Neighbor Joining Algorithms for Inferring Phylogenies via LCA-Distances

Ilan Gronau Shlomo Moran

June 17, 2007

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 edge-weighted trees by distances to LCAs (*Least Common Ancestors*). This characterization enables a direct application of ultrametric reconstruction techniques to trees which are not necessarily ultrametric. A simple and natural neighbor joining criterion based on this observation is used to provide a family of efficient neighbor-joining algorithms. These algorithms are shown to reconstruct a refinement of the Buneman tree, which implies optimal robustness to noise under criteria defined by Atteson. In this sense, they outperform many popular algorithms such as Saitou&Nei's NJ. One member of this family is used to provide a new simple version of the 3-approximation algorithm for the closest additive metric under the l_∞ norm.

A byproduct of our work is a novel technique[1] which yields a time optimal $O(n^2)$ implementation of common clustering algorithms such as UPGMA.

1 Introduction

Phylogenetic reconstruction methods attempt to find the evolutionary history of a given set of extant species (taxa). This history is usually described by an edge-weighted tree whose internal vertices represent past speciation events (extinct species) and whose leaves correspond to the given set of taxa. The amount of evolutionary change between two subsequent speciation events is indicated by the weight of the edge connecting them. The topology of the tree (which is defined by the set of positively weighted edges) is often assumed to be *fully resolved*, meaning that it forms a binary tree (i.e. each internal vertex is of degree 3). Distance-based phylogenetic reconstruction methods typically try to reconstruct this evolutionary tree from estimates of pairwise distances.

[1] After publication of this paper, it was brought to our attention that this technique was already presented in [30]. A proof that this technique implies a correct implementation of UPGMA and other clustering algorithms appears in [23].

A distance metric consistent with some positively edge-weighted tree is said to be *additive*, and a distance-based algorithm which always returns a tree given its induced additive metric is said to be *consistent*.

One of the most popular distance-based reconstruction techniques is *neighbor-joining*. Neighbor-joining is an agglomerative clustering approach, in which at each stage two neighboring elements are joined to one cluster; this new cluster then replaces them in the set of elements, and clustering continues recursively on this reduced set. One of the most important components of a neighbor-joining algorithm is the criterion by which elements are chosen to be joined. The simplest neighbor-joining criterion is probably the 'closest-pair' criterion, which is used in several well known clustering algorithms such as UPGMA, WPGMA [35] and the single-linkage algorithm [26, 4]. While this criterion is inconsistent in general, it is consistent for the special case of *ultrametric trees*, which contain a point (root) which is equidistant from all taxa. Ultrametric reconstruction algorithms typically have very efficient implementations: $O(n^2 log(n))$ for UPGMA and WPGMA, and $O(n^2)$ for the single linkage algorithm. Neighbor-joining algorithms which consistently reconstruct **general** trees (which are not necessarily ultrametric) typically use more complex neighbor joining criteria, significantly increasing their running time.

The problem of consistent reconstruction can be reduced to the special case of ultrametric reconstruction by applying the *Farris transform* [18]. The Farris transform converts any additive metric into an *ultrametric* while conserving the topology of the corresponding tree (see Fig. 1). After applying the Farris transform, ultrametric reconstruction methods (such as the ones listed above) can be used to obtain an intermediate ultrametric tree. Finally, in order to obtain the desired tree, the weights of external edges need to be restored. This approach leads to several time optimal consistent reconstruction algorithms (see e.g. [24, 1]). In this paper we introduce an alternative technique for reducing the problem of consistent reconstruction to the problem of ultrametric reconstruction. Using distances to *least common ancestors* (LCAs), this technique directly reconstructs the desired tree, thus bypassing the intermediate ultrametric tree mentioned above. This direct approach enables the proof of certain robustness properties which are strictly stronger than consistency alone.

Consistency is a natural and basic requirement, guaranteeing correct reconstruction when distance estimates are accurate. However, in practice we are rarely able to obtain accurate distance estimates, and the input from which trees are reconstructed is seldom additive. The input dissimilarity matrix is often regarded as a noisy version of some original additive metric, and distance-based reconstruction methods are required to be *robust* to this noise. Informally, robustness of an algorithm to noise is measured by the amount of noise under which it is still guaranteed correct reconstruction of the tree's topology (or parts of it).

One notion of robustness is defined by the ability to reconstruct the correct topology given *nearly additive* input. A dissimilarity matrix D is said to be *nearly additive* with respect to a *binary* edge-weighted tree T (whose induced additive metric is denoted by D_T), if $||D, D_T||_\infty < \frac{1}{2} \cdot \min_{e \in T}\{w(e)\}$ [2]. The

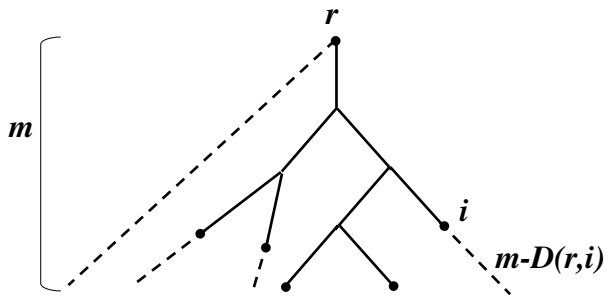

Figure 1: **The Farris Transform.** Given a dissimilarity matrix D, a taxon r and some value $m \geq max_i\{D(r,i)\}$, the Farris-transform defines a dissimilarity matrix U s.t. $U(i,j) = 2m + D(i,j) - D(r,i) - D(r,j)$. If D is additive, consistent with some tree T, then U is consistent with an ultrametric tree achieved by elongating the external edges of T (elongation marked by dashed line).

topology of T is **uniquely** determined by any dissimilarity matrix D which is nearly additive with respect to it. This is because the topology of T is uniquely determined by the configurations of all taxon-quartets in the tree, and a matrix D which is nearly additive w.r.t. T is also *quartet-consistent* with it in the following sense:

Definition 1.1 (*Quartet consistency*)**.** *Let D be a dissimilarity matrix, then:*

- *D is* consistent with quartet-configuration $(ij:kl)$*, if:*
$$D(i,j) + D(k,l) < \min\{D(i,k) + D(j,l)\ ,\ D(i,l) + D(j,k)\}.$$
- *D is* quartet-consistent *with some tree T if it is consistent with all quartet-configurations induced by T.*

When the input is not quartet-consistent with any tree, it may still be consistent with certain edges of the tree in the sense introduced by Buneman [9]. In this context, an edge is identified with the *split* it induces over the taxon set, and a split $(P|Q)$ is implied by dissimilarity matrix D, if D is consistent with **all** quartet-configurations $(ij:kl)$ s.t. $i,j \in P$ and $k,l \in Q$. The *Buneman tree* of D is a tree which contains exactly all edges which are thus implied by D (these edges are called *Buneman edges*). Buneman's original algorithm in [9] constructs the Buneman tree in $\Omega(n^4)$ time. A more efficient $\theta(n^3)$ algorithm was later introduced in [5]. This reconstruction approach is considered to be very conservative in the sense that it typically results in a highly unresolved topology consisting only of few edges. Nevertheless, it is intuitively expected that a robust reconstruction algorithm correctly reconstruct all Buneman edges.

