Manhattan Path-Difference Median Trees

Alexey Markin
Department of Computer Science
Iowa State University
Ames, IA 50011, USA
amarkin@iastate.edu

Oliver Eulenstein
Department of Computer Science
Iowa State University
Ames, IA 50011, USA
oeulenst@iastate.edu

ABSTRACT

Median tree problems are a powerful tool for inferring large-scale phylogenetic trees that hold enormous promise for society at large. Such problems seek a median tree for a given collection of input trees under some problem-specific distance. Here, we introduce this problem for the classic Manhattan path-difference distance and show that this problem is NP-hard. To address this inherent time complexity we devise an ILP formulation, and an effective local search heuristic that is based on solving a local search problem exactly. Our algorithm for the local search problem improves asymptotically by a factor of $n/\log n$ on the best-known (naïve) solution, where n is the size of the input trees. Finally, we demonstrate the ability of our heuristic in a comparative study using large-scale published empirical data sets, and showing its accuracy for small phylogenetic studies by using exact ILP solutions.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Trees*; F.2 [**Theory of Computation**]: Analysis of algorithms and problem complexity

General Terms

Algorithms, Performance

Keywords

Path-Difference distance, median trees, supertrees, Manhattan distance, local search, phylogenetics

1. INTRODUCTION

A basic tenet of all biological disciplines is the common history of all life forms, including all extant species. The evolutionary relationships among such entities are usually represented as a bifurcating tree, which is referred to as a species tree. Large-scale species trees for thousands of entities hold enormous promise for society at large. While such trees are fundamental throughout biology, their analysis and predictive power is also benefitting various other disciplines, such as agronomy, biochemistry, epidemiology, environmental sciences, and medical sciences [17, 18, 27, 33]. At the same time inferring large-scale species trees confronts us with some of the most difficult computational challenges raised in the field of evolutionary biology today.

Prior to the genomic era, a species tree for a given set of species was built by constructing an evolutionary tree from a common gene sampled from those species. Such trees are called gene trees. The implicit assumption of this approach is that the evolution of the chosen gene mimics the evolution of the species themselves. A major problem with this approach is that gene trees for distinct genes can be discordant for the same set of species, and thus, may not accurately reflect the actual species tree. Such discordance is frequently caused by erroneous gene trees, but can also be caused by genes that have evolved differently due to complex evolutionary processes that have shaped the species' genomes. Furthermore, genes are often only partially sequenced and not expressed in every species, which results in typically smaller gene trees.

The advent of new sequencing technologies provided an unprecedented wealth of new gene sequences, holding the promise of building more resolved and less biased species trees [29]. In contrast to building a species tree based on only one gene tree, discordance can now be averaged out based on the cumulative evidence from thousands of genes sampled from across the species genomes [14, 24].

Median tree problems have emerged as a powerful tool for assembling large-scale species trees from discordant gene trees, which may represent the evolution of different sets of species [4]. Such problems seek a median tree for a given set of input (gene) trees based on some problem-specific distance; that is, a tree with the minimum overall distance to the input trees.

Median tree problems have been studied extensively for many of the well-established distances in comparative phylogenetics. Notable exceptions, however, are some of the oldest distances in comparative phylogenetics, namely the path-difference distances. Given two trees, path-difference distances measure the differences of the path-lengths between any pair of leaves between the two trees. Various norms can be applied to this distance, such as the Manhattan norm (also known as the Taxicab norm) and the Euclidean norm, called the *Manhattan path-difference distance* and the

BCB '16, October 02-05, 2016, Seattle, WA, USA

DOI: http://dx.doi.org/10.1145/2975167.2975209

Euclidean path-difference distance respectively. The appeal of path-difference distances is that, unlike many other distances in comparative phylogenetics [38], they can be used to compare any type of trees, including rooted trees, unrooted trees, and non-binary trees. In fact, while most distances are not defined to compare trees with branch-lengths, path-difference distances allow the comparison of such trees.

Despite the appeal and tradition of path-difference distances the analysis of median trees under these distances is still in its infancy. Only recently the median tree problem under the Euclidean path-difference distance has been analyzed. While this problem is NP-hard, local search heuristics have been developed that allow the inference of credible species tree estimates [23]. However, median tree problems under any other norm for the path-differences have not been analyzed.

