D(i,l) + D(k,j)\,. \tag{29}$$

In addition, $w(kl; ij) > 0$, and hence

$$D(i,k) + D(j,l) < D(i,j) + D(k,l)\,. \tag{30}$$

Combining Eq. (29) and (30) yields the four point condition in (28).

### 5.2 The quartet determinant and the SNJ merging criterion

Let $A$ and $B$ be non-overlapping sets that are each equal to the observed nodes of a clan in a tree, and let $C = A \cup B$. The following lemma relates the $\sigma_2(R^C)$ criterion for merging $A$ and $B$ and the sum over quartet values $w(ik; jl)$.

**Lemma 5.1.** *For the population matrix R, The SNJ criterion $\sigma_2(R^C)$ can be written in terms of the quartet scores as follows,*

$$\sigma_2(R^C)^2 = \frac{1}{4\sigma_1(R^C)^2} \sum_{i,k \in A \cup B} \ \sum_{j,l \in (A \cup B)^c} w(ik; jl)^2\,.$$

Lemma 5.1 sheds new light on the spectral neighbor joining criterion for merging subsets of terminal nodes. At each iteration, SNJ merges two subsets $A, B$ that minimize a *weighted quartet score*, where $w(ik; jl)^2$ serves as a measure of consistency between the quartet $i, j, k, l$ and the potential merge of $A$ and $B$. Thus, similar to quartet methods, the result of each step of SNJ is a merge that maximizes the consistency across all possible quartets $i, k \in A \cup B$ and $j, l \in (A \cup B)^c$.

### 5.3 The maximum quartet score and finite sample guarantees

Inspired by Lemma 5.1, we suggest the following criterion for merging subsets of terminal nodes,

$$M(A,B) = \max_{i,k \in A \cup B;\, j,l \notin A \cup B} |\, w(ik;jl)\,| \ . \tag{31}$$

In words, we propose a different NJ type algorithm where we replace the *sum of squared quartets* criterion in Lemma 5.1 with *the max quartet* criterion. Clearly, the algorithm is consistent. Given the exact similarity matrix $R$, if $A \cup B$ forms a clan,

$$w(ik;jl) = 0 \ \ \forall (i \in A, k \in B, j,l \notin A,B),$$

and hence $M(A, B) = 0$. On the other hand, if $A \cup B$ does not form a clan, there is at least one pair of nodes $k,l \in A \cup B$ such that for any pair $i, j \in A \cup B$ the topology is $ik; jl$, see illustration in Figure 6. Let $h_1, h_2$ be the two nodes that split between $(i, k)$ and $(j, l)$ as in the right panel of Figure 6. The criterion $|w(ij; kl)|$ is equal to

$$\begin{aligned} |\, w(ij;k,l)\,| \ &= \ |\, R(i,k)R(j,l) - R(i,l)R(k,j)\,| \\ &= R(i,h_1)R(k,h_1)R(j,h_2)R(l,h_2)(1 - R(h_1,h_2)^2) > 0\,. \end{aligned}$$

Thus, if $A \cup B$ is not a clan, the criterion is proportional to the product of similarities between $h_1, h_2$ and the four taxa. To further analyse this expression, we denote by depth($\mathcal{T}$) the *depth of a tree* $\mathcal{T}$, which was defined in [15] in the following way. For an edge $e(h_i, h_j)$, let $\boldsymbol{x}_A(h_i, h_j)$, $x_B(h_i, h_j)$ denote a partition of the taxa induced by $e(h_i, h_j)$. We denote by $g(h_i, h_j)$ the maximum between two values: (i) the number of edges from $h_i$ to the closest taxon in $\boldsymbol{x}_A$ and (ii) the number of edges from $h_j$ to the closest taxon in $\boldsymbol{x}_B$. Finally, the depth of a tree $\mathcal{T}$ is defined by

$$\text{depth}(\mathcal{T}) = \max_{e(h_i,h_j) \in \mathcal{T}} g(h_i,h_j)\,. \tag{32}$$

The following theorem addresses the statistical efficiency of the max quartet NJ algorithm.

**theorem 5.2.** *Assume that the similarity between adjacent nodes is bounded as in* Eq. (2) *and that the data is generated according to the Jukes-Cantor model. The number of samples sufficient for an accurate reconstruction of the tree by the max quartet approach scales as*

$$n = O\Big(\log(m) \,/\, \delta^{4(depth(\mathcal{T})+1)}\Big). \tag{33}$$

Similar guarantees to Theorem 5.2 were derived for quartet based approaches such as [15] and [3], and the Recursive Grouping algorithm [11]. In addition, [15] showed that under two common tree distributions, the depth of almost all random trees scales as $O(\log \log m)$. Indeed for such a tree, the guarantee in Theorem 5.2 is polynomial in $\log m$. However, for cases such as binary symmetric trees where depth($\mathcal{T}$) = $\log m$, then $n = O(\log(m)/m^{4 \log \delta})$, which, for $\delta^2 < 0.5$ is similar to the bound for SNJ in Theorem 4.10.

The finite sample guarantee of $O(\log(m)/\delta^{4\text{depth}(T)})$ for quartet based methods such as [15, 3] is achieved by analyzing only *short quartets* [15] where the distance between siblings is smaller than 2depth($\mathcal{T}$). The drawback is that finding short quartets requires a costly search of all combinations of four terminal nodes. For example, [15] prove that their computational complexity is $O(m^5 \log m)$. Similarly, the computation of the max quartet criterion requires, for some subsets, a search of $O(m^4)$, making the algorithm intractable for large trees. In contrast, computing the sum of quartets in Lemma 5.1 can be done efficiently by computing the singular values.

Figure 7 shows the RF distance and runtime of both approaches on trees generated according to the coalescent model. The accuracy of SNJ is similar to the max-quartet approach, with a much lower runtime.

## 6 Comparison between Atteson's NJ guarantee and its SNJ analogue

Here, we make a qualitative comparison between the NJ sufficient condition for perfect tree recovery in Eq. (8) and its SNJ analogue in Theorem 4.9. We make two simplifying assumptions: (i) the affinity between all adjacent nodes is equal to $\delta$, and (ii) $\delta^2 \geq 0.5$. Our main insight is that to guarantee perfect recovery for trees with a large diameter, SNJ requires fewer samples than NJ.

The comparison between the two guarantees is done in two steps. First, in Eqs. (36) and (37) we derive requirements for the accuracy of $\widehat{R}$, that are *less strict* than Eq. (8) (Atteson's condition). In other words, if $R$ satisfies (8) , it also satisfies Eqs. (36) and (37). Then, these requirements are compared to Theorem 4.9.

