BMC Bioinformatics

BioMed Central

Research article

Open Access

Shortest triplet clustering: reconstructing large phylogenies using representative sets

Le Sy Vinh[2] and Arndt von Haeseler*[1,2]

Address: [1]Heinrich-Heine-Universität Düsseldorf, WE Informatik, Universitätstr. 1, D-040225 Düsseldorf, Germany and [2]Forschungszentrum Jülich, Germany

Email: Le Sy Vinh - vinh@cs.uni-duesseldorf.de; Arndt von Haeseler* - haeseler@cs.uni-duesseldorf.de

* Corresponding author

Published: 08 April 2005

BMC Bioinformatics 2005, **6**:92 doi:10.1186/1471-2105-6-92

Received: 29 November 2004

Accepted: 08 April 2005

This article is available from: http://www.biomedcentral.com/1471-2105/6/92

Abstract

Background: Understanding the evolutionary relationships among species based on their genetic information is one of the primary objectives in phylogenetic analysis. Reconstructing phylogenies for large data sets is still a challenging task in Bioinformatics.

Results: We propose a new distance-based clustering method, *the shortest triplet clustering algorithm (STC)*, to reconstruct phylogenies. The main idea is the introduction of a natural definition of so-called *k-representative sets*. Based on *k*-representative sets, *shortest triplets* are reconstructed and serve as building blocks for the STC algorithm to agglomerate sequences for tree reconstruction in $O(n^2)$ time for n sequences.

Simulations show that STC gives better topological accuracy than other tested methods that also build a first starting tree. STC appears as a very good method to start the tree reconstruction. However, all tested methods give similar results if balanced nearest neighbor interchange (BNNI) is applied as a post-processing step. BNNI leads to an improvement in all instances. The program is available at http://www.bi.uni-duesseldorf.de/software/stc/.

Conclusion: The results demonstrate that the new approach efficiently reconstructs phylogenies for large data sets. We found that BNNI boosts the topological accuracy of all methods including STC, therefore, one should use BNNI as a post-processing step to get better topological accuracy.

Background

Reconstructing the evolutionary relationships among species based on their genetic information is one of the primary objectives in phylogenetic analysis. In recent years, numerous heuristics to reconstruct phylogenies for large data sets have been proposed [1-11]. In addition, parallel tree-reconstruction programs have been implemented [12-15].

To date, distance-based methods introduced by Cavalli-Sforza and Edwards [16] and by Fitch and Margoliash [17] appear most appropriate to reconstruct phylogenies based on thousands of sequences. These methods are a compromise between computational speed and topological accuracy [1,3,5-7] and run typically in $O(n^3)$ time for n sequences [1,3,5] or in $O(n^2)$ for recently suggested approaches [6,7]. Clustering algorithms form a major class of distance-based methods [18]. They do not have an explicit objective function that needs to be optimized.

They rather group sequences (or taxa) iteratively to reconstruct a distance-based phylogenetic tree. UPGMA is a popular method to infer phylogenies with the constraint that a molecular clock is imposed on the evolutionary process. Other clustering approaches have been proposed to relax the molecular clock assumption [1,3,5,19-21].

An attempt to boost the accuracy and to reduce the computational burden is the introduction of *k*-representative set concepts [10,11]. *k*-representative sets consist of at most *k* elements but retain the most important information from whole sets. In this paper, we extend our original approach [10] by introducing a more natural *k*-representative set concept. In a nutshell, representative sets are regarded as components to construct shortest triplets, each of which comprises three closely related sequences from three *k*-representative sets. The collection of shortest triplets serves as building block for a new distance-based clustering method called shortest triplet clustering algorithm (STC).

Results

Simulations were run on a PC cluster with 16 nodes. Each node has two 1.8 GHz processors and 2 GB RAM. Seq-Gen [22] was used to evolve sequences along trees using the Kimura two-parameter model [23] with a transition/transversion ratio of 2.0. We generated 100 simulated data sets of 500 sequences each with sequence lengths 500, 1000 and 2000 nucleotides (nt), respectively. As one model tree, we used the rbcl gene tree with diameter 0.36 substitutions per site as inferred from an alignment of 500 rbcl-genes [10]. We call this *the rbcl-simulation*.

In a second experiment, the so-called *large simulation*, tree topologies were drawn from the Yule-Harding distribution [24], and edge lengths were drawn from an exponential distribution and subsequently rescaled such that the mean diameter of the tree was either 0.1, 0.5, 1.0, or 1.5. For each value of the diameter we generated 100 trees with 1000 sequences and 100 trees with 5000 sequences. Thus, a total of 800 trees were used.

Finally, we tested the accuracy and runtime of the STC and compared it with six other commonly used distance-based methods. More specifically, we investigate the performance of the Neighbor-Joining method (NJ) [1] implemented in PAUP* 4.0 [25], BIONJ [3], Weighbor 1.2 [5], Harmony Greedy Triplet and Four Point Condition (HGT/FP) [7] as well as Greedy Minimum Evolution (GME) and Balanced Minimum Evolution (BME) [6]. Unfortunately, no distance-based program is available for the disc-covering method [4]. All methods were combined with DNADIST version 3.5 [26] and pairwise distances were corrected for multiple hits according to the model used in the simulation. Moreover, we examined the performance of all methods when the balanced nearest neighbor interchange (BNNI) [6] is used as a post-processing step.

Further, to illustrate the performance of STC we re-analyzed the 96-taxon alignments of sequence length 500 nt, that were analyzed in [6] and available at http://www.lirmm.fr/~guindon/simul/. The 6000 trees were split into three groups called "slow" (0.2 substitutions per site), "moderate" (0.4 substitutions per site) and "fast" (1.0 substitutions per site). We call this *the re-analyzed simulation*.

The accuracy of a tree reconstruction method for a simulated data set is measured by the Robinson and Foulds (*RF*) distance [27] between the inferred tree and the model tree used to generate the data set. The RF distance between two trees is the number of bi-partitions present in one of the two trees but not the other, divided by the number of possible bi-partitions. Thus, the smaller the RF distance between two trees the closer are their topologies. In other words, the smaller the RF distance is between the inferred tree and the model tree the higher is the topological accuracy of the tree reconstruction method.

In the following we discuss the results of *the rbcl-simulation*, and *the large simulation* and *the re-analyzed simulation*.

rbcl-simulation

Table 1 shows that the STC outperforms all other methods analyzed in terms of topologlcal accuracy. For instance, the RF distance between the STC-tree and the model tree is on average 0.177 (with respect to the sequence length of 500 nt) and better than NJ (0.190), slightly better than the second best method BME (0.184) and much better than HGT/FP (0.512). Table 1 also demonstrates that all tested methods including STC give higher topologlcal accuracy when the sequence length is increased. However, Table 2 shows that other methods in combination with BNNI outperform STC without BNNI. The combination of STC and BNNI shows similar performance as the combinations of NJ (BIONJ, Weighbor) and BNNI and, slightly better results than the combination of GME (HGT/FP) and BNNI.

Large simulation

Due to the increase in runtime, Weighbor could not be tested. Table 3 and 4 show that STC gives better results than the other methods independent of the diameter. All methods display a decrease in accuracy when the number of sequences changes from 1000 to 5000. As shown in Table 5 and 6, BNNI boosts the accuracy of all methods including STC. All methods give similar results when being used together with BNNI.

Table 1: The average Robinson and Foulds distance of 100 simulated data sets of 500 sequences each with sequence lengths 500, 1000 and 2000 nt (rbcl simulation). Methods are used without BNNI.

sequence length	NJ	BIONJ	Weighbor	HGT/FP	GME	BME	**STC**$^{k=5}$
500	.190	.188	.194	.512	.240	.184	**.177**
1000	.100	.098	.099	.409	.144	.096	**.088**
2000	.049	.048	.050	.313	.082	.046	**.040**

Table 2: The average Robinson and Foulds distance of 100 simulated data sets of 500 sequences each with sequence lengths 500, 1000 and 2000 nt (rbcl simulation). Methods are used with BNNI.

sequence length	NJ	BIONJ	Weighbor	HGT/FP	GME	BME	**STC**$^{k=5}$
500	.162	.162	.162	.166	.163	.163	**.162**
1000	.079	.079	.079	.079	.080	.079	**.079**
2000	.035	.035	.035	.036	.036	.035	**.035**

Table 3: The average Robinson and Foulds distance of 100 simulated data sets of 1000 taxa for each tree diameter 0.1, 0.5, 1.0 and 1.5 and with sequence length 1000 nt (large simulation). Methods are used without BNNI.

number sequences	NJ	BIONJ	HGT/FP	GME	BME	**STC**$^{k=5}$
1000 (0.1)	.146	.146	.378	.168	.143	**.139**
1000 (0.5)	.093	.089	.193	.126	.075	**.066**
1000 (1.0)	.094	.090	.188	.132	.074	**.062**
1000 (1.5)	.097	.091	.182	.138	.073	**.061**

Table 4: The average Robinson and Foulds distance of 100 data sets of 5000 taxa for each tree diameter 0.1, 0.5, 1.0 and 1.5 and with sequence length 1000 nt (large simulation). Methods are used without BNNI.

number sequences	NJ	BIONJ	HGT/FP	GME	BME	**STC**$^{k=5}$
5000 (0.1)	.178	.179	.442	.207	.173	**.170**
5000 (0.5)	.109	.105	.210	.156	.084	**.072**
5000 (1.0)	.107	.102	.192	.155	.073	**.064**
5000 (1.5)	.112	.106	.188	.164	.072	**.063**

Re-analyzed simulation

Except for STC, the accuracies for the other methods displayed in Table 7 and 8 were taken from [6]. Table 7 shows that STC outperforms the other methods in terms of topological accuracy with the exception that Weighbor is slightly better than STC with respect to the slow simulation group. If BNNI is applied, all methods exhibit an almost identical performance (see Table 8).

