Path-Difference Median Trees

Alexey Markin and Oliver Eulenstein

Department of Computer Science, Iowa State University, Ames, IA 50011, USA
{amarkin,oeulenst}@iastate.edu

Abstract. Synthesizing large-scale phylogenetic trees is a fundamental problem in evolutionary biology. Median tree problems have evolved as a powerful tool to reconstruct such trees. Given a tree collection, these problems seek a median tree under some problem-specific tree distance. Here, we introduce the median tree problem for the classical path-difference distance. We prove that this problem is NP-hard, and describe a fast local search heuristic that is based on solving a local search problem exactly. For an effective heuristic we devise a time efficient algorithm for this problem that improves on the best-know (naïve) solution by a factor of n, where n is the size of the input trees. Finally, we demonstrate the performance of our heuristic in a comparative study with other commonly used methods that synthesize species trees using published empirical data sets.

Keywords: Phylogenetic trees, median trees, supertrees, path-difference distance, local search.

1 Introduction

Large-scale phylogenetic trees that represent the evolutionary relationships, or genealogy, among thousands of species offer enormous promise for society's advancements. While such species trees are fundamental to evolutionary biology, they are also benefiting many other disciplines, such as agronomy, biochemistry, conservation biology, epidemiology, environmental sciences, genetics, genomics, medical sciences, microbiology, and molecular biology [13, 14, 21]. However, despite these promises, synthesizing large-scale species trees is confronting us with one of the most difficult computational challenges in evolutionary biology today. Here, we are focusing on synthesizing large species trees from a given collection of typically smaller phylogenetic trees.

Traditionally, a species tree for a set of species is inferred by first selecting a gene that is common to them, and then inferring the evolutionary history for this gene, which is called a *gene tree*. Gene trees describe partial evolutionary histories of the species genomes, and therefore, it is often assumed that gene trees have evolved along the edges of the species tree, imitating it. However, a major shortcoming of the traditional approach is that different gene trees for the same set of species can describe discordant evolutionary histories. Such discordance is frequently caused by erroneous gene trees, or can be the result of genes which have evolved differently due to complex evolutionary processes

that have shaped the species genomes [23]. To confront these challenges, median tree problems (also called supertree problems [4]) have emerged as a powerful tool for inferring species trees from a collection of discordant gene trees. These problems seek a tree, called a median tree, that is minimizing the overall distance to the input trees based on some problem-specific distance measure. Typically, measures that have been well-established in comparative phylogenetics are used to compute median trees [4], and one of the oldest measures to compare trees is the path-difference distance. However, despite the tradition and popularity of the path-difference distance, median trees under this measure and their computation have not been analyzed.

In this work we are studying the computation of median trees under the path-difference distance. We show that computing median trees under the path-difference distance is an NP-hard problem for rooted as well as unrooted input trees. While most median tree problems used in practice are NP-hard, they have been effectively addressed by standard local search heuristics that solve a local search problem thousands of times. Encouraged by these promising results we introduce a novel local search heuristic to compute median trees under the path-difference distance. The heuristic is based on our $\Theta(kn^3)$ time algorithm (introduced here) that solves the corresponding local search problem exactly, where n and k is the size and number of trees in a given instance of the median tree problem respectively. Our new local search heuristic allows to compute the first large-scale median trees under the path-difference distance. Finally, we demonstrate the performance of our heuristic in a comparative study on several published empirical data sets, and demonstrate that it outperforms other standard heuristics in minimizing the overall path-difference. Software implementing our local search heuristic is freely available from the authors.

Related work. *Median tree problems* are a popular tool to synthesize large-scale species trees from a collection of smaller trees. Given a collection of input trees, such problems seek a tree, called a *median tree*, that minimizes the sum of its distances to each of the input trees. Since the ultimate goal of median tree problems is to synthesize accurately species trees of enormous scale, a large body of work has focussed on the biological, mathematical, and algorithmic properties of median tree problems adopting numerous definitions of distance measures from comparative phylogenetics [4]. One of the oldest such measures, however, is the path-difference distance [5,12,26,30], and median trees under this distance have not been analyzed. The *path-difference distance* between two trees is defined through the Euclidean distance between their path-length vectors. Each such vector represents the pairwise distances between all leaves of the corresponding tree (i.e., the number of edges on a simple path between leaves). Steel and Penny [30] have studied the distribution of the path-difference distance for un-rooted trees. Complementing this work, Mir and Rosello [19] computed the mean value of this distance for fully resolved unrooted trees with n leaves, and showed that this mean value grows in $O(n^3)$. Variants of the path-difference distance are the Manhattan distance of the path-length vectors [33], and their correlation [24].