Under the assumption that the similarity between all adjacent nodes is $\delta$, the NJ sufficient condition (8) simplifies to

$$\left| \log \widehat{R}(i,j) - \log R(i,j) \right| = \left| \log \frac{\widehat{R}(i,j)}{R(i,j)} \right| \leq -\frac{\log \delta}{2} = \log \delta^{-0.5} \qquad \forall i,j\,. \tag{34}$$

Taking an exponent on both sides and simple algebraic manipulations give

$$(1-\delta^{-0.5})R(i,j) < R(i,j) - \widehat{R}(i,j) < (1-\delta^{0.5})R(i,j) \qquad \forall i,j\,. \tag{35}$$

Since $0 < \delta < 1$, if Eq. (35) holds, then

$$\left| R(i,j) - \widehat{R}(i,j) \right| \leq \delta^{-0.5} R(i,j) \qquad \forall i,j\,. \tag{36}$$

Let diam($\mathcal{T}$) denote the diameter of $\mathcal{T}$, defined as the maximal number of edges between a pair of terminal nodes. Let $i^*, j^*$ be a pair of terminal nodes with diam($\mathcal{T}$) edges on the path between them such that $R(i^*, j^*) = \delta^{\text{diam}(\mathcal{T})}$. The requirement in Eq. (36) is for all pairs $i, j$, and hence a necessary condition for $\widehat{R}(i^*, j^*)$ is

$$| R(i^*, j^*) - \widehat{R}(i^*, j^*) | \leq \delta^{\mathrm{diam}(\mathcal{T}) - 0.5}. \tag{37}$$

Next we recall SNJs theoretical guarantee in Theorem 4.9. In our setting $\delta = \xi$ and $\delta^2 \geq 0.5$, hence SNJ recovers the tree if

$$\| R - \widehat{R} \| \leq \frac{1}{2}\delta^3(1 - \delta^2). \tag{38}$$

We point out two differences between Eq. (38) and the corresponding NJ requirements in (36) and (37). First, the inequality in Eq. (38) is on the spectral norm, while in (36) it is on every element in the similarity matrix. Second, the requirement for SNJ does not depend on the number of terminal nodes $m$ or the tree topology. In contrast, the NJ guarantee requires an accuracy of order $O(\delta^{\mathrm{diam}(\mathcal{T})})$.

Let us consider two extreme cases. For trees similar to the caterpillar tree, the diameter is of order $O(m)$. In this case the entries $\widehat{R}(i, j)$ must be extremely accurate as the right hand side in Eq. (37) decays *exponentially* in $m$, a significantly stricter condition than for SNJ. At the other end, consider a tree similar to the binary symmetric tree, with a diameter of $B \log m$, for some constant $B$. In this case, the required accuracy in Eq. (37) is of order $O(m^{B \log \delta})$. This condition is comparable to SNJ for low values of $B$ and high values of $\delta$ which corresponds, respectively, to trees with a small diameter and low mutation rate. For cases with high mutation rate, or if $B$ is large, we expect SNJ to have an advantage over NJ.

In Figures 10, 12 we compare SNJ to NJ for caterpillar trees with $\mathrm{diam}(\mathcal{T}) = m - 1$. The results show that the SNJ is considerably more accurate than NJ for this setting. In Figures 9,11 we compare SNJ to NJ to the binary symmetric tree. Here, the advantage of SNJ is not as significant as in the case of the caterpillar tree, but increases with higher mutation rate. Thus, the simulation results match the qualitative comparison of the two guarantees. A more rigorous comparison between the two methods may be an interesting direction for future research.

## 7 Simulation results

We compare the performance of SNJ to the following methods: (i) standard neighbor joining, equipped with the log-determinant distance (ii) Recursive Grouping (RG) [11] (iii) the Binary Forrest algorithm [27] and (iv) the Tree SVD algorithm [16]. The algorithms are tested on the following tree models: (i) perfect binary trees with equal similarity between all adjacent nodes, and (ii) caterpillar trees, where the non terminal nodes form a path graph. Due to the prohibitive runtime of some of these methods when applied to large trees, we divided the simulation section to three parts:

1. Comparing SNJ and NJ for large trees and $d = 4$ states.
2. Comparing SNJ, NJ and Recursive Grouping for medium sized trees and $d = 4$ states. For this part, in addition to perfect binary and caterpillar trees, we test the

methods on trees generated according to Kingman's coalescent model [64], a common model in phylogeny.

3. Compare SNJ and NJ for heterogeneous mutation rates

In addition, in Section F we compare SNJ,NJ, Tree SVD and Binary Forrest for small trees and $d = 2$ states. In all experiments, the transition matrices between adjacent nodes follow the Jukes-Cantor model. The code for SNJ and scripts to reproduce our results can be found at https://github.com/NoahAmsel/spectral-tree-inference. All simulations were done with the Python phylogenetic computing library Dendropy [60]. The accuracy of a recovered tree is evaluated by the Robinson-Foulds (RF) distance [17], a popular measure for comparison between trees. The RF distance between two trees $\mathcal{T}_1$ and $\mathcal{T}_2$ counts the number of partitions in $\mathcal{T}_1$ that are not present in $\mathcal{T}_2$ and the number of partitions in $\mathcal{T}_2$ not present in $\mathcal{T}_1$.

### Comparison to NJ for large trees and $d = 4$ states.

Figure 9 shows, for the case of a perfect binary tree with $m = 512$ terminal nodes, the RF distance between the tree and its NJ and SNJ estimates, as a function of the sequence length $n$. The similarity between adjacent nodes is $\delta = 0.85, 0.9$. The results are averaged over 5 realizations of the tree model. As expected from the theoretical analysis in Section 6, the advantage of SNJ over NJ increases for trees with high mutation rates.

Next, we consider caterpillar trees. In general, these trees are considered more challenging to recover than balanced ones, see [36]. As shown in Figure 10, the advantage of SNJ over NJ, for both high and low mutation rates is much more apparent in these trees compared to the perfect binary trees. Figure 11 and 12 show the RF distance as a function of the number of terminal nodes $m$, on perfect binary and caterpillar trees, respectively. The number of samples $n$ is fixed to 400 and 800 for the binary and caterpillar trees, respectively and the similarity between adjacent nodes is $\delta = 0.85, 0.9$. The advantage of SNJ increases with the tree size. For perfect binary small trees, the performance of SNJ and NJ is similar.

### Comparison to NJ and RG for medium size trees with $d = 4$ states.

Figure 13 shows, for the case of a perfect binary tree with $m = 128$ terminal nodes, the RF distance between the tree and its NJ, SNJ and RG estimates, as a function of the sequence length $n$ and for $\delta = 0.85, 0.9$. The results are averaged over 5 realizations. The SNJ and NJ algorithms both outperform RG for this tree.

Next, in Figure 14 we show the results for caterpillar trees with $m = 128$ terminal nodes and $\delta = 0.85,0.9$. Here, RG outperforms NJ for high mutation rate. The SNJ method, however, outperforms RG even in this case. As discussed in Section 5, the required number of samples of quartet based algorithms increase exponentially with the depth of the tree as defined in (32). For perfect binary trees, the depth is of order $O(\log m)$, and for caterpillar trees is equal to one. Thus, we expect quartet methods such as RG to require more samples for accurately recover a perfect binary tree, compared to caterpillar tree.

Figure 15 shows the results for trees generated according to the coalescent model. The SNJ slightly outperforms NJ with low mutation rate. For higher mutation rate - the results for both methods are similar. Figures 16, 17 and 18 show the performance, as a function of number of terminal nodes $m$ for perfect binary, caterpillar, and coalescent trees respectively. The number of samples $n$ is fixed to 400, 800 and 1000 for the binary, caterpillar and coalescent trees, respectively. For this range of tree sizes, the performance of SNJ and NJ is similar. Finally, Figure 19 compares the runtime of NJ, SNJ and RG as a function of the number of terminal nodes on a logarithmic scale. As expected, the runtime of RG is much higher than the runtimes of NJ and SNJ.