Another look at the performance

Instead of looking at the average RF distance, we suggest to take a closer look at the simulated data. For each simulated data set, that is subjected to the STC and six other tree reconstruction methods mentioned above, we compute the RF distance between the reconstructed tree and the model tree for all methods. Figure 1 illustrates the results for the large simulation when comparing STC with

Table 5: The average Robinson and Foulds distance of 100 simulated data sets of 1000 taxa for each tree diameter 0.1, 0.5, 1.0 and 1.5 and with sequence length 1000 nt (large simulation). Methods are used with BNNI.

number sequences	NJ	BIONJ	HGT/FP	GME	BME	**STC**$^{k=5}$
1000 (0.1)	.137	.137	.137	.137	.137	**.138**
1000 (0.5)	.061	.061	.061	.061	.061	**.061**
1000 (1.0)	.057	.057	.057	.057	.057	**.056**
1000 (1.5)	.055	.055	.055	.055	.055	**.055**

Table 6: The average Robinson and Foulds distance of 100 data sets of 5000 taxa for each tree diameter 0.1, 0.5, 1.0 and 1.5 and with sequence length 1000 nt (large simulation). Methods are used with BNNI.

number sequences	NJ	BIONJ	HGT/FP	GME	BME	**STC**$^{k=5}$
5000 (0.1)	.168	.168	.168	.168	.168	**.168**
5000 (0.5)	.066	.066	.066	.066	.066	**.066**
5000 (1.0)	.057	.057	.057	.057	.057	**.057**
5000 (1.5)	.055	.055	.055	.055	.055	**.055**

Table 7: The average RF distance of the 96-taxon alignments of sequence length 500 nt, that were analyzed in [6]. The 6000 trees were split into three groups called "slow" (0.2 substitutions per site), "moderate" (0.4 substitutions per site) and "fast" (1.0 substitutions per site). Except for STC, the accuracies for the other methods were taken from [6]. Methods are used without BNNI.

number sequences	NJ	BIONJ	Weighbor	HGT/FP	GME	BME	**STC**$^{k=5}$
96 (slow)	.183	.180	.178	.512	.199	.186	**.179**
96 (moderate)	.136	.134	.129	.480	.158	.137	**.125**
96 (fast)	.115	.112	.103	.465	.144	.117	**.102**

Table 8: The average RF distance of the 96-taxon alignments of sequence length 500 nt, that were analyzed in [6]. The 6000 trees were split into three groups called "slow" (0.2 substitutions per site), "moderate" (0.4 substitutions per site) and "fast" (1.0 substitutions per site). Except for STC, the accuracies for the other methods were taken from [6]. Methods are used with BNNI.

number sequences	NJ	BIONJ	Weighbor	HGT/FP	GME	BME	**STC**$^{k=5}$
96 (slow)	.173	.173	.173	.175	.173	.173	**.173**
96 (moderate)	.119	.118	.118	.123	.118	.118	**.116**
96 (fast)	.090	.090	.091	.098	.091	.090	**.090**

NJ (left column) and STC with the second best method BME (right column). In each diagram specified by the number of taxa and reconstruction methods, 400 points are displayed, that resulted from 100 simulations for each of the tree-diameters (0.1, 0.5, 1.0 and 1.5). Although four tree-diameters were studied only two clouds of points are discernible, where the cloud in the north-east corner of each diagram represents the simulations with the tree-diameter 0.1. The remaining 300 points gather in the south-west cloud because the RF-distances from trees with diameter 0.5, 1.0, 1.5 are not substantially different from each other (see Table 3 and 4). More precisely, the horizontal and vertical axes indicate the RF distances of STC and NJ (or BME), respectively. Each point in the graph

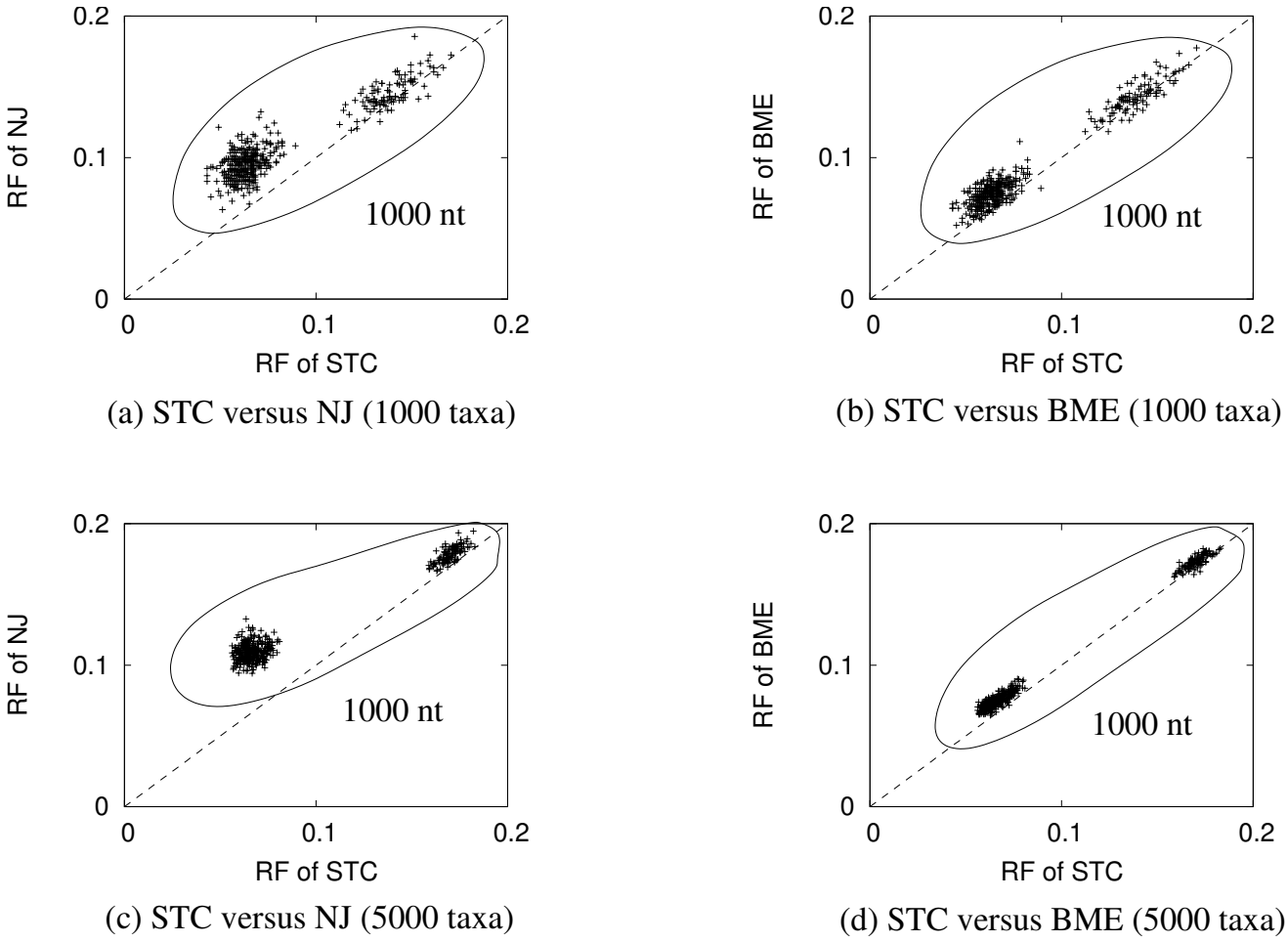
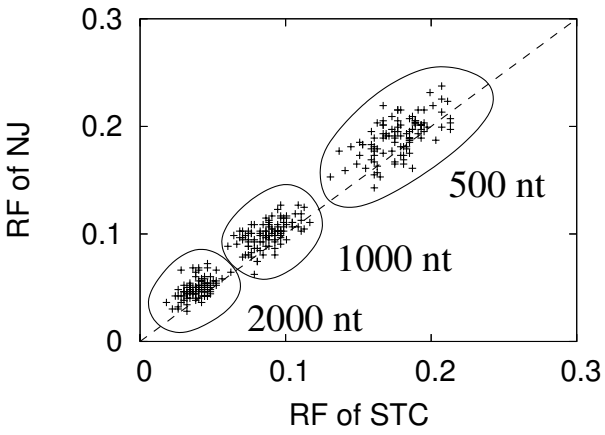