The focus of this work is studying the computation of median trees under the Manhattan path-difference distance, which we call the *Manhattan median tree problem*. We show that this problem is NP-hard by proving the more general statement that the median tree problem is NP-hard under each norm of path-differences. However, while almost all median tree problems that are typically 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 to compute large-scale median trees. Given the promising results achieved by these heuristics, we introduce a local search heuristic for the Manhattan median tree problem. However, the best-known (naïve) solution for the corresponding local search problem requires $\Theta(kn^4)$ time, where n and k is the size and number of trees in a given instance of the problem, which is prohibitive for most large-scale instances of the problem. Here we describe a novel algorithm for this local search problem that improves asymptotically on the original runtime by a factor of $n/\log n$, which allows to compute the first large-scale Manhattan median trees. Further, we provide an exact *integer linear programming (ILP)* formulation of the Manhattan median tree problem with $O(n^3 + kn^2)$ variables and constraints that allows us to compute exact solutions for instances with up to 8 species on a standard workstation within a few minutes. In order to analyze distances the knowledge about their distribution is vital. Therefore, we have provided a sample distribution of the distances for Manhattan median trees. Finally, we demonstrate the ability of our local search heuristic. In a comparative study on published empirical data sets we show that our heuristic outperforms other standard heuristics in minimizing the overall Manhattan median tree distance. Further, we show that our heuristic solves the Manhattan median tree problem exactly for instances with 8 species by using our exact ILP solutions for comparison. The software for our local search heuristic is freely available from the authors.

1.1 Related Work

There has been a large body of work focusing on the biological, mathematical, and algorithmic properties of median trees adopting various definitions of distance measures that have been effectively used in comparative phylogenetics [4]. Path-difference distances describe some of the oldest such distances [5, 13, 32, 38] and are defined by a distance-specific norm of the difference between the path-lengths vectors of the two trees whose distance is measured. Each of these *vectors* represents the pairwise distances between the leaves of the corresponding tree that is the number of edges on the simple path between the leaves. Steel and Penny [38] have studied the distribution of the Euclidean (L_2 norm) path-difference distance for unrooted trees. Later, Mir and Rosello [25] computed the mean value of this distance for fully resolved unrooted trees with n leaves, and showed that this mean value grows in $O(n^3)$. Recently it was shown that the median tree problem under this distance is NP-hard and effective local search heuristics have been developed [23]. Variants of path-difference distances have been described, including the Manhattan (L_1 norm) path-difference distance [42], and their correlation [30].

While most median tree problems are NP-hard [4], they have been effectively addressed by local search heuristics [1, 10, 21, 22, 41], which have provided credible estimates of large-scale species trees [22, 41]. Such a search heuristic starts with some initial candidate species tree, and finds a tree with the minimum overall distance to the given input trees in the local neighborhood of the initial tree. This constitutes a *local search step*. The tree found in a local search step becomes the starting point for the following local search step, and so on, until a local minima is reached, which is reported by the heuristic. To find a tree with minimum distance in a local search step an instance of a *local search problem* is solved exactly. The time complexity of this local search problem depends on the tree edit operation that defines the local neighborhood, as well as on the computation time of the tree distance measure that is used. The classic *subtree prune and regraft (SPR)* tree edit operation where a subtree is pruned from the edited tree and then regrafted back into this tree at another location has been well-studied [34]. The *SPR neighborhood* of a tree T is the set of all trees into which T can be transformed by at most one SPR edit operation, which consists of $\Theta(n^2)$ trees, where n is the size of T. Now, solving the local search problem for the Manhattan median tree problem naïvely requires $\Theta(kn^4)$ time, since the best-known algorithm to compute the Manhattan pairwise distance for one candidate tree and k input trees requires $\Theta(kn^2)$ time. However, computing large-scale species trees requires typically solving thousands of instances of the local search problem, and a local search runtime of $\Theta(kn^4)$ becomes prohibitive.

1.2 Contribution

We introduce the Manhattan median tree problem. This problem is NP-hard for its rooted and unrooted variants, since, as we show, each of these variants contains an NP-hard problem as a special case. Consequently, we apply a standard SPR based local search heuristic to the Manhattan median tree problem. Our efficient algorithm for the corresponding SPR local search problem makes this heuristic suitable for phylogenetic analyzes that allowed us to compute the first large-scale Manhattan median trees. This algorithm is based on an efficient algorithm that solves the SPR local search problem for the Euclidean median tree problem [23]. The baseline analysis of the effect of a single SPR operation on path-differences between two trees used for Euclidean local search algorithm was given in [23]. The difference between the Manhattan distance that involves absolute values and the Euclidean distance, however, prevents us from applying the same precomputation idea, which was the core algorithmic part of the Euclidean median tree heuristic.

Hence, in this work we developed a more general precomputation method that allows us to efficiently answer more elaborate queries to path-difference matrices as required by the Manhattan median tree heuristic. The algorithm solves an instance of the local search problem in $O(kn^3 \log n)$ time, which is a significant improvement in comparison to the best-known naïve approach when dealing with large-scale tree assembly problems.

As for the ILP formulation of the problem, the search space is encoded using the matrix-based tree representation introduced in [8]. Building on top of that we introduced variables and constraints allowing to infer path-lengths between all pairs of leaves. The main challenge in this formulation was to account for the minus method [12] that is used in this work for the path-difference distance calculation.

We evaluate the efficiency and effectiveness of our local search algorithm in a comparative study involving other popular supertree methods and different cost functions on standard phylogenetic datasets. Additionally, we show the accuracy of our local search heuristic by comparing its results with the exact solutions provided by our ILP formulation on small simulated data sets.