### Heterogeneity in mutation rates: comparison between NJ and SNJ

Our last simulation compares the performance of NJ and SNJ for the case of heterogeneity in mutation rates along the sequence. The simulation was done on a binary symmetric tree with $m = 128$ terminal nodes. The similarity between all pairs of adjacent nodes is equal to $\delta_r$, with $\delta$ fixed at 0.95 and $r$ sampled according to a Gamma distribution with a mean value of one. Figure 20 shows the RF distance as a function of number of samples $n$ for two values of $\beta$, which denotes the *shape* of the Gamma distribution. A high value of $\beta$ indicates a higher degree of concentration in the mutation rates. Based on the observed data, the distance matrix $D$ was estimated via RAxML [56]. Finally NJ was applied based on the estimated distance matrix, and SNJ based on the corresponding similarity $R(i, j) = \exp(-D(i, j))$. The SNJ algorithm outperforms NJ in this scenario for both values of $\beta$.

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## Acknowledgements

BN is incumbent of the William Petschek professorial chair of mathematics. Part of this work was done while BN was on sabbatical at the Institute for Advanced Study at Princeton. He gratefully acknowledges the support from the Charles Simonyi Endowment. Y.K. acknowledges support by NIH grants R61DA047037, R01HG008383, R01GM131642, UM1DA051410 and 2P50CA121974. The authors would like to thank Junhyong Kim, Stefan Steinerberger and Ronald Coifman for their help in various aspects of the paper.

## References

[1]. Allman ES, Kubatko LS, and Rhodes JA, Split scores: a tool to quantify phylogenetic signal in genome-scale data, Systematic Biology, 66 (2017), pp. 620–636. [PubMed: 28123114]

[2]. Allman ES and Rhodes JA, Molecular phylogenetics from an algebraic viewpoint, Statistica Sinica, (2007), pp. 1299–1316.

[3]. Anandkumar A, Chaudhuri K, Hsu DJ, Kakade SM, Song L, and Zhang T, Spectral methods for learning multivariate latent tree structure, in Advances in Neural Information Processing Systems, 2011, pp. 2025–2033.

[4]. Aris-Brosou S and Excoffier L, The impact of population expansion and mutation rate heterogeneity on DNA sequence polymorphism, Molecular Biology and Evolution, 13 (1996), pp. 494–504. [PubMed: 8742638]

[5]. Atteson K, The performance of neighbor-joining methods of phylogenetic reconstruction, Algorithmica, 25 (1999), pp. 251–278.

[6]. Bryant D, On the uniqueness of the selection criterion in neighbor-joining, Journal of Classification, 22 (2005), pp. 3–15.

[7]. Camin JH and Sokal RR, A method for deducing branching sequences in phylogeny, Evolution, 19 (1965), pp. 311–326.

[8]. Cavender JA and Felsenstein J, Invariants of phylogenies in a simple case with discrete states, Journal of Classification, 4 (1987), pp. 57–71.

[9]. Chang JT, Full reconstruction of markov models on evolutionary trees: identifiability and consistency, Mathematical Biosciences, 137 (1996), pp. 51–73. [PubMed: 8854662]

[10]. Chang JT and Hartigan JA, Reconstruction of evolutionary trees from pairwise distributions on current species, in Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface, Interface Foundation Fairfax Station, VA, 1991, pp. 254–257.

[11]. Choi MJ, Tan VY, Anandkumar A, and Willsky AS, Learning latent tree graphical models, Journal of Machine Learning Research, 12 (2011), pp. 1771–1812.

[12]. Day WH and Sankoff D, Computational complexity of inferring phylogenies by compatibility, Systematic Biology, 35 (1986), pp. 224–229.

[13]. Delsuc F, Brinkmann H, and Philippe H, Phylogenomics and the reconstruction of the tree of life, Nature Reviews Genetics, 6 (2005), pp. 361–375.

[14]. Durbin R, Eddy SR, Krogh A, and Mitchison G, Biological sequence analysis: probabilistic models of proteins and nucleic acids, Cambridge University Press, 1998.

[15]. Erdős PL, Steel MA, Székely LA, and Warnow TJ, A few logs suffice to build (almost) all trees (i), Random Structures & Algorithms, 14 (1999), pp. 153–184.

[16]. Eriksson N, Tree construction using singular value decomposition., in Algebraic statistics for computational biology, Pachter L and Sturmfels B, eds., Cambridge University Press, 2005, pp. 347–358.

[17]. Estabrook GF, McMorris F, and Meacham CA, Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units, Systematic Zoology, 34 (1985), pp.193–200.

[18]. Felsenstein J, Evolutionary trees from dna sequences: a maximum likelihood approach, Journal of Molecular Evolution, 17 (1981), pp. 368–376. [PubMed: 7288891]

[19]. Felsenstein J, Inferring phylogenies, Sinauer associates Sunderland, MA, 2nd ed., 2004.

[20]. Fernandez-Sanchez J and Casanellas M, Invariant versus classical quartet inference when evolution is heterogeneous across sites and lineages, Systematic Biology, 65 (2016), pp. 280–291. [PubMed: 26559009]

[21]. Fitch WM, Toward defining the course of evolution: minimum change for a specific tree topology, Systematic Biology, 20 (1971), pp. 406–416.

[22]. Gascuel O and Steel M, Neighbor-joining revealed, Molecular Biology and Evolution, 23 (2006), pp. 1997–2000. [PubMed: 16877499]

[23]. Gascuel O and Steel M, A ‘stochastic safety radius’ for distance-based tree reconstruction, Algorithmica, 74 (2016), pp. 1386–1403.

[24]. Gu X, Fu YX, and Li WH, Maximum likelihood estimation of the heterogeneity of substitution rate among nucleotide sites., Molecular Biology and Evolution, 12 (1995), pp. 546–557. [PubMed: 7659011]

[25]. Guindon S and Gascuel O, A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood, Systematic Biology, 52 (2003), pp. 696–704. [PubMed: 14530136]

[26]. Hajdinjak M, Fu Q, Hübner A, Petr M, Mafessoni F, Grote S, Skoglund P, Narasimham V, Rougier H, Crevecoeur I, et al., Reconstructing the genetic history of late neanderthals, Nature, 555 (2018), pp. 652–656. [PubMed: 29562232]

[27]. Harmeling S and Williams CK, Greedy learning of binary latent trees, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (2010), pp. 1087–1097.

[28]. Huang F, UN N, Perros J, Chen R, Sun J, and Anandkumar A, Scalable latent tree model and its application to health analytics, Machine Learning in Healthcare NIPS Workshop 2015, (2015).

[29]. Jaffe A, Fetaya E, Nadler B, Jiang T, and Kluger Y, Unsupervised ensemble learning with dependent classifiers, in Artificial Intelligence and Statistics, 2016, pp. 351–360.

[30]. Jaffe A, Nadler B, and Kluger Y, Estimating the accuracies of multiple classifiers without labeled data, in Artificial Intelligence and Statistics, 2015, pp. 407–415.

[31]. Jaffe A, Weiss R, Carmi S, Kluger Y, and Nadler B, Learning binary latent variable models: A tensor eigenpair approach, Proceedings of the 35th International Conference on International Conference on Machine Learning, (2018), pp. 2196–2205.