Median tree problems are typically NP-hard [4], and therefore are, in practice, approached by using local search heuristics [1,9,17,18,32] that make truly large-scale phylogenetic analyses feasible [18,32]. Effective local search heuristics have been proposed and analyzed [1,9,17,18,32], and provided various credible species trees [18,32]. Given an instance I of a median tree problem, such heuristics start with some initial candidate species tree T and find a minimum cost tree for I under the tree distance measure of the problem in the (local) neighborhood of T, and so on, until a local minima is reached. At each local search step, the heuristic solves an instance of a local search problem. The time complexity of this local search problem depends on the tree edit operation that defines the neighborhood, as well as on the computation time of the tree distance measure that is used. A classical and well-studied tree edit operation is the *subtree prune and regraft (SPR)* operation [27] where a subtree of the edited tree is pruned and regrafted back into the tree at another location. The *SPR neighborhood* of T is the set of all trees into which T can be transformed by one SPR operation, and this neighborhood contains $\Theta(n^2)$ trees. Further, the best-known algorithm to compute the path-difference distance between two trees with n leaves requires $\Theta(n^2)$ time [30]. Therefore, given an instance of k trees over n different taxa of the SPR based local search problem, this problem can be naïvely solved by complete enumeration in $\Theta(kn^4)$ time, which is the best-known algorithm. However, when faced with heuristically estimating larger median trees this runtime becomes prohibitive.

Our Contribution. We introduce the *path-difference median tree problem* under the classical path-difference distance to synthesize large-scale phylogenetic trees. To prove its NP-hardness for rooted and unrooted input trees we are using polynomial time mapping-reductions from the maximum triplet consistency problem and from the quartet compatibility problem respectively. To solve large-scale instances of the path-difference median tree problem, we have devised a standard SPR local search heuristic. For time efficiency, we design a $\Theta(kn^3)$ time algorithm for an instance of the local search problem that improves on the best-known (naïve) solution by a factor of n, where n and k is the size and number of the input trees of the median tree problem respectively. Finally, we demonstrate the performance of our new local search heuristic through comparative studies using empirical data sets.

2 Basics and Preliminaries

Basic definitions. A *(phylogenetic) tree* T is a rooted full binary tree. We denote its node set, edge set, leaf set, and root, by $V(T)$, $E(T)$, $\mathsf{L}(T)$, and $\mathsf{Rt}(T)$ respectively. Given a node $v \in V(T)$, we denote its parent by $\mathsf{Pa}_T(v)$, its set of children by $\mathsf{Ch}_T(v)$, its sibling by $\mathsf{Sb}_T(v)$, the subtree rooted at v by $T(v)$, and $T|v$ is the phylogenetic tree that is obtained by pruning $T(v)$ from T. Note that we identify the leaf set with the respective set of leaf-labels (taxa).

Let $L \subseteq \mathsf{L}(T)$ and T' be the minimal subtree of T with leaf set L. We define the *leaf-induced subtree* $T[L]$ of T to be the tree obtained from T' by successively removing each node of degree two (except for the root) and adjoining its two neighbors.

Path-difference distance. Given a tree T and two leaves $u, v \in \mathsf{L}(T)$, let $\mathsf{d}_{u,v}(T)$ denote the length in edges of the unique path between u and v in T. Let $\mathsf{d}(T)$ be an associated vector obtained by a fixed ordering of pairs i, j [30], e.g., $\mathsf{d}(T) = (\mathsf{d}_{1,2}(T), \mathsf{d}_{1,3}(T), \ldots, \mathsf{d}_{n-1,n}(T))$, where n is the number of leaves. Then the *path-difference distance (PDD)* between two trees G and S over the same leaf set is defined as $\mathsf{d}(G, S) := ||\,\mathsf{d}(G) - \mathsf{d}(S)||_2$.

We also define $\mathsf{PLM}(T)$ to be the matrix of path-lengths between each two leaves in T. That is, a matrix of size $|\,\mathsf{L}(T)| \times |\,\mathsf{L}(T)|$, where rows and columns represent leaves of T, and $\mathsf{PLM}_{u,v}(T) = \mathsf{d}_{u,v}(T)$. Let G and S be trees over the same leaf set, then we define $\Delta(G, S) := \mathsf{PLM}(G) - \mathsf{PLM}(S)$ to be the *matrix of path-length differences.*

3 Path-difference median tree problem

Let $\mathcal{P}$ be a set of trees $\{G_1, \ldots, G_k\}$. We define $\mathsf{L}(\mathcal{P}) := \cup_{i=1}^{k} \mathsf{L}(G_i)$ to be the *leaf set of* $\mathcal{P}$. A tree S is called a *supertree* of $\mathcal{P}$, if $\mathsf{L}(S) = \mathsf{L}(\mathcal{P})$. Further, we extend the definition of the path-difference distance to a set of trees. Note, we defined PDD only for two trees over the same leaf set. However, we do not want to enforce such a restriction on the set of input trees, since it is generally not the case for real world data. Therefore, in order to compare two trees S and G, where $\mathsf{L}(G) \subseteq \mathsf{L}(S)$ we use the *minus method* [11]. That is, we calculate a distance between G and the subtree of S induced by $\mathsf{L}(G)$: $\mathsf{d}(S, G) = \mathsf{d}(S[\mathsf{L}(G)], G)$. We now define PDD for an input set $\mathcal{P}$ and a supertree S as a sum $\mathsf{d}(\mathcal{P}, S) := \sum_{i=1}^{k} \mathsf{d}(G_i, S[\mathsf{L}(G_i)])$, which is used to establish the following problem.