In a related work, Atteson [2] introduced the concepts of *l_∞-radius* and *edge l_∞-radius*, which provide numerical scales for robustness. The *l_∞-radius* of a reconstruction algorithm $\mathcal{A}$ is the maximal ε s.t. for every dissimilarity matrix D and binary tree T, if $||D, D_T||_\infty < \varepsilon \cdot \min_{e \in T}\{w(e)\}$ then $\mathcal{A}$ is guaranteed to return a tree with the same topology as T when receiving D as input. An

algorithm $\mathcal{A}$ is said to have *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 $\frac{1}{\varepsilon}||D, D_T||_\infty$. Notice that the edge l_∞-radius of an algorithm is bounded from above by its l_∞-radius, and it is shown in [2] that they both cannot be greater than $\frac{1}{2}$. It is clear by definition that an algorithm which guarantees correct reconstruction from nearly additive input has an optimal l_∞-radius of $\frac{1}{2}$. Moreover, any algorithm which correctly reconstructs all Buneman edges has an optimal edge l_∞-radius of $\frac{1}{2}$, since all edges in T of weight greater than $2||D, D_T||_\infty$ are Buneman edge of D.

1.1 Related Work

Consistent reconstruction of phylogenetic trees has been studied since the early seventies [9, 37, 34]. In general, this task requires $\Omega(n^2)$ time, and $\Omega(nlog(n))$ for the special case of trees with fully resolved topologies (where n denotes the number of taxa) [11]. An $O(n^2)$ algorithm was already proposed in [37], and $O(nlog(n))$ algorithms for reconstructing fully-resolved topologies were proposed only later in [25, 6].

The neighbor joining scheme was first used in the context of consistent distance-based reconstruction by the $\Omega(n^4)$ ADDTREE algorithm [34]. Later, Saitou and Nei proposed the famous $\theta(n^3)$ neighbor-joining algorithm commonly called NJ [33, 36]. Since then, numerous algorithms were developed in hope of outperforming the original NJ algorithm on noisy input matrices (e.g. BIONJ [20], NJML [31] and Weighbor [7] to name a few). This improved performance is typically not proven analytically, but rather demonstrated on actual data generated via some simulation of the evolutionary process. A consistent $O(n^2)$ neighbor joining algorithm was recently proposed in [15]; this algorithm, called FastNJ, uses a neighbor-selection criterion similar to the one used by NJ, while reducing the total time complexity by a factor of n. Experimental results reported there show that the reduction in running time has only a minor affect on the accuracy of reconstruction.

One approach for analytically evaluating the performance of a distance-based reconstruction algorithm on non-additive input is by observing the distance between the input dissimilarity matrix and the metric induced by the output tree. This distance is typically measured using some metric norm. Unfortunately, the consequent optimization problem of finding the **closest** tree to the input matrix was shown to be NP-hard for several such norms (ℓ_1, ℓ_2 in [13] and ℓ_∞ in [1]). The only constant-rate approximation known to us in this area is the 3-approximation algorithm for the ℓ_∞ norm presented in [1]. This algorithm uses the Farris transform and an algorithm for finding the closest ultrametric to a given dissimilarity matrix [26, 17].

Another indication for robustness to noise, which was mentioned earlier, is Atteson's l_∞-radius, and in particular the ability to reconstruct the correct topology given a nearly additive dissimilarity matrix. The conditions under which a dissimilarity matrix calculated over biological sequences is guaranteed (with high probability) to be nearly additive were studied in [16]. Much con-

sequent research [12, 29, 10] was done using this result and assuming near-additivity of the input or parts of it. In [2] Atteson shows that many distance-based algorithms (including NJ and its variants) have an optimal l_∞-radius of $\frac{1}{2}$, meaning that they return the correct topology given nearly additive input. He also analyzes the edge l_∞-radius of these algorithms, and shows that NJ has edge l_∞-radius no greater than $\frac{1}{4}$. Only recently in [28], it was proven that the edge l_∞-radius of NJ is exactly $\frac{1}{4}$.

1.2 Our Contribution

In this paper we introduce a characterization of tree metrics by *LCA-distances*, which are distances from a selected root-taxon to the least common ancestors of all taxon-pairs. Despite not obeying the basic distance-metric requirements (such as the triangle inequality), LCA-distances bear some of the nice properties of ultrametric distances, with the additional advantage of being able to represent general trees and not only ultrametric trees. A simple and natural neighbor joining criterion based on this observation is used to provide a family of efficient neighbor-joining algorithms – *Deepest Least Common Ancestor* (DLCA). DLCA algorithms can be seen as a simpler and more direct implementation of the Farris transform: rather than transforming the tree-metric and going through an intermediate ultrametric tree, it uses LCA distances (which can be calculated from the original metric) to directly reconstruct the desired tree.

The DLCA family allows a large variety of consistent reconstructions algorithms, each of which is distinctive in the way it reduces the input matrix at each recursive step. These algorithms have a time optimal $O(n^2)$ implementation based on a novel technique, which may also be used to provide $O(n^2)$ implementations of UPGMA, WPGMA and other similar clustering algorithms. We concentrate on a large natural sub-family of DLCA algorithms called *conservative* algorithms, which are all shown to reconstruct a refinement of the corresponding Buneman tree, implying optimal l_∞-radius and edge l_∞-radius of $\frac{1}{2}$. We note that although the different algorithms in this sub-family all have identical reconstruction guarantees, their performance may still differ significantly when executed on actual data.