[32]. Jia F, Lo N, and Ho SYW, The impact of modelling rate heterogeneity among sites on phylogenetic estimates of intraspecific evolutionary rates and timescales, PLoS One, 9 (2014), pp. 1–8.

[33]. Jiang T, Kearney P, and Li M, A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application, SIAM Journal on Computing, 30 (2001), pp. 1942–1961.

[34]. John KS, Warnow T, Moret BM, and Vawter L, Performance study of phylogenetic methods: (unweighted) quartet methods and neighbor-joining, Journal of Algorithms, 48 (2003), pp. 173–193.

[35]. Jukes TH and Cantor CR, Evolution of protein molecules, Mammalian Protein Metabolism, 3 (1969), pp. 21–132.

[36]. Lacey MR and Chang JT, A signal-to-noise analysis of phylogeny estimation by neighbor-joining: insufficiency of polynomial length sequences, Mathematical Biosciences, 199 (2006), pp. 188–215. [PubMed: 16412478]

[37]. Lake JA, Reconstructing evolutionary trees from dna and protein sequences: paralinear distances, Proceedings of the National Academy of Sciences, 91 (1994), pp. 1455–1459.

[38]. Lanciotti RS, Lambert AJ, Holodniy M, Saavedra S, and Signor L. d. C. C., Phylogeny of zika virus in western hemisphere, 2015, Emerging Infectious Diseases, 22 (2016), pp. 933–935. [PubMed: 27088323]

[39]. Mihaescu R, Levy D, and Pachter L, Why neighbor-joining works, Algorithmica, 54 (2009), pp. 1–24.

[40]. Mossel E and Roch S, Learning nonsingular phylogenies and hidden Markov models, in Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, 2005, pp. 366–375.

[41]. Mourad R, Sinoquet C, Zhang NL, Liu T, and Leray P, A survey on latent tree models and applications, Journal of Artificial Intelligence Research, 47 (2013), pp. 157–203.

[42]. Nei M and Kumar S, Molecular evolution and phylogenetics, Oxford University Press, 2000.

[43]. Parisi F, Strino F, Nadler B, and Kluger Y, Ranking and combining multiple predictors without labeled data, Proceedings of the National Academy of Sciences, 111 (2014), pp.1253–1258.

[44]. Pauplin Y, Direct calculation of a tree length using a distance matrix, Journal of Molecular Evolution, 51 (2000), pp. 41–47. [PubMed: 10903371]

[45]. Pearl J and Tarsi M, Structuring causal trees, Journal of Complexity, 2 (1986), pp. 60–77.

[46]. Rannala B and Yang Z, Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference, Journal of Molecular Evolution, 43 (1996), pp. 304–311. [PubMed: 8703097]

[47]. Ranwez V and Gascuel O, Quartet-based phylogenetic inference: improvements and limits, Molecular Biology and Evolution, 18 (2001), pp. 1103–1116. [PubMed: 11371598]

[48]. Rhodes JA, Topological metrizations of trees, and new quartet methods of tree inference, IEEE/ACM Transactions on Computational Biology and Bioinformatics, (2019).

[49]. Roch S, A short proof that phylogenetic tree reconstruction by maximum likelihood is hard, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3 (2006), pp. 92–94. [PubMed: 17048396]

[50]. Rusinko JP and Hipp B, Invariant based quartet puzzling, Algorithms for Molecular Biology, 7 (2012).

[51]. Saitou N and Nei M, The neighbor-joining method: a new method for reconstructing phylogenetic trees., Molecular Biology and Evolution, 4 (1987), pp. 406–425. [PubMed: 3447015]

[52]. Semple C and Steel M, Phylogenetics, vol. 24, Oxford University Press on Demand, 2003.

[53]. Smith AB, Rooting molecular trees: problems and strategies, Biological Journal of the Linnean Society, 51 (1994), pp. 279–292.

[54]. Snir S, Warnow T, and Rao S, Short quartet puzzling: A new quartet-based phylogeny reconstruction algorithm, Journal of Computational Biology, 15 (2008), pp. 91–103. [PubMed: 18199023]

[55]. Sokal RR, A statistical method for evaluating systematic relationships, University of Kansas Science Bulletin, 38 (1958), pp. 1409–1438.

[56]. Stamatakis A, RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models, Bioinformatics, 22 (2006), pp. 2688–2690. [PubMed: 16928733]

[57]. Steel M, Phylogeny: discrete and random processes in evolution, SIAM, 2016.

[58]. Strimmer K and von Haeseler A, Accuracy of neighbor joining for n-taxon trees, Systematic Biology, 45 (1996), pp. 516–523.

[59]. Strimmer K and von Haeseler A, Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies, Molecular Biology and Evolution, 13 (1996), pp. 964–969.

[60]. Sukumaran J and Holder MT, Dendropy: a python library for phylogenetic computing, Bioinformatics, 26 (2010), pp. 1569–1571. [PubMed: 20421198]

[61]. Susko E, Inagaki Y, and Roger AJ, On inconsistency of the neighbor-joining, least squares, and minimum evolution estimation when substitution processes are incorrectly modeled, Molecular Biology and Evolution, 21 (2004), pp. 1629–1642. [PubMed: 15155796]

[62]. Tamura K, Nei M, and Kumar S, Prospects for inferring very large phylogenies by using the neighbor-joining method, Proceedings of the National Academy of Sciences, 101 (2004), pp.11030–11035.

[63]. Waddell PJ and Steel M, General time-reversible distances with unequal rates across sites: mixing $\gamma$ and inverse gaussian distributions with invariant sites, Molecular Phylogenetics and Evolution, 8 (1997), pp. 398–414. [PubMed: 9417897]

[64]. Wakeley J, Coalescent theory: an introduction, no. 575: 519.2 WAK, 2009.

[65]. Wilkinson M, McInerney JO, Hirt RP, Foster PG, and Embley TM, Of clades and clans: terms for phylogenetic relationships in unrooted trees, Trends in Ecology & Evolution, 22 (2007), pp. 114–115. [PubMed: 17239486]

[66]. Yang Z and Rannala B, Molecular phylogenetics: principles and practice, Nature Reviews Genetics, 13 (2012), pp. 303–314.