Problem 1 (PD median tree (supertree) – decision version).
Instance: a set of input trees $\mathcal{P}$ and a real number p;
Question: determine whether there exist a supertree S, such that $\mathsf{d}(\mathcal{P}, S) \leq p$.

3.1 The PD median tree problem is NP-hard.

We show this by a polynomial time reduction from the MaxRTC problem.

Problem 2 (Maximum Compatible Subset of Rooted Triplets – MaxRTC).
Instance: a set of rooted triplets R and an integer $0 \leq c \leq |R|$;
Question: Is there a subset $R' \subseteq R$, such that R' is compatible and $|R'| \geq c$.

A *rooted triplet* is a (rooted full binary) tree with exactly three leaves. A set of trees $\mathcal{P}$ is called *compatible* if there exist a supertree T consistent with every tree in $\mathcal{P}$, and a tree T is *consistent* with a tree G if $T[\mathsf{L}(G)] \equiv G$.

Theorem 1. *The PD median tree problem is NP-hard.*

Proof. We map an instance $\langle R, c\rangle$ of the MaxRTC problem to an instance $\langle R, \sqrt{2}(|R| - c)\rangle$ of the PD median tree problem. The MaxRTC problem is known to be NP-complete [6]. This transformation works due to the following property. Assume that S is a supertree of a set of rooted triplets $R = \{T_1, \ldots, T_k\}$. Then we observe that $\mathsf{d}(S[\mathsf{L}(T_i)], T_i)$ is 0, when S is *consistent* with T_i, and is $\sqrt{2}$ otherwise. Therefore, $\mathsf{d}(R, S) = \sqrt{2}(|R| - c')$, where c' is the number of triplets in R, which are consistent with S. That is, there are at least c' compatible triplets in R. Now, we can conclude the proof.

(i) If $\langle R, c\rangle$ is a *yes*-instance of the MaxRTC problem, then there exist a tree S, such that S is consistent with $|R'| \geq c$ triplets. As we shown above, in that case $\mathsf{d}(P, S) \leq \sqrt{2}(k - c)$. Therefore, $\langle R, \sqrt{2}(|R| - c)\rangle$ is a *yes*-instance of the PD median tree problem.
(ii) Clearly, the same argument works in the other direction. □

In practice median trees are sometimes computed for multi-sets of trees. However, our results, shown for sets of input trees, easily extend to multi-sets.

4 Local search for PD median tree problem

As stated in the introduction, we address the NP-hardness by devising a new SPR based local search heuristic. Next, we introduce needed definitions.

4.1 SPR-based Local search

Definition 1. *Given a node $v \in V(S)\backslash\{\mathsf{Rt}(S)\}$, and a node $u \in V(S)\backslash(V(S(v))\cup\{\mathsf{Pa}(v)\})$, $SPR_S(v, u)$ is a tree obtained as follows:*
(i) Prune the subtree $S(v)$ by (i) removing the edge $\{\mathsf{Pa}(v), v\}$, and (ii) removing $\mathsf{Pa}(v)$ by adjoining its parent and child.
(ii) If u is a root of $S|v$, then a new root w' is introduced, so that u is a child of w'. Otherwise, an edge $(\mathsf{Pa}(u), u)$ is subdivided by a new node w'.
(iii) Connect the subtree $S(v)$ to the node w'.

In addition, we introduce the following useful notation

$SPR_S(v) := \bigcup_u SPR_S(v, u)$; $SPR_S := \bigcup_{v,u} SPR_S(v, u)$. SPR_S is called an *SPR-neighborhood* of a tree S, and $|SPR_S| = O(n^2)$, where $n = |\,\mathsf{L}(S)|$.

Given a set of input trees $\mathcal{P} = \{G_1, \ldots, G_k\}$, the search space in the median tree problem can be viewed as a graph $\mathcal{T}$, where nodes represent supertrees of $\mathcal{P}$. There is an edge $\{S_1, S_2\}$ in $\mathcal{T}$, if S_1 could be transformed to S_2 with a single SPR operation. As was mentioned in the introduction, local search is designed to terminate at a local minimum of $\mathcal{T}$. More formally, at each iteration the following problem is solved

Problem 3 (PD local search).
Instance: An input set $\mathcal{P}$ and a supertree S;
Find: $S' = \underset{S' \in SPR_S}{\arg\min}\ \mathsf{d}(\mathcal{P}, S')$.

Next we describe an algorithm for the PD local search problem that improves on the best-known naïve algorithm (see Introduction) by a factor of n.