Figure 1
The comparisons of topological accuracy between STC, NJ and BME for the *large simulation*. Each point in the graph presents the Robinson and Foulds (RF) distance for a simulated data set. Points above the dotted line are examples where the RF distance of the STC-tree is less than the RF distance of the NJ-tree or BME-tree. Thus, the STC gives higher topological accuracy than NJ or BME with respect to the simulated data set.

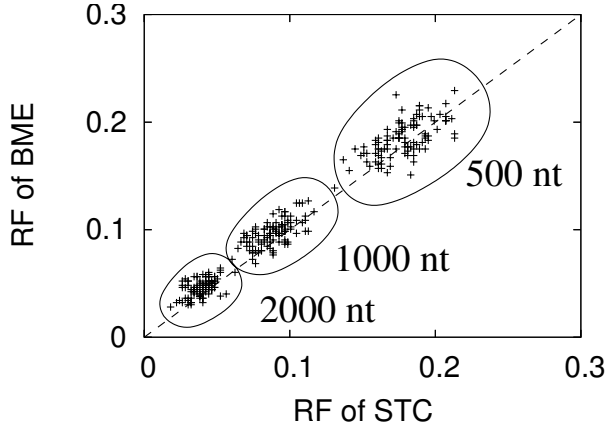
presents the RF distance for a simulated data set. Points above the dotted line are examples where the RF distance of the STC-tree is less than the RF distance of the NJ-tree or BME-tree. Thus, the STC gives higher topological accuracy than NJ or BME with respect to the simulated data set. For example, Figure 1a illustrates the comparison between STC and NJ with respect to 1000 taxa data sets. 379 out of 400 points are above the diagonal, thus, STC gives better results than NJ in about 95% of the simulations. For the remaining 21 alignments (points), two methods showed the same RF distance. Finally, we found 19 points below the diagonal in which case NJ outperforms STC. For the *large simulation* (5000 taxa), NJ is worse than STC in all cases. However, the second best method BME is better than STC in 11% and 5% of the cases with respect to 1000 and 5000 sequence data sets.

Figure 2 shows the same analysis for the rbcl simulation. It shows that with increasing sequence length the cloud of points moves towards zero. From Figure 2 we learn that in some instances NJ (or BME) performs better (with regard to the RF distance) than STC, i.e. 20%, 12%, 8% (or 34%, 17%, 14%) of the simulations for sequence lengths 500, 1000 and 2000 nt, respectively.

Similar results hold for the other methods. These results are summarized in Table 9 where we show the percentage of simulations in which STC is at least as good as the other methods.



(a) STC versus NJ (500 sequences)



(a) STC versus BME (500 sequences)

Figure 2
The comparisons of topological accuracy between STC, NJ and BME for the *rbcl simulation*. Each point in the graph presents the Robinson and Foulds (RF) distance for a simulated data set. Points above the dotted line are examples where the RF distance of the STC-tree is less than the RF distance of the NJ-tree or BME-tree. Thus, the STC gives higher topological accuracy than NJ or BME with respect to the simulated data set.

Table 9: The percentage of cases where STC is at least as good as other tested methods in terms of RF distance. The number in parentheses is the percentage of cases where STC is equally good as other tested methods. Methods are used without BNNI.

number sequences	NJ	BIONJ	Weighbor	HGT/FP	GME	BME
96 (500 nt)	68 (16)	65 (15)	57 (16)	100 (0)	73 (10)	70 (14)
500 (500 nt)	80 (4)	76 (4)	88 (3)	100 (0)	100 (0)	66 (1)
500 (1000 nt)	88 (3)	79 (4)	84 (4)	100 (0)	100 (0)	83 (6)
500 (2000 nt)	92 (6)	90 (4)	92 (3)	100 (0)	100 (0)	86 (9)
1000 (1000 nt)	95 (2)	95 (1)	n.d.	100 (0)	100 (0)	89 (15)
5000 (1000 nt)	100 (0)	99 (0)	n.d.	100 (0)	100 (0)	95 (1)

Table 10: The percentage of cases where STC is better than other tested methods in terms of RF distance. The number in parentheses is the percentage of cases where STC is worse than other tested methods. Methods are used with BNNI.

number sequences	NJ	BIONJ	Weighbor	HGT/FP	GME	BME
96 (500 nt)	9 (8)	8 (8)	10 (10)	12 (10)	10 (8)	10 (9)
500 (500 nt)	34 (37)	35 (39)	35 (36)	59 (29)	46 (33)	41 (39)
500 (1000 nt)	22 (19)	17 (23)	18 (22)	23 (28)	30 (20)	24 (20)
500 (2000 nt)	10 (13)	8 (7)	10 (8)	9 (8)	12 (10)	7 (10)
1000 (1000 nt)	30 (28)	27 (29)	n.d.	28 (22)	30 (24)	28 (27)
5000 (1000 nt)	48 (40)	42 (44)	n.d.	45 (45)	52 (37)	43 (43)

Again, if BNNI is applied we observe that no substantial difference among the various approaches. The accuracy of the methods is mostly determined by BNNI (see Table 10).

Conclusion

We are presenting *k*-representative sets which allow us to design a fast and accurate method to reconstruct phylogenies from large data sets with 1000 or more taxa. Simulations show that STC gives better results than other tested methods in terms of topological accuracy. However, if BNNI is introduced as a subsequent optimization step, the differences in the performance disappear. All methods show more or less the same accuracy. Thus, one should apply BNNI to improve the topological accuracy.

The time to reconstruct a tree of up to 1000 sequences is not really an issue for all tested distance-based methods, with the exception of Weighbor. Weighbor needed about 19 minutes to reconstruct a tree with 500 sequences, thus it is only applicable to data sets with up to some hundred sequences. For data sets with up to 1000 sequences, the remaining methods needed less than one minute to output a tree, thus the difference between methods in terms of runtime is not significant. For data sets with 5000 sequences, STC (GME, HGT/FP or BME) with BNNI took about 2.0 (2.5, 3.0 or 3.5) minutes to reconstruct a tree. NJ (BIONJ) with BNNI were slower and consumed approximately six minutes to output a tree. In short, the combination of STC and BNNI efficiently reconstruct trees for large data sets in both terms of topological accuracy and runtime.

Finally, we did not systematically evaluate the impact of the number of representatives *k*. We present some preliminary results for k = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. Figure 3 shows that the RF distance of STC decreases when k grows from 1 to 5. This proves our intuition that a too small number of triplets leads to an inaccurate estimate of path lengths and edge lengths. When k ranges from 5 to 10, the RF distance remains more or less unchanged. For $k \geq 10$, the RF distance increases steadily indicating a loss of accuracy. The decrease in accuracy is explained by the inclusion of triplets with large distances which include noise and disturb the reconstruction. Thus, we chose $k = 5$ as a good compromise between the accuracy and computational complexity for all data sets. That is, the practical complexity of the STC algorithm is only $O(n^2)$.

Methods

In this section we introduce a new clustering algorithm to reconstruct phylogenies based on distance matrices.

Additive distances

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of n objects (typically contemporary sequences/taxa), let $D = D(uv)$ be a distance matrix where $D(uv)$ is the distance between two objects u and v.

Definition 1

The distance matrix D is additive if and only if it satisfies the four-point condition [28]: *for any quartet* $\{u, v, w, x\}$,

$$D(uv) + D(wx) \leq max\{D(uw) + D(vx), D(ux) + D(vw)\}.$$

In this case, the objects $s \in S$ are related by a tree $T = (V, E)$ where V is the set of vertices such that $S \subset V$ and $E = \{\{v_1, v_2\} | v_1, v_2 \in V\}$ is the set of edges. A vertex with one adjacent edge is called a *leaf*, all other vertices are called *internal nodes*. We let $L \subset V$ be the leaf set of the tree T. Note that we typically require $L \subseteq S$ in the phylogenetic setting.