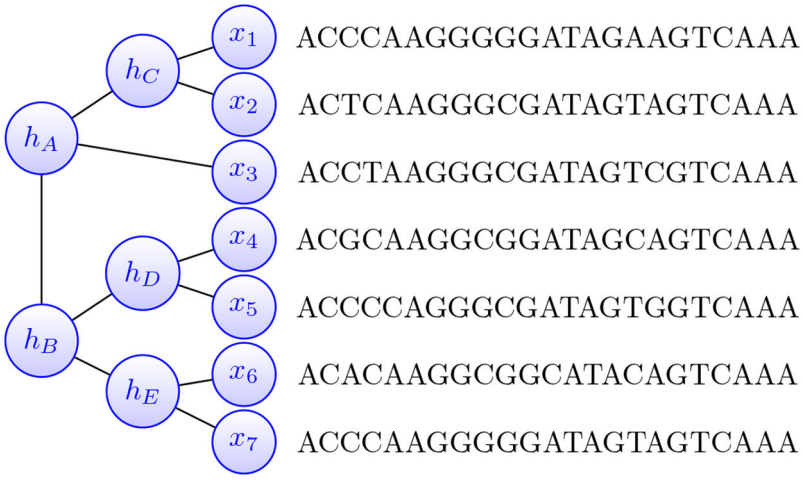


**Figure 1:**

*A tree with $m = 7$ observed nodes. In a typical phylogenetic application, the data consists of a sequence of characters for every terminal node.*

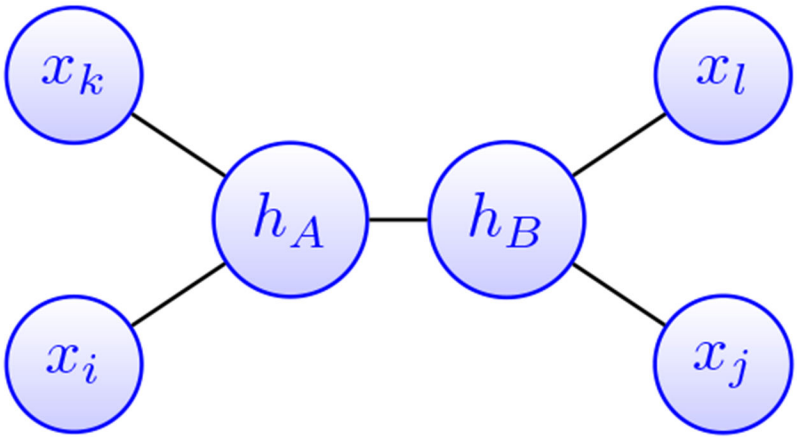


**Figure 2:**
*A subtree with* $m = 4$ *observed nodes.*

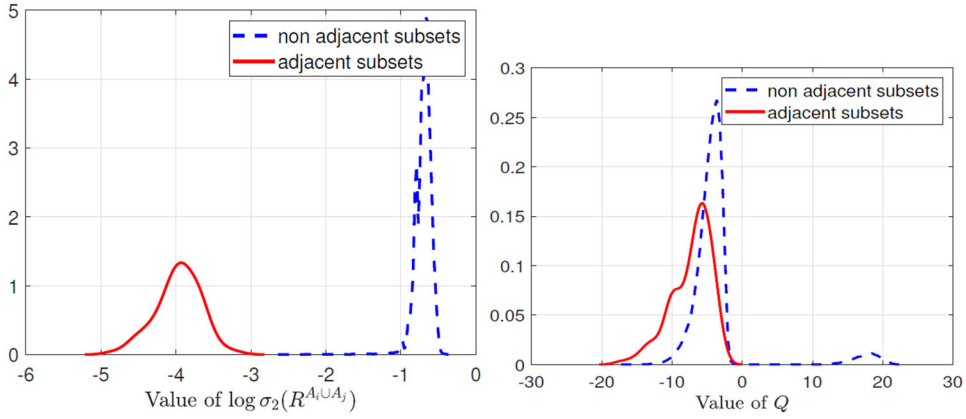


**Figure 3:**

*The red and blue lines are the empirical distributions of the $\sigma_2$ ($R^{A_i \cup A_j}$) criterion from SNJ (left), and the Q criterion from NJ (right), for cases where $A_i$, $A_j$ are adjacent and non-adjacent pairs of singleton sets.*

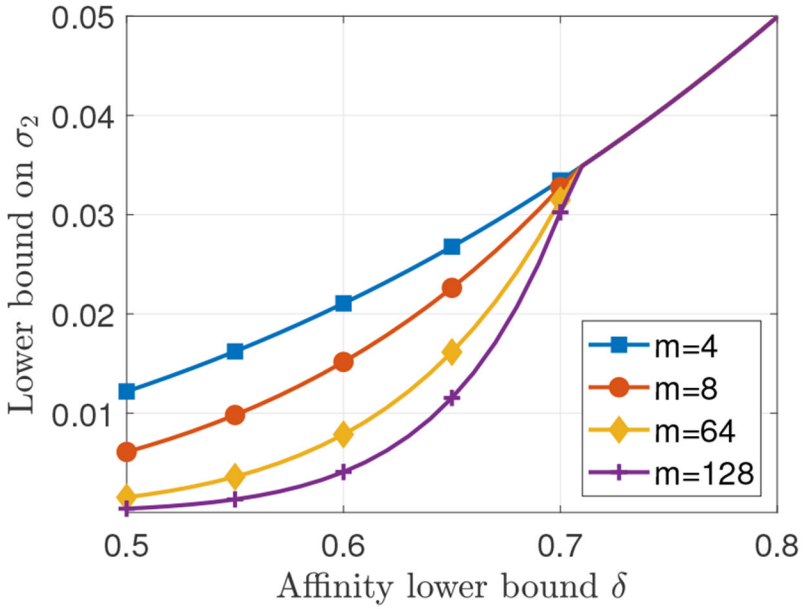


**Figure 4:**

*The lower bound* (14) *on the second largest eigenvalue of* $R^{A \cup B}$ *for non adjacent clans A, B as a function of the affinity lower bound* $\delta$*, at a fixed value* $\xi = 0.95$.

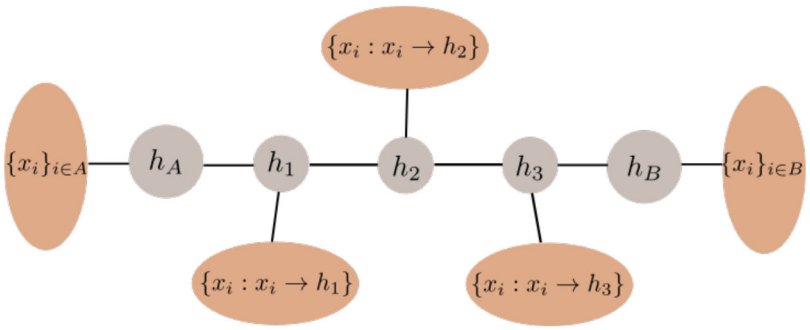


**Figure 5:**

*An example of two non adjacent clans A and B. Every observed node in* $(A \cup B)^c$ *is assigned to the closest node on the path between* $h_A$ *and* $h_B$.

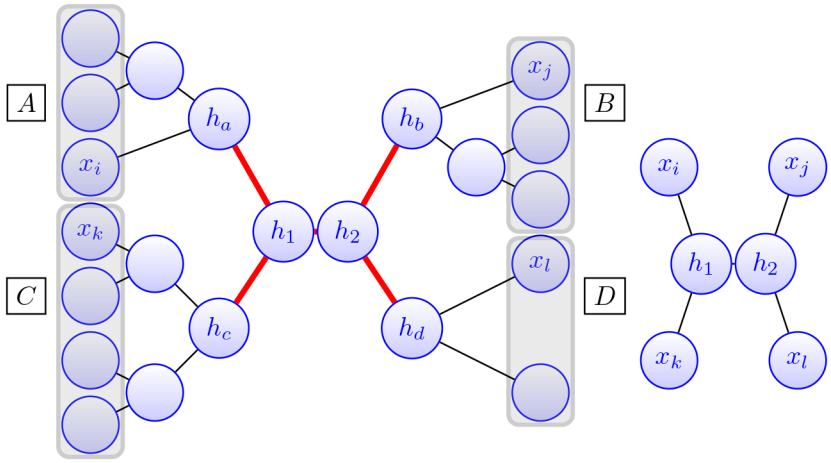


**Figure 6:**

*Computing the max quartet score for merging subset A and subset B. We can find at least one pair k, l* ∈ *C*∪*D and one pair i, j* ∈ *A*∪*B that together satisfy two properties: (i) The topology of the quartet is as in the right subtree, and (ii) The number of edges from the splitting edge e*($h_1$, $h_2$) *to the quartets is at most depth*($\mathcal{T}$) + 1.

