

Pivotal Neighbor Joining Algorithms for Inferring Phylogenies via LCA-Distances

Ilan Gronau and Shlomo Moran

May 8, 2006

Abstract

Reconstructing phylogenetic trees efficiently and accurately from distance estimates is an ongoing challenge in computational biology from both practical and theoretical considerations. We study algorithms which are based on a characterization of weighted trees by distances to LCAs (*Least Common Ancestors*). This characterization combines the theoretical advantages of ultrametrics, with the practical advantages of neighbor joining algorithms.

A simple and natural neighbor joining criterion based on the above characterization is used to provide a family of consistent neighbor-joining algorithms which are simpler and more efficient than Saitou&Nei's NJ. A large subclass of this family is shown to be optimal under Atteson's robustness criterion for reconstruction of 'sufficiently long' edges; in this respect it outperforms NJ. A specific algorithm in this subclass is shown to provide a simpler version of the known 3-approximation algorithm of an arbitrary metric by an additive metric.

Our neighbor joining algorithms are pivotal, in the sense that when the input is not consistent with some tree, the output tree may depend on a root-taxon selected by the algorithm. We present experimental results for two variants of our algorithm on simulated data generated according to a well accepted evolutionary model. These experiments indicate that for the right selection of the root, the tree returned by our algorithm is likely to be topologically closer to the true tree than the one returned by NJ. The experimental results also indicate that selecting a root-taxon closer to the origin of evolution is likely to produce a more accurate tree. An interesting phenomenon demonstrated by our results is that in this evolutionary model, trees which best approximate the input distances are usually not the trees which best approximate the correct topology.

1 Introduction

Phylogenetic reconstruction methods attempt to find the evolutionary history of a given set of species (taxa). This history is usually described by an edge-weighted tree, where edges correspond to different branches of evolution, and

the weight of an edge corresponds to the amount of evolutionary change on that particular branch. To avoid ambiguity, it is usually assumed that internal edges have strictly positive weights. Distance-based phylogenetic reconstruction methods try to find this tree using estimates on distances along the tree; each such distance corresponds to the total weight of a path in the tree. Most known methods construct trees from pairwise-distance matrices, i.e. estimates on weights of paths connecting taxon-pairs.

Distance-based methods typically deal with two scenarios: accurate data and noisy data. In the first scenario, we assume that there is a unique tree which is consistent with the input. Algorithms which guarantee accurate reconstruction of this tree are said to be *consistent*. In the second (more realistic) scenario, there may be no tree consistent with the input. In such a case, the goal is to reconstruct a tree fitting the input in some way.

A distance metric consistent with some tree is called *additive* [6]. It is known that such a metric defines a unique tree topology (with strictly positive internal edge weights). After the introduction of this notion in [6], numerous algorithms were proposed, which reconstruct the tree given its additive metric [6, 33, 36]. One class of such algorithms reduces the construction of trees from additive distances to construction of ultrametrics. This reduction (also known as the *Farris transform* [16]) implies $O(n^2)$ algorithms for constructing trees from additive metrics, and is useful for obtaining various properties of additive metrics (e.g. [20, 1]).

Another common technique for constructing phylogenetic trees, which usually has larger complexity but proves to be reliable in practice, is the neighbor joining scheme. Neighbor-joining is an agglomerative clustering approach, in which at each stage two neighbor-elements are joined to one cluster; this new cluster then replaces them in the set of elements. This approach is used in hierarchical-clustering algorithms such as UPGMA [34]. The neighbor-joining scheme was first used to reconstruct trees from additive metrics by the AD-TREE algorithm [33]. Later, Saitou and Nei proposed the famous neighbor-joining algorithm [32, 22], which is commonly called NJ. Since then, the neighbor-joining scheme has been used in numerous algorithms (such as BIONJ [18], NJML [29] and Weighbor [4] to name a few) which were developed in hope of outperforming the original NJ algorithm on noisy (non-additive) input matrices. Another interesting approach was taken by [12], which presents a consistent algorithm which makes an exact simulation of NJ on near additive data, but is considerably faster than NJ. Other works (e.g. [9, 35]) present efficient algorithms which are more accurate than NJ on simulated data; these algorithms are usually more involved and harder to analyze.

When the input matrix is noisy, one approach is to return a tree whose implied metric is ‘close’ to the input metric under a certain distance norm (such as the ℓ_p norm). Finding the closest tree, however, was shown to be NP-hard for several such norms (ℓ_1, ℓ_2 in [8] and ℓ_∞ in [1]). A 3-approximation algorithm for ℓ_∞ is presented in [1]. This is the only constant-rate approximation known to us in this area, and is based on the reduction of additive distances to ultrametrics by the aforementioned Farris transform. A natural question in this context is

whether a good approximation of the noisy input matrix is likely to provide an accurate reconstruction of the original tree. Experiments reported in this paper indicate that this may very well not be the case. A complementary approach studied in [2, 28] defines the robustness of a phylogenetic reconstruction algorithm by limitations that must be imposed on the noisy input matrix, in order to guarantee correct reconstruction of the original tree or a specific edge in that tree.

Another approach in dealing with noisy input is to test the performance of tree reconstruction algorithms on actual data. Since real phylogenetic data are scarce, it is common to use data generated by simulation of evolution under certain accepted probabilistic models [17][14][27]. Notable are the large-scale simulated datasets which have been generated during the past years in [9][10][31] which we use in our experiments.

In this paper, we introduce a characterization of tree metrics by *LCA-distances*, i.e. distances from a selected root to the least common ancestors of all pairs of leaves. This characterization preserves many nice properties of ultrametrics, while avoiding the transformation of additive distances to ultrametric distances. A simple and natural neighbor joining criterion based on this characterization is used to provide a family of neighbor-joining algorithms - *Deepest Least Common Ancestor* (DLCA) algorithms - which are simpler and more efficient than Saitou&Nei's NJ. One specific DLCA variant ('maximal-value') is shown to return a tree-topology best fitting the estimated LCA-distances under several common norms. This tree is then used to provide a simpler version of the 3-approximation result from [1]. Moreover, a large subclass of DLCA algorithms is shown to have optimal *edge l_∞ -radius*, a robustness criterion introduced by Atteson in [2]. Informally, this is the maximum error bound, under which we can guarantee correct reconstruction of an edge of specified length (exact definition will be given in Section 4.3). In this respect, our algorithms outperform NJ, since it is shown in [2] that NJ does not have optimal edge l_∞ -radius. In fact, our algorithms are shown to be optimal under generalizations of Atteson's criteria defined in [28].

An inherent difference between previously studied neighbor joining algorithms and our DLCA algorithms is that the latter are *pivotal*, in the sense that when the input is not consistent with some tree, the output tree may depend on a root-taxon selected by the algorithm [7]. This allows for algorithms which may output more than one tree when the input matrix is noisy. We present experimental results of two DLCA variants on the aforementioned simulated datasets [9, 31]. These results indicate that for the right selection of the root, the tree returned by the algorithm is likely to be topologically closer to the true tree than the one returned by NJ. The simulations also indicate that selecting a root-taxon closer to the origin of evolution is likely to produce a more accurate tree. An interesting phenomenon observed in our experiments is that trees which best approximate the input distances are not the same trees which best approximate the correct topology.

The rest of the paper is organized as follows. The next subsection provides the needed notations and definitions. In Section 2 we present our character-

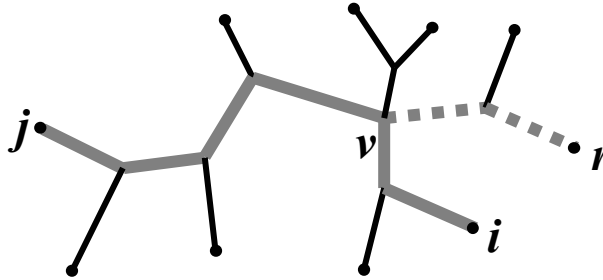


Figure 1: **Distance estimates in trees.** The path connecting taxa i, j is marked. $D_T(i, j)$ is the total weight of edges in this path. v is the center vertex of the 3-finger claw spanning r, i, j . The path connecting r, v is marked by a dashed line. $D_T(r; ij) = D_T(r, v)$ is the total weight of edges in this path.

ization theorem. The DLCA family of pivotal neighbor joining algorithms is presented in Section 3, and its theoretical properties are then analyzed in Section 4. In particular, one variant of these algorithms is shown to provide a simple proof of the 3-approximation result of [1], and analysis is given regarding Atteson’s robustness criteria. In Section 5 we briefly describe the probabilistic model of evolution used in [9, 31], and then present and analyze experimental results of our neighbor joining algorithms on these simulated datasets.

1.1 Definitions and Notations

Let S be a finite set (the set of taxa). A phylogenetic tree over S is an undirected weighted tree $T = (V, E, w : E \rightarrow \mathcal{R}^+ \cup \{0\})$ whose leaves are the elements of S . An edge is *external* if one of its endpoints is a leaf, and is *internal* otherwise; it is usually assumed that internal edges have strictly positive weight. Let r, i, j be three (not necessarily distinct) vertices in a tree T . $D_T(i, j)$, the distance in T between i and j , is the length of the path connecting i and j in T . Similarly, $D_T(r; ij)$ is the length of the path connecting r and the center vertex of the 3-finger claw spanning r, i, j (see Fig. 1); when T is rooted at r , this center vertex is the least common ancestor of i and j , (note that $D_T(r; ii) = D_T(r, i)$). A matrix over S is a square matrix A whose rows and columns are indexed by the elements of S . For a subset $S' \subseteq S$, $A(S')$ denotes the principal submatrix of A induced by the indices in S' . For matrices A, B over S , $A \leq B$ means that $\forall i, j : A(i, j) \leq B(i, j)$. All matrices referred to in this paper are assumed to be symmetric.