2. BASICS AND PRELIMINARIES

2.1 Basic definitions

Throughout this paper we adhere to the definitions and notation introduced in [23]. A *(phylogenetic) tree* T is a rooted tree, where each leaf is uniquely labeled with a taxon, each internal node v has exactly two children nodes denoted by $\mathsf{Ch}_T(v)$, and each node u except for the root has a single parent node denoted by $\mathsf{Pa}_T(v)$. In addition, we denote the node set, edge set and leaf set of T by $V(T)$, $E(T)$ and $\mathsf{L}(T)$ respectively. We denote the root by $\mathsf{Rt}(T)$ and a sibling of each non-root node u by $\mathsf{Sb}(u)$. We also set $T(v)$ to be a subtree of T rooted at $v \in V(T)$, and $T|v$ to be a phylogenetic tree obtained by pruning $T(v)$ from T.

A set of leaves $\mathsf{L}(T(v))$ is called a *cluster* of the node v. When v is not a leave and not a root of T we say that the corresponding cluster is *non-trivial*. Note that for convenience we identify the leaves in a phylogenetic tree with the respective 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 (a parent and a child).

Let $\mathcal{P}$ be a set of phylogenetic trees $\{G_1, \ldots, G_k\}$. We extend the definition of a leaf set to a set of trees as follows: $\mathsf{L}(\mathcal{P}) := \cup_{i=1}^{k} \mathsf{L}(G_i)$. A tree S is called a *supertree* of $\mathcal{P}$, if $\mathsf{L}(S) = \mathsf{L}(\mathcal{P})$. 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 $T[\mathsf{L}(G)] \equiv G$.

2.2 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 *path-length vector* obtained by a fixed ordering of pairs i, j [38], 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 (for a fixed $p \in [1, \infty)$) as

$$\mathsf{d}_p(G, S) := ||\, \mathsf{d}(G) - \mathsf{d}(S)||_p$$

Where $||\cdot||_p$ denotes an L_p norm of a vector.

Further, we define $\Delta(G, S)$ to be a *path-difference matrix* of size $|\,\mathsf{L}(T)| \times |\,\mathsf{L}(T)|$. Thats is, $\Delta_{i,j}(G, S) = \mathsf{d}_{i,j}(G) - \mathsf{d}_{i,j}(S)$.

3. PATH-DIFFERENCE MEDIAN TREES

Here we introduce a class of path-difference based median tree problems defined under different vector norms.

We extend the definition of the path-difference distance to a set of trees. Note that so far PDD is defined 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, because tree-size variations are typically present in real world data. Therefore, in order to calculate a path-difference distance under an L_p norm between two trees S and G, where $\mathsf{L}(G) \subseteq \mathsf{L}(S)$ we use the *minus method* [12]. That is, we calculate a distance between G and the subtree of S induced by $\mathsf{L}(G)$: $\mathsf{d}_p(S, G) = \mathsf{d}_p(S[\mathsf{L}(G)], G)$. We now define PDD for an input set $\mathcal{P}$ and a supertree S as a sum

$$\mathsf{d}_p(\mathcal{P}, S) := \sum_{i=1}^{k} \mathsf{d}_p(G_i, S[\mathsf{L}(G_i)]).$$

Given that we establish the following general problem:

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

The PD median tree problem is NP-complete.
We show NP-hardness by a polynomial time reduction from the MaxRTC problem, which is know to be NP-complete (see [6]).

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 $|R'| \geq c$.

Where a *rooted triplet* is a phylogenetic tree with exactly three leaves.

THEOREM 3.1. *The PD median tree problem under an L_p norm is NP-complete for any $1 \leq p < \infty$.*

PROOF. Clearly, this problem belongs to NP: given a supertree S we can in polynomial time determine whether $\mathsf{d}_p(R, S) \leq q$ by directly calculating the distance. The rest of the proof is almost identical to the proof of the special case of this theorem for the L_2 norm given in [23]. Below we provide its brief generalization.

We map an instance $\langle R, c\rangle$ of the MaxRTC problem to an instance $\langle R, 2^{\frac{1}{p}}(|R| - c)\rangle$ of the PD median tree problem. To explain why this transformation works we observe the following. Assume that S is a supertree of a set of rooted triplets $R = \{T_1, \ldots, T_k\}$. Then $\mathsf{d}_p(S[\mathsf{L}(T_i)], T_i)$ is 0, when S is *consistent* with T_i, and is $2^{\frac{1}{p}}$ otherwise. Therefore, $\mathsf{d}_p(R, S) = 2^{\frac{1}{p}}(|R| - c')$, 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. □

Unrooted median trees. While in this work we focus on rooted phylogenetic trees, it is not difficult to see that the unrooted version of the PD median tree problem is NP-hard as well. To show that we use the classic NP-complete quartet compatibility problem [37].

Problem 3 (Quartet compatibility).
Instance: A set of quartets Q;
Question: Is there a supertree S that is consistent with every quartet in Q.