The rest of the paper is organized as follows. The next subsection provides the needed notations and definitions. In Section 2 we describe our characterization and present the generic DLCA neighbor joining algorithm based on this characterization. An efficient $O(n^2)$ implementation is presented in Subsection 2.1. In Section 3 we analyze the robustness of our algorithms, and in Section 4 we provide specific analysis of the DLCA algorithm which uses 'maximal-value' reduction; among other things, this variant of DLCA is used to produce a simple proof of the 3-approximation result of [1]. Section 5 summarizes the results and discusses future research directions.

1.3 Definitions and Notations

Let S be a finite set (the set of taxa). A phylogenetic tree over S is an undirected weighted tree $T = (V, E, w : E \to \mathcal{R}^+ \cup \{0\})$ 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 weights. 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. 2); 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 $A(i,j) \leq B(i,j)$ for all $i, j \in S$. All matrices referred to in this paper are assumed to be symmetric.

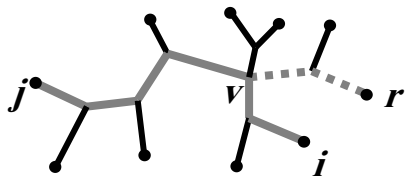

Figure 2: **Distance estimates in trees.** $D_T(i,j)$ is the total weight of the path connecting taxa i, j in T. $D_T(r; ij)$ is the total weight of the path connecting r and the center vertex (v) of the 3-finger claw spanning r, i, j.

2 The DLCA Family of Algorithms

Given an edge-weighted tree T over a set of taxa S and a taxon $r \in S$, LCA^r_T is a matrix over $S \setminus \{r\}$ holding all LCA-distances in T from root-taxon r: $LCA^r_T(i,j) = D_T(r; ij)$. LCA-distances may be estimated from taxon-sequences in two ways. The first option is to obtain them from a pairwise dissimilarity matrix D (computed using standard methods) by the transformation $LCA(D, r)$ in Definition 2.1 below. This transformation preserves consistency such that if D is an additive metric consistent with tree T, then $LCA(D, r) = LCA^r_T$.

Definition 2.1. *Given a dissimilarity 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:*

$$L(i,j) = \frac{1}{2}(D(r,i) + D(r,j) - D(i,j)) \ .$$

LCA-distances may also be estimated directly from taxon-sequences by applying standard maximum likelihood techniques (e.g. [19]) over triplets of sequences. Previous works [32, 27] indicate that distance estimates obtained directly over sequence-triplets are more accurate than the ones obtained from sequence-pairs, potentially leading to more accurate reconstruction.

A characterization of matrices of the form LCA^r_T, to be denoted *LCA-matrices*, is given by Definition 2.2 and Theorem 2.3 below.

Definition 2.2 (LCA-matrix)**.** *A symmetric non-negative matrix L over a set S is an LCA-matrix if 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[2]*).*

The above 3-point condition can also 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 appears at least twice.*

Theorem 2.3. *A symmetric non-negative matrix L over a set of taxa S is an LCA-matrix iff there exists an edge-weighted tree T over the expanded set of taxa $S \cup \{r\}$ s.t. $L = LCA^r_T$, 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 $L = LCA^r_T$. It is clear that $\forall i,j \in S : D_T(r,i) \geq D_T(r; ij)$, which implies Property 1 of Definition 2.2. Now observe the subtree spanning r, i, j, k. If its topology is a star (Fig. 3a), 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 (Fig. 3b) then $L(i,j) = L(i,k) < L(j,k)$, and the minimum value appears twice. The same can be argued for the other two possible topologies of this subtree, proving Property 2.

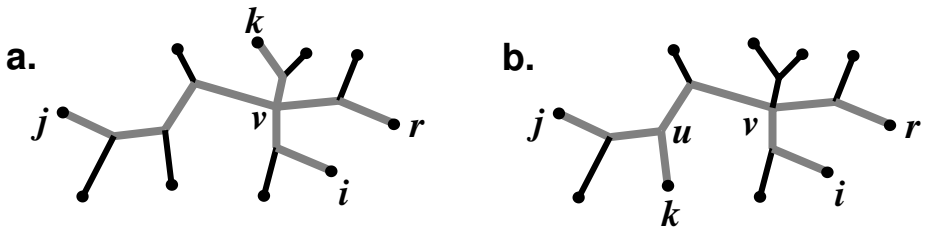

Figure 3: **The 3-point condition for LCA-distances.** Observe the subtree spanning r, i, j, k (marked edges). **a)** If its topology is a star (4-finger claw), with center-vertex v, then $D_T(r; ij) = D_T(r; ik) = D_T(r; jk) = D_T(r, v)$. **b)** Otherwise, w.l.o.g. i is paired up with r as illustrated,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

$\Rightarrow$ The proof of the other direction is constructive. Given a matrix L which satisfies both conditions, we show that any variant of the generic *Deepest Least Common Ancestor* (DLCA) neighbor joining algorithm described in Table 1 constructs a tree T s.t. $LCA^r_T = L$. It is clear that such an algorithm returns a tree rooted at r with S as its set of leaves. We prove by induction on $|S|$ that this tree is consistent with the input LCA-matrix L.

[2]Matrices satisfying this condition are referred to in [24] as *min-ultrametrics*.

Deepest LCA Neighbor Joining (DLCA):

Input: A symmetric nonnegative matrix L over a set (of taxa) S.

1. **Stopping condition:** If $L = [w]$ return a tree consisting of a single edge of weight w, connecting the root r to the single taxon in S.
2. **Neighbor selection:** Select a pair of distinct taxa i, j, s.t. $L(i, j)$ is a maximal off-diagonal entry in rows i, j of L.
   (i.e. for all $k \neq i, j$: $L(i, j) \geq \max\{L(i, k), L(j, k)\}$)
3. **Reduction:** Remove i, j and add v to the taxon-set.
   – Set $L(v, v) \leftarrow L(i, j)$.
   – For all $k \neq v$, set $L(v, k) \leftarrow \alpha_k L(i, k) + (1 - \alpha_k)L(j, k)$.
   – Recursively call DLCA on the reduced matrix L.
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 $w(v, j) = \max\{0, L(j, j) - L(i, j)\}$.

Table 1: **The DLCA algorithm.** The recursive procedure above describes a *generic* DLCA algorithm. Each variant of this algorithm is determined by the way α_k is calculated in step 3. This calculation may depend on the identities of i, j, k, on the input matrix L, and on any other data kept by the algorithm.