2 Characterization of Trees Using LCA-Distances

Our characterization is mostly based on the concept of an *LCA-matrix*, which will be shown to be equivalent to a representation of a weighted tree by distances

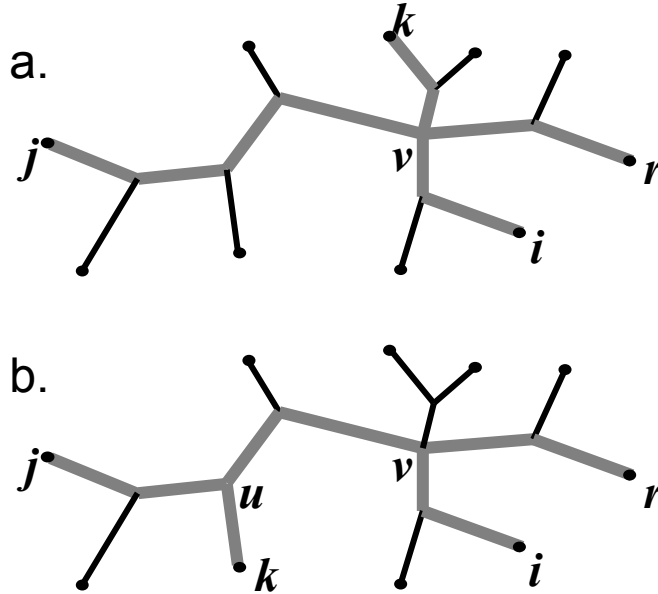


Figure 2: **The 3-point condition for LCA-distances.** Observe the subtree spanning r, i, j, k (marked edges). **a)** If its topology is a star (4-finger claw), with center-vertex v , then $D_T(r; ij) = D_T(r; ik) = D_T(r; jk) = D_T(r, v)$. **b)** Otherwise, w.l.o.g. i is paired up with r as illustrated, and $D_T(r; ij) = D_T(r; ik) = D_T(r, v) < D_T(r, u) = D_T(r; jk)$.

from a root-taxon to the least common ancestors of all pairs of leaves. For a given edge-weighted tree T over a set of taxa S and a taxon $r \in S$, the LCA-matrix of T from r , LCA_T^r , is a matrix over $S \setminus \{r\}$ defined by $LCA_T^r(i, j) = D_T(r; ij)$. The properties of such matrices are given in the following definition:

Definition 2.1. A symmetric non-negative matrix L over a set S is said to be an LCA-matrix iff it satisfies the following properties:

1. for all taxa $i \in S$, $L(i, i) = \max_{j \in S} L(i, j)$
2. For every triplet of distinct taxa (i, j, k) in S , $L(i, j) \geq \min\{L(i, k), L(j, k)\}$ (this property will be termed the 3-point condition).

Matrices satisfying the above 3-point condition are referred to in [20] as *min-ultrametrics*. This condition can be phrased as follows: *In every three entries of L of the form $\{L(i, j), L(i, k), L(j, k)\}$, the minimal value must appear at least twice.* While min-ultrametrics are not metrics (e.g, they do not satisfy the triangle inequality), they share many nice properties of *ultrametric distances*, with the additional advantage that together with Property 1, they enable representation of general tree metrics (rather than only ultrametrics).

Theorem 2.2. *A symmetric non-negative matrix L over a set of taxa S is an LCA-matrix iff there exists a weighted tree T over the expanded set of taxa $S \cup \{r\}$ s.t. $L = LCA_T^r$, i.e. $\forall i, j \in S, D_T(r; ij) = L(i, j)$.*

Proof. \Leftarrow Suppose that T is a weighted tree over the taxon-set $S \cup \{r\}$, and let $L = LCA_T^r$. It is clear that $\forall i, j \in S : D_T(r, i) \geq D_T(r; ij)$, which implies Property 1 of the definition. Now observe the subtree spanned by r, i, j, k . If its topology is a star, then $L(i, j) = L(i, k) = L(j, k)$, and the minimum value appears in $\{L(i, j), L(i, k), L(j, k)\}$ three times. If i is paired up with r in this quartet then $L(i, j) = L(i, k) < L(j, k)$ (see Fig. 2), and the minimum value appears twice. The same can be argued for the other two possible topologies of this subtree, proving Property 2.

\Rightarrow To prove the other direction, suppose L satisfies both conditions. We will use the following lemma, which follows immediately from the 3-point condition:

Lemma 2.3. *Let L be an LCA-matrix over S , and let \bar{i}, \bar{j} be two distinct elements of S s.t. $\forall k \neq \bar{i}, \bar{j} : L(\bar{i}, \bar{j}) \geq \max\{L(\bar{i}, k), L(\bar{j}, k)\}$. Then $\forall k \neq \bar{i}, \bar{j} : L(\bar{i}, k) = L(\bar{j}, k)$.*

Using Lemma 2.3 above we prove by induction on the number of taxa, that there is a tree T over $S \cup \{r\}$ satisfying the theorem. If $|S| = 1$ then L is a scalar $[w]$, and T consists of a single edge of weight w . If $|S| > 1$, let $L(\bar{i}, \bar{j}) = \max_{i \neq j} \{L(i, j)\}$ be a maximal off-diagonal entry of L . Clearly, $L(\bar{i}, \bar{j})$ satisfies the assumption of Lemma 2.3. Now, let v be a new vertex, and L' be the symmetric matrix over $S' = S \setminus \{\bar{i}, \bar{j}\} \cup \{v\}$, defined as follows:

$$\begin{aligned} L'(k, l) &= L(k, l) & k, l \neq v \\ L'(k, v) &= L(k, \bar{i}) & k \neq v \\ L'(v, v) &= L(\bar{i}, \bar{j}) \end{aligned}$$

Since all the entries of L' except $L'(v, v)$ are identical to the corresponding entries of $L(S \setminus \{\bar{j}\})$ (where index v in L' corresponds to index \bar{i} of L), Property 2 of Definition 2.1 holds for L' as it holds for L . For the same reason Property 1 holds for all indices in $S' \setminus \{v\}$. Property 1 holds for v as well, since $L'(v, v) = L(\bar{i}, \bar{j}) \geq \max_{k \in S \setminus \{\bar{i}, \bar{j}\}} L(\bar{i}, k) = \max_{k \in S' \setminus \{v\}} L'(v, k)$.

By the induction hypothesis, there is a tree T' over $S' \cup \{r\}$ s.t. for all $i, j \in S'$, $D_{T'}(r; ij) = L'(i, j)$. Let T be the tree obtained from T' by making v an internal vertex with two daughters \bar{i}, \bar{j} , where $w(v, \bar{i}) = L(\bar{i}, \bar{i}) - L(\bar{i}, \bar{j}) \geq 0$ and $w(v, \bar{j}) = L(\bar{j}, \bar{j}) - L(\bar{i}, \bar{j}) \geq 0$. It remains to show that $\forall i, j \in S : L(i, j) = D_T(r; ij)$. For all $i, j \in S \setminus \{\bar{i}, \bar{j}\}$ this holds by the induction hypothesis and the fact that $L(i, j) = L'(i, j)$. For $k \neq \bar{i}, \bar{j}$ we have that:

$$D_T(r; \bar{j}k) = D_T(r; \bar{i}k) = D_{T'}(r; vk) = L'(v, k) = L(\bar{i}, k) = L(\bar{j}, k).$$

We are left to prove the equality for the entries $(\bar{i}, \bar{j}), (\bar{i}, \bar{i}), (\bar{j}, \bar{j})$. By the definition of T' we have that $D_T(r; \bar{i}\bar{j}) = D_{T'}(r, v) = L'(v, v) = L(\bar{i}, \bar{j})$. Finally, $D_T(r, \bar{i}) = D_{T'}(r, v) + w(v, \bar{i}) = L(\bar{i}, \bar{i})$ by the definition of $w(v, \bar{i})$, and by a similar argument $D_T(r, \bar{j}) = L(\bar{j}, \bar{j})$. \square

Note that this theorem, as currently phrased, does not guarantee that each LCA-matrix has a **unique** tree representing it. Nevertheless, the tree constructed by the above proof can be shown to be the unique tree consistent with the input LCA-matrix. As implied by the proof, this tree can be constructed in $O(n^2)$ time: Start by computing the maximal off-diagonal entry for each row in L , and then proceed via $n - 1$ iterations, where at each iteration a neighboring pair \bar{i}, \bar{j} is chosen and connected to a common parent v , and the matrix L is reduced as follows:

1. Find a maximal off-diagonal entry $L(\bar{i}, \bar{j})$, by scanning the n row-maxima
2. Reduce the matrix L to L' by replacing rows \bar{i}, \bar{j} with row v as described.
3. Recompute the maximal off-diagonal entries for each row in L' , and set $L \leftarrow L'$.

Observe that computing the maximal off-diagonal entry for a row $k \neq v$ in L' is doable in constant time: Let $L(k, i)$ be the maximal off diagonal entry of row k in L . Then the maximal off diagonal entry of row k in L' is $L'(k, i)$ if $i \neq \bar{i}, \bar{j}$, and $L'(k, v)$ otherwise. Thus each iteration requires $O(n)$ time.

3 The DLCA Algorithms

Since in practice, we are rarely able to obtain accurate distance estimates, it is crucial that reconstruction algorithms produce meaningful output even when the input is not consistent with any tree metric. The algorithm implied by the proof of Theorem 2.2 is easily adjusted for noisy input matrices. In fact, it fits in the frame of the ‘neighbor-joining’ approach, which is widely used for reconstructing phylogenies from noisy data. Neighbor-joining algorithms consist of a main loop which has three stages:

1. **Neighbor selection:** Select a pair of taxa i, j optimizing some criterion.
2. **Reduction:** Replace the taxon-pair with a new vertex v in the taxon-set. Define distances from v to all other taxa, and recursively solve the problem on this smaller set.
3. **Neighbor connection:** In the returned solution, make v an internal vertex and connect i and j to v with two edges of some defined length.

The number of taxa is reduced by one in each recursive call, until the stopping condition is met. This process eventually yields a full-binary tree rooted at r . To obtain a canonical undirected version of the tree, we contract all zero-weight internal edges. Our algorithm can be presented in the neighbor joining scheme as follows:

Deepest LCA Neighbor Joining (DLCA):

Input: A symmetric nonnegative matrix L .