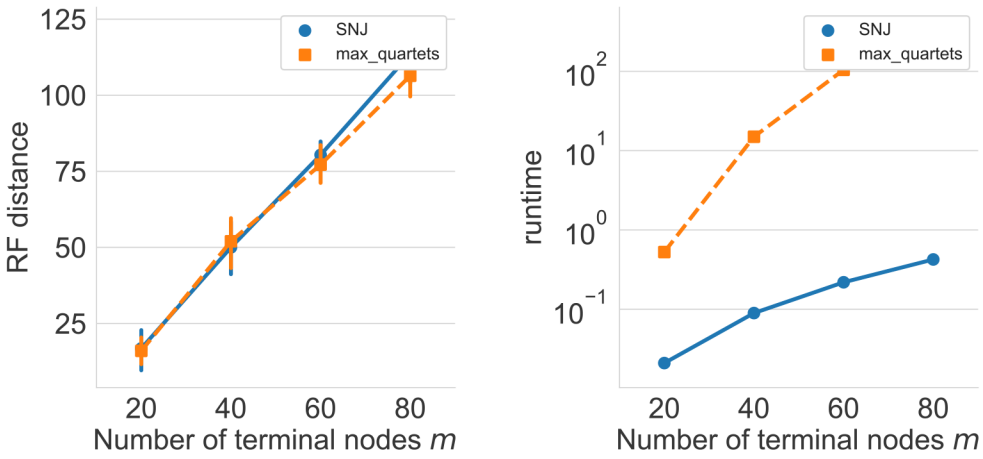


**Figure 7:**

Comparison between SNJ and the max quartet method for recovering trees generated according to the coalescent model. The laft panel shows the RF distance between the reconstructed and ground truth tree as a function of the number of terminal nodes m. The right panel shows the runtime of both methods.

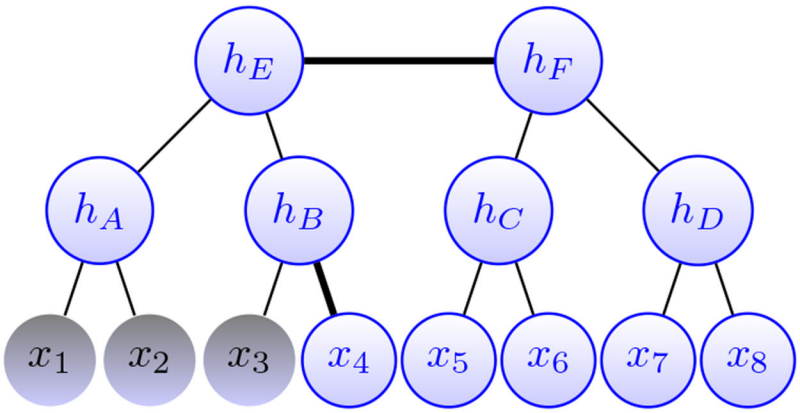


**Figure 8:**

*A perfect binary tree model with* $m = 8$ *terminal nodes. For the proof of Lemma 3.2, the terminal nodes in* $x_A$ *are colored in darker shade of gray. The thick edges form the minimal set that separates* $x_A$ *from* $x_{A^c}$*. The quartet* $i = 1$, $k = 3$, $j = 4$ and $l = 5$ *satisfies* $i, k \in A$, $j, l \in A^c$ *but its topology is not as in* Figure 2.

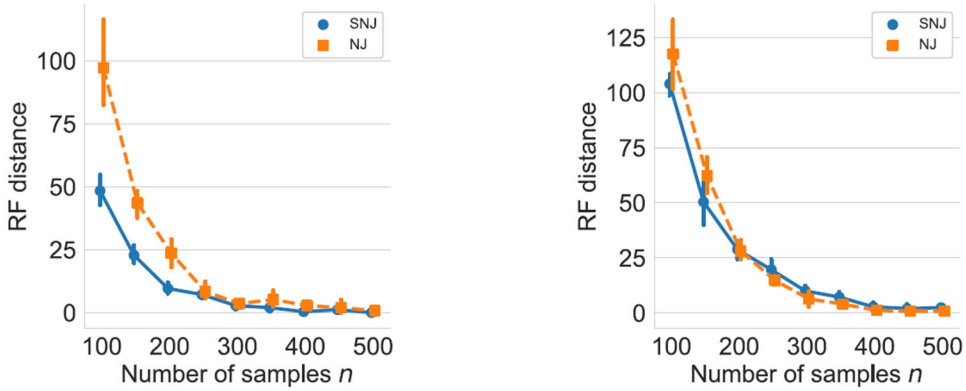


**Figure 9:**

*Comparison between NJ and SNJ for perfect binary trees with* $m = 512$ *nodes, and* $\delta = 0.85$ *(left) and* $\delta = 0.9$ *(right).*

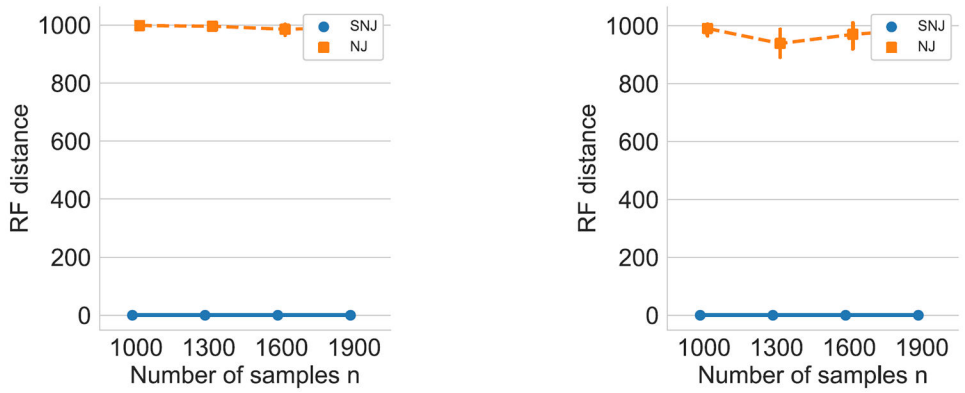


**Figure 10:**

*Comparison between NJ and SNJ for caterpillar trees with* $m = 512$ *nodes and* $\delta = 0.85$ *(left) and* $\delta = 0.9$ *(right).*

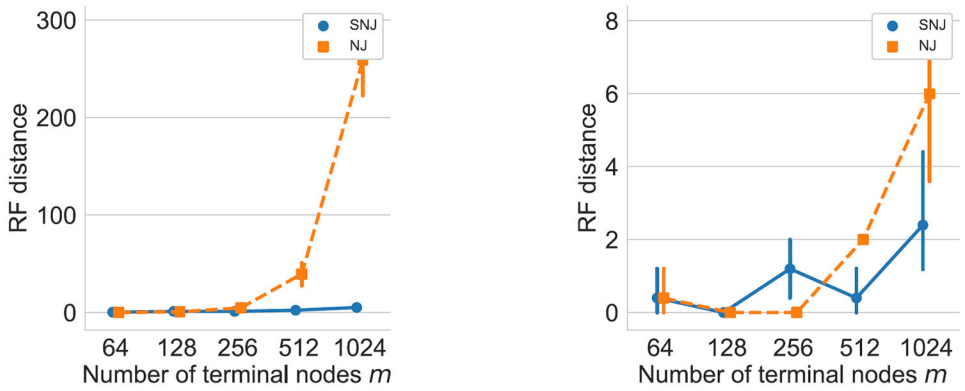


**Figure 11:**

*Comparison between NJ and SNJ for binary trees of different size, and for* $\delta = 0.85$ *(left) and* $\delta = 0.9$ *(right). The number of samples is fixed to* $n = 400$.