Base case: $|S| = 1$. $L = [w]$, and by the stopping condition we have $LCA^r_T = [w]$. For the induction step, observe the following lemma, which follows immediately from the 3-point condition:

Lemma 2.4. *Let L be an LCA-matrix over S, and let i, j be two distinct elements of S s.t. $\forall k \neq i, j : L(i, j) \geq \max\{L(i, k), L(j, k)\}$. Then $\forall k \neq i, j$: $L(i, k) = L(j, k)$.*

Now, suppose that $|S| > 1$ and let i, j be the taxon-pair chosen by the algorithm (in step 2). By Lemma 2.4 we have $L(i, k) = L(j, k)$ for all $k \neq i, j$. Hence in step 3 of the algorithm we get $L(v, k) \leftarrow L(i, k)$, regardless of the value assigned to α_k. We now argue that the reduced matrix L' over $S' = S \setminus \{i, j\} \cup \{v\}$ defined by step 3 of the algorithm is an LCA-matrix as well. Since all the entries of L' except $L'(v, v)$ are identical to the corresponding entries of $L(S \setminus \{j\})$ (where index v in L' corresponds to index i of L), Property 2 of Definition 2.2 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 for all $k \in S' \setminus \{v\}$, $L'(k, v) = L(k, i) \leq L(i, j) = L'(v, v)$.

Given that L' is an LCA-matrix, the induction hypothesis implies that the tree T' over $S' \cup \{r\}$ returned by the recursive call at the end of step 3 satisfies $LCA^r_{T'} = L'$. Using this we show that $LCA^r_T = L$. Recall that T is obtained

from T' in step 4 by adding two edges $(v,i),(v,j)$ with weights $L(i,i)-L(i,j)$ and $L(j,j)-L(i,j)$ respectively (these weights are non-negative due to Property 1 of LCA-matrices). Now for all $k,l \in S \setminus \{i,j\}$ we have:

$$\begin{aligned} LCA^r_T(k,l) &= LCA^r_{T'}(k,l) = L'(k,l) = L(k,l) \ , \\ LCA^r_T(k,i) &= LCA^r_{T'}(k,v) = L'(k,v) = L(k,i) \ , \\ LCA^r_T(k,j) &= LCA^r_{T'}(k,v) = L'(k,v) = L(k,j) \ . \end{aligned}$$

We are left to prove the equality for the entries $(i,j),(i,i),(j,j)$ of L:

$$\begin{aligned} LCA^r_T(i,j) &= LCA^r_{T'}(v,v) = L'(v,v) = L(i,j) \ , \\ LCA^r_T(i,i) &= LCA^r_{T'}(v,v) + w(v,i) = L(i,j) + L(i,i) - L(i,j) = L(i,i) \ , \\ LCA^r_T(j,j) &= LCA^r_{T'}(v,v) + w(v,j) = L(i,j) + L(j,j) - L(i,j) = L(j,j) \ . \end{aligned}$$

□

In order to complete the proof of consistency for the DLCA algorithm, it is enough to show that each LCA-matrix represents a **unique** edge-weighted tree (with strictly positive internal edge weights). This is implied by the fact that the taxa i,j chosen in step 2 of the algorithm must be neighbors in **all** trees consistent with the input LCA-matrix (proof details are omitted).

The degree of freedom in the choice of reduction formula (defined by the value assigned to α_k in step 3) implies a wide family of consistent algorithms (the DLCA family). The discussion in this paper is confined to algorithms which use only *conservative* reductions. A conservative reduction step is achieved by first calculating some value for $\alpha \in [0,1]$, and then applying one of the following:

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

Although not all consistent reductions are conservative, most interesting reductions are. We will mainly be interested in two specific conservative variants:

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

The deepest LCA neighbor-joining scheme proposed here relates to the well known closest-pair neighbor-joining scheme for ultrametric reconstruction. The closest-pair criterion is based on the 3-point condition for ultrametrics much the same way that the deepest-LCA criterion is based on the 3-point condition for LCA-matrices. This simple relation allows us to convert many known algorithms which reconstruct ultrametric trees from pairwise-distances to algorithms which reconstruct general trees from LCA-distances. The aforementioned 'mid-point' variant can actually be viewed as such a conversion of the WPGMA algorithm[3]. The 'maximal-value' variant similarly relates to the single linkage algorithm presented in [26, 17].

[3] A variant of the DLCA algorithm which similarly relates to UPGMA can be achieved by a slight modification of the mid-point reduction.

2.1 A Time Optimal Implementation of DLCA

Given an input matrix L over a set of n taxa S, DLCA performs $n-1$ iterations (recursive calls). Each such iteration involves selecting a neighboring taxon-pair and reducing the input matrix. It is easy to see that the reduction step can be implemented in linear time[4]. Thus, the running time of the algorithm is typically dominated by the time required for the neighbor selection steps. A naive approach for neighbor selection, which requires $\theta(n^2)$ time in **each iteration** (and a total time complexity of $\theta(n^3)$) scans the matrix L for a maximum off-diagonal entry and selects the taxon-pair corresponding to it.

The time complexity can be reduced to $O(n^2 log(n))$ by maintaining for each $i \in S$ an index j s.t. $L(i,j)$ is a maximal off diagonal entry in row i of L, as follows. Let $MAX_L(i) = \max_{k \neq i} L(i,k)$ denote the maximal off-diagonal value in row i of L. An ordered taxon-pair (i,j) is a *maximal pair* (in row i) of L if $L(i,j) = MAX_L(i)$. Finding a maximal pair for each row in L can be done in $O(n^2)$ time. Once a maximal pair is kept for each row, a taxon-pair satisfying the neighbor-selection criterion is found in linear time by scanning the set of maximal pairs and selecting a pair (i,j) for which $L(i,j)$ is maximized. Updating the set of maximal pairs after the reduction of L (in step 3 of the algorithm) can be done in $O(nlog(n))$ time by maintaining the entries in each row of L in a heap, thus resulting in total time complexity of $O(n^2 \log n)$.