1. **Stopping condition:** If $L = [w]$ return a tree consisting of a single edge of weight w , rooted at r .
2. **Neighbor selection:** Select a pair of distinct taxa i, j , s.t. $L(i, j)$ is a maximal off-diagonal value in L .
3. **Reduction:** Remove i, j from the taxon-set and add v . Set $L(v, v) \leftarrow L(i, j)$, and $\forall k \neq v : L(v, k) \leftarrow \frac{1}{2}(L(i, k) + L(j, k))$. Recursively call DLCA on the resulting matrix.
4. **Neighbor connection:** In the returned tree, add i and j as daughters of v , with edge-weights: $w(v, i) = \max\{0, L(i, i) - L(i, j)\}$ and $w(v, j) = \max\{0, L(j, j) - L(i, j)\}$.

Notice that if L is an LCA-matrix, then by Lemma 2.3 $\forall k \neq i, j : L(i, k) = L(j, k)$ in step 2 of the algorithm, thus follows the equivalence of this algorithm to the one described in the proof of Theorem 2.2. Hence we can claim:

Theorem 3.1. *The DLCA algorithm is consistent. I.e., if the input matrix L is LCA_T^r , for some edge-weighted tree T and a taxon r of that tree, then the DLCA algorithm returns T .*

The observation used by the algorithm indicating that a taxon-pair with the deepest *LCA* are neighbors, is well known and was used in the past (see e.g. [35]). It is the natural connection between LCA-distances and ultrametrics that provides many theoretical and experimental advantages to this approach. In ultrametric trees, the taxon-pair closest to each other are neighbors. This is the basis of many ultrametric reconstruction algorithms such as UPGMA. In such trees, pairwise distances inversely correlate with the depth of the *LCA*, so the taxon-pair closest to each other is the taxon-pair with **deepest LCA**. However, when considering distances to *LCA*'s, we no longer need to assume an ultrametric tree. We can therefore use this neighboring criterion to reconstruct general trees. In general, *any algorithm which reconstructs ultrametric trees from pairwise distances can be converted to an algorithm which reconstructs general trees from LCA-distances*. The DLCA algorithm as described above can be viewed as such a conversion of the WPGMA algorithm (a known variant of UPGMA). Note that this algorithm can be generalized in two points without compromising its consistency:

- **Neighbor selection:** Rather than selecting the maximal off-diagonal entry, we can select any taxon pair satisfying the condition of Lemma 2.3. In fact, this criterion gives us a simple method for pinpointing **all** neighbor-pairs in T (which do not include r). We do not have to choose the ‘deepest’ pair in the tree, as suggested by the original description of DLCA. In this paper we confine ourselves to the original, deepest pair selection.
- **Reduction:** Since for the selected neighbors i, j it holds that $\forall k \neq i, j : LCA_T^r(i, k) = LCA_T^r(j, k)$ (by Lemma 2.3), we can set $L(v, k)$ to any

affine combination of $L(i, k)$ and $L(j, k)$. In other words, a **consistent** reduction formula is of the form $L(v, k) \leftarrow \alpha L(i, k) + (1 - \alpha)L(j, k)$, where the value of α may depend on i, j, k . We will mainly be interested in two specific variants:

1. The original (mid-point) reduction: $L(v, k) \leftarrow \frac{1}{2}(L(i, k) + L(j, k))$
2. The maximal-value reduction: $L(v, k) \leftarrow \max\{L(i, k), L(j, k)\}$

As previously mentioned, the ‘mid-point’ variant of DLCA relates to the WPGMA algorithm for ultrametrics. Similarly, the ‘maximal-value’ variant (discussed in further detail in Subsection 4.1) relates to an algorithm proposed in [23, 15]. A variant of the DLCA algorithm which similarly relates to UPGMA would use a weighted average in the reduction step; i.e: $L(v, k) \leftarrow \alpha L(i, k) + (1 - \alpha)L(j, k)$, where $\alpha, 1 - \alpha$ correlate with the number of offspring-taxa of i, j respectively.

The running time of the algorithm may depend on the reduction stage. Specifically, it remains $O(n^2)$ if $L(v, k)$ is set to $\max\{L(i, k), L(j, k)\}$ (*maximal-value*), since this enables simple updating of the maximal off-diagonal values as previously described. However, in other cases we may have to sort the entries of each row, therefore adding a factor of $\log(n)$ to the complexity, which becomes $O(n^2 \log(n))$.

It is interesting to note that NJ’s selection criterion acts, in a sense, as a non-pivotal version of DLCA: given a pairwise distance matrix consistent with some tree T , it selects a pair of taxa which maximizes the sum $\sum_{r \neq i, j} d_T(r; ij) + d(i, j)$. When all external edges of T have same weights (or when these weights are ignored), it selects the pair of taxa with the average deepest *LCA* (the addition of the term $d(i, j)$ is necessary to make the criterion consistent - see e.g. [5]).

3.1 Reconstruction from Pairwise Distances

The DLCA algorithm can be used to reconstruct trees from pairwise distances, by transforming pairwise distances over a set S into estimates of *LCA*-distances as follows:

Definition 3.2. *Given a distance matrix D over a set of taxa S and a taxon $r \in S$, $L = LCA(D, r)$ is the matrix over $S \setminus \{r\}$ defined by $L(i, j) = \frac{1}{2}(D(r, i) + D(r, j) - D(i, j))$.*

We can now execute any of the variants of the DLCA algorithm on $LCA(D, r)$, to get a tree. In such a case we say the algorithm is *run on D from taxon r* . If D is an additive distance matrix consistent with some tree T , then $LCA_T^r = LCA(D, r)$, and the DLCA algorithm is guaranteed by previous discussion to return T (for all choices of root-taxon r).

This method can also be generalized to choose other points (not taxa) as roots. In [7], Farach and Cohen present the Dual-Pivot (DP) method, which allows a pivotal choice of the mid-point on a path connecting an arbitrary taxon-pair. This method is applied to the 3-approximation algorithm from [1]. The

approach we present here is slightly more general, as it allows the choice of **any** point located on such a path, and not only the mid-point.

Lemma 3.3. *Let T be an edge-weighted tree, and a, b a taxon-pair in T . Given a point x on the path connecting a, b , (specified by its distances from a and b), we have for all taxa i :*

$$D_T(x, i) = \begin{cases} D_T(a, i) - D_T(a, x) & | D_T(a; bi) > D_T(a, x); \\ D_T(b, i) - D_T(b, x) & | \text{Otherwise.} \end{cases}$$

Proof. Follows from a simple observation of the the 3-finger claw spanning a, b, i .

Note that since $D_T(a; bi) = \frac{1}{2}(D_T(a, b) + D_T(a, i) - D_T(b, i))$ this lemma can be used to calculate, given an additive metric, the distances from any such point x to all taxa. Once these distances are known, x acts as any other taxon, and our additive metric is expanded. The formula in Definition 3.2 can now be used to provide the LCA-distances from x , and the DLCA algorithm can be applied to reconstruct the tree. Note that this approach significantly increases the number of possible choices for a root.

The potential advantage of this method lies when applied to non-additive distance matrices. Whereas when the distance matrix is additive, the DLCA algorithm is guaranteed to return the same tree (no matter which point is chosen for root), when it is non-additive, we may get different outputs for different choices of roots. By providing a controlled method to increase the number of possible pivotal choices, we potentially increase the number of outputs. This can be useful when you have a good method for selecting the ‘best’ tree out of the set of outputs, as done e.g. in [7] (see also discussion in Subsection 5.3).

4 Performance of the DLCA Algorithm on Noisy Distance Estimates

In this section we present some performance guarantees of the DLCA algorithm on noisy input distances. In Subsection 4.1, the performance of the aforementioned ‘maximal-value’ variant of DLCA is analyzed. Specifically, we show that it yields a tree whose LCA-matrix is the unique *dominant* LCA-matrix of the input matrix A (see Definition 4.1). We then show in Subsection 4.2 how this tree can be used to provide a simpler version of the known 3-approximation algorithm of an arbitrary metric by an additive metric under the ℓ_∞ norm [1]. Finally, in Subsection 4.3, we analyze the robustness of a large class of DLCA algorithms under criteria defined in [2][28].

4.1 The Dominant LCA-Matrix

Our definitions and analysis in this subsection are similar to those used in [23, 15] for ultrametrics. Let $\mathcal{U}(A)$ be the set of all LCA-matrices which are larger or equal to an input matrix A .

Definition 4.1. An LCA-matrix L is said to be dominant to a matrix A if it is a minimal element in $\mathcal{U}(A)$, i.e. if $L \in \mathcal{U}(A)$ and for every LCA-matrix $L' \in \mathcal{U}(A)$, if $L' \leq L$ then $L' = L$.

$\mathcal{U}(A)$ is compact and bounded from below (by A), hence it contains a minimal element. It is also easy to see that if the matrices L_1, L_2 are in $\mathcal{U}(A)$, then the matrix L defined by $L(i, j) = \min\{L_1(i, j), L_2(i, j)\}$ is also in $\mathcal{U}(A)$. This implies that $\mathcal{U}(A)$ contains a *unique* minimal element - the unique LCA-matrix dominant to A , denoted by L^{dom} . The uniqueness of L^{dom} implies that it is closest to A among all LCA-matrices in $\mathcal{U}(A)$, under any distance-metric d which satisfies the following intuitive requirement: if $A \leq A_1 \leq A_2$ then $d(A, A_1) \leq d(A, A_2)$ (this includes, for instance, all the ℓ_p norms). It is also easy to see that that L^{dom} is an LCA-matrix closest to A under the *maximal distortion* measure defined by: $MaxDist(A, L) = \max_{i,j} \frac{L(i,j)}{A(i,j)} \cdot \max_{i,j} \frac{A(i,j)}{L(i,j)}$ [3].