Where a *quartet* is an unrooted phylogenetic tree with four leaves. Note that through the unrooted PD median tree problem we can solve the quartet compatibility problem, since for a supertree S, $\mathsf{d}_p(S,Q) = 0$ if and only if S is consistent with every tree in Q. Therefore, unrooted PD median tree problem is NP-hard.

Although results in this section are shown for *sets* of input trees, they easily extend to *multi-sets*. Note that further in this work we focus on an L_1 norm PDD, or as we additionally refer to it, *Manhattan PDD (MPDD)*.

4. LOCAL SEARCH

One of the most widely used ways to approach NP-hard supertree problems is a local search heuristic (hill climbing heuristic). Local search on supertrees is generally defined in terms of tree edit operations. In this paper we focus on *SPR* – one of the most applied tree edit operations (see [2,39,40]).

DEFINITION 4.1. *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 by the following modifications of the tree $S' = S|v$:*

1. *If u is a root of S', 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'.*
2. *Connect the pruned subtree $S(v)$ to the node w'.*

In addition, we introduce a useful notation

$$SPR_S(v) := \bigcup_u SPR_S(v,u); \quad SPR_S := \bigcup_{v,u} SPR_S(v,u)$$

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 $\mathcal{P} = \{G_1, ..., G_k\}$, the search space in the supertree problem could be viewed as a graph $\mathcal{T}$, where nodes represent supertrees of $\mathcal{P}$. $\{S_1, S_2\}$ is an edge in $\mathcal{T}$, if S_1 could be transformed to S_2 with a single SPR operation.

At each iteration local search heuristic finds a supertree S' in the neighborhood of a current tree S, such that S' minimizes the Manhattan PD distance. In case $S \equiv S'$, the local search stops (reaches a local minimum). Otherwise, it proceeds to the next iteration with a tree S'. An instance (single iteration) of the SPR-based local search algorithm could be formalized as the following problem:

Problem 4 (PD Local Search).
Instance: An input set $\mathcal{P}$ and a supertree S
Find: find $S' = \arg\min_{S' \in SPR_S} \mathsf{d}_p(\mathcal{P}, S')$.

Naïve algorithm for the PD Local Search problem. Given two trees S and G, one can compute $\mathsf{d}_p(S,G)$ in $O(n^2)$ time for any p. Therefore, direct computation of the $\mathsf{d}_p(\mathcal{P}, S')$ score for each $S' \in SPR_S$ would take $O(n^4k)$ time, where $n = |\,\mathsf{L}(\mathcal{P})|$ and $k = |\,\mathcal{P}\,|$. Next, we show to improve on this complexity.

Euclidean median tree algorithm description. In [23] authors present an algorithm for the PD local search problem under the L_2 norm that works an order of n faster than the naïve algorithm. To summarize the main idea behind this algorithm we need to first establish the settings and notation:
Let S_i be an initial supertree in the i-th iteration of a local search and $G \in \mathcal{P}$ be a fixed input tree. Through the rest of the section we will refer to $S_i[\mathsf{L}(G)]$ as simply S. We further note that due to [9] for a tree $N := SPR_S(v, \mathsf{Rt}(S))$, a local SPR-neighborhood $SPR_N(v)$ is equivalent to $SPR_S(v)$ for a fixed $v \in V(S)$. Finally, we let C_u denote $\mathsf{L}(N(u))$ for any $u \in V(N)$.

The algorithm in [23] exploits the idea that for a fixed node $v \in V(S)$, we can efficiently find $\mathsf{d}_2(T,G)$ for all $T \in SPR_S(v)$ by precomputing certain values for all vertices in S beforehand.

More formally, let's 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 $u_t = y$. Then Table 1 captures the way path-lengths change with regrafting v above y. That is, it represents the $\Delta(T,G) - \Delta(T,G)$ matrix. It becomes immediately apparent from the table that path-lengths change by some constants only between $O(t)$ pairs of node clusters, which together with a precomputation step allow for an efficient traversal of the SPR-neighborhood.

While dealing with an L_1 norm (Manhattan distance), however, the same precomputation idea does not immediately apply. This is caused by the nature of the absolute-value function, as we will see later. Next, we present a similar, but more involved precomputation idea that allows us to overcome that obstacle.

Manhattan median tree local search algorithm. 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 clusters), 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 $\Delta_{i,j}$.

$$\mathsf{d}_1(T,G) - \mathsf{d}_1(N,G) = \sum_{\forall\{A,B\}} \sum_{\substack{i\in A\\ j\in B}} |\Delta_{i,j} + dif_{A,B}| - |\Delta_{i,j}|$$

$$= \sum_{\forall\{A,B\}} \begin{pmatrix} dif_{A,B} \cdot \#\{(i\in A, j\in B) | \Delta_{i,j} \geq -dif_{A,B})\} \\ -dif_{A,B} \cdot \#\{(i\in A, j\in B) | \Delta_{i,j} < -dif_{A,B})\} \\ +2 \sum\limits_{i\in A, j\in B: -dif_{A,B} \leq \Delta_{i,j} < 0} \Delta_{i,j} \\ -2 \sum\limits_{i\in A, j\in B: 0 \leq \Delta_{i,j} < -dif_{A,B}} \Delta_{i,j} \end{pmatrix} \quad (1)$$

The above equation shows that in order to efficiently compute a difference $\mathsf{d}_1(T,G) - \mathsf{d}_1(N,G)$ (e.g., in linear time over n) we need to know the values of

- $\#\{(i\in A, j\in B) | \Delta_{i,j} \geq -dif_{A,B})\}$,
- $\sum\limits_{i\in A, j\in B} \Delta_{i,j}$
  where $min(0, -dif_{A,B}) \leq \Delta_{i,j} < max(0, -dif_{A,B})$.