For the 'maximal value' reduction, the running time of the above algorithm can be reduced to $O(n^2)$ by techniques similarly employed in the single linkage algorithm [17, 4]. In a nutshell, this is done by updating the set of maximal pairs in linear time during each reduction step. This is possible since when reducing the matrix L to a matrix L' by replacing i, j with v, the maximal value reduction guarantees that if (k,i) or (k,j) is a maximal pair of L, then (k,v) is a maximal-pair of L'. Unfortunately, this is not true for other conservative reductions. We are able to get $O(n^2)$ running time for other conservative reductions by the observation that the selected pair i, j should correspond to a maximal off diagonal entry in rows i, j, but not necessarily in the entire matrix. To find such pairs efficiently, we maintain a *complete ascending path*. A sequence of distinct taxa $P = (i_1, i_2, \ldots, i_l)$ is an *ascending path* with respect to L if all its edges (i_r, i_{r+1}) are maximal pairs, implying also that $MAX_L(i_r) \leq MAX_L(i_{r+1})$. An ascending path is *complete* if the above inequality holds with equality for the last taxon-pair in the sequence (i.e. $MAX_L(i_{l-1}) = MAX_L(i_l)$).

Observation 2.5. *If $P = (i_1, \ldots, i_l)$ is a complete ascending path of L, then $L(i_{l-1}, i_l)$ is a maximal off-diagonal entry in rows i_{l-1}, i_l of L.*

Observation 2.5 implies that neighbor selection can be implemented by maintaining a complete ascending path. A method for constructing and maintaining such a path throughout the execution of the algorithm in overall $O(n^2)$ time will imply the desired bound on the total time complexity. Our method is based on the following *basic extension operation*: given an ascending path $P = (i_1, \ldots, i_l)$

[4]We exclude the computation of α_k in step 3 from our analysis as it is typically done in constant time for commonly used reductions.

of L, compute $m = MAX_L(i_l)$; if $m = L(i_{l-1}, i_l)$ then terminate extension; otherwise, extend the path P by adding to it any vertex i_{l+1}, s.t. $L(i_l, i_{l+1}) = m$. By repeating this basic extension operation until termination, we obtain a **complete** ascending path.

Given an input matrix L of dimension $n > 1$, a complete ascending path is constructed by initializing an ascending path P in an arbitrary taxon (i.e. $P = (i_1, i_2)$ s.t. $i_1 \in S$ and $L(i_1, i_2) = MAX_L(i_1)$), and then extending P as described above. Given a complete ascending path $P = (i_1, \ldots, i_l)$ of a matrix L, consider a reduction step in which the taxon-pair $(i, j) = (i_{l-1}, i_l)$ is replaced by a new taxon v. Let L' be the matrix obtained by this reduction. We observe that if the reduction is convex, meaning that for all $k \neq v$ the value of $L'(v, k)$ lies between $L(i, k)$ and $L(j, k)$[5], then the path $\bar{P} = (i_1, \ldots, i_{l-2})$ is a (possibly empty) ascending path of the reduced matrix L'. This observation follows from the fact that all consecutive pairs in $\bar{P}$ remain maximal with respect to L'. Thus a complete ascending path P' can be computed for L' by iteratively extending $\bar{P}$ by basic extension operations, until the termination condition is met.

We now analyze the total time complexity of the process described above. This process consists of a series of basic extension operations, some of which lead to termination, whereas the rest lead to an extension of P by an additional vertex. Each operation requires the computation of $MAX_L(i)$ (for some taxon i), which can be done in linear time. Thus, the total time complexity of maintaining P is determined by the total number of basic extension operations invoked throughout the execution of the algorithm. $n-1$ such operations leads to termination (one in each iteration), whereas the rest result in an extension of the path by a single vertex. Now, since in each iteration only two vertices are removed from P, and by the time the execution concludes this path is emptied (up to a single vertex), the total number of vertices added to P throughout the execution is no more than $2n - 2$. Thus the total number of basic extension operations is no more than $3n - 3$, leading to a total running time of $O(n^2)$.

Note: Complete ascending paths can also be used to achieve optimal $O(n^2)$ implementations of some well known clustering algorithms such as UPGMA and WPGMA. To the best of our knowledge, these are the first $O(n^2)$ *faithful* implementations of these algorithms which completely preserve their input-output specifications for **all** possible inputs (see [23]).

3 Optimal Robustness of DLCA

In this section we discuss the robustness of DLCA. In particular, we consider an execution of an arbitrary conservative DLCA algorithm on input of the form $LCA(D, r)$, and prove that in such an execution the algorithm returns a tree which refines the Buneman tree of D. We then show that this implies optimal robustness under Atteson's criteria. We start by defining the concepts of *clades* and *LCA-clusters*, which play a central role in the analysis.

[5]Observe that each conservative reduction is convex.

Definition 3.1 (Clades)**.** *Given a tree T rooted at r and a vertex v in T, denote by $\mathcal{L}_r(v)$ (the* clade *of v) the set of leaves which are descendants of v in T.*

Definition 3.2 (LCA clusters)**.** *Let L be a symmetric matrix over S. A proper subset $X \subset S$ is an* LCA-cluster *of L if it satisfies the following condition:*

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

The following lemma characterizes the connection between clades and LCA-clusters.

Lemma 3.3. *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 in 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 $|S| > |X| > 1$.

The induction step is carried out by observing that conservative reductions preserve LCA-clusters. 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 $L(i,j)$ in rows i, j of L implies that either $\{i,j\} \subseteq X$, or $\{i,j\} \subseteq S \setminus X$. Now 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' be the reduced version of X, such that $X' = X \setminus \{i,j\} \cup \{v\}$ if $\{i,j\} \subseteq X$ and $X' = X$ otherwise. We prove now that X' is an LCA-cluster of L', i.e:

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

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

- $\{i,j\} \subseteq S \setminus X$ (and hence $v \in S' \setminus X'$): If $z \neq v$ the claim follows from the inductive 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 since $L'(x,y) = L(x,y)$, the convexity of the reduction step guarantees that $L'(x,y) > L'(x,v)$. Similar argument can be used to show that $L'(x,y) > L'(y,v)$ as well.

- $\{i,j\} \subseteq X$ (and hence $v \in X'$): If $x, y \neq v$ the claim follows from the inductive assumption on X. We are left to show that $L'(x,v) > \max\{L'(x,z), L'(v,z)\}$ for all $x \in X' \setminus \{v\},\ z \notin X'$.
  Let x, z be as above. Since X is an LCA-cluster of L, we have $L(x,i), L(x,j) > L(x,z)$. Again, the convexity of the reduction step guarantees $L'(x,v) > L'(x,z)$. 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 is of the form $L(v,k) \leftarrow \alpha L(i,k) + (1-\alpha)L(j,k)$, 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 if of the second form $(L(v,k) \leftarrow \alpha \min\{L(i,k), L(j,k)\} + (1-\alpha)\max\{L(i,k), L(j,k)\})$, 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 denote by T' the rooted tree returned by DLCA when run on L'. Since X' is an LCA-cluster of L', the induction hypothesis implies that there is a vertex u in T', 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 $\mathcal{L}_r(u) = X$. □

After establishing the connection between LCA-clusters and clades, the following lemma completes the picture by providing the desired connection between Buneman edges and LCA-clusters.