The following lemma states another nice property of the unique dominant LCA matrix, which will be used later:

Lemma 4.2. Let L^{dom} be the dominant LCA matrix of a symmetric matrix A .

$$\text{Then:} \quad \forall i : L^{dom}(i, i) = \max_j \{A(i, j)\}.$$

Proof. Let $m_i = \max_j \{A(i, j)\}$, and let L be the matrix over S defined as follows:

$$L(i, j) = \min\{L^{dom}(i, j), m_i, m_j\}.$$

Since $\forall i, j : A(i, j) \leq \min\{m_i, m_j\}$, we have that $A \leq L \leq L^{dom}$. Moreover, $L(i, i) = m_i$ (since $L^{dom}(i, i) = \max_j \{L^{dom}(i, j)\} \geq \max_j \{A(i, j)\} = m_i$). Thus, if we show that L is an LCA-matrix, then $L = L^{dom}$ (by the dominance of L^{dom}) and the lemma follows.

Property 1 of LCA-matrices (see Definition 2.1) holds for L since $\forall i, j : \min\{L^{dom}(i, i), m_i\} \geq \min\{L^{dom}(i, j), m_i, m_j\}$. To see that property 2 holds as well, consider an arbitrary triplet i, j, k . Let

$$m = \min\{L^{dom}(i, j), L^{dom}(i, k), L^{dom}(j, k)\},$$

and assume w.l.o.g. that $m_i \leq m_j \leq m_k$. If $m \leq m_i$, the two minimal entries in $\{L(i, j), L(i, k), L(j, k)\}$ are as in L^{dom} (and equal to m). Otherwise, $L(i, j) = L(i, k) = m_i \leq \min\{m_j, L^{dom}(j, k)\} = L(j, k)$. In both cases the two minimal entries in $\{L(i, j), L(i, k), L(j, k)\}$ hold the same value. \square

Next we show how to transform L^{dom} into an LCA-matrix closest to A under the ℓ_∞ norm (out of all LCA-matrices, not just these in $\mathcal{U}(A)$).

Lemma 4.3. Given a matrix A , and its unique dominant LCA-matrix L^{dom} , denote by $\varepsilon = \|A, L^{dom}\|_\infty = \max_{i,j} \{L^{dom}(i, j) - A(i, j)\}$. Then L^∞ defined by $L^\infty(i, j) = \max\{L^{dom}(i, j) - \frac{\varepsilon}{2}, 0\}$ is an LCA-matrix closest to A under the ℓ_∞ norm.

Proof. First, it is easy to see that L^∞ is an LCA-matrix, and that $\|A, L^\infty\|_\infty = \frac{\varepsilon}{2}$. Now, given any LCA-matrix L , define by $\varepsilon_L = \|A, L\|_\infty$. We need to prove that $\frac{\varepsilon}{2} \leq \varepsilon_L$. Denote by L' the matrix defined as follows: $L'(i, j) = L(i, j) + \varepsilon_L$. It is again easy to verify that $A \leq L'$, and that $\|A, L'\|_\infty \leq 2\varepsilon_L$. Now, since L' is an LCA-matrix, and L^{dom} is the dominant LCA-matrix of A , we have $A \leq L \leq L'$. This means that $\varepsilon = \|A, L^{dom}\|_\infty \leq \|A, L'\|_\infty \leq 2\varepsilon_L$. \square

Notice that the transformation implied by Lemma 4.3 does not change the topology of the tree corresponding to the LCA-matrix, with the exception that it may set some edge-weights to zero. This means that for each matrix A there is a unique tree topology - the one defined by the unique dominant LCA-matrix - which minimizes various distance-measures to A (e.g. ℓ_∞ and $MaxDist$). Note that edge weights may vary in the different trees according to the specific distance-measure. We conclude this discussion by proving that this tree is the one constructed by the maximal-value variant of the DLCA algorithm.

Theorem 4.4. *Let A be a symmetric matrix over S , and let T be the tree over $S \cup \{r\}$ reconstructed from A by the **DLCA** algorithm with maximal-value reduction. Then LCA_T^r is the unique dominant LCA-matrix of A .*

Proof. By induction on $|S|$. If $|S| = 1$, then $A = [w]$, T is a tree with one edge of weight w , and $LCA_T^r = [w] = A$. Assume now that $|S| > 1$, and let i, j be the taxon-pair chosen in step 2 of the DLCA algorithm. Denote by $S' = S \setminus \{i, j\} \cup v$ the reduced taxon-set, by A' the reduced matrix, and by T' the tree returned by the algorithm given A' as input. Let $L = LCA_T^r$ and $L' = LCA_{T'}^r$. By the induction hypothesis, L' is the unique dominant LCA-matrix of A' . We will use this to show that L is dominant to A .

First, we show that $L \geq A$ and $L(i, j) = A(i, j)$. By the induction hypothesis ($L' \geq A'$) and the maximal-value reduction of A to A' , we have:

$$\begin{aligned} \forall k, l \neq i, j : \quad & L(k, l) = L'(k, l) \geq A'(k, l) = A(k, l). \\ \forall k \neq i, j : \quad & L(k, i) = L(k, j) = L'(k, v) \geq A'(k, v) \geq A(k, i), A(k, j). \\ & L(i, j) = L'(v, v) \geq A'(v, v) = A(i, j). \end{aligned}$$

Observe that the neighbor-selection criterion and reduction formula guarantee that $A'(v, v) = \max_k \{A'(v, k)\}$. Since L' is dominant to A' , Lemma 4.2 implies that $L'(v, v) = A'(v, v)$, and the third inequality above turns into an equality ($L(i, j) = A(i, j)$).

We are left to prove that if M is an LCA-matrix and $A \leq M \leq L$, then $M = L$. Given such a matrix M , we use the equality $L(i, j) = A(i, j)$ and the fact that $\forall k \neq i, j : L(i, k) = L(j, k) \leq L(i, j)$, to show that for every $k \neq i, j$:

$$M(i, k) \leq L(i, k) \leq L(i, j) = A(i, j) \leq M(i, j), \quad \text{and similarly } M(j, k) \leq M(i, j).$$

Thus we have that $\forall k \neq i, j : M(i, j) \geq \max\{M(i, k), M(j, k)\}$. This implies, by Lemma 2.3, that $\forall k \neq i, j : M(i, k) = M(j, k)$. Hence the matrix M can be

reduced to a matrix M' over S' by replacing rows i, j by row v , as in the proof of Theorem 2.2. The matrix M' is an LCA-matrix and $A' \leq M' \leq L'$. From the induction hypothesis on L' we have that $M' = L'$, and this implies that $M = L$ by the following equalities:

$$\begin{aligned} \forall k, l \neq i, j : \quad & M(k, l) = M'(k, l) = L'(k, l) = L(k, l). \\ \forall k \neq i, j : \quad & M(k, i) = M(k, j) = M'(k, v) = L'(k, v) = L(k, i) = L(k, j). \\ & M(i, j) = M'(v, v) = L'(v, v) = L(i, j). \end{aligned}$$

□

4.2 3-Approximation of Distances by Additive Metrics

We now show how the ‘maximal-value’ variant of the DLCA algorithm can be used to get an $O(n^2)$ 3-approximation algorithm for the closest additive metric under the ℓ_∞ norm. This result was originally presented in [1] using the Farris transform, however, our formulation implies simpler analysis.

Given a metric (satisfying the triangle inequality) D over a taxon-set S , our algorithm acts as follows:

1. Choose some arbitrary taxon r , and calculate $L = LCA(D, r)$.
2. Find an LCA-matrix L^∞ closest to L under ℓ_∞ , which satisfies:
 $\forall i \in S, \quad L^\infty(i, i) = L(i, i)$.
3. Return the tree T^∞ , represented by L^∞ .

Stage 2 of this algorithm can be achieved as follows: First, L^{dom} - the dominant LCA-matrix of L , is constructed by running the ‘maximal-value’ variant of the DLCA algorithm (see Theorem 4.4). Notice that since D is a metric, L is nonnegative and $D(r, i) = L(i, i) = \max_j L(i, j)$ for every taxon i . Therefore, by Lemma 4.2 we have that $L^{dom}(i, i) = \max_j L(i, j) = L(i, i)$. Hence, when constructing L^∞ from L^{dom} , we invoke the transformation described in Lemma 4.3, except that $\frac{\varepsilon}{2}$ is not subtracted from the diagonal. This results in an LCA-matrix L^∞ , s.t. $\|L, L^\infty\|_\infty = \frac{\varepsilon}{2}$ and:

$$\forall i : D_{T^\infty}(r, i) = L^\infty(i, i) = L^{dom}(i, i) = L(i, i) = D(r, i). \quad (4.1)$$

The following theorem uses this equation in proving the approximation ratio of the algorithm:

Theorem 4.5. *Let D be a metric over a taxon-set S , and let T^∞ be the edge-weighted tree constructed from L^∞ in the above algorithm. Denote by D_{T^∞} the additive metric implied by T^∞ . Then for every additive metric D' :*

$$\|D, D_{T^\infty}\|_\infty \leq 3 \cdot \|D, D'\|_\infty$$

Proof. Denote by $L = LCA(D, r)$; the proof consists of two simple claims:

Claim 4.6. $\|D, D_{T^\infty}\|_\infty = 2 \cdot \|L, L^\infty\|_\infty$.

Proof. For an arbitrary taxon-pair i, j , we use the formula in Definition 3.2 and Equation 4.1 to show that:

$$\begin{aligned} D(i, j) - D_{T^\infty}(i, j) &= \\ (D(r, i) + D(r, j) - 2L(i, j)) - (D_{T^\infty}(r, i) + D_{T^\infty}(r, j) - 2L^\infty(i, j)) &= \\ 2(L^\infty(i, j) - L(i, j)) \end{aligned}$$

which implies that $\|D, D_{T^\infty}\|_\infty = 2 \cdot \|L, L^\infty\|_\infty$. □

Now, denote by T' the edge-weighted tree which realizes D' , and let $L' = LCA_{T'}^r$. Note that L' is an LCA-matrix due to Theorem 2.2.