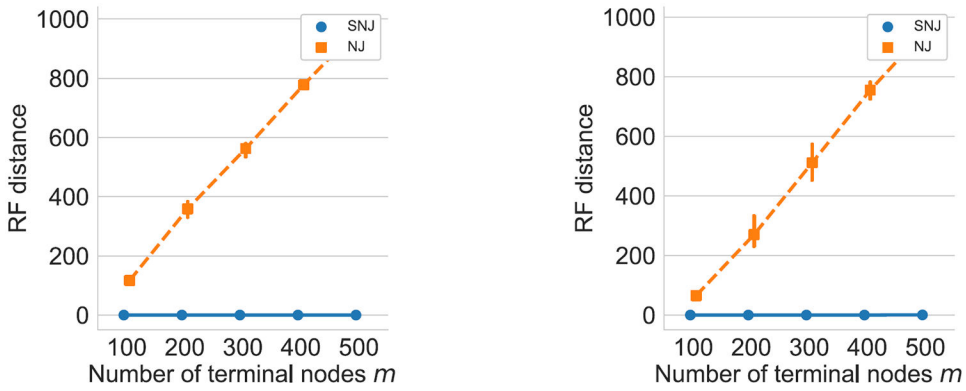


**Figure 12:**

*Comparison between NJ and SNJ for caterpillar trees of different sizes and* $\delta = 0.85$ *(left) and,* $\delta = 0.9$ *(right). The number of samples is fixed to* $n = 800$.

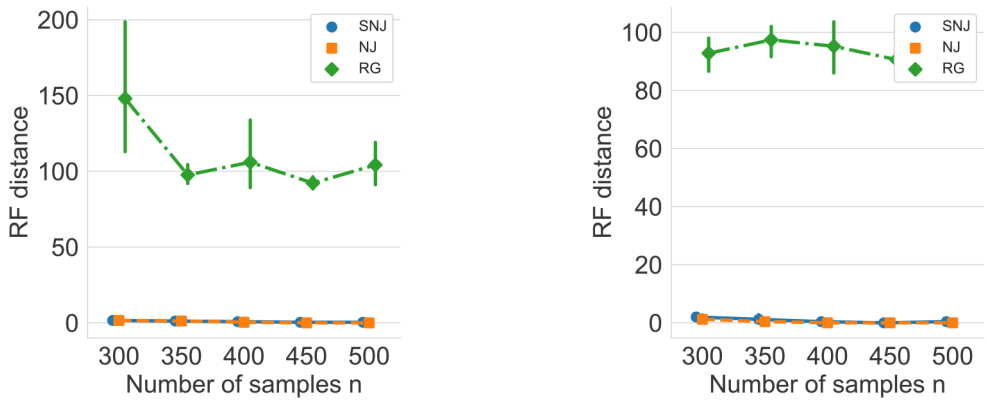


**Figure 13:**

*Comparison between NJ, SNJ and RG for perfect binary trees with* $m = 128$ *nodes and for* $\delta = 0.85$ *(left) and* $\delta = 0.9$ *(right).*

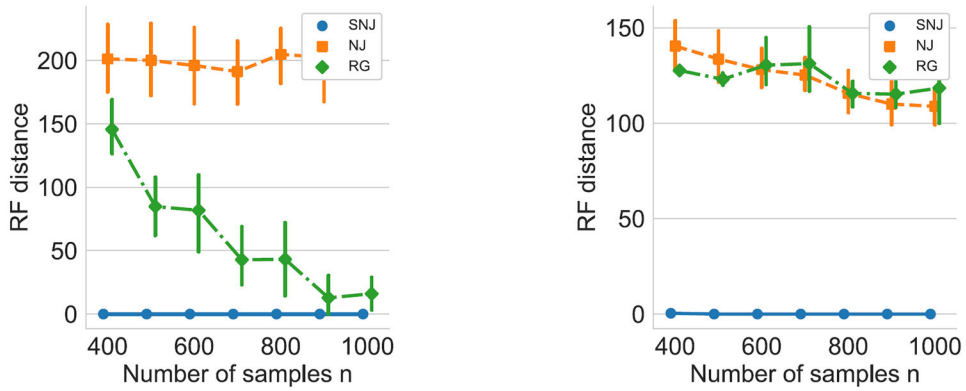


**Figure 14:**

*Comparison between NJ,RG and SNJ for caterpillar trees with* $m = 128$ *nodes and for* $\delta = 0.85$ *(left) and* $\delta = 0.9$ *(right).*

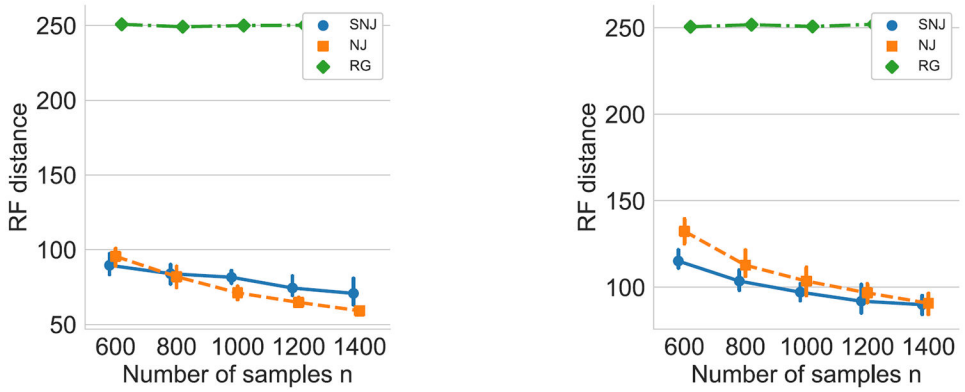


**Figure 15:**

*Comparison between NJ,RG and SNJ for trees with* $m = 128$ *nodes, generated according to the coalescent model and for* $\delta = 0.85$ *(left) and* $\delta = 0.9$ *(right).*