If D is additive, then there exists a map $\varphi : S \mapsto V$ and a length function $\ell : E \mapsto R_+$ such that

$$D(uv) = \sum_{e \in p(\varphi(u), \varphi(v))} \ell(e) \equiv D_T(\varphi(u), \varphi(v))$$

for all $u, v \in S$ where $p(\phi(u), \phi(v))$ is the unique path connecting $\phi(u)$ and $\phi(v)$ in T and $D_T : V \times V \mapsto R_+$ denotes the distances between vertices in T (cf. [29]). $\ell(e)$ is called edge length of the edge e. To avoid unnecessary complication, we consider only one-to-one maps from S on the leaf set L of T. If D is additive, the reconstruction of tree T and ℓ is trivial. If D is not additive, methods are available that try to fit a tree T to D with respect to an objective function (cf. [30]). Thus, in the following we consider arbitrary distance matrices and we want to reconstruct a tree $\hat{T}$ together with a length function $\hat{\ell}$.

Estimating edge lengths using triplets

We consider a subset X of S, then $\varphi(X) : S \mapsto L$ induces a map on a subtree of T such that the relationships of objects in X are displayed by the subtree with leaf set $\phi(X)$. The complement $S_0(X) = S - X$ we will call the *unclassified object set*, because the relationships of objects in $S_0(X)$ to X is not known from the subtree. Note that we will use S_0 instead of $S_0(X)$ if X is clear from the context.

Let denote $T_r = (V_r, E_r)$ a rooted tree with root r and leaf set L_r, and let S_r be a subset of S such that $\phi(S_r) = L_r$. For convenience, we use S_r and L_r interchangeably.

Now, we consider the most simple edge length estimation problem. That is, we would like to estimate the edge lengths for a triplet tree $\{a, b, c\}$ with distance matrix D (see Figure 4a). Edge lengths are estimated as follows

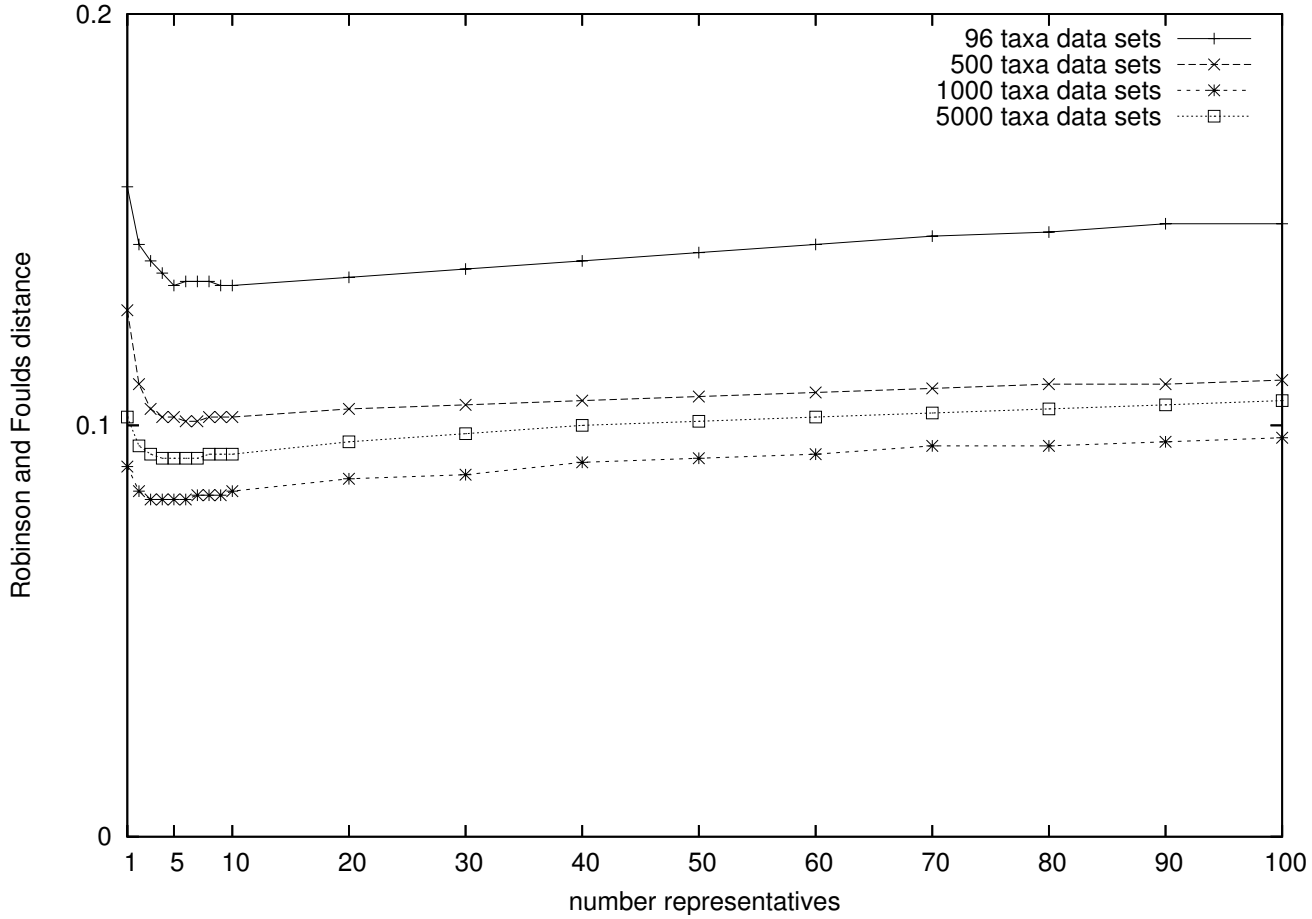


Figure 3
The impact of the number of representatives *k*. The RF distance of STC decreases when *k* grows from 1 to 5. When *k* ranges from 5 to 10, the RF distance remains more or less unchanged. For $k \geq 10$, the RF distance increases steadily indicating a loss of accuracy.

$$\ell(ar \mid abc) = \frac{1}{2}(D(ab) + D(ac) - D(bc)) \quad (1a)$$

$$\ell(br \mid abc) = \frac{1}{2}(D(ab) + D(bc) - D(ac)) \quad (1b)$$

$$\ell(cr \mid abc) = \frac{1}{2}(D(ac) + D(bc) - D(ab)) \quad (1c)$$

Now consider a rooted T_r with the inferred tree-like metric D_{T_r}. The rooted tree T_r consists of two rooted subtrees T_{r_1} and T_{r_2} (see Figure 4b). For convenience, we will use T_i instead of T_{r_i} if r_i is clear from the context. The leaf set $S_r = \{S_1 \cup S_2\}$ where $S_r \subset S$ and $S_0 = S - S_r$ is not represented in T_r. Then we can compute

$$\ell(s_0 r \mid s_0 s_1 s_2) = \frac{1}{2}(D(s_0 s_1) + D(s_0 s_2) - D(s_1 s_2)) \quad (2a)$$

$$\ell(s_1 r \mid s_0 s_1 s_2) = \frac{1}{2}(D(s_0 s_1) + D(s_1 s_2) - D(s_0 s_2)) \quad (2b)$$

$$\ell(s_2 r \mid s_0 s_1 s_2) = \frac{1}{2}(D(s_0 s_2) + D(s_1 s_2) - D(s_0 s_1)) \quad (2c)$$

for each triplet $(s_0, s_1, s_2) \in (S_0 \times S_1 \times S_2)$.