Claim 4.7. $\|L, L'\|_\infty \leq \frac{3}{2} \cdot \|D, D'\|_\infty$.

Proof. The proof simply follows from the fact that $L'(i, j) = \frac{1}{2}(D'(r, i) + D'(r, j) - D'(i, j))$ and $L(i, j) = \frac{1}{2}(D(r, i) + D(r, j) - D(i, j))$. □

Now, since by definition $\|L, L^\infty\|_\infty \leq \|L, L'\|_\infty$, we have that:

$$\|D, D_{T^\infty}\|_\infty = 2 \cdot \|L, L^\infty\|_\infty \leq 2 \cdot \|L, L'\|_\infty \leq 3 \cdot \|D, D'\|_\infty$$

□

4.3 Optimal Robustness of the DLCA Algorithm

In this section, we give some reconstruction guarantees of the DLCA algorithm when run on a pairwise-distance matrices which are not additive. We consider robustness criteria which were originally applied in [13] and then formally defined and studied in [2], as well as additional criteria recently studied in [28]. In [2] Atteson defines the l_∞ -radius of a reconstruction algorithm \mathcal{A} as the maximal ε , for which \mathcal{A} is guaranteed to return a tree T , given a distance matrix D , s.t. $\|D, D_T\|_\infty < \varepsilon \cdot \min_{e \in T} \{w(e)\}$. It is shown in [2] that no algorithm has l_∞ -radius greater than $\frac{1}{2}$, and that most common algorithms (including NJ and its variants) have an optimal l_∞ -radius of $\frac{1}{2}$.

Atteson also defines the *edge l_∞ -radius* of an algorithm, which gives a reconstruction guarantee for sufficiently long edges. We say that algorithm \mathcal{A} has edge l_∞ -radius of ε if for each input matrix D and tree T , \mathcal{A} correctly reconstructs all edges in T of weight strictly greater than $\frac{1}{\varepsilon} \|D, D_T\|_\infty$. Correct reconstruction of an edge in T means that the reconstructed tree has an edge inducing the same split (partition) on the taxon-set. Notice that the edge l_∞ -radius of an algorithm is bounded from above by its l_∞ -radius. In [2] it is shown that NJ has edge l_∞ -radius no greater than $\frac{1}{4}$, and in [28] it is proven that the edge l_∞ -radius of NJ is exactly $\frac{1}{4}$. The result in [28] actually uses weaker assumptions on the input matrix than the ones made by Atteson's formulation. The following consistency criteria for a distance matrix D are adapted from these weaker assumptions.

- D is *consistent with quartet* $(ij : kl)$, if $D(i, j) + D(k, l) < \min\{D(i, k) + D(j, l), D(i, l) + D(j, k)\}$.
- D is $(P|Q)$ -*consistent*, for a partition $(P|Q)$ of the taxon-set, if D is consistent with all quartets $(ij : kl)$, s.t. $i, j \in P$ and $k, l \in Q$.
- D is *quartet-consistent* with some tree T if it is consistent with all quartets induced by T .

Lemmas 4.9 and 4.10 below show that given a $(P|Q)$ -consistent input matrix, a DLCA algorithm which uses only *conservative* reductions is guaranteed to reconstruct a tree containing an edge inducing the split $(P|Q)$; this, in turn, implies optimal l_∞ -radius and edge l_∞ -radius of $\frac{1}{2}$. A reduction step which replaces a pair of taxa i, j by their parent taxon v is said to be *conservative* if there is a constant $\alpha \in [0, 1]$ (which may depend on i, j) s.t:

$$\begin{array}{ll} \text{either} & \forall k \neq v : L(v, k) \leftarrow \alpha L(i, k) + (1 - \alpha) L(j, k), \\ \text{or} & \forall k \neq v : L(v, k) \leftarrow \alpha \max\{L(i, k), L(j, k)\} + (1 - \alpha) \min\{L(i, k), L(j, k)\}. \end{array}$$

Note that not all **consistent** reductions (as defined in Section 3) are **conservative** as well. However, most interesting reductions are conservative. For instance, the maximal value reduction satisfies the second rule (with $\alpha = 1$), whereas the mid-point reduction satisfies both rules (with $\alpha = \frac{1}{2}$). A DLCA algorithm is said to be *conservative* if it uses a conservative reduction. In the rest of the paper we confine our discussion to conservative DLCA algorithms.

Given a tree T rooted at r , and a vertex in that tree v , denote by $\mathcal{L}_r(v)$ the set of leaves of T , which are descendants of v . Such a set is called a *clade* of T . The following lemma characterizes subsets of taxa which are guaranteed to be clades of the tree reconstructed by the DLCA algorithm.

Definition 4.8. *Let L be a symmetric non-negative matrix over S . A proper subset $X \subset S$ is an LCA-cluster of L if it satisfies the following condition:*

$$\forall \{x, y\} \subseteq X, z \in S \setminus X : L(x, y) > L(x, z)$$

Lemma 4.9. *Let L be a symmetric non-negative matrix over S , and let T be the rooted tree returned by a conservative DLCA algorithm when run on L . Then every LCA-cluster of L is a clade of T .*

Proof. Let X be an LCA-cluster of L . We prove by induction on $|S|$ that X is a clade in T . This claim holds vacuously for $|S| = 1$, so assume that $|S| > 1$. If $|X| = 1$, then $X = \{x\}$ for some taxon x , and clearly $\{x\} = \mathcal{L}_r(x)$ is a clade in T . So we may assume that $|S| > |X| > 1$.

Let $i, j \in S$ be the taxon-pair selected by the algorithm. Since X is an LCA-cluster of S and $|X| > 1$, the maximality of $L(i, j)$ implies that either $\{i, j\} \subseteq X$, or $\{i, j\} \subseteq S \setminus X$. Denote by v the parent vertex of i, j , by $S' = S \setminus \{i, j\} \cup v$ the reduced set of taxa, and by L' the reduced matrix. Let $X' = X$ if $\{i, j\} \subseteq S \setminus X$

and $X' = X \setminus \{i, j\} \cup \{v\}$, otherwise. We prove now that X' is an LCA-cluster of L' , i.e:

$$\forall \{x, y\} \subseteq X', z \in S' \setminus X' : L'(x, y) > L'(x, z).$$

Let x, y, z as above be given. We distinguish between two cases:

- $\{i, j\} \subseteq S \setminus X \Rightarrow v \in S' \setminus X'$: If $z \neq v$ the claim follows directly from the assumption on X , so assume that $z = v$. We need to show that for an arbitrary pair $\{x, y\} \subseteq X'$ it holds that $L'(x, y) > L'(x, v)$. First, we note that $\{x, y\} \subseteq X$ as well; hence $L(x, y) > L(x, i), L(x, j)$ since X is an LCA-cluster of S . Now, $L'(x, y) = L(x, y)$ and the reduction step guarantees that $L'(x, v)$ is set to a value between $L(x, i)$ and $L(x, j)$. Therefore $L'(x, y) > L'(x, v)$.

- $\{i, j\} \subseteq X \Rightarrow v \in X'$: If $x, y \neq v$ the claim follows directly from the assumption on X . We are left to show that $\forall x \in X' \setminus \{v\}, z \notin X' : L'(x, v) > L'(x, z), L'(v, z)$.

Let x, z be as above. Since X is an LCA-cluster of L , we have $L(x, i), L(x, j) > L(x, z)$. Since $L'(x, v)$ is set to a value between $L(x, i)$ and $L(x, j)$, we have that $L'(x, v) > L'(x, z)$. We are left to prove that $L'(x, v) > L'(v, z)$. Since X is an LCA-cluster we have that $L(i, x) > L(i, z)$ and $L(j, x) > L(j, z)$. Assume first, that the conservative reduction satisfies the first rule, then:

$$L'(v, x) = \alpha L(i, x) + (1 - \alpha)L(j, x) > \alpha L(i, z) + (1 - \alpha)L(j, z) = L'(v, z).$$

A similar argument applies also when the reduction satisfies the second rule, using the fact that $\max\{L(i, x), L(j, x)\} > \max\{L(i, z), L(j, z)\}$ and $\min\{L(i, x), L(j, x)\} > \min\{L(i, z), L(j, z)\}$.

Now, since X' is an LCA-cluster of S' , we can use the induction hypothesis to get that in T' , the rooted tree returned by DLCA when run on L' , there is a vertex u , s.t. $\mathcal{L}_r(u) = X'$. The tree T is obtained from T' by adding i, j as two daughters of v . Therefore, in T we get $\mathcal{L}_r(u) = X$. \square

Given a pairwise distance matrix D , the following lemma characterizes subsets of taxa which are guaranteed to be LCA-clusters of $LCA(D, r)$.