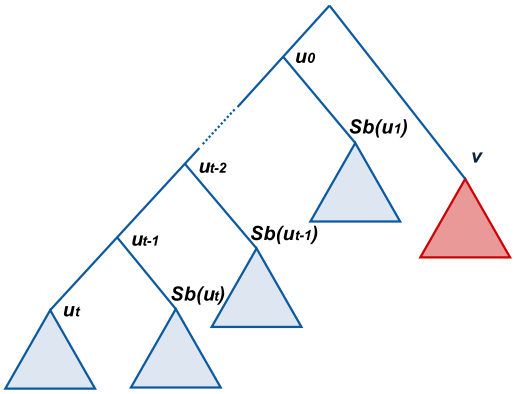

Figure 1: Scheme of the $N = SPR_S(v, \mathsf{Rt}(S))$ tree, depicting how the leaf set was partitioned to create Table 1.

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 $j \in C_{u_t}$: $\mathsf{d}_{i,j}(T) = \mathsf{d}_{i,j}(N) - t$.

	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

for every pair of distinct A, B, such that $dif_{A,B} \neq 0$.

Precomputation. Basically, as the Equation 1 implies, we need to be able to efficiently answer queries of kind: find a sum/count of elements of a certain submatrix of Δ, such that those elements are between specified bounds. In order to do that, we preprocess a tree N and a matrix Δ by precomputing some sums and counts.

First, we introduce a useful notation. Let $\mathsf{L}_1, \mathsf{L}_2 \subseteq \mathsf{L}(N)$, then $\Sigma_{\geq}(\mathsf{L}_1, \mathsf{L}_2)$ is a vector indexed from $-(n-2)$ to $n-2$, such that

$$\Sigma_{\geq}(\mathsf{L}_1, \mathsf{L}_2)[x] = \sum_{\substack{i \in \mathsf{L}_1 - \mathsf{L}_2: \ \Delta_{i,j} \geq x \\ j \in \mathsf{L}_2 - \mathsf{L}_1}} \Delta_{i,j}$$

That is, a sum of path-differences across two subsets of leafs, such that those path-differences are greater than or equal to a parameter x. The reason why we need such vectors becomes immediately clear, when we observe, for example, the following relation (assuming that $dif_{A,B} \geq 0$):

$$\sum_{\substack{i \in A \\ j \in B: \ -dif_{A,B} \leq \Delta_{i,j} < 0}} \Delta_{i,j} = \Sigma_{\geq}(A, B)[-dif_{A,B}] - \Sigma_{\geq}(A, B)[0].$$

Clearly, by having those vectors we will be able to answer sum-related queries. Further, we introduce a similar notation for "counts": $\#_{\geq}(\mathsf{L}_1, \mathsf{L}_2)$ is a vector indexed from $-(n-2)$ to $n-2$, such that

$$\#_{\geq}(\mathsf{L}_1, \mathsf{L}_2)[x] = \#\{(i \in \mathsf{L}_1 - \mathsf{L}_2, j \in \mathsf{L}_2 - \mathsf{L}_1) | \Delta_{i,j} \geq x\}$$

According to the Table 1 in order to find the $\mathsf{d}_1(T, G) - \mathsf{d}_1(N, G)$ difference in $O(t)$ time for any $T \in SPR_S(v)$, we need to precompute

1. the vectors $\Sigma_{\geq}(C_v, C_u)$ and $\#_{\geq}(C_v, C_u)$,
2. values $\#_{\geq}(C_{u_0}, C_u)[-1]$ and $\#_{\geq}(C_{u_0}, C_u)[0]$

for all $u \in V(N|v)$. The second item is restricted to only those two values, because a distance between a cluster C_u and all other leaves (except for C_v) can either remain unchanged or increase by 1 (see the second column of the Table 1). In that case

$$\sum_{\substack{i \in C_u - C_v: \ -1 \leq \Delta_{i,j} < 0 \\ j \in C_u}} \Delta_{i,j} = -\begin{pmatrix} \#_{\geq}(C_{u_0}, C_u)[-1] \\ -\#_{\geq}(C_{u_0}, C_u)[0] \end{pmatrix}.$$

Let's fix an internal node $u \in V(N|v)$ with two children c_1 and c_2. We now observe that the above values can be computed dynamically using the following optimal substructure:

1. $\Sigma_{\geq}(C_v, C_u) = \Sigma_{\geq}(C_v, C_{c_1}) + \Sigma_{\geq}(C_v, C_{c_1})$,
2. $\#_{\geq}(C_v, C_u) = \#_{\geq}(C_v, C_{c_1}) + \#_{\geq}(C_v, C_{c_1})$,
3. $\#_{\geq}(C_{u_0}, C_u) = \#_{\geq}(C_{u_0}, C_{c_1}) + \#_{\geq}(C_{u_0}, C_{c_1}) - 2\#_{\geq}(C_{c_1}, C_{c_2})$

Time complexity. While analyzing the time complexity of the precomputation step we are mostly interested in the base cases, i.e., the computation of $\Sigma_{\geq}$ and $\#_{\geq}$ vectors for leaf-nodes, because a linear combination of vectors (where the number of vectors is constant) of size $O(n)$ could be computed in $O(n)$ time.

- $\Sigma_{\geq}(C_v, C_u)$ and $\#_{\geq}(C_v, C_u)$ for a leaf-node u could be computed by first sorting the array of path-differences between u and leaves in C_v and then filling out the vectors in linear time over $|C_v|$. This results in the base case time complexity for a single leaf-node of $O(n \log(n))$. Overall, computation of the vectors for all $u \in V(N|v)$ takes $O(n^2 \log(n))$ time.

- Computation of values $\#_{\geq}(C_{u_0}, C_u)[-1]$ and $\#_{\geq}(C_{u_0}, C_u)[0]$ for a leaf-node u could be performed in $O(n)$ time by going through all the corresponding entries in the matrix Δ. It is also easy to see that computation of $\#_{\geq}(C_{c_1}, C_{c_2})[-1]$ and $\#_{\geq}(C_{c_1}, C_{c_2})[0]$ values for all pairs of sibling-nodes (c_1, c_2) takes $O(n^2)$ time. Therefore, the computation of $\#_{\geq}(C_{u_0}, C_u)[-1]$ and $\#_{\geq}(C_{u_0}, C_u)[0]$ for all $u \in V(N|v)$ takes $O(n^2)$ time overall.

Summing it up, we obtain the total complexity of a precomputation step of $O(n^2 \log(n))$. Using the precomputed vectors one can traverse the $SPR_S(v)$ neighborhood in $O(n^2)$ time. This analysis was performed for a fixed input tree G and a fixed prune-node v. Thus, the overall time complexity of a single iteration of the local search algorithm is $\mathbf{O(kn^3 \log n)}$.

5. ILP FORMULATION

In this section we present an ILP formulation for the Manhattan median tree problem, which allows us to solve small instances of the problem *exactly*.

In order to encode a space of phylogenetic trees we utilize the structure presented in [8]. A phylogenetic tree is encoded as a binary hierarchy – a binary matrix of size $n \times n-2$, where rows represent taxa (one row for each taxon), and columns represent non-trivial clusters of the tree. A binary variable $M_{i,c} = 1$ if and only if a taxon i is contained in the cluster c.

To ensure that a matrix contains only non-trivial clusters, we add the following constraint:

$$2 \leq \textstyle\sum_{i=1}^{n} M_{i,c} \leq n-2, \quad \forall 1 \leq c \leq n-2.$$

Compatibility. Further, the *three-gamete* condition [16] are used in order to make sure that a matrix M represents a valid phylogenetic tree, i.e., clusters are compatible. The $(0,1)$, $(1,0)$ and $(1,1)$ *gametes* are inferred using the following variables for all $1 \leq i \leq n$, $1 \leq c_1, c_2 \leq n-2$ (see [8] for more details):

$$\begin{aligned} C_{01}(c_1,c_2) &\geq -M_{i,c_1} + M_{i,c_2} \\ C_{10}(c_1,c_2) &\geq M_{i,c_1} - M_{i,c_2} \\ C_{11}(c_1,c_2) &\geq M_{i,c_1} + M_{i,c_2} - 1 \end{aligned}$$

Now to enforce compatibility for any two clusters c_1 and c_2 we add the constraint $C_{01}(c_1,c_2)+C_{10}(c_1,c_2)+C_{11}(c_1,c_2) = 2$.

Uniqueness. If we rearrange the columns of a matrix M, a corresponding tree would not change. Therefore, to enforce unique matrix representation of a tree we supply a linear order on the columns (treating columns as binary integers), and thereby prohibiting column rearrangements:

$$\sum_{i=1}^{n} 2^{i-1} M_{i,c} \geq \sum_{i=1}^{n} 2^{i-1} M_{i,c+1}, \quad \forall 1 \leq c \leq n-3$$

Path-Difference variables. Note that all the variables that we are going to introduce are integers. Let $PL(i,j)$ be a variable that equals to a path-length between two taxa i and j in a tree S represented by the matrix M. We define it as follows:

$$PL(i,j) = D(i) + D(j) - 2D(i,j), \quad \forall 1 \leq i,j \leq n$$

Here $D(i)$ is a depth of a taxon i in S (length of the path between $\mathsf{Rt}(S)$ and i). While $D(i,j)$ is a depth of a least common ancestor of i and j in S. $D(i)$ could be calculated by counting the number of clusters, where i appears, i.e.