With $D_{T_1}(s_1 r_1)$ and $D_{T_2}(s_2 r_2)$ we denote the known distances of s_1 and s_2 to their roots r_1 and r_2. Thus, we can

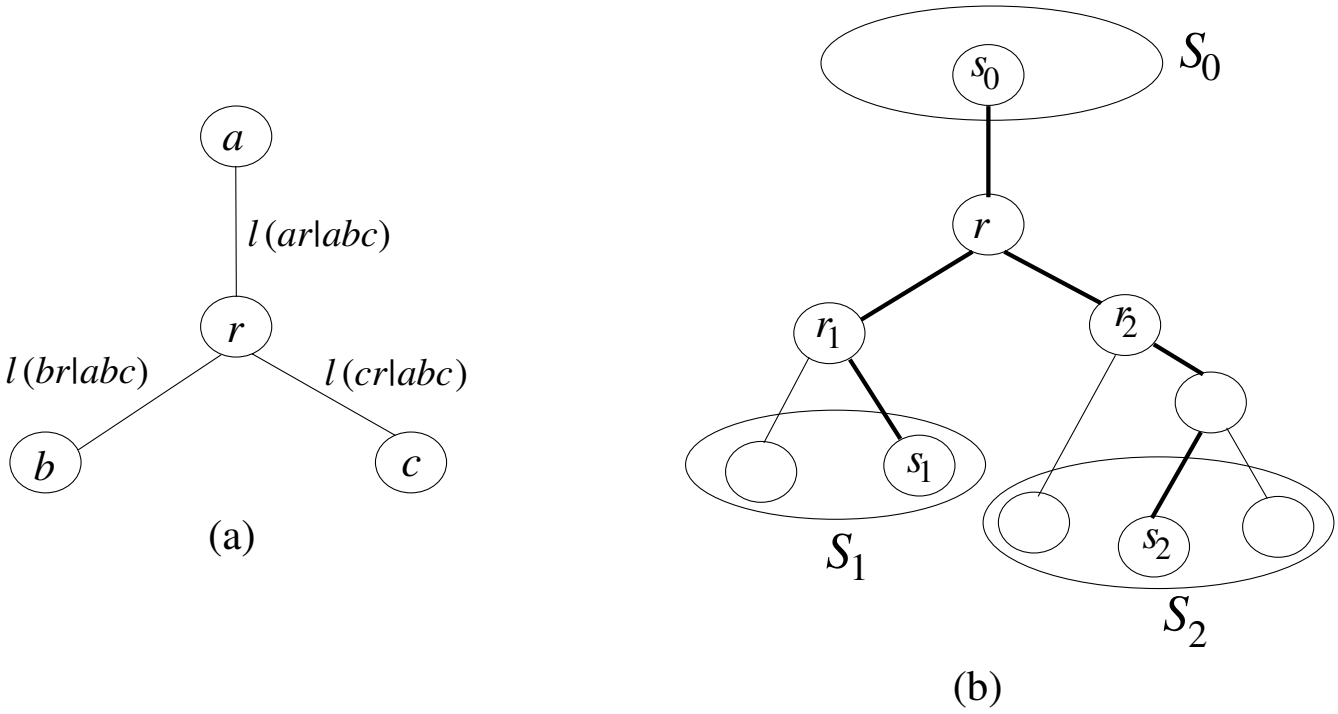


Figure 4
On the left, estimation of edge lengths $\ell(ar|abc)$, $\ell(br|abc)$ and $\ell(cr|abc)$ of the triplet tree $\{a, b, c\}$. On the right, estimation of path length $\ell(s_0r|s_0s_1s_2)$ and edge lengths $\ell(r_1r|s_0s_1s_2)$, $\ell(r_2r|s_0s_1s_2)$ based on the triplet tree $\{s_0, s_1, s_2\}$.

compute for each triplet $\{s_0, s_1, s_2\}$ the lengths $\ell(r_1r)$ and $\ell(r_2r)$ as

$$\ell(r_1r \mid s_0s_1s_2) = \ell(s_1r \mid s_0s_1s_2) - D_{T_1}(s_1r_1) \quad (3a)$$

$$\ell(r_2r \mid s_0s_1s_2) = \ell(s_2r \mid s_0s_1s_2) - D_{T_2}(s_2r_2). \quad (3b)$$

Note that, if D is additive and T_1, T_2 are isometric subtrees of T, the lengths $\ell(r_1r)$ and $\ell(r_2r)$ do not depend on the choice of the triplet $\{s_0, s_1, s_2\}$.

Regardless of additivity considerations, we may define the average length for a fixed $s_0 \in S_0$ as

$$\ell(s_0r \mid s_0S_1S_2) \equiv \frac{1}{|S_1||S_2|} \sum_{(s_1,s_2)\in S_1\times S_2} \ell(s_0r \mid s_0s_1s_2) \quad (4)$$

We can estimate edge lengths $\ell(r_1r)$ and $\ell(r_2r)$ by using all possible triplets as

$$\ell(r_1r \mid S_0S_1S_2) \equiv \frac{1}{|S_0||S_1||S_2|} \sum_{(s_0,s_1,s_2)\in S_0\times S_1\times S_2} \ell(r_1r \mid s_0s_1s_2) \quad (5a)$$

$$\ell(r_2r \mid S_0S_1S_2) \equiv \frac{1}{|S_0||S_1||S_2|} \sum_{(s_0,s_1,s_2)\in S_0\times S_1\times S_2} \ell(r_2r \mid s_0s_1s_2) \quad (5b)$$

Recovering a tree from a distance matrix

The largest path length criterion

We want to reconstruct a tree $T = (V, E)$ with respect to a distance matrix D such that D_T represents D. To this end, we use triplets and the notation of a rooted tree T_r together with Equations 4 and 5.

Our algorithm starts with the observation that if we take an arbitrarily rooted tree T_m with $m \in S$ and length function ℓ_{T_m}, then there must be a pair of leaves (*neighboring leaves*) that share an immediate most recent common ancestor *mrca* which is farthest away from the root m with respect to ℓ_{T_m}. In Figure 5, the pair (3, 4) satisfies this condition, we say this pair fulfills the *largest path length criterion*. The largest path length criterion easily generalizes to arbitrarily rooted subtrees T_i and T_j of T_m, where all descendants from the roots of T_i and T_j are in the vertex sets V_i or V_j, respectively.

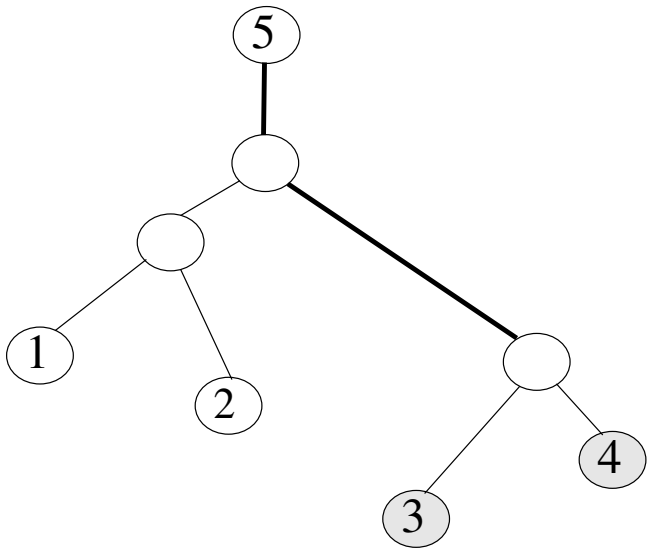


Figure 5
The tree is rooted at leaf 5. In the tree, leaves 3 and 4 with the largest path length from their most recent common ancestor to the root 5 are neighbors.

Let $\mathcal{T}_S$ be the set of rooted subtrees of T_m (each leaf $l \in L_m$ is considered as a rooted subtree T_l). Now consider two disjoint rooted subtrees T_i and T_j of T_m where $i, j \in V_m$. Then the distance $\ell(m, mrca|mS_iS_j)$ from the *mrca* of T_i and T_j to m is computed according to Equation 4, where S_i and S_j are the leaf sets of T_i and T_j, respectively. Then we pick

$$(T_{i_0}, T_{j_0}) = argmax\{\ell(m, mrca|mS_iS_j) \mid T_i, T_j \in \mathcal{T}_S\} \quad (6)$$

as a pair of neighbors (if we detect more than one pair, we randomly select one). By construction, (T_{i_0}, T_{j_0}) fulfills the largest path length criterion.

If D is additive, $\ell(m, mrca|mS_iS_j)$ is exactly the path length from the *mrca* of (T_i, T_j) to m. In other words, the path length from the *mrca* of (T_{i_0}, T_{j_0}) to m is large stand (T_{i_0}, T_{j_0}) is a true neighboring pair. However, in real applications D is rarely additive, therefore the root m is selected so as to avoid noise from stochastic errors involved with large distance estimates [17]. To this end, m is selected such that the distance from the farthest object to root m is minimal,

$$med = argmin_{m' \in S}\{max\{D(m'x)|x = 1,..., n\}\} \quad (7)$$

med is called *a median object*.