4.2 Local Search based on an SPR semi-structure

Let $\mathbf{G} \in \mathcal{P}$ be a fixed input tree, and let S_i be a supertree in the i-th iteration of the local search. Throughout this section we refer to the restricted tree $S_i[\mathsf{L}(G)]$ as $\mathbf{S}$. To reduce the complexity of the naïve algorithm, we exploit a *semistructure* of an SPR-neighborhood initially introduced in [8]. Let $N := SPR_S(v, \mathsf{Rt}(S))$ for some $v \in V(S)$, then $SPR_N(v)$ is equivalent to $SPR_S(v)$. This property is essential for the further analysis, which is motivated by the following theorem.

Theorem 2. *Given $\Delta(N, G)$, $\mathsf{d}(T, G)$ is computable in $\Theta(n)$ time for any $T \in SPR_S(v)$ with a single precomputation step of time complexity $\Theta(n^2)$.*

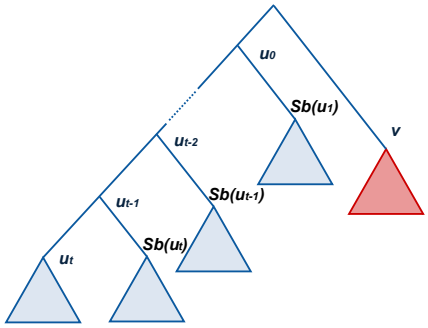

Fig. 1: Scheme of the $N = SPR_S(v, \mathsf{Rt}(S))$ tree, depicting how the leaf set was partitioned to create Table 1.

	C_v	C_{u_t}	$C_{\mathsf{Sb}(u_p)}$
C_v	0	$-t$	$-2p+1$ $+t$
C_{u_t}	$-t$	0	$+1$
$C_{\mathsf{Sb}(u_p)}$	$-2p+1$ $+t$	$+1$	0

Table 1: Note that $1 \leq p \leq t$. Values inside the table indicate the difference in path lengths between leaves from different subsets, i.e., for $i \in C_v$ and $j \in C_{u_t}$: $\mathsf{d}_{i,j}(T) = \mathsf{d}_{i,j}(N) - t$.

This theorem implies that for a fixed input tree G and a fixed prune node $v \in V(S)$ we can compute the PD distance for every $T \in SPR_S(v)$ in $\Theta(n^2)$ time. Therefore, computing $\mathsf{d}(T, G)$ for **all** $T \in SPR_S$ and **all** $G \in \mathcal{P}$ takes $\mathbf{\Theta(n^3k)}$ time, where $k = |\mathcal{P}|$. This is the time complexity of our algorithm for the PD local search problem. In the remainder of this section we detail the precomputation idea and prove Theorem 2.

Consider a tree $T := SPR_S(v, y)$, where $y \in V(S|v)$, and let $(u_0, ..., u_t)$ be a simple path in $S|v$, where $u_0 = \mathsf{Rt}(S|v)$ and $u_t = y$. Note, this path is also also a path in N, since $S|v$ is a subtree of N.

For convenience, let C_u denote $\mathsf{L}(N(u))$ for any $u \in V(N)$. Thus, we have $C_v = \mathsf{L}(N(v)) = \mathsf{L}(S(v))$. Table 1 shows a path-length difference matrix $\mathsf{PLM}(T) - \mathsf{PLM}(N)$. Using this table, it is possible to derive the difference between $\mathsf{d}(T, G)$ and $\mathsf{d}(N, G)$. It was constructed by partitioning the leaf set of S as follows (see also Figure 1): $\mathsf{L}(S) = C_v \cup (C_{\mathsf{Sb}(u_1)} \cup \ldots \cup C_{\mathsf{Sb}(u_t)}) \cup C_{u_t}$.

In order to explain Table 1 we need to explore how the path between two leaves changes when regrafting node v. We consider all possibilities for a pair of leaves i and j (except for the cases, when i and j belong to the same subset from the table, since the path does not change in that case).

(i) $i \in C_v$, $j \in C_{u_t}$. In N the path between i and j could be denoted by $A_i \sqcup (\mathsf{Pa}(v), u_0, \ldots, u_t) \sqcup B_j$. Note that partial paths A_i and B_j are not changed by the regrafting operation. In T the path between i and j is $A_i \sqcup (\mathsf{Pa}_T(v), u_t) \sqcup B_j$. The number of edges in the path is decreased by t.

(ii) $i \in C_v$, $j \in C_{\mathsf{Sb}(u_p)}$, where $1 \leq p \leq t$. Again, we denote the path between i and j in N by $A_i \sqcup (\mathsf{Pa}(v), u_0, \ldots, u_{p-1}, \mathsf{Sb}(u_p)) \sqcup B_j$. Then the corresponding path in T is $A_i \sqcup (\mathsf{Pa}_T(v), u_{t-1}, \ldots, u_p, u_{p-1}, \mathsf{Sb}(u_p)) \sqcup B_j$. It is easy to see that the path length increased by $(t - p) - (p - 1)$.