Lemma 4.10. *Let D be a distance matrix over S which is $(P|Q)$ -consistent (for some partition $(P|Q)$ of S), and let r be a taxon in P . Then Q is an LCA-cluster of $LCA(D, r)$.*

Proof. Let $L = LCA(D, r)$. In order to show that Q is an LCA-cluster, we need to prove that for every $x, y \in Q, z \in P$ we have $L(x, y) > L(x, z)$. Since $r \in P$ and D is $(P|Q)$ -consistent, we have that $D(r, z) + D(x, y) < \min\{D(r, x) + D(y, z), D(r, y) + D(x, z)\}$. In particular, we get the following:

$$\begin{aligned} D(r, y) + D(x, z) &> D(r, z) + D(x, y) && \Rightarrow \\ D(r, y) - D(x, y) &> D(r, z) - D(x, z) && \Rightarrow \\ D(r, x) + D(r, y) - D(x, y) &> D(r, x) + D(r, z) - D(x, z) && \Rightarrow L(x, y) > L(x, z). \end{aligned}$$

\square

Lemmas 4.9 and 4.10 imply that if the input pairwise-distance matrix is $(P|Q)$ -consistent, then the tree returned by a conservative DLCA algorithm contains an edge inducing the split $(P|Q)$. This, in turn, implies the following result:

Theorem 4.11. *All conservative DLCA algorithms, when executed on a pairwise distance matrix from an arbitrary root-taxon r , have an optimal edge l_∞ -radius (and hence also optimal l_∞ radius) of $\frac{1}{2}$.*

Proof. Let T be an edge-weighted tree over taxon-set S , D a distance matrix over S , and e an edge in T s.t. $w(e) > 2\|D, D_T\|_\infty$. Denote by $(P|Q)$ the split induced by e , and assume w.l.o.g. that $r \in P$. It is easy to see that since $\|D, D_T\|_\infty < \frac{1}{2}w(e)$, we have that D is $(P|Q)$ -consistent. By Lemmas 4.9, 4.10 the tree returned by the DLCA algorithm when run on D contains an edge inducing the split $(P|Q)$. \square

We now wish to compare our results for DLCA algorithms with related results presented in [2, 28] for NJ. Regarding reconstruction of ‘long-edges’ (i.e. edge l_∞ -radius), we showed that DLCA is optimal and hence superior to NJ (whose edge l_∞ -radius is $\frac{1}{4}$). Regarding reconstruction of the entire tree, both algorithms have an optimal l_∞ -radius. However, our analysis implies that quartet consistency of the input matrix guarantees correct reconstruction of the tree by DLCA, whereas [28] gives an example where NJ does not reconstruct the correct tree from a quartet-consistent input matrix. In fact, [28] shows that in addition to quartet consistency, another condition termed ‘*quartet-additivity*’ is needed to guarantee correct reconstruction by NJ, and thus DLCA is superior to NJ in this respect as well.

Note: As shown Subsection 4.2, the maximal-value variant of the DLCA algorithm provides a 3-approximation algorithm which is equivalent to the one presented in [1]. Therefore, Theorem 4.11 demonstrates that the l_∞ -radius of this algorithm is $\frac{1}{2}$. We note that in [2] it is argued that its l_∞ -radius cannot be larger than $\frac{1}{6}$. This claim is based on an example given in [13], which presents a distance matrix D over 4 taxa as well as two trees with different topologies. One tree (T) satisfies $\|D, D_T\|_\infty = \frac{1}{6} \cdot \min_{e \in T} \{w(e)\}$, whereas the other (T') is shown to give a 3-approximation to D under l_∞ . We observe that this example only demonstrates that, *a-priori*, a 3-approximation algorithms for this problem is not guaranteed to have an l_∞ -radius larger than $\frac{1}{6}$. However, it does not exclude the possibility that such an algorithm may indeed have a larger l_∞ -radius, and hence it does not contradict our result.

5 Simulations and Results

When considering the DLCA algorithm until now, we did not discuss the initial step of choosing a root-taxon r from which to execute the algorithm. As mentioned earlier, when distance estimates are not consistent with some tree (as often is the case), the output may vary for different choices of a root. Moreover,

the reduction formula used by the algorithm may affect the output as well. As will be demonstrated by results in this section, these factors may bear crucial influence on the performance of the algorithm.

We use two types of simulated data: **random trees** (sampled according to the Yule-Harding distribution) and **model-trees**. Both datasets were downloaded from the LIRMM ‘Methods and Algorithms in Bioinformatics’ (MAB) website [19]. For model trees, we use six trees over 12 taxa as described in [9]. Three of these trees (A/B group: ‘AA’, ‘AB’ and ‘BB’) are ultrametric, and the other three (C/D group: ‘CC’, ‘CD’ and ‘DD’) are not (see Fig. 3). These trees are based on smaller model-trees (‘A’, ‘B’, ‘C’ and ‘D’) proposed earlier in [29]. Each tree was scaled in 4 different ways, according to different maximum pairwise divergence (MD), resulting in a total of 24 model-trees (see [31] for more detail). The ‘random-trees’ dataset was obtained in [9] as follows: $2 \times 2,000$ trees (over sets of 24 and 96 taxa) were generated using the stochastic speciation process described in [24], corresponding to the Yule-Harding distribution over trees [17, 14]. Edge-lengths were slightly modified in order to adjust deviation from molecular-clock. Each of the 4,000 tree was rescaled to obtain ‘slow’, ‘moderate’ and ‘fast’ rates of evolution (see details in [9]).

In both cases, SeqGen [30] was used to generate sequences, simulating the process of evolution along each tree according to the Kimura two parameter model [27] with transition/transversion ratio of 2.0. For each random tree, a single simulation was performed resulting in 500_b long sequences for all taxa. For each of the 24 different model-trees, $2 \times 1,000$ simulations were performed, resulting in 300_b and 600_b long sequences. For each such set of sequences we computed a pairwise distance-matrix using DNADIST from the PHYLIP (3.63) package [21]. We obtained estimates on LCA-distances from these distance matrices using the transformation stated in Definition 3.2. We note that [25, 31] suggest more accurate estimates can be obtained directly over ‘*maximum-likelihood taxon-triplets*’. However, our tests indicate that the difference between estimates obtained by both approaches on our datasets is not significant.

For each distance matrix, we hold the original (correct) tree from which it is originated, in order to test the accuracy of reconstruction. Accuracy is measured by the Robinson-Foulds (RF) distance [11] between the original and reconstructed topologies. Since both topologies are fully resolved, half the RF-distance is the number of edges of the original tree not appearing in the reconstructed topology. We term this number the *RF-score*. The RF-score of a reconstruction algorithm on a dataset is the average RF-score of all reconstructed topologies. Note that the smaller the RF-score, the better the algorithm is.

5.1 Results on Random Trees

When executing a DLCA algorithm on an $n \times n$ distance-matrix, n different trees can potentially be obtained. We first wish to see how good is the best tree obtained, how bad is the worst tree, and how the average-case behaves. We tested two conservative variants of DLCA, (*mid-point* and *maximum-value*), and compared their performances to Saitou&Nei’s NJ for reference. Summary

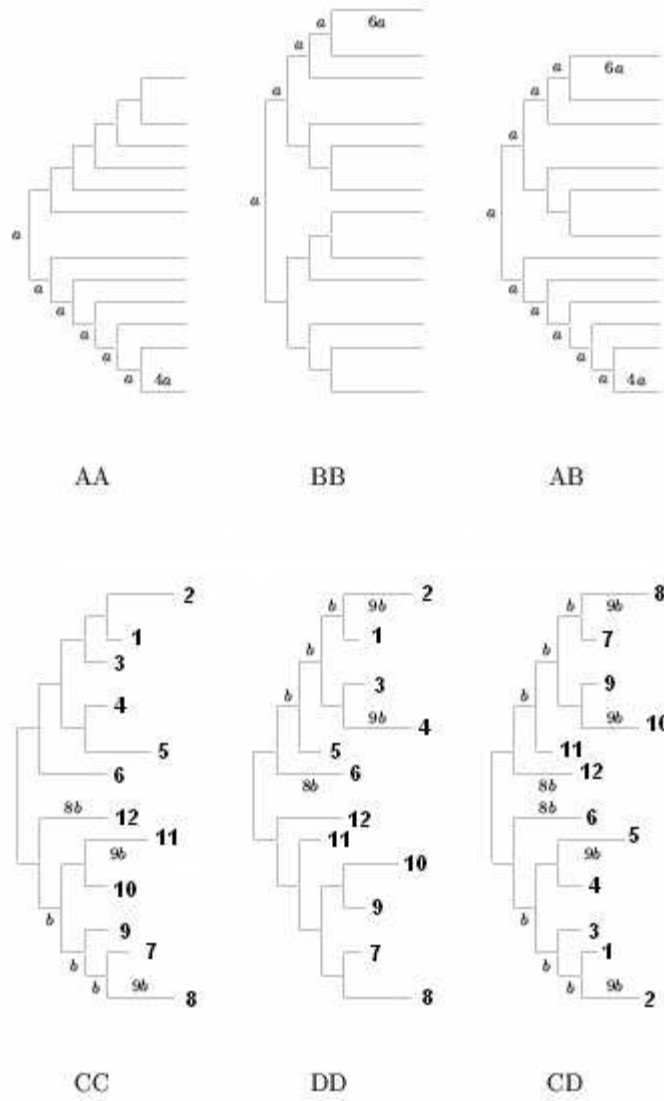


Figure 3: **Model trees.** Short branches are of uniform length (a for the A/B group and b for the C/D group). Lengths of long branches are specified. Note that trees in the A/B group are ultrametric, and trees in C/D are not. Taxa of trees in the C/D group are labelled by 1-12.

of these results appear in Table 1 and Fig. 4. In Table 1 we can see that change in the scale of the trees (rate of evolution) has little influence on the relative performance of the algorithms. They all seem to perform better when the rate of evolution is faster. This behavior is also observed in [9]. It is also notable that the ‘mid-point’ variant consistently outperforms ‘max-value’. Hence, from this point on we will concentrate on results obtained by the mid-point variant of DLCA.

The relative performance of the average-case root is not influenced dramatically by the size of the tree (number of taxa). In both 24-taxa and 96-taxa trees the RF-score of the average root (using mid-point reduction) is about 25% higher than the RF-score of NJ. On the other hand, the relative performance of the best-case root significantly deteriorates in bigger trees. In 24-taxa trees the RF-score of the best root (using mid-point reduction) is about 50% lower than the RF-score of NJ, and in 96-taxa trees it is merely 15% lower.

This observation can also be seen in Fig. 4. We ranked the taxa in each data instance according to the RF-scores that DLCA achieves when using them as roots: rank = 1 for the taxon resulting in the lowest score, and rank = $\#taxa$ for taxon resulting in the highest score. In Fig. 4 we see average RF-scores plotted against *root-rank* of the taxa. The results shown here correspond to trees with moderate rate of evolution, however very similar behavior can be seen for other rates as well. We observe that in both cases the best topology obtained by DLCA (‘mid-point’ variant) is on average better than the one obtained by NJ. However, in 96-taxa trees a smaller portion of the taxa (8/96) yield lower average scores than NJ, compared with 24-taxa trees (8/24).