Moreover, to reduce the computational complexity of finding a pair of neighbors (T_{i_0}, T_{j_0}) using Equation 6, we store for each $T_i \in \mathcal{T}_S$ its *potential neighbor* $T_{i'} \in \mathcal{T}_S$ such that

$$T_{i'} = argmax\{\ell(med, mrca|med, S_i, S_j)|T_j \in \mathcal{T}_S\}. \quad (8)$$

Now the neighboring pair (T_{i_0}, T_{j_0}) fulfilling the largest path length criterion is determined as follows

$$(T_{i_0}, T_{j_0}) = argmax\{\ell(med, mrca|med, S_{i'}, S_i)|T_i \in \mathcal{T}_S\}. \quad (9)$$

In the following, we present a natural clustering algorithm to reconstruct trees based on distance matrices and the largest path length criterion

Clustering Algorithm

• **Initial step:** Find the median object *med* using Equation 7. Set $\mathcal{T}_S = \{T_1,..., T_n\} - \{T_{med}\}$. Find for each $T_i \in \mathcal{T}_S$ its *potential neighbor* $T_{i'} \in \mathcal{T}_S$ using Equation 8.

• **Selection step (largest path length criterion):** Find the neighboring pair (T_{i_0}, T_{j_0}) using Equation 9.

• **Agglomeration step:** Combine T_{i_0} and T_{j_0} into a new rooted tree $T_{\{i_0j_0\}}$ with root i_0j_0, and estimate new edge lengths of $T_{\{i_0j_0\}}$ using Equation 5. Delete T_{i_0} and T_{j_0} and add $T_{\{i_0j_0\}}$ to $\mathcal{T}_S$. Find the potential neighbor for the new rooted tree $T_{\{i_0j_0\}}$. using Equation 8, and replace $T_{i'}$ for each $T_i \in \mathcal{T}_S$ by $T_{\{i_0j_0\}}$ if $T_{\{i_0j_0\}}$ is its potential neighbor.

• **Stopping step:** If $|\mathcal{T}_S| > 1$ goto the Selection step, otherwise output the tree.

This algorithm is similar to approaches described elsewhere [19-21], however, an essential difference is that we estimate path lengths and edge lengths by using triplets.

Local rearrangement

The heart of the clustering algorithm is the largest path length criterion, at which the path length from the *mrca* of (T_i, T_j) to *med* is estimated by $\ell(med, mrca|med, S_i, S_j)$ using Equation 4. Thus, as path length we take the average of the lengths obtained from at most $O(n^2$ triplets $\{med, s_i,$

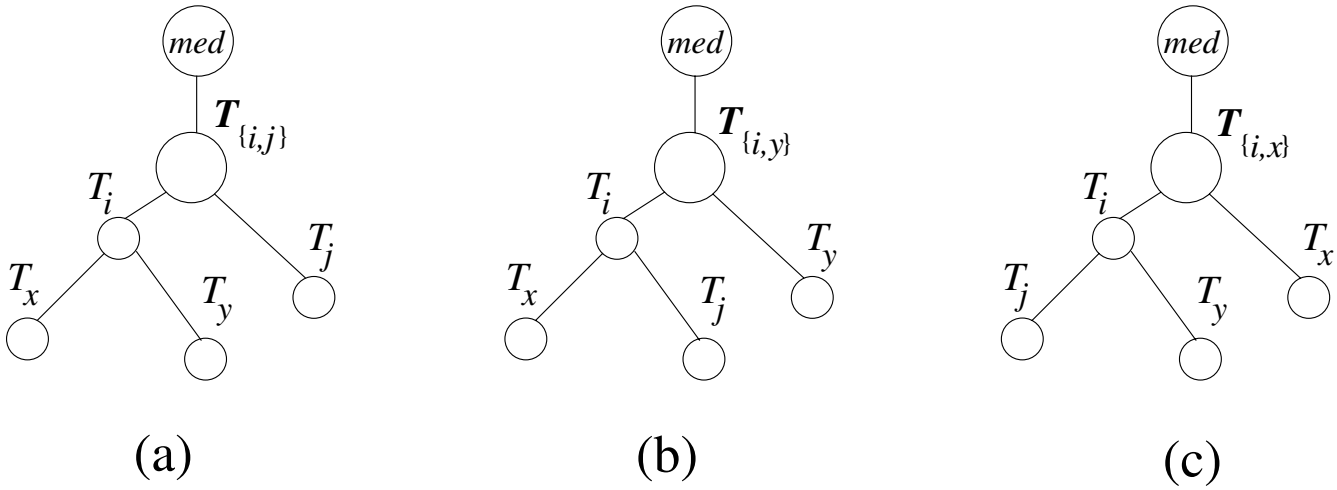


Figure 6
Reconstruction of new rooted tree $T_{\{ij\}}$ using the the preorder traversal procedure based on the largest average path length criterion. If (T_x, T_y) is the neighboring pair, we stick to the suggested grouping of T_i and T_j (see Figure 6a). Otherwise, if (T_x, T_j) or (T_y, T_j) is the neighboring pair, we switch to the trees displayed in Figure 6b or 6c, respectively.

$s_j\} \in med \times S_i \times S_j$. This average may not be the representative estimate of the true path length. Moreover the root *med* may be too far way from the *mrca* and this leads to an inaccurate estimate of the path length.

To take these problems into account, we extend the clustering algorithm. To this end, imagine the algorithm has clustered T_i and T_j with corresponding disjoint leaf sets S_i, $S_j \subset S$ (we have finished the agglomeration step). Thus, we have created the newly rooted tree $T_{\{ij\}}$ with leaf set $S_{ij} = \{S_i \cup S_j\}$ and the set of unclassified objects $S_0(S_{ij}) = S - S_{ij}$. In the following, we describe the nearest neighbor interchange operation around the root of T_i upon condition that T_i consists of two rooted subtrees T_x, T_y (Figure 6a). First, we estimate average path lengths from the unclassified object set $S_0(S_{ij})$ to the *mrca* of (T_x, T_y), (T_x, T_j) and (T_y, T_j) as

$$\ell(S_0(S_{ij})S_xS_y \mid S_xS_y) \equiv \frac{1}{|S_0(S_{ij})||S_x||S_y|} \sum_{(s_0,s_x,s_y)\in S_0(S_{ij})\times S_x\times S_y} \ell(s_0 r \mid s_0 s_x s_y) \quad (10a)$$

$$\ell(S_0(S_{ij})S_xS_j \mid S_xS_j) \equiv \frac{1}{|S_0(S_{ij})||S_x||S_j|} \sum_{(s_0,s_x,s_j)\in S_0(S_{ij})\times S_x\times S_j} \ell(s_0 r \mid s_0 s_x s_j) \quad (10b)$$

$$\ell(S_0(S_{ij})S_yS_j \mid S_yS_j) \equiv \frac{1}{|S_0(S_{ij})||S_y||S_j|} \sum_{(s_0,s_y,s_j)\in S_0(S_{ij})\times S_y\times S_j} \ell(s_0 r \mid s_0 s_y s_j). \quad (10c)$$

For convenience, we will use $\ell(S_0(S_{ij})|S_xS_y)$ instead of $\ell(S_0(S_{ij})S_xS_y|S_xS_y)$. We now use the average path lengths from Equation 10 to decide which pair of subtrees among (T_x, T_y), (T_x, T_j) and (T_y, T_j) is preferred. More specifically, if

$$\ell(S_0(S_{ij})|S_xS_y) \geq max\{\ell(S_0(S_{ij})|S_xS_j),\ \ell(S_0(S_{ij})|S_yS_j)\}$$

we stick to the suggested grouping of T_x and T_y (see Figure 6a). Otherwise, if $\ell(S_0(S_{ij})|S_xS_j)$ or $\ell(S_0(S_{ij})|S_yS_j)$ is larger than the remaining average path lengths, we swap T_y and T_j or T_x and T_j as displayed in Figure 6b or 6c, respectively. Note that, this decision can be considered as a correction of the largest path length criterion by taking all possible triplets into account. We call the correction the *largest average path length criterion*.

We now explain the preorder traversal procedure [31] to reconstruct the rooted tree T_i using the nearest neighbor interchange operation based on the largest average path length criterion (T_i is a subtree of $T_{\{ij\}} = (T_i, T_j)$):

Preorder traversal procedure (T_i)

• **Step 1:** If T_i is a single leaf, return.

• **Step 2:** Otherwise, T_i consists of two subtrees T_x and T_y. Do the nearest neighbor interchange operation around the root of T_i based on the largest average path length criterion (Equation 10). If T_x and T_j (or T_y and T_j) were exchanged, estimate new edge lengths using Equation 5.

• **Step 3:** Apply the preorder traversal procedure to two rooted subtrees of T_i.