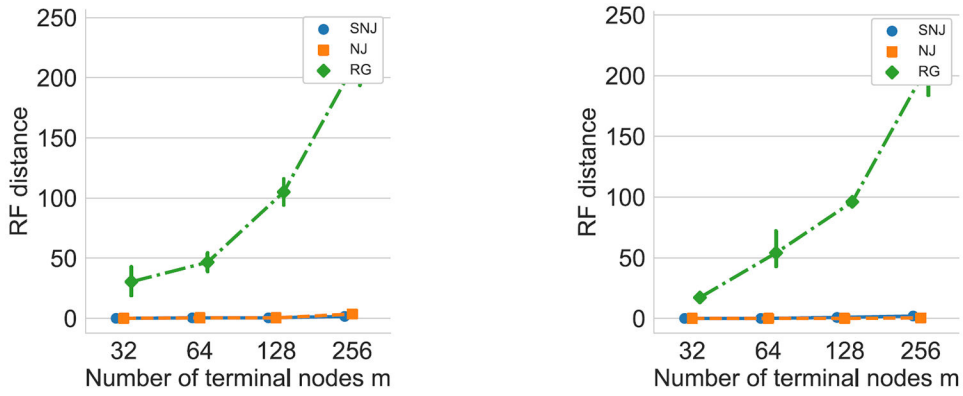


**Figure 16:**

*Comparison between NJ, SNJ and RG for binary trees of different sizes, and for* $\delta = 0.85$ *(left) and* $\delta = 0.9$ *(right). The number of samples is fixed with* $n = 400$ *samples.*

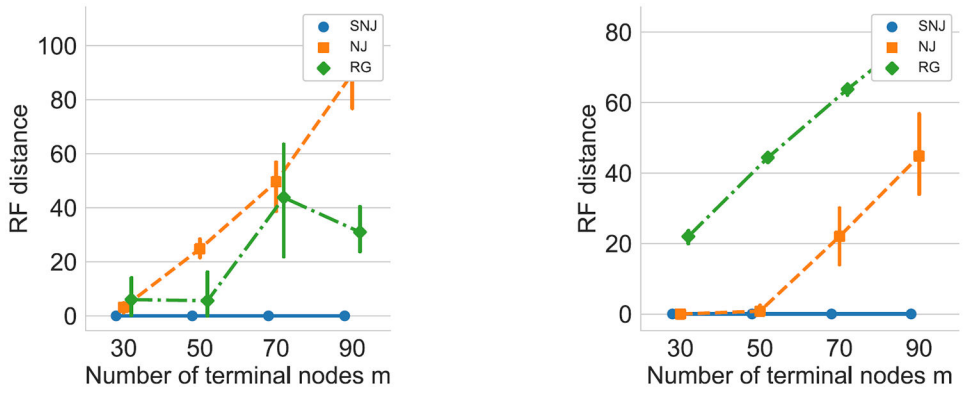


**Figure 17:**

*Comparison between NJ, SNJ and RG for caterpillar trees of different sizes and for* $\delta = 0.85$ *(left) and* $\delta = 0.9$ *(right). The number of samples is fixed with* $n = 800$ *samples.*

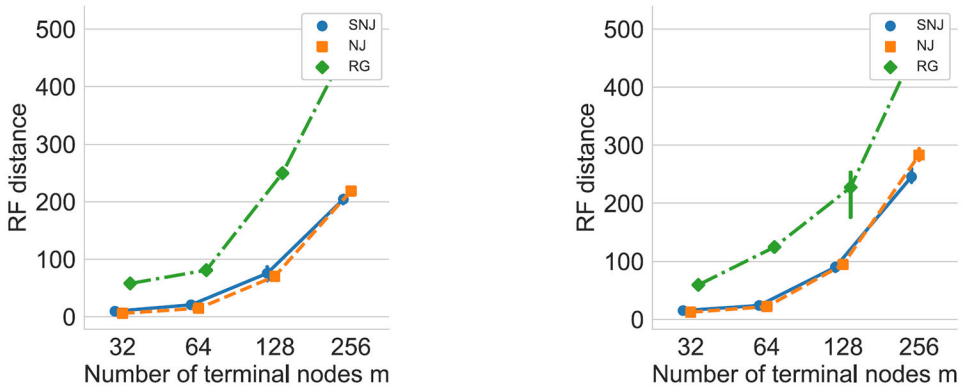


**Figure 18:**

*Comparison between NJ,SNJ and RG for trees with different sizes, generated according to the coalescent model, and for* $\delta = 0.85$ *(left) and* $\delta = 0.9$ *(right). The number of samples is fixed to* $n = 1000$.

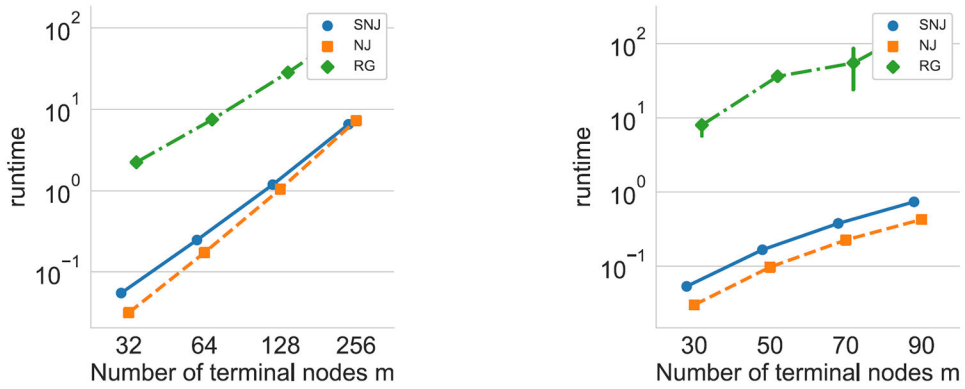


**Figure 19:**

Comparison between the runtime of NJ, SNJ and RG for the experiment on binary and caterpillar trees shown in Figure 16 and 17.

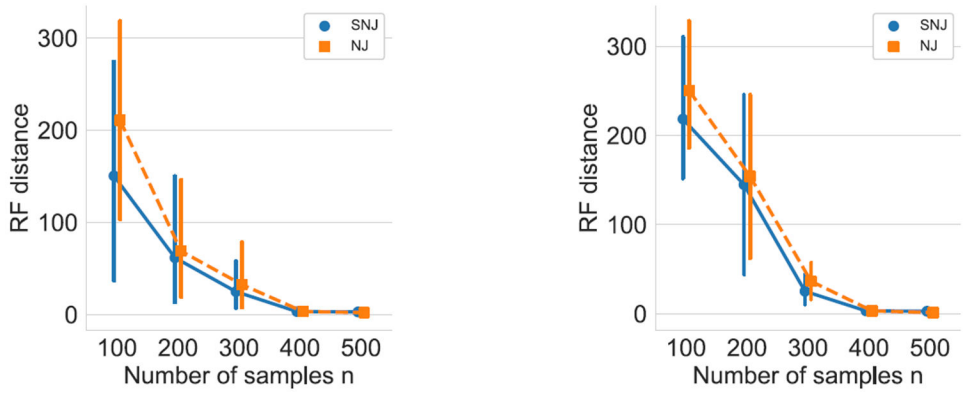


**Figure 20:**

*Heterogeneity in mutation rates. Comparison between NJ and SNJ on a binary tree with* $m = 256$ *terminal nodes. The mutation rates were sampled according to the Gamma distribution with a shape parameter* $a = 5$ *(left) and* $a = 10$ *(right).*