$$D(i) = \sum_{c=1}^{n-2} M_{i,c} + 1, \quad \forall 1 \leq i \leq n.$$

In order to calculate $D(i,j)$ we need to introduce a few more variables:

$$\begin{aligned} &In(c,i,j) \geq M_{i,c} + M_{j,c} - 1, && \forall 1 \leq i,j \leq n,\ 1 \leq c \leq n-2 \\ &In(c,i,j) \leq M_{i,c}, && \forall 1 \leq i,j \leq n,\ 1 \leq c \leq n-2 \\ &In(c,i,j) \leq M_{j,c}, && \forall 1 \leq i,j \leq n,\ 1 \leq c \leq n-2 \end{aligned}$$

$In(c,i,j)$ is a binary variable, which is 1 if and only if a cluster c contains both taxa i and j. Note that the first inequality assures that $In(c,i,j)$ equals one, when c contains i and j, while the second and third constraints assure that $In(c,i,j)$ is 0 otherwise. Now, we can calculate $D(i,j)$ as follows:

$$D(i,j) = \sum_{c=1}^{n-2} In(c,i,j), \quad \forall 1 \leq i,j \leq n$$

Note that if all the trees in the input to our median tree problem were over the same taxa set, the ILP formulation would be almost complete. However, when it is not the case, we need to accommodate for the minus method used for calculating path-difference distance. Therefore, we introduce the following binary variables:

$$\begin{aligned} &Above(k,i) \geq D(k,i) - D(k) + 1 && \forall 1 \leq i,k \leq n \\ &\overline{Above}(k,i) \geq \frac{(D(k) - D(k,i))}{n-2} && \forall 1 \leq i,k \leq n \\ &Above(k,i) + \overline{Above}(k,i) = 1 && \forall 1 \leq i,k \leq n \end{aligned}$$

Here $Above(k,i)$ and $\overline{Above}(k,i)$ are two complementary variables. The constraints assert that $Above(k,i) = 1$ if and only if the parent of a node with a taxon k lies on the path from $\mathsf{Rt}(S)$ to i.

Let now k,i,j be three taxa in the tree S represented by a matrix M. Then a binary variable $Affects(k,i,j)$ equals 1 if and only if $\mathsf{Pa}(k)$ is on the path from i to j in S. The following constraints enforce this relationship:

$$\begin{aligned} &Max\_above(k,i,j) && \geq && Above(k,i) \\ &Max\_above(k,i,j) && \geq && Above(k,j) \\ &Min\_above(k,i,j) && \leq && Above(k,i) \\ &Min\_above(k,i,j) && \leq && Above(k,j) \\ &Max\_above(k,i,j) + Min\_above(k,i,j) && = && Above(k,i) \\ & && && +Above(k,j) \\ &Max\_above(k,i,j), Min\_above(k,i,j) && \in && \{0,1\} \end{aligned}$$

$$Affects(k,i,j) = Max\_above(k,i,j) - Min\_above(k,i,j)$$

Now we are ready to introduce path-difference variables that would be used in the objective function. Let G be some tree from the input $\mathcal{P}$ and let M_G be a set of "missing" taxa in G. That is, $M_G = \mathsf{L}(\mathcal{P}) - \mathsf{L}(G)$. We introduce a variable $PD(G,i,j)$ for two taxa i and j which equals $d_{i,j}(G,S)$ if $i,j \in \mathsf{L}(G)$ and 0 otherwise. That is

$$PD(G,i,j) = \begin{cases} \mathsf{d}_{ij}(G) - PL(i,j) & \\ \quad + \sum_{k \in M_G} Affects(k,i,j), & i,j \in \mathsf{L}(G) \\ 0, & \text{otherwise.} \end{cases}$$

This leads us to the following objective function:

$$\min \sum_{G \in \mathcal{P}} \sum_{i<j} |PD(G,i,j)|.$$

Finally, we need to eliminate absolute values from the objective function. In order to do it we introduce a set of auxiliary variables:

$$\begin{aligned} APD(G,i,j) &\geq PD(G,i,j) \\ APD(G,i,j) &\geq -PD(G,i,j). \end{aligned}$$

It is easy to verify that the following objective function is equivalent to the original one:

$$\min \sum_{G \in \mathcal{P}} \sum_{i<j} APD(G,i,j).$$

Table 2: Supertree methods evaluation. The best scores under each objective function are shown in bold.