(iii) $i \in C_{u_t}$, $j \in C_{\mathsf{Sb}(u_p)}$, where $1 \leq p \leq t$. We denote a path between i and j in N by $A_i \sqcup (u_{t-1}, \ldots, u_p, u_{p-1}, \mathsf{Sb}(u_p)) \sqcup B_j$. Then the corresponding path

in T is $A_i \sqcup (\mathsf{Pa}_T(v), u_{t-1}, \ldots, u_p, u_{p-1}, \mathsf{Sb}(u_p)) \sqcup B_j$. Exactly one edge was added to the path (as a result of regrafting v above u_t).

(iv) $i \in C_{\mathsf{Sb}(u_p)}$, $j \in C_{\mathsf{Sb}(u_q)}$, where $1 \leq p, q \leq t$. Clearly, the path between i and j is not affected by the regrafting operation.

Let A and B be two elements from $\{C_v, C_{u_t}, C_{\mathsf{Sb}(u_t)}, \ldots, C_{\mathsf{Sb}(u_1)}\}$ (set of disjoint subsets), and $dif_{A,B}$ be the corresponding value according to Table 1. For convenience we will refer to $\Delta_{i,j}(N, G)$ as simply $\Delta_{i,j}$.

$$\begin{aligned} \mathsf{d}^2(T, G) - \mathsf{d}^2(N, G) &= \sum_{\forall\{A,B\}} \sum_{\substack{i \in A \\ j \in B}} (\Delta_{i,j} + dif_{A,B})^2 - (\Delta_{i,j})^2 \\ &= \sum_{\forall\{A,B\}} (|A||B| dif_{A,B}^2 + 2 dif_{A,B} \sum_{\substack{i \in A \\ j \in B}} \Delta_{i,j}). \end{aligned} \tag{1}$$

Precomputation. The above equation shows that in order to efficiently calculate $\mathsf{d}(T, G)$ for an arbitrary $T \in SPR_v(S)$ with a fixed v, we need to know $\sum_{\substack{i \in A \\ j \in B}} \Delta_{i,j}$ for every pair of distinct A, B, such that $dif_{A,B} \neq 0$. Note that there are only $O(t)$ such pairs. Further, we observe that those sums can be exhaustively precomputed as the following values for each $u \in V(N)$.

- $\mathsf{BDist}(u) := \sum_{\substack{i \in C_v \\ j \in C_u}} \Delta_{i,j}$, for any $u \in V(S|v)$. This sum is called the *Base Distance (BDist)*: sum of path-length differences between all pairs of leaves from a subset C_u and C_v.
- $\mathsf{RDist}(u) := \sum_{\substack{i \in C_u \\ j \in \mathsf{L}(S|v) \setminus C_u}} \Delta_{i,j}$, for any $u \in V(S|v)$. This sum is called the *Remaining Distance (RDist)*: sum of path-length differences between all pairs of leaves from a subset C_u, and all the other leaves in N (excluding C_v).

Consider any node $u \in V(S|v)$. If u is not a leaf, then we denote its children as c_1 and c_2. $BDist(u)$ and $RDist(u)$ can be equivalently computed as follows.

- $\mathsf{BDist}(u) = \begin{cases} \mathsf{BDist}(c_1) + \mathsf{BDist}(c_2), & p \text{ is not a leaf;} \\ \sum_{i \in C_v} \Delta_{i,u}, & \text{otherwise.} \end{cases}$
- $\mathsf{RDist}(u) = \begin{cases} \mathsf{RDist}(c_1) + \mathsf{RDist}(c_2) - 2 \cdot \mathbf{SDist}(\mathbf{c_1}), & p \text{ is not a leaf;} \\ \sum_{i \in C_{u_0}} \Delta_{i,u}, & \text{otherwise.} \end{cases}$
- $\mathsf{SDist}(c_1) = \mathsf{SDist}(c_2) := \sum_{i \in C_{c_1}, j \in C_{c_2}} \Delta_{i,j}$ (*sibling distance*).

Time complexity. The precomputation step is divided into three sub-steps according to the relations for the distances BDist, SDist and RDist. Below we assess their computation time separately.

– BDist is calculated in constant time for internal nodes and in $O(|C_v|)$ time for leaves. Therefore, it requires $\Theta(n^2)$ operations to claculate BDist for all $u \in V(N)$.
– RDist is similar to BDist: it is calculated in constant time for internal nodes and in $O(|C_{u_0}|)$ for leaves. Once again, the overall time complexity is $\Theta(n^2)$.
– SDist is calculated across all siblings, and thus requires the computation of sums over multiple sub-matrices of Δ. However, these sub-matrices do not overlap for different pairs of siblings. Hence, the overall time complexity to compute SDist is $O(n^2)$.

After the precomputation step, the sums in Equation 1 can be substituted with BDist and RDist values in order to calculate $\mathsf{d}^2(T,G) - \mathsf{d}^2(N,G)$ in time $O(t)$, where $t \leq n$ for any $G \in SPR_S(v)$. This concludes the proof of Theorem 2.