Lemma 3.4. *Let D be a dissimilarity matrix over a taxon-set S, and let $(P|Q)$ be a partition of S induced by some edge in the Buneman tree of D. Then Q is an LCA-cluster of $LCA(D,r)$ for every taxon r in P.*

Proof. Let $L = LCA(D,r)$. In order to show that Q is an LCA-cluster of L, we need to prove that for every $x, y \in Q,\ z \in P$ we have $L(x,y) > L(x,z)$. Since $(P|Q)$ is induced by a Buneman edge, then for all $x, y \in Q$ and $w, z \in P$, we have $D(w,z) + D(x,y) < \min\{D(w,x) + D(y,z)\ ,\ D(w,y) + D(x,z)\}$. When assigning $w = r$ we get the following:

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

□

Lemmas 3.3 and 3.4 rather straightforwardly imply the following theorem, which contains our main result.

Theorem 3.5. *Any conservative DLCA algorithm, when executed on $LCA(D,r)$ (for arbitrary dissimilarity matrix D and a root-taxon r), has an optimal edge l_∞-radius (and hence also optimal l_∞ radius) of $\frac{1}{2}$.*

Proof. Let T be an edge-weighted tree, D be a dissimilarity matrix over taxon-set S, and let e be an edge in T s.t. $w(e) > 2||D, D_T||_\infty$. It is required to show that the tree reconstructed by DLCA has an edge inducing the same split $(P|Q)$ as e. It is easy to see that since $w(e) > 2||D, D_T||_\infty$, the Buneman tree of D contains an edge inducing the split $(P|Q)$. Now, assume w.l.o.g. that the root-taxon r (from which DLCA is executed) is in P, then Lemma 3.4 implies that Q is an LCA-cluster of $LCA(D,r)$. By Lemma 3.3, Q is a clade of the tree returned by DLCA, meaning that some edge in this tree induces the split $(P|Q)$. □

We conclude this section by comparing the robustness of (conservative) DLCA algorithms with that of NJ (as reported in [2, 28]). 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, it was demonstrated in [28] that NJ does not always correctly reconstruct a tree from a dissimilarity matrix which is quartet-consistent with it. DLCA, on the other hand, is guaranteed correct reconstruction in such a case, demonstrating yet again its superior robustness to noise compared with NJ.

4 The 'Maximal-Value' variant of DLCA

This section is devoted to a detailed discussion of a specific member of the DLCA family – the 'maximal-value' variant. As mentioned earlier, the 'maximal-value' variant of DLCA relates to the single linkage ultrametric reconstruction algorithm from [26, 17]. Its Farris-transform equivalent was used as part of the $O(n^3)$ algorithm for reconstructing the Buneman tree [5], and as part of the $O(n^2)$ 3-approximation algorithm from [1]. In this section we prove that this variant possesses some interesting properties, which are implied by the fact that it yields a tree whose LCA-matrix is the *unique dominant LCA-matrix* of the input matrix[6]. Among other things, our analysis provides a simple proof of the 3-approximation result from [1].

Let $\mathcal{U}(A)$ be the set of all LCA-matrices which are greater or equal to a matrix A. Since $\mathcal{U}(A)$ is closed and bounded from below (by A), it contains a minimal element. It is also easy to see that for any two matrices L_1, L_2 in $\mathcal{U}(A)$, 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 dominant LCA-matrix* of 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 ℓ_p norms). It is also easy to see that L^{dom} is an LCA-matrix closest to A (among all matrices, not just those in $\mathcal{U}(A)$) 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 dominant LCA-matrices:

Lemma 4.1. *Let L^{dom} be the dominant LCA-matrix of a symmetric matrix A.*

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

Proof. Let $m_i = \max_j\{A(i,j)\}$, and let L be the matrix over S defined by: $L(i,j) = \min\{L^{dom}(i,j), m_i, m_j\}$. Then we have that for all $i, j \in S$,

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

[6]This concept is dual to the unique sub-dominant ultrametric defined in [26].

meaning 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 by the dominance of L^{dom}, $L = L^{dom}$ and the lemma follows.

Property 1 of LCA-matrices (see Definition 2.2) holds for L since for all 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. □

We now turn to prove that the LCA-matrix of the tree reconstructed by the 'maximal-value' variant of DLCA is dominant to the input matrix.

Theorem 4.2. *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 using the maximal-value reduction. Then LCA^r_T 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 $LCA^r_T = [w] = A$. 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. By the induction hypothesis, $L' = LCA^r_{T'}$ is the unique dominant LCA-matrix of A'. We will use this to show that $L = LCA^r_T$ 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:

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

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 4.1 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, and an arbitrary taxon $k \neq i,j$, we use the fact that $L(i,j) = A(i,j)$ and $L(i,k) = L(j,k) \leq L(i,j)$, to show that:

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

Thus we have that $M(i,j) \geq \max\{M(i,k), M(j,k)\}$ for all $k \neq i,j$. This implies, by Lemma 2.4, that $M(i,k) = M(j,k)$. Hence the matrix M can be reduced to a matrix M' over S' by replacing rows i, j by row v, and as argued in the proof of Theorem 2.3, this reduced matrix is an LCA-matrix. Now since

$A' \leq M' \leq L'$, the induction hypothesis on L' implies that $M' = L'$, and this in turn implies that $M = L$ by the following equalities:

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

□

The following lemma demonstrates how to transform L^{dom} into an LCA-matrix closest to A under the ℓ_∞ norm (out of all LCA-matrices, not just the ones 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 ℓ_∞.*

Proof. First, it is easy to see that L^∞ is an LCA-matrix, and that $||A, L^\infty||_\infty = \frac{\varepsilon}{2}$. Given an arbitrary LCA-matrix L, we need to prove that $\frac{\varepsilon}{2} \leq \varepsilon_L$, where $\varepsilon_L = ||A, L||_\infty$. 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 $A \leq L^{dom} \leq L'$. This means that $\varepsilon = ||A, L^{dom}||_\infty \leq ||A, L'||_\infty \leq 2\varepsilon_L$. □