Representative sets and shortest triplets

For a set S of sequences (or taxa), the (genetic) distance matrix D is typically not additive due to stochastic errors

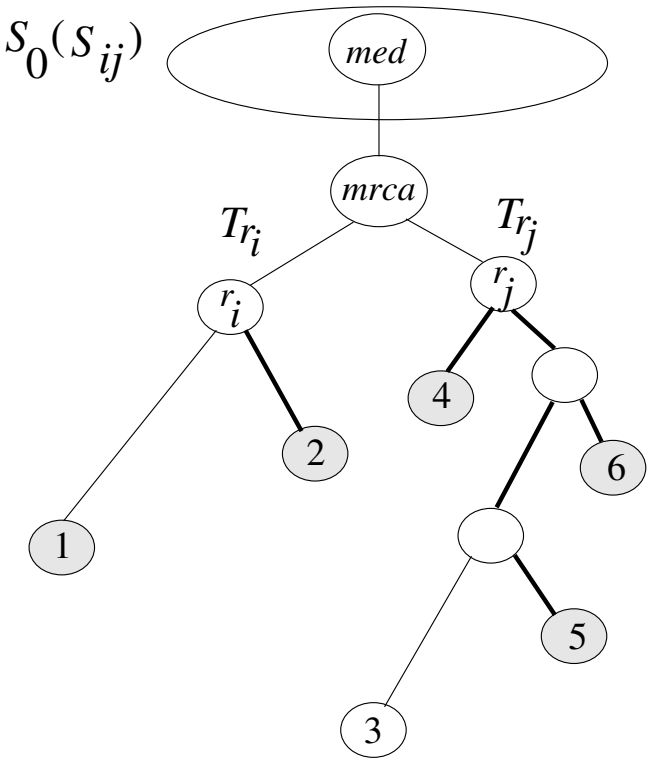


Figure 7
we select only *min*(*k*, |S_i|) and *min*(*k*, |S_j|) closest leaves to the root of T_i and T_j with respect to the path length, respectively, i.e. for *k* = 3 we pick {1, 2} from T_{r_i} and {4, 5, 6} from T_{r_j}. The leaf set {1, 2} (or {4, 5, 6}) is the 3-representative leaf set of the rooted subtree T_{r_i}. (or T_{r_j}).

[17]. Larger distances between two sequences are less accurately estimated. This leads to a low performance of both the clustering algorithm and the preorder traversal procedure for divergent data sets.

In earlier work [10,11], we have presented simple representative concepts to reduce stochastic error involved in large distances. Here, we extend our work by introducing the so-called *k-representative set* concept. We use now genetic distances instead of topological distances (all edges have length 1). Our motivation is to reduce the computational complexity and to exclude objects far away from the root under consideration. In the clustering algorithm, the path length from the *mrca* of (T_i, T_j) to *med* (see Figure 7) can be estimated by two approaches. The first method picks randomly one pair (s_i, s_j) ∈ S_i × S_j then computes

$$\ell(med, mrca \mid med, s_i, s_j) = \frac{1}{2}(D(med, s_i) + D(med, s_j) - D(s_i s_j)). \quad (11)$$

The second approach takes the average distance

$$\ell(med, mrca \mid med, S_i, S_j) \equiv \frac{1}{|S_i||S_j|} \sum_{(s_i, s_j) \in S_i \times S_j} \ell(med, mrca \mid med, s_i, s_j). \quad (12)$$

Both approaches suffer from noise. Estimating the path length using Equation 11 may be inaccurate because it randomly picks a pair (s_i, s_j) which may not be really representative. Equation 12 may be problematic, especially since it might be susceptible to noise, due to the possibility of including long- distances with large stochastic errors.

To overcome these problems, we select only *min*(*k*, |S_i|) and *min*(*k*, |S_j|) closest leaves to the root of T_i and T_j with respect to the path length, respectively. To illustrate, for *k* = 3 we pick {1, 2} from T_i and {4, 5, 6} from T_j in Figure 7.

We now define S_i^k as the set of *min*(*k*, |S_i|) closest leaves to the root of T_i. S_i^k is called the *k-representative leaf set*. Hereafter, we estimate similar to Equation 4 the path length from the *mrca* of (T_i, T_j) to *med* as

$$\ell(med, mrca \mid med, S_i^k, S_j^k) = \frac{1}{|S_i^k||S_j^k|} \sum_{(s_i^k, s_j^k) \in S_i^k \times S_j^k} \ell(med, mrca \mid med, s_i^k, s_j^k) \quad (13)$$

which is only based on the *k*-representative leaf sets. Now we can perform the clustering algorithm with reduced complexity. However, we also want to improve the preorder traversal procedure. The average path length from the unclassified object set $S_0(S_{ij})$ to the *mrca* of (T_i, T_j) is estimated by Equation 10 which also suffers from noise. To overcome this problem, we select only *min*(*k*, |$S_0(S_{ij})$|) unclassified objects closest to the root of tree $T_{\{ij\}}$ with respect to distances $\ell(s_0 r \mid s_0 S_i^k S_j^k)$ where $s_0 \in S_0(S_{ij})$. We call the subset, denoted $S_0^k(S_{ij})$, *k-representative unclassified object set*.

We now define $\{s_0^k, s_i^k, s_j^k\} \in S_0^k(S_{ij}) \times S_i^k \times S_j^k$ a *shortest triplet*, which contains three representatives of the three *k*-representative sets. By construction, s_0^k, s_i^k, s_j^k are close to the root of $T_{\{ij\}}$ and close to each other. Therefore, the edge length estimates based on shortest triplet $\{s_0^k, s_i^k, s_j^k\}$ are less susceptible to estimation errors.

We now rewrite Equation 10 to estimate the average path length from the representative unclassified object set $S_0^k(S_{ij})$ to the *mrca* of (T_i, T_j) using only shortest triplets as

$$\ell(S_0^k(S_{ij}) \mid S_x^k S_y^k) \equiv \frac{1}{|S_0^k(S_{ij})||S_x^k||S_y^k|} \sum_{(s_0^k, s_x^k, s_y^k) \in S_0^k(S_{ij}) \times S_x^k \times S_y^k} \ell(s_0^k r \mid s_0^k s_x^k s_y^k) \quad (14a)$$

$$\ell(S_0^k(S_{ij}) \mid S_x^k S_j^k) \equiv \frac{1}{|S_0^k(S_{ij})||S_x^k||S_j^k|} \sum_{(s_0^k, s_x^k, s_j^k) \in S_0^k(S_{ij}) \times S_x^k \times S_j^k} \ell(s_0^k r \mid s_0^k s_x^k s_j^k) \quad (14b)$$

$$\ell(S_0^k(S_{ij}) \mid S_y^k S_j^k) \equiv \frac{1}{|S_0^k(S_{ij})||S_y^k||S_j^k|} \sum_{(s_0^k, s_y^k, s_j^k) \in S_0^k(S_{ij}) \times S_y^k \times S_j^k} \ell(s_0^k r \mid s_0^k s_y^k s_j^k) \quad (14c)$$

In short, the preorder traversal procedure uses only shortest triplets to estimate path lengths as well as edge lengths.

Shortest triplet clustering algorithm (STC)

We introduce now the shortest triplet clustering algorithm by combining the clustering algorithm, the local rearrangement, the *k*-representative sets, and the shortest triplets approach.

Shortest triplet clustering algorithm (STC)

• **Initial step:**

- **(i):** Find the median object *med* using Equation 7.

- **(ii):** Set $\mathcal{T}_S = \{T_1, ..., T_n\} - \{T_{med}\}$ and for each $T_i \in \mathcal{T}_S$ its representative leaf set $S_i^k = \{i\}$.

- **(iii):** Find for each $T_i \in \mathcal{T}_S$ its *potential neighbor* $T_{i'} \in \mathcal{T}_S$ using Equation 8.

• **Selection step (largest path length criterion):** Find the neighboring pair (T_{i_0}, T_{j_0}) using Equation 9.

• **Agglomeration step:**

- **(i):** Combine T_{i_0} and T_{j_0} into a new rooted tree $T_{\{i_0 j_0\}}$ with root $i_0 j_0$, and estimate new edge lengths of $T_{\{i_0 j_0\}}$ using Equation 5 based on shortest triplets.

- **(ii):** Compute the *k*-representative leaf set $S_{i_0 j_0}^k$ of $T_{\{i_0 j_0\}}$. based on *k*-representative leaf sets $S_{i_0}^k$ and $S_{j_0}^k$ of T_{i_0} and T_{j_0}, respectively.

- **(iii):** Compute the *k*-representative unclassified object set $S_0^k(S_{i_0 j_0})$ of $T_{\{i_0 j_0\}}$.

- **(iv):** Delete T_{i_0} and T_{j_0} and add $T_{\{i_0 j_0\}}$ to $\mathcal{T}_S$.