Unrooted case. The PD median tree problem for unrooted trees is NP-hard, which follows from a straightforward polynomial time reduction from the NP-hard quartet compatibility problem [29] (as in the rooted case, we observe that the PD distance is 0 when all input trees are consistent with a supertree). Moreover, our local search algorithm for rooted trees can be extended to work with unrooted trees as well. We are describing the key ideas of this algorithm, and omitting details for brevity. The semi-structure of the SPR-neighborhood can be exploited in the unrooted case as well: one can root a supertree at an edge $(\mathsf{Pa}(v), v)$, where v is the "prune" node, and traverse the SPR-neighborhood in the same way as in the rooted case. Table 1 would be slightly changed to account for the artificial root, though the same precomputation idea is still applicable.

5 Experimental Evaluation

Median tree methods under the path-difference objective have never been studied before. Therefore, we adhere to a classical evaluation approach by comparing our median tree heuristic against standard supertree methods with different objectives [2, 20]. We processed two published baseline phylogenetic datasets, the Marsupials dataset [7] and the Cetartiodactyla dataset [25]. These datasets have been actively used for experimental supertree evaluations throughout the evolutionary community (see, for example, [2, 10, 16, 28]).

Following the experiments presented in one of the recent supertree papers [16], we compare our PD local search method against the following supertree methods: the maximum representation with parsimony (MRP) heuristic [31], the modified min-cut (MMC) algorithm [22], and the triplet supertree heuristic [16]. MRP heuristics are addressing the NP-hard MRP problem [20], and are among the most popular supertree methods in evolutionary biology [4]. For our evaluation we use the MRP local search heuristic implemented in PAUP* [31] with Tree Bisection and Reconnection (TBR) branch swapping [16]. The *TBR edit operation* is an extension of the SPR operation, where the pruned subtree is allowed to be re-rooted before regrafting it. The MMC algorithm computes supertrees (that satisfy certain desirable properties) in polynomial time, which makes this method especially attractive for large-scale phylogenetic analysis [22]. The triplet supertree heuristic is a local search heuristic that is addressing the well-studied

NP-hard triplet supertree problem [16]. We are using the triplet heuristic based on SPR and TBR local searches, called TH(SPR) and TH(TBR) respectively.

Hybrid heuristic. In a classical local search scenario there are two major steps. In the first step a supertree is constructed incrementally. Typically, the process is initiated with some t taxa, and an optimal supertree over the chosen taxa is computed exactly. Here, t is typically small, e.g., three. Next, on each iteration t new taxa are added to the partial supertree (an optimum among all possible ways to add t leaves to the tree is picked). The second major step is to run the actual local search starting with the tree obtained in step one.

Clearly, the first step is rather slow, especially when it is costly to compute the distance measure for a supertree (as in our case). Even though many ideas could be suggested to accelerate the first step, we want to emphasize that it is not necessary to separate the two steps in the first place. That is, the local search heuristic, which is the main optimization engine, could be applied on every step of construction of the start tree. It could be argued that SPR-based local search brings in more flexibility than simply trying to add new taxa to a tree with a fixed structure. For estimation of PD median trees we implemented this novel hybrid heuristic using the introduced here local search algorithm.

Results and Discussion. Table 2 summarizes the results that we obtained from the conducted experiments with our heuristic PDM(SPR) in comparison with the published results for MMC, MRP, TH(SPR), and TH(TBR) [16]. As expected, all of the methods stand their ground. The MRP method proves to be most effective according to the parsimony objective. In addition, MRP supertrees show the best fit over the input data in terms of our computed MAST-similarity scores – which could be seen as an "independent" objective in our evaluation. At the same time, both triplet heuristics, TH(SPR) and TH(TBR), produced the best supertrees under the triplet similarity objective. As for our method – PDM(SPR) – it was able to produce best supertrees with regards to the PD distance.

Data set	**Method**	PD score	Triplet-sim	MAST-sim	Pars. score
Marsuplial 158 input trees 272 taxa	MMC	16,670.45	51.73 %	53.4 %	3901
	MRP	5,694.59	98.29 %	**71.6** %	**2274**
	TH(SPR)	5,866.27	**98.99** %	70.2 %	2317
	TH(TBR)	5,888.22	**98.99** %	70.5 %	2317
	PDM(SPR)	**4,677.99**	68.43 %	63.4 %	3339
Cetartiodactyla 201 input trees 299 taxa	MMC	16,206.17	70.03 %	51.5 %	4929
	MRP	6,991.36	95.84 %	**64.7** %	**2603**
	TH(SPR)	7,630.03	**97.28** %	63.1 %	2754
	TH(TBR)	7,591.13	**97.28** %	63.0 %	2754
	PDM(SPR)	**6,051.13**	59.49 %	52.2 %	4162

Table 2: Summary of the experimental evaluation. The best scores under each objective are highlighted in bold.

In order to rigorously assess the results of our heuristic, we should contemplate them with the distribution of the path-difference distance for the two

datasets. However, such distributions, even for a single input tree, remain unknown [19]. Thus, following the approach from Steel and Penny [30], we estimate PD distance distributions based on sample data. For each dataset we generated two collections with 20,000 random supertrees using PAUP*. One collection was generated under the uniform binary tree distribution, and the other one was generated using the Markovian branching process [3]. Then, each collection was processed to obtain sample datasets with PD distance scores for every generated supertree. The obtained results are outlined in Figure 2.