Note that if $L^{dom}(i,i) = A(i,i)$ for all i, then we can modify the definition of L^∞ in Lemma 4.3 s.t. $\frac{\varepsilon}{2}$ is subtracted only from **off-diagonal** entries of L^{dom}, and thus for all i, $L^\infty(i,i) = A(i,i)$ as well. Both versions of the transformation of L^{dom} to L^∞ do not change the topology of the tree corresponding to L^{dom}, with the exception of setting some edge-weights to zero. We now show that the tree T^∞ corresponding to L^∞ provides the desired 3-approximation. Given a metric D over a taxon-set S, our 3-approximation algorithm acts as follows:

1. Choose some arbitrary taxon r, and calculate $L = LCA(D, r)$.
2. Execute the 'maximal-value' variant of DLCA on L to get a tree T^{dom}. Let $L^{dom} = LCA^r_{T^{dom}}$.
3. Apply on L^{dom} the transformation of Lemma 4.3 which subtracts $\frac{\varepsilon}{2}$ only from off-diagonal entries. Return the tree T^∞ corresponding to the resulting LCA-matrix L^∞.

Note that all stages of the algorithm can be implemented in $O(n^2)$ time.

Theorem 4.4. *Let D be a metric over a taxon-set S, and let T^∞ be the tree returned by 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 $L = LCA(D, r)$, and by T' the edge-weighted tree which realizes D'. Note that $L' = LCA^r_{T'}$ is an LCA-matrix due to Theorem 2.3. Our proof consists of two simple claims:

Claim 4.5. $||D, D_{T^\infty}||_\infty = 2 \cdot ||L, L^\infty||_\infty$.

Proof. We first need to show that for all taxa i, $D_{T^\infty}(r, i) = D(r, i)$. Notice that since D satisfies the triangle inequality, L is nonnegative and $\forall i : D(r, i) = L(i, i) = \max_j L(i, j)$. Therefore, by Lemma 4.1 we have that $\forall i : L^{dom}(i, i) = \max_j L(i, j) = L(i, i)$. Hence, when invoking the transformation implied by Lemma 4.3, $\frac{\varepsilon}{2}$ is not subtracted from the diagonal, and we get:

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

Now we use the above equality and the formula in Definition 2.1 to show that the following holds for every taxon-pair $i, j \in S \setminus \{r\}$:

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

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

Claim 4.6. $||L, L'||_\infty \leq \frac{3}{2} \cdot ||D, D'||_\infty$.

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 $||L, L^\infty||_\infty \leq ||L, L'||_\infty$, the above claims imply:

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

□

Note: Theorem 3.5 in the previous section implies that the 'maximal-value' variant of DLCA has optimal l_∞-radius and edge l_∞-radius of $\frac{1}{2}$. Since the transformation in step 3 of the above 3-approximation algorithm does not change the topology of the tree, the same robustness result applies to this algorithm as well. We note that in [16] it is argued that a 3-approximation algorithm cannot have l_∞-radius greater than $\frac{1}{6}$. This claim is based on an example which consists of a dissimilarity matrix D and two trees with different topologies over the same set of 4 taxa. 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 of the closest additive metric to D under ℓ_∞. We observe that this example only demonstrates that, a-priori, a 3-approximation algorithm is not guaranteed to have an l_∞-radius greater than $\frac{1}{6}$. However, it does not exclude the possibility that such an algorithm may indeed have a greater l_∞-radius, and hence it does not contradict our result.

5 Conclusion and Discussion

In this paper we discussed a characterization of edge-weighted trees using LCA-distances. We showed that any tree can be uniquely defined by distances from an arbitrary taxon to the least common ancestors of all taxon-pairs (Theorem 2.3). These LCA-distances obey a 3-point condition dual to the 3-point ultrametric condition, providing us with a simple neighbor-joining criterion (Deepest Least Common Ancestor). Using this criterion, we defined a family of neighbor joining algorithms (DLCA), and then presented an $O(n^2)$ time implementation of these algorithms using the technique of complete ascending paths. The same technique can be used to implement various clustering algorithms such as UPGMA and WPGMA in optimal $O(n^2)$ time as well.

A major part of our discussion was dedicated to exploring the robustness of DLCA algorithms to noise in the input distance-estimates. DLCA algorithms using conservative reduction steps were shown to possess various optimal robustness properties. In this respect, they outperform Saitou&Nei's NJ algorithm. Specific analysis was given for one conservative variant of DLCA – 'maximal value'. This variant was shown to yield a tree-topology best fitting the input LCA-distances under several interesting measures. It was also used to provide a new simple $O(n^2)$ 3-approximation algorithm for the closest additive metric under the ℓ_∞ norm. The optimal robustness of conservative DLCA algorithms mentioned above applies to this 3-approximation algorithm as well.

Apart from their being efficient and robust, DLCA algorithms are distinctive in their pivotal nature, which may hold an advantage when executing them on actual data. DLCA algorithms allow an arbitrary choice of the root-taxon, however, preliminary experiments indicate that accuracy of reconstruction is very much influenced by this choice as well as the choice of reduction formula. In our experiments we used datasets described in [14, 32], which were downloaded from the LIRMM 'Methods and Algorithms in Bioinformatics' website [21]. A detailed account of our experimental setup and results can be found in [22].

The results of these experiments indicate that accuracy of reconstruction varies very much among the different choices of root taxon. Accuracy is also highly influenced by the reduction formula used by the algorithm: the 'mid-point' variant of DLCA typically yields significantly better reconstruction than the 'maximal-value' variant despite the theoretical guarantees shown for the latter in Section 4. On average, both variants were observed to yield less accurate reconstruction compared with NJ (averaging over all possible choices of root-taxon). It is plausible that the relative superiority of both NJ and the 'mid-point' variant is due to the use of averaging, which plays an important role in 'smoothing' noise in the input. Averaging appears in the reduction steps of both NJ and the 'mid-point' variant, and in NJ it also appears in the neighbor-selection criterion. Given a dissimilarity matrix D, NJ selects a pair of taxa maximizing $D(i,j) + \sum_{r \neq i,j} L_r(i,j)$, where $L_r = LCA(D,r)$. Intuitively, this criterion gives priority to pairs of taxa with **average** deepest LCA[7].

[7]The introduction of the term $D(i,j)$ is necessary to make the selection criterion consistent (see e.g. [8]).