- **(v):** Find the potential neighbor for the new rooted tree $T_{\{i_0 j_0\}}$ using Equation 8 based on representative sets, and replace $T_{i'}$ for each $T_i \in \mathcal{T}_S$ by $T_{\{i_0 j_0\}}$ if $T_{\{i_0 j_0\}}$ is its potential neighbor.

• **Local rearrangement step:** Apply the preorder traversal procedure to the rooted subtrees T_{i_0} and T_{j_0} of the new rooted tree $T_{\{i_0 j_0\}}$ based on only shortest triplets.

• **Stopping step:** If $|\mathcal{T}_S| > 1$, goto Selection step, otherwise output the tree.

The complexity of STC

Now we briefly describe the complexity of the STC. At the initial step, (i), (ii), and (iii) are done in $O(n^2)$, $O(n)$ and $O(n^2)$ time, respectively. Thus, the complexity of the initial step is $O(n^2)$. The selection step is done in $O(n)$. At the agglomeration step, (i), (ii), (iii), (iv), and (v) are done in $O(k^3)$, $O(k)$, $O(nk^2)$, $O(1)$, and $O(nk^2)$ time, respectively. Thus, the complexity of the agglomeration step is $O(nk^2 + k^3)$. Finally, we are estimating the complexity of the preorder traversal procedure based on only shortest triplets. Step 1 is done in constant time. Step 2, the nearest neighbor interchange operation around the root of T_i costs $O(k^3)$. Estimating new edge lengths is done in $O(k^3)$ time. Re-computing the *k*-representative leaf set S_i^k of T_i based on *k*-representative leaf sets of its rooted subtrees T_x and T_y costs $O(k)$ time. Finally, re-computing the *k*-representative unclassified object set $S_0^k(S_i)$ of T_i based on the *k*-representative leaf set S_i^k of T_j and the *k*-representative unclassified object set $S_0^k(S_{ij})$ of $T_{\{ij\}}$ is done in $O(k)$ time. Thus, the complexity of step 2 is $O(k^3)$. Step 3 is done in constant time. Step 1, step 2, and step 3 are repeated $O(n)$ times so the complexity of the preorder traversal procedure is $O(nk^3)$.

In the STC algorithm, the selection step, the agglomeration step and the local rearrangement step are repeated $(n - 2)$ times so the overall complexity of the STC algorithm is $O(n^2k^3)$. Practically, we chose $k = 5$ as a good compromise between the accuracy and computational complexity for all data sets. That is, the practical complexity of the STC algorithm is only $O(n^2)$.

Authors' contributions

Both authors participated in the design of the study and performed the analysis. LSV implemented the algorithms. Both authors wrote and approved the final manuscript.

Acknowledgements

We would like to express special thanks to Heiko Schmidt for his technical support. We thank Gunter Weiss, Ingo Ebersberger, Tanja Gesell and Jutta Buschbom for carefully reading the manuscript. We acknowledge the use of supercomputing resources of the ZAM/NIC at the Research Center Jülich. We thank three anonymous referees for helpful comments.

References

1. Saitou N, Nei M: **The Neighbor – joining Method: A New Method for Reconstructing Phylogenetic Trees.** *Mol Biol Evol* 1987, **4:**406-425.
2. Strimmer K, von Haeseler A: **Quartet Puzzling: A Quartet Maximum – Likelihood Method for Reconstructing Tree Topologies.** *Mol Biol Evol* 1996, **13:**964-969.
3. Gascuel O: **BIONJ: An Improved Version of the NJ Algorithm Based on a Simple Model of Sequence Data.** *Mol Biol Evol* 1997, **14:**685-695.
4. Huson DH, Nettles SM, Warnow TJ: **Disk-Covering, a Fast-Converging Method for Phylogenetic Reconstruction.** *J Comput Biol* 1999, **6:**369-386.
5. Bruno WJ, Socci ND, Halpern AL: **Weighted Neighbor Joining: A Likelihood Based-Approach to Distance-Based Phylogeny Reconstruction.** *Mol Biol Evol* 2000, **17:**189-197.
6. Desper R, Gascuel O: **Fast and Accurate Phylogeny Reconstruction Algorithms Based on the Minimum-Evolution Principle.** *J Comput Biol* 2002, **9:**687-706.
7. Csürös M: **Fast Recovery of Evolutionary Trees with Thousands of Nodes.** *J Comput Biol* 2002, **9:**277-297.
8. Guindon S, Gascuel O: **A Simple, Fast and Accurate Algorithm to Estimate Large Phylogenies by Maximum Likelihood.** *Syst Biol* 2003, **52:**696-704.
9. Stamatakis A, Ludwig T, Meier H: **RAxML-III: A fast program for maximum likelihood-based inference of large phylogenetic trees.** *Bioinformatics* 2005, **21:**456-463.
10. Vinh LS, von Haeseler A: **IQPNNI: Moving fast through tree space and stopping in time.** *Mol Biol Evol* 2004, **21:**1565-1571.
11. Vinh LS, Schmidt HA, von Haeseler A: **PhyNav: A Novel Approach to Reconstruct Large Phylogenies.** In *Proceedings of the 28th Annual German Classification Society Conference (GfKl 2004)* Dortmund, Germany; 2004 in press.
12. Charleston MA: **Hitch-Hiking: A Parallel Heuristic Search Strategy, Applied to the Phylogeny Problem.** *J Comput Biol* 2001, **8:**79-91.
13. Brauer MJ, Holder MT, Dries LA, Zwickl DJ, Lewis PO, Hillis DM: **Genetic Algorithms and Parallel Processing in Maximum-Likelihood Phylogeny Inference.** *Mol Biol Evol* 2002, **19:**1717-1726.
14. Schmidt HA, Strimmer K, Vingron M, von Haeseler A: **TREE-PUZZLE: Maximum likelihood phylogenetic analysis using quartets and parallel computing.** *Bioinformatics* 2002, **18:**502-504.
15. Schmidt HA, von Haeseler A: **Maximum Likelihood Analysis Using TREE-PUZZLE.** In *Current Protocols in Bioinformatics* Edited by: Baxevanis AD, Davison DB, Page RDM, Stormo G, Stein L. New York, USA: Wiley and Sons; 2003:6.6.1-6.6.25.
16. Cavalli-Sforza L, Edwards AWF: **Phylogenetic analysis: Models and estimation procedures.** *Am J Hum Genet* 1967, **19:**233-257.
17. Fitch W, Margoliash E: **Construction of Phylogenetic trees.** *Science* 1967, **155:**279-284.
18. Hartigan AJ: *Clustering Algorithms* John Wiley and Sons, Inc; 1975.
19. Farris J: **On the phenetic approach to vertebrate classification.** 1977, **17:**823-850.
20. Klotz NKRB LC, Mitchell RM: **Calculation of evolutionary trees from sequence data.** *Proc Natl Acad Sci USA* 1979, **76:**4516-4520.
21. Li WH: **Simple method for constructing phylogenetic trees from distance matrices.** *Proc Natl Acad Sci USA* 1981, **78:**1085-1089.
22. Rambaut A, Crassly NC: **Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees.** *Comput Appl Biosci* 1997, **13:**235-238.
23. Kimura M: **A Simple Method for Estimating Evolutionary Rates of Base Substitutions through Comparative Studies of Nucleotide Sequences.** *J Mol Evol* 1980, **16:**111-120.
24. Harding EF: **The probabilities of rooted tree-shapes generated by random bifurcation.** *Adv Appl Prob* 1971, **3:**44-77.
25. Swofford DL, Olsen GJ, Waddell PJ, Hillis DM: **Phylogeny Reconstruction.** In *Molecular Systematics* 2nd edition. Edited by: Hillis DM, Moritz C, Mable BK. Sunderland, Massachusetts: Sinauer Associates; 1996:407-514.
26. Felsenstein J: **PHYLIP (Phylogeny Inference Package) version 3.5c.** *Department of Genetics, University of Washington, Seattle* 1993 [http://evolution.genetics.washington.edu/phylip.html].
27. Robinson DR, Foulds LR: **Comparison of phylogenetic trees.** *Mathematical Biosciences* 1981, **53:**131-147.
28. Buneman P: **The recovery of trees from measures of dissimilarity.** In *Mathematics in the archaeological and historical sciences* Edited by: Hodson, Lendall, Tautu. Edinburgh: Edinburgh university press; 1971.
29. Semple C, Steel M: *Phylogenetics* OXFORD univerity press; 2003.
30. Felsenstein J: *Inferring Phylogenies* Sunderland, Massachusetts: Sinauer Associates; 2004.
31. Aho AV, Hopcroft JE, Ullman JD: *The Design and Analysis of Computer Algorithms* Addison-Wesley Publishing Company; 1974.