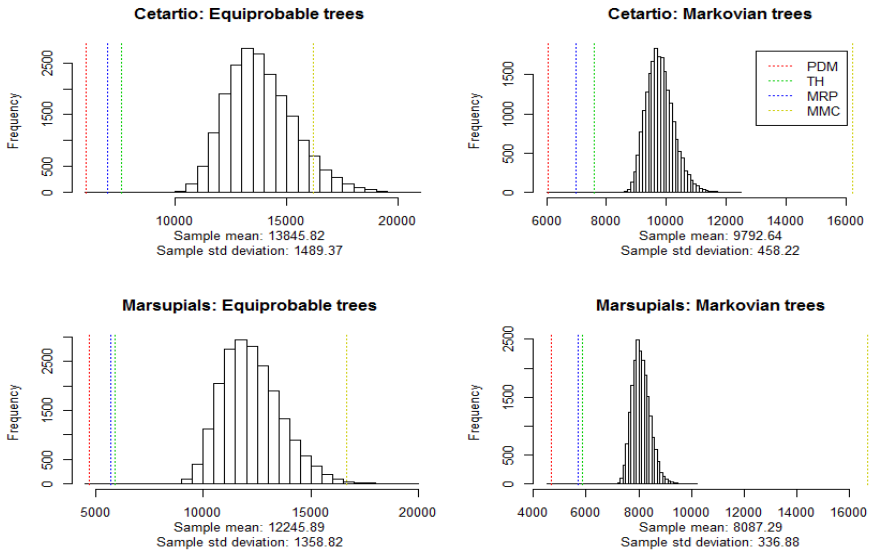

Fig. 2: Histograms of the PD distance based on the generated tree samples. All methods used for the evaluation are marked on each histogram with dotted lines.

The figure makes it clear that even though our heuristic was able to obtain the best results for the two datasets, MRP and Triplet heuristics still produce trees that are significantly better than any of the randomly generated trees. As for the MMC algorithm, it performs much worse under the PD objective than simply constructing Markovian Binary trees; and, what is more, worse than drawing random trees from the uniform distribution.

Figure 2 suggests that there exists a positive correlation between the Parsimony, Triplet-similarity, and PD distance supertree objectives. On the other hand, according to Table 2, better PD supertrees do not necessarily score well in terms of parsimony and triplet measures. Thus, the PD heuristic might produce structurally new phylogenetic trees that have not been analyzed previously.

6 Conclusion

We synthesized the first large-scale median trees under one of the oldest and widely popular tree distance metrics — the path-difference distance. While we show that the corresponding PD median tree problem is NP-hard, we demonstrated that it can be successfully approached by using our new SPR based local

search heuristic. To make the heuristic applicable to real-world phylogenetic datasets, we have significantly improved its time complexity in comparison to the best known naïve approach.

Currently, no mainstream supertree method can construct edge-weighted supertrees. However, there has been an increased interest in such tools due to fast developing databases of time-annotated evolutionary trees (e.g., TimeTree [15]). The path-difference distance on the other hand, is naturally extendable to account for edge-weights in phylogenetic trees. The introduced PD heuristic, in turn, could also be adapted to deal with edge-weighted supertrees. This property makes the PD distance even more appealing as a median tree objective and suggests further investigation in its theoretical and algorithmic means.

7 Acknowledgments

The authors would like to thank the two anonymous reviewers for their constructive comments that helped to improve the quality of this work.

References

1. Bansal, M.S., Burleigh, J.G., Eulenstein, O.: Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models. BMC Bioinformatics 11 Suppl 1, S42 (2010)
2. Bansal, M.S., Burleigh, J.G., Eulenstein, O., Fernández-Baca, D.: Robinson-foulds supertrees. Algorithms for Molecular Biology 5(1), 1–12 (2010)
3. Bean, N.G., Kontoleon, N., Taylor, P.G.: Markovian trees: properties and algorithms. Annals of Operations Research 160(1), 31–50 (2007)
4. Bininda-Emonds, O.R. (ed.): Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life, Computational Biology, vol. 4. Springer Verlag (2004)
5. Bluis, J., Shin, D.: Nodal distance algorithm: Calculating a phylogenetic tree comparison metric. In: 3rd IEEE International Symposium on BioInformatics and BioEngineering (BIBE 2003), 10-12 March 2003, Bethesda, MD, USA. pp. 87–94. IEEE Computer Society (2003)
6. Bryant, D.: Hunting for trees in binary character sets: efficient algorithms for extraction, enumeration, and optimization. J Comput Biol 3(2), 275–288 (1996)
7. Cardillo, M., Bininda-Emonds, O.R.P., Boakes, E., Purvis, A.: A species-level phylogenetic supertree of marsupials. Journal of Zoology 264, 11–31 (2004)
8. Chaudhari, R., Burleigh, G.J., Eulenstein, O.: Efficient Algorithms for Rapid Error Correction for Gene Tree Reconciliation using Gene Duplications, Gene Duplication and Loss, and Deep Coalescence. BMC Bioinformatics 13 Suppl 10, S11 (2012)
9. Chaudhary, R., Bansal, M.S., Wehe, A., Fernández-Baca, D., Eulenstein, O.: iGTP: a software package for large-scale gene tree parsimony analysis. BMC Bioinformatics 11, 574 (2010)
10. Chen, D., Eulenstein, O., Fernández-Baca, D., Burleigh, J.: Improved heuristics for minimum-flip supertree construction. Evolutionary Bioinformatics 2 (2006)
11. Cotton, J.A., Wilkinson, M.: Majority-rule supertrees. Syst Biol 56(3), 445–452 (2007)
12. Farris, J.: A successive approximations approach to character weighting. Systematic Zoology 18, 374–385 (1969)