5.2 Results on Model Trees

It would be very helpful to know, prior to execution, from which taxa the DLCA algorithm is likely to yield a more accurate reconstruction. This could save us the trouble of running it from all taxa, and improve the chance for more accurate reconstruction. In order to approach this challenge, we used the ‘model trees’ dataset since it contains multiple (1,000) simulations for each tree. In each tree, we looked for taxa which consistently lead to better reconstruction. Figure 5 summarizes the results achieved for all trees under the fastest rate of evolution (M.D = 2.0, see [31]), with 300_b long sequences. Other settings yielded similar results.

Each taxon receives an RF-score, which is the average RF-score over all topologies reconstructed using DLCA from that taxon. It is apparent from our results that in ultrametric trees (A/B group) all taxa receive similar average RF-scores which are notably higher than the score NJ receives. In such trees, there is no taxon which seems to significantly outperform the others. In the other group of trees (C/D), which are far from ultrametric, we see high fluctuations in scores; some taxa significantly outperform others. In all three trees, there are taxa which receive average RF-scores similar to NJ, whereas the rest receive much higher scores. It is easy to see that the taxa which yield more accurate reconstruction are the ones closer to the actual root of the tree. In ‘CD’, for

24 taxa				
		slow rate	moderate	fast
NJ	–	2.3365	1.8085	1.7235
max-value	best	1.5295	1.0685	1.1280
	average	3.2936	2.9151	3.3481
	worst	5.2040	4.9880	5.9955
mid-point	best	1.3165	0.8565	0.7815
	average	2.7154	2.2794	2.3970
	worst	4.3680	4.0380	4.5445

96 taxa				
		slow rate	moderate	fast
NJ	–	16.983	12.312	10.669
max-value	best	17.935	13.839	14.093
	average	24.077	20.916	23.107
	worst	30.848	29.249	35.143
mid-point	best	14.348	10.124	9.3025
	average	19.644	15.936	16.005
	worst	25.659	22.875	24.650

Table 1: **Average performance on random trees.** Results for NJ and DLCA are estimated using the RF-score. Each result is averaged over 2,000 random instances. DLCA was executed using both mid-point and max-value reductions. For each variant, we show the score obtained by the best root, worst root, and average score over all roots. The upper table corresponds to random trees over 24 taxa, and the lower one to trees over 96 taxa.

instance, we see that taxa 1, 3, 4, 7, 9, 11 receive scores similar to NJ (~ 2), taxa 2, 5, 8, 10 receive a much higher score (~ 4.4), and taxa 6, 12 receive a somewhat better score (~ 3.8). This correlates well with distances from the root of the tree: $\sim 4b$ for the first group, $\sim 13b$ for the second, and $9b$ for the last. Notice that taxon 3 receives a slightly higher score than 4. This difference, though small, seems to be significant, and it cannot be explained by distance from the root (as taxa 3,4 are equidistant from the root). There may be other factors which come into play here.

Given an input distance-matrix, therefore, we wish to predict which taxa are closer to the root of evolution. In the model trees we studied, the taxa which are closer to the root are relatively closer to other taxa as-well. A taxon i minimizing $\max_{j \in S} \{D(i, j)\}$ is likely to be close to the root. In Table 2 we can see that choosing a taxon via this criterion yields better results than choosing a taxon by random. In the A/B group this difference is rather small, however in the C/D group the improvement is much more significant. We tried using the same criterion to find ‘good taxa’ in our random-trees dataset, however, we did not get a significant improvement over the average-case. Note, however, that in some scenarios of phylogenetic reconstruction, we may add an outgroup taxon, which we know to be relatively close to the root.

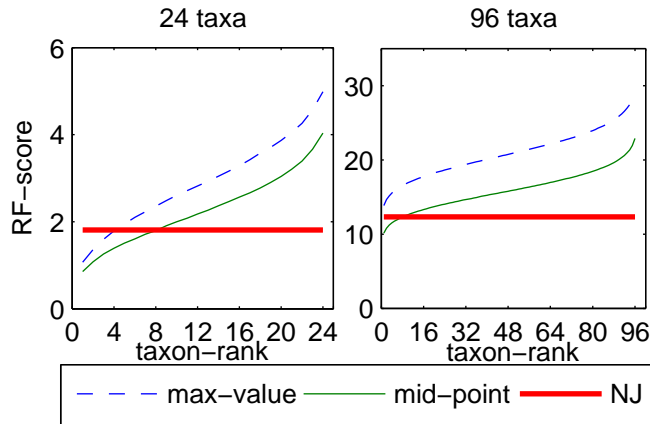


Figure 4: **Performance from various roots on random trees.** The horizontal line corresponds to the score obtained by NJ. The x-axis corresponds to root-rank. rank=1 means that we chose the root yielding the smallest RF-score, and rank=#*taxa* means we chose the worst-case root. Graphs describe performance on 24-taxa and 96-taxon random trees (both with moderate rate evolution).

	AA	BB	AB	CC	DD	CD
NJ	2.924	3.534	3.247	2.106	2.089	2.103
average taxon	4.020	4.986	4.471	3.125	3.175	3.186
chosen taxon	3.688	4.420	4.000	2.219	2.181	2.262

Table 2: **Choosing a root-taxon.** Results for NJ and DLCA (mid-point variant) are estimated by the RF-score. For DLCA, we show the average score over all roots, and the score received by the root i minimizing the criterion: $\max_{j \in S} \{D(i, j)\}$.

5.3 Reconstruction Accuracy vs. Best-Fit to Input

In case we do not know a-priori which taxa are more likely to yield a better tree, we can run DLCA from all taxa, and choose one of the n resulting trees which is predicted to be closer to the correct tree by some criterion. One approach is to test the fit of a tree to the input matrix. Hopefully, the better it fits the input, the closer it is to the original topology. We tested this assumption on our random trees over 24 taxa, by computing the ℓ_2 -distances from each input matrix to all trees reconstructed from it. Since we wish to test the reconstructed **topology**, we disregard the edge-weights assigned during reconstruction, and use *least-squares* estimates of edge weights instead (as suggested in [33]). These weights minimize, for a given topology, the ℓ_2 -distance from the input matrix. A similar approach was taken by Farach and Cohen in [7] regarding ℓ_1 and ℓ_∞ (using Linear Programming to re-calculate edge weights). The results are

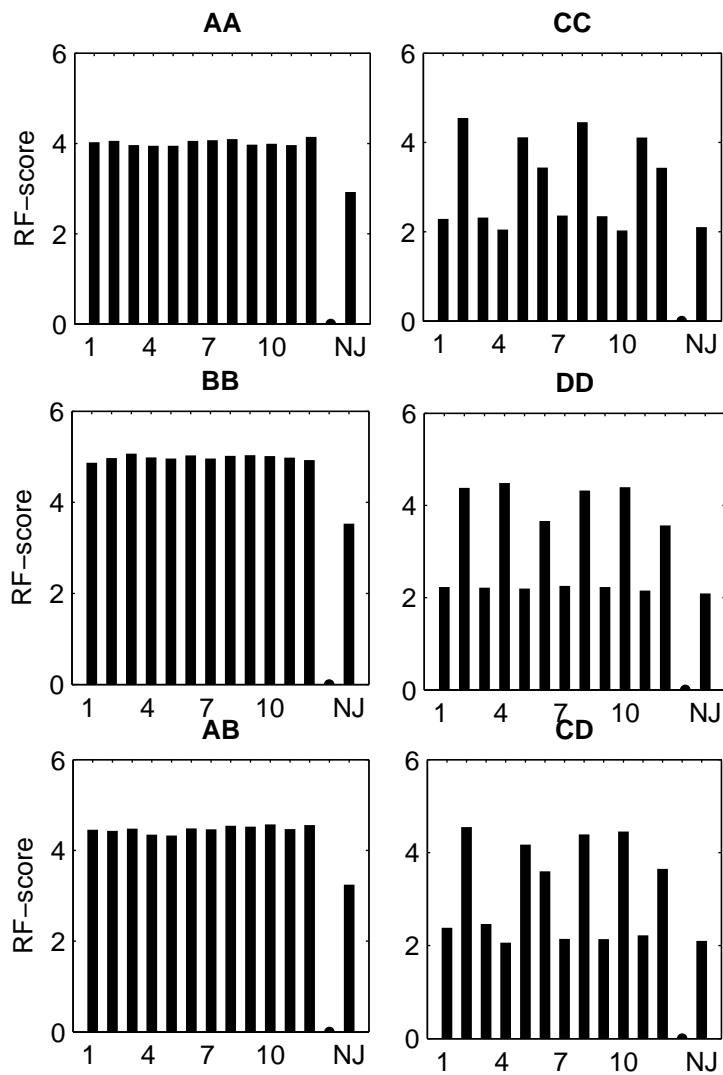


Figure 5: **Characterizing good roots.** Summary of results obtained for all 6 model trees. In each graph, the first 12 bars correspond to all 12 taxa. The rightmost bar corresponds to NJ. Each taxon receives an RF-score which is the average RF-score over all topologies reconstructed using DLCA from that taxon.

	slow	moderate	fast
improvement ratio of chosen tree	0.131	0.165	0.189
DLCA beats NJ (MID)	93%	94%	93%
(MAX)	84%	86%	80%
DLCA beats original tree	92%	90%	91%

Table 3: **Reconstruction accuracy vs. best-fit to input under ℓ_2 .** The values in the top row are the ratios: $\frac{AVERAGE_{RF} - CHOSEN_{RF}}{AVERAGE_{RF} - BEST_{RF}}$, where $AVERAGE_{RF}$ is the RF-score of the average case, $BEST_{RF}$ is the smallest RF-score, and $CHOSEN_{RF}$ is the RF-score of the tree best approximating the input matrix under ℓ_2 . RF-scores were obtained by running the mid-point variant of DLCA, and averaging over 2,000 data instances (see Table 1). The rest of the table shows in what ratio of the data DLCA achieves better fit to the input-matrix, compared with NJ, and the original tree. NJ is compared to both variants of DLCA (‘maximal value’ and ‘mid-point’). All results were obtained on the random-trees dataset over 24 taxa.

presented in the upper row of Table 3. We see that the average RF-score of the topology ‘closest’ to the input matrix is much closer to the one received by the average-case root, than the one received by the best topology. Therefore, among the topologies reconstructed by DLCA, that which fits best the input matrix, is rarely a good candidate for approximating the original topology.