As mentioned above, when the root taxon is chosen uniformly at random, reconstruction done by the DLCA algorithm is typically less accurate than that of Saitou&Nei's NJ. However, almost every instance in our dataset contained a taxon from which the 'mid-point' variant of DLCA yields a tree closer to the true tree than the one returned by NJ. This phenomenon suggests two possible courses of action. The first option is to run DLCA from all taxa to obtain n possibly different trees, and then select from these trees the one most likely to be closest to the true topology. While there is no straightforward way to perform such a selection, certain natural criteria come into mind, such as fit to the input matrix, parsimony score and likelihood. An apparent disadvantage of this approach is that it introduces an additional factor of n to the running time of the algorithm. Another approach is to choose the root-taxon according to some criterion which is expected to lead to better reconstruction. Our experiments indicate that taxa closer to the origin of evolution are more likely to lead to better reconstruction.

Their relative simplicity and proven robustness are apparent advantages of DLCA algorithms. The main conclusion we draw from our preliminary experimental results is that a better use of the pivotal nature of DLCA may lead to competitive reconstruction in practice. This venue of research is still to be pursued.

Acknowledgement

We would like to thank Isaac Elias and Satish Rao for interesting discussions, and Satish Rao also for drawing our attention to [28].

References

[1] R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. 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*, 25:251–278, 1999.

[3] Y. Bartal, N. Linial, M. Mendel, and A. Naor. Low dimensional embeddings of ultrametrics. *Eur. J. Comb.*, 25(1):87–92, 2004.

[4] J. Barthelemy and A. Guenoche. *Trees and proximities representations.* Wiley, 1991.

[5] V. Berry and D. Bryant. Faster reliable phylogenetic analysis. In *RECOMB '99: Proceedings of the third annual international conference on Computational molecular biology*, pages 59–68, New York, NY, USA, 1999. ACM Press.

[6] G. Brodal, R. Fagerberg, C. Pedersen, and A. stlin. The complexity of constructing evolutionary trees using experiments. In *Proc. 28th International Colloquium on Automata, Languages, and Programming*, volume 2076 of *Lecture Notes in Computer Science*, pages 140–151. 2001.

[7] W. Bruno, N. Socci, and A. Halpern. Weighted Neighbor Joining: a likelihood-based approach to distance-based phylogeny reconstruction. *Mol Biol Evol*, 17(1):189–197, 2000.

[8] D. Bryant. On the uniqueness of the selection criterion in neighbor-joining. *Journal of Classification*, 22(1):3–15, 2005.

[9] P. Buneman. The recovery of trees from measures of dissimilarity. *Mathematics in the Archeological and Historical Sciences*, pages 387–395, 1971.

[10] A. Jaffe R. Mihaescu E. Mossel S. Rao C. Daskalakis, C. Hill. Maximal accurate forests from distance matrices. In *RECOMB*, pages 281–295, 2006.

[11] J. Culberson and P. Rudnicki. A fast algorithm for constructing trees from distance matrices. *Information Processing Letters*, 30(4):215–220, February 1989.

[12] T. Warnow D. Huson, S. Nettles. Disk-Covering, a fast-converging method for phylogenetic tree reconstruction. *J Comp Biol*, 6:369–386, 1999.

[13] W. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of Mathematical Biology*, 49(4):461–467, 1987.

[14] R. Desper and O. Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *J Comp Biol*, (5):687–705, 2002.

[15] I. Elias and J. 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.

[16] P. Erdos, M. Steel, L. Szekely, and T. Warnow. A few logs suffice to build (almost) all trees (II). *Theoretical Computer Science*, 221:77–118, 1999.

[17] M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13(1/2):155–179, January 1995.

[18] J. Farris. A probability model for inferring evolutionary trees. *Systematic Zoology*, 22:250–256, 1973.

[19] J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol*, 17(6):368–376, 1981.

[20] 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.

[21] O. Gascuel and S. Guindon. The methods and algorithms in bioinformatics (MAB) lab. Le Laboratoire d'Informatique, de Robotique et de Microlectronique de Montpellier http:/www.lirmm.fr/mab/sommaire_english.php3.

[22] I. Gronau and S. Moran. Pivotal neighbor joining algorithms for inferring phylogenies via LCA-distances. Technical Report CS-2006-11, Technion, May 2006. http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2006/CS/CS-2006-11.pdf.

[23] I. Gronau and S. Moran. Optimal implementations of UPGMA and other common clustering algorithms. Technical Report CS-2007-06, Technion, May 2007. http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2007/CS/CS-2007-06.pdf.

[24] D. Gusfield. *Algorithms on Strings, Trees, and Sequences.* Cambridge University Press, 1997.

[25] S. Kannan, E. Lawler, and T. Warnaw. Determining the evolutionary tree using experiments. *Journal of Algorithms*, 21:26–50, 1996.

[26] M. Křivánek. The complexity of ultrametric partitions on graphs. *Inform. Process. Lett.*, 27:265–270, 1988.

[27] D. Levy, R. Yoshida, and L. Pachter. Beyond pairwise distances: Neighbor-joining with phylogenetic diversity estimates. *Mol Biol Evol*, 23(3):491–498, 2006.

[28] R. Mihaescu, D. Levy, and L. Pachter. Why neighbor-joining works, 2006.

[29] E. Mossel. Phase transitions in phylogeny. *Trans Amer Math Soc*, 356:2379–2404, 2004.

[30] F. Murtagh. Complexities of hierarchic clustering algorithms: state of the art. *Computational Statistic Quarterly*, 1(2):101–113, 1984.

[31] S. Ota and W. Li. NJML: a hybrid algorithm for the neighbor-joining and maximum-likelihood methods. *Mol Biol Evol*, 17(9):1401–1409, 2000.

[32] V. Ranwez and O. Gascuel. Improvement of distance-based phylogenetic methods by a local maximum likelihood approach using triplets. *Mol Biol Evol*, 19(11):1952–1963, 2002.

[33] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4:406–425, 1987.

[34] S. Sattath and A. Tversky. Additive similarity trees. *Psychometrica*, 42(3):319–345, 1977.

[35] P. Sneath and R. Sokal. *Numerical Taxonomy : the principles and practice of numerical classification.* W. H. Freeman, San Francisco, 1973.

[36] J. Studier and K. Keppler. A note on the neighbor-joining algorithm of Saitou and Nei. *Mol Biol Evol*, 5(6):729–731, 1988.

[37] M. Waterman, T. Smith, M. Singh, and W. Beyer. Additive evolutionary trees. *J Theor Biol*, 64(2):199–213, January 1977.