13. Harris, S.R., Cartwright, E.J., Török, M.E., Holden, M.T., Brown, N.M., Ogilvy-Stuart, A.L., Ellington, M.J., Quail, M.A., Bentley, S.D., Parkhill, J., Peacock, S.J.: Whole-genome sequencing for analysis of an outbreak of meticillin-resistant staphylococcus aureus: a descriptive study. Lancet Infect Dis 13(2), 130–6 (2013)
14. Hufbauer, R.A., Marrs, R.A., Jackson, A.K., Sforza, R., Bais, H.P., Vivanco, J.M., Carney, S.E.: Population structure, ploidy levels and allelopathy of Centaurea maculosa (spotted knapweed) and C. diffusa (diffuse knapweed) in North America and Eurasia. In: Proceedings of the XI International Symposium on Biological Control of Weeds, Canberra Australia. pp. 121–126. USDA Forest Service. Forest Health Technology Enterprise Team, Morgantown, WV. (2003)
15. Leaché, A.D.: The timetree of life. S. Blair Hedges and Sudhir Kumar, editors. Integrative and Comparative Biology 50(1), 141–142 (2010)
16. Lin, H.T., Burleigh, J.G., Eulenstein, O.: Triplet supertree heuristics for the tree of life. BMC Bioinformatics 10(Suppl 1), S8 (2009)
17. Lin, H.T., Burleigh, J.G., Eulenstein, O.: Consensus properties for the deep coalescence problem and their application for scalable tree search. BMC Bioinformatics 13 Suppl 10, S12 (2012)
18. Maddison, W.P., Knowles, L.L.: Inferring phylogeny despite incomplete lineage sorting. Syst Biol 55(1), 21–30 (2006)
19. Mir, A., Rosselló, F.: The mean value of the squared path-difference distance for rooted phylogenetic trees. CoRR abs/0906.2470 (2009)
20. Moran, S., Rao, S., Snir, S.: Using semi-definite programming to enhance supertree resolvability. In: Proceedings of the 5th International Conference on Algorithms in Bioinformatics. pp. 89–103. WABI'05, Springer-Verlag, Berlin, Heidelberg (2005)
21. Nik-Zainal, S., et al.: The life history of 21 breast cancers. Cell 149(5), 994–1007 (2012)
22. Page, R.D.M.: Modified mincut supertrees. In: Proceedings of the Second International Workshop on Algorithms in Bioinformatics. pp. 537–552. WABI '02, Springer-Verlag, London, UK, UK (2002)
23. Page, R.D., Holmes, E.: Molecular evolution: a phylogenetic approach. Blackwell Science (1998)
24. Phipps, J.B.: Dendogram topology. Systematic Zoology 20, 306–308 (1971)
25. Price, S.A., Bininda-Emonds, O.R.P., Gittleman, J.L.: A complete phylogeny of the whales, dolphins and even-toed hoofed mammals (cetartiodactyla). Biological Reviews 80(3), 445–473 (2005)
26. Puigbò, P., Garcia-Vallvé, S., McInerney, J.O.: TOPD/FMTS: a new software to compare phylogenetic trees. Bioinformatics 23(12), 1556–1558 (2007)
27. Semple, C., Steel, M.A.: Phylogenetics. University Press, Oxford (2003)
28. Snir, S., Rao, S.: Quartets maxcut: A divide and conquer quartets algorithm. IEEE/ACM TCBB 7(4), 704–718 (2010)
29. Steel, M.: The complexity of reconstructing trees from qualitative characters and subtrees. Journal of Classification 9(1), 91–116 (1992)
30. Steel, M.A., Penny, D.: Distributions of tree comparison metrics - some new results. Systematic Biology 42(2), 126–141 (1993)
31. Swofford, D.L.: PAUP*. Phylogenetic analysis using parsimony (*and other methods). Version 4. Sinauer Associates, Sunderland, Massachusetts. (2002)
32. Than, C., Nakhleh, L.: Species tree inference by minimizing deep coalescences. PLoS Comput Biol 5(9), e1000501 (2009)
33. Williams, W., Clifford, H.: On the Comparison of Two Classifications of the Same Set of Elements. Taxon 20(4), 519–522 (1971)