We also checked to see if DLCA yields a better ℓ_2 -fit to the input matrix, compared with NJ. Indeed, for more than 90% of the distance-matrices in our dataset, the best tree reconstructed by DLCA is closer to the input matrix than the one reconstructed by NJ. Interestingly, more than 90% of the times this tree is even closer to the input than the **original tree**. This supports the observation we made earlier, that best-fit to the input matrix rarely implies the most accurate reconstruction. Farach and Cohen show by simulations [7] that their pivotal-algorithm (SP-DP) yields a better fit to the input matrix than NJ under ℓ_1, ℓ_∞ . Note that the SP algorithm is equivalent to the ‘maximal-value’ variant of DLCA, and DP is a variant of SP which allows for midpoints between two taxa to act as roots, as well as the taxa themselves. We see here that ‘maximal-value’ also yields good results considering ℓ_2 . However, it seems the ‘mid-point’ variant is slightly better than it. This fact adds up with our initial observation that the ‘mid-point’ variant is superior to ‘maximal-value’ regarding the RF-score, despite the good theoretical properties we proved for the latter in Section 4.

6 Discussion and Conclusion

In this paper, we presented a characterization of edge-weighted trees using LCA-distances. LCA-distances of trees obey a 3-point condition dual to the 3-point ultrametric condition. This duality enables us to provide a simple neighbor-joining criterion (Deepest Least Common Ancestor). Using this criterion, we de-

fined a family of efficient neighbor joining algorithms (DLCA), with $O(n^2 \log(n))$ running time. These algorithms are pivotal in the sense that when the input is not consistent with some tree, the output may vary according to the choice of root-taxon (from which LCA-distances are estimated).

We showed that conservative DLCA algorithms possess various optimal robustness properties, defined in [2, 28], and in this respect they outperform NJ. The ‘maximal value’ value variant of DLCA can be implemented in $O(n^2)$ time, and yields a topology best-fitting the estimated LCA-distances under several interesting measures of fitness. Using these properties, this variant was shown to provide a new simple $O(n^2)$ 3-approximation algorithm for the closest additive metric under the ℓ_∞ norm, and to prove its optimality under the abovementioned robustness criteria.

We used simulated data to test the performance of DLCA against Saitou&Nei’s NJ algorithm. Using the fact that DLCA algorithms are pivotal, we obtained n potentially different trees from each data-instance. We see that the pivotal approach almost always yields topologies better fitting the input matrix than NJ. This is shown here (Table 3) regarding the ℓ_2 norm, and was shown earlier in [7] regarding ℓ_1, ℓ_∞ norms.

The main goal of our experiments was to test the accuracy of topological reconstruction. In this respect, we observe that the ‘mid-point’ variant of DLCA, which uses averaging, is significantly superior to the ‘maximal-value’ variant, while the latter has better theoretical guarantees. In the same respect, note that NJ, which as noted earlier tends to select neighboring leaves with maximal average LCA distance, was shown to be theoretically less robust than DLCA. On the other hand, it turned out to yield better results on average than DLCA on our simulated data.

As in the case of approximating the input matrices, in most cases at least one of the n trees returned by the mid-point DLCA is closer to the original topology than the one returned by NJ. A natural problem raised by this scenario is that there is no obvious way to select an output tree which best approximates the original topology. This calls for some heuristic solution which is able to pinpoint either a topology which is close to the original one, or a root-taxon which is likely to yield such a topology. A possible solution is to use some other optimization criterion, like fitness to the input distance matrix or maximum parsimony of the underlying sequences [26]. Our results indicate that fitness to the input distance matrix (specifically under ℓ_2) does not seem to imply similarity to the original topology, and hence may be inappropriate for the specified evolutionary model. On the other hand, we were able to illustrate on several model trees that taxa closer to the origin of evolution are more likely to yield better topologies. This observation suggests selecting as a root - an outgroup-taxon known to be closer to that origin.

Acknowledgement

We would like to thank Satish Rao for an interesting discussion and for drawing our attention to [28].

References

- [1] Richa Agarwala, Vineet Bafna, Martin Farach, Mike Paterson, and Mikkel Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM Journal on Computing*, 28(3):1073–1085, June 1999.
- [2] K. Atteson. The performance of neighbor-joining methods of phylogenetic reconstruction. *ALGORITHMICA: Algorithmica*, 25, 1999.
- [3] Yair Bartal, Nathan Linial, Manor Mendel, and Assaf Naor. Low dimensional embeddings of ultrametrics. *Eur. J. Comb.*, 25(1):87–92, 2004.
- [4] William J. Bruno, Nicholas D. Socci, and Aaron L. Halpern. Weighted Neighbor Joining: A Likelihood-Based Approach to Distance-Based Phylogeny Reconstruction. *Mol Biol Evol*, 17(1):189–197, 2000.
- [5] D. Bryant. On the uniqueness of the selection criterion in Neighbor-joining. *Journal of Classification*, 22(1):3–15, 2005.
- [6] Peter Buneman. The recovery of trees from measures of dissimilarity. *Mathematics in the Archeological and Historical Sciences*, pages 387–395, 1971.
- [7] Jaime Cohen and Martin Farach. Numerical taxonomy on data: Experimental results. *Journal of Computational Biology*, 4(4):547–558, 1997.
- [8] W. H. E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of Mathematical Biology*, 49(4):461–467, 1987.
- [9] Richard Desper and Olivier Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *J Comp Biol*, (5):687–705, 2002.
- [10] Richard Desper and Olivier Gascuel. Theoretical foundation of the balanced minimum evolution method of phylogenetic inference and its relationship to weighted least-squares tree fitting. *Mol Biol Evol*, (3):587–598, 2004.
- [11] Robinson D.F. and Foulds L.R. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.
- [12] Isaac Elias and Jens Lagergren. Fast neighbor joining. In *Proc. of the 32nd International Colloquium on Automata, Languages and Programming (ICALP’05)*, volume 3580 of *Lecture Notes in Computer Science*, pages 1263–1274. Springer-Verlag, July 2005.
- [13] P. L. Erdos, M. A. Steel, L. A. Szekely, and T. J. Warnow. A few logs suffice to build (almost) all trees (II). Technical Report 97-72, DIMACS, October 17 1997. Mon, 14 Sep 1998 20:00:00 GMT.

- [14] Harding E F. The probabilities of rooted tree shapes generated by random bifurcation. *Adv. Appl. Prob.*, 3:44–77, 1971.
- [15] M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13(1/2):155–179, January 1995.
- [16] Kluge A.G. Farris J.S and Eckardt M.J. A numerical approach to phylogenetic systematics. *Systematic Zoology*, 19:172–189, 1970.
- [17] Yule Undy G. A Mathematical Theory of Evolution, based on the Conclusions of Dr. J. C. Willis, F.R.S. *Philosophical Transactions of the Royal Society of London Ser. B*, 213:21–87, 1925.
- [18] O Gascuel. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol Biol Evol*, 14(7):685–695, 1997.
- [19] Olivier Gascuel and Stphane Guindon. The methods and algorithms in bioinformatics (MAB) lab. Le Laboratoire d’Informatique, de Robotique et de Microelectronique de Montpellier
[http : /www.lirmm.fr/mab/sommaire_english.php3](http://www.lirmm.fr/mab/sommaire_english.php3).
- [20] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.
- [21] Felsenstein J. PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics*, 5:164–166, 1989.
- [22] Studier JA and Keppler KJ. A note on the neighbor-joining algorithm of Saitou and Nei. *Mol Biol Evol*, 5(6):729–731, 1988.
- [23] Mirko Krivanek. The complexity of ultrametric partitions on graphs. *Inform. Process. Lett.*, 27:265–270, 1988.
- [24] MK Kuhner and J Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates [published erratum appears in *Mol Biol Evol* 1995 May;12(3):525]. *Mol Biol Evol*, 11(3):459–468, 1994.
- [25] Dan Levy, Ruriko Yoshida, and Lior Pachter. Beyond pairwise distances: Neighbor joining with phylogenetic diversity estimates. *Molecular Biology and Evolution*, November 2005.
- [26] Katherine St. John Jerry Sun Luay Nakhleh, Usman Roshan and Tandy Warnow. The performance of phylogenetic methods on trees of bounded diameter. In *WABI*, pages 214–224, 2001.
- [27] Kimura M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J Mol Evol*, 16(2):111–120, December 1980.

- [28] Radu Mihaescu, Dan Levy, and Lior Pachter. Why neighbor-joining works, 2006.
- [29] Satoshi Ota and Wen-Hsiung Li. NJML: A Hybrid Algorithm for the Neighbor-Joining and Maximum-Likelihood Methods. *Mol Biol Evol*, 17(9):1401–1409, 2000.
- [30] Grassly NC Rambaut A. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput Appl Biosci*, 13:235–238, 1997.
- [31] Vincent Ranwez and Olivier Gascuel. Improvement of Distance-Based Phylogenetic Methods by a Local Maximum Likelihood Approach Using Triplets. *Mol Biol Evol*, 19(11):1952–1963, 2002.
- [32] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4:406–425, 1987.
- [33] S. Sattath and A. Tversky. Additive similarity trees. *Psychometrika*, 42(3):319–345, 1977.
- [34] Peter H. A. Sneath and Robert R. Sokal. *Numerical Taxonomy : The principles and practice of numerical classification*. W. H. Freeman, San Francisco, 1973.
- [35] Le Sy Vinh and Arndt von Haeseler. Shortest triplet clustering: reconstructing large phylogenies using representative sets. *BMC Bioinformatics*, 6(1):92, 2005.
- [36] Singh M Waterman MS, Smith TF and Beyer WA. Additive evolutionary trees. *J Theor Biol*, 64(2):199–213, January 1977.