Data set	Method	L_1 PD score	L_2 PD score	Triplet-sim	MAST-sim	Pars. score
Marsuplial 158 input trees 272 taxa	MMC	1,681,015	16,670.45	51.73 %	53.4 %	3901
	MRP	515,257	5,694.59	98.29 %	**71.6** %	**2274**
	TH(SPR)	515,906	5,866.27	**98.99** %	70.2 %	2317
	TH(TBR)	517,274	5,888.22	**98.99** %	70.5 %	2317
	EMT	333,642	**4,677.99**	68.43 %	63.4 %	3339
	MMT	**330273**	5,376.06	52.25 %	52.2 %	4370
Cetartiodactyla 201 input trees 299 taxa	MMC	918,639	16,206.17	70.03 %	51.5 %	4929
	MRP	365,870	6,991.36	95.84 %	**64.7** %	**2603**
	TH(SPR)	403,233	7,630.03	**97.28** %	63.1 %	2754
	TH(TBR)	401,327	7,591.13	**97.28** %	63.0 %	2754
	EMT	274,632	**6,051.13**	59.49 %	52.2 %	4162
	MMT	**270,807**	6,672.98	39.43 %	47.9 %	5191

Complexity analysis. The $Affects$ and PD, APD variable groups contribute the largest number of variables as well as constraints. There are $O(n^3)$ $Affects$ variables and $O(n^3)$ constraints are needed to guarantee its correctness. Further, there are $O(kn^2)$ of PD and APD variables and the same order of constraints involving them. In total, we have $O(n^3 + kn^2)$ variables and constraints in our ILP formulation.

6. EXPERIMENTAL EVALUATION

In this section we evaluate the efficiency and effectiveness of the presented local search heuristic. In our first study we validate our method against other popular supertree software on empirical phylogenetic datasets, and compare resulting trees under multiple relevant objectives. The second study demonstrates the performance of our ILP formulation and compares the heuristic results with the corresponding exact ones on simulated data.

6.1 Local search heuristic evaluation

Here we analyze how well our heuristic achieves its objective using standard empirical data sets that are publicly available. Given the scale of the data sets used, we are using a comparative study evaluating how heuristics with other objectives perform under the L_1 PD distance. Indeed, we expect that our Manhattan median tree heuristic performs best in this study. However, a negative outcome would question the ability of our method.

Data Sets. Following the pioneer work on the path-difference median trees [23] we evaluate the presented algorithm on the following baseline phylogenetic datasets, the Marsupials dataset [7] and the Cetartiodactyla dataset [31]. These datasets are considerably large in size and appear frequently in phylogenetic studies (see, for example, [2, 11, 20, 35]).

Experimental setting. In order to evaluate the performance of our Manhattan median tree (MMT) heuristic we compare it against other phylogenetic methods including the Euclidean (L_2 norm) median tree (EMT) heuristic. In addition, we included the traditional maximum representation with parsimony (MRP) supertree heuristic [39], the exact modified min-cut (MMC) algorithm [28], and the triplet heuristic (TH) for the triplet median tree problem [20].

While the MRP supertree problem is NP-hard [26], existing MRP heuristics are among the most applied supertree methods in evolutionary biology [4]. In this study we use supertrees obtained by the MRP local search algorithm implemented in PAUP* [39] under the tree bisection and reconnection (TBR) edit operation [20]. Note that the *TBR edit operation* is an extension of SPR, where the pruned subtree is allowed to be re-rooted before regrafting it. The MMC algorithm is the only polynomial-time supertree algorithm in this study. In addition, this algorithm was shown to satisfy certain "desirable" properties formulated by Steel et. al. [36], and therefore was suggested for its use on large-scale phylogenetic inference problems [28]. Finally, the triplet heuristic is a local search algorithm that addresses the well-studied NP-hard triplet median tree problem [20]. We use supertrees constructed using the triplet heuristic based on both SPR and TBR local searches, abbreviated by TH(SPR) and TH(TBR) respectively.

Results. The evaluation results are summarized in Table 2. The phylogenetic supertrees obtained by different methods were assessed using relevant objectives: Manhattan (L_1) and Euclidean (L_2) PD distances, triplet similarity (which is used as a maximization criterion in the triplet heuristic), parsimony score (minimization criterion for the MRP heuristic). In addition, we present the maximum agreement subtree (MAST) scores (i.e., the average percentage over the largest agreement subtrees between a supertree and each input tree in proportion to the total tree size) for the computed supertrees. As expected, we observe that our MMT algorithm produced the best supertrees in regards to the Manhattan PD distance. The results also suggest that the supertrees constructed under Manhattan and Euclidean path-difference distances are highly correlated. For example, L_1 scores for the MMT and EMT trees are very close to each other.

In general, having exact distance distributions over supertree spaces is highly beneficial while evaluating experimental results. However, there are no currently known fast algorithms that would allow us to obtain PDD distributions even for a single input tree under any vector-norm [25]. Therefore, to further analyze the results we estimate the distribution of Manhattan PD distances on random sample data for both datasets as it was performed previously for the Euclidean distance [23]. That is, we generated two supertree collections with 5000 trees for each phylogenetic dataset discussed above. One collection was generated under the uniform binary tree distribution, and another collection using the Markovian branching process [3]. Next, we processed each supertree collection and obtained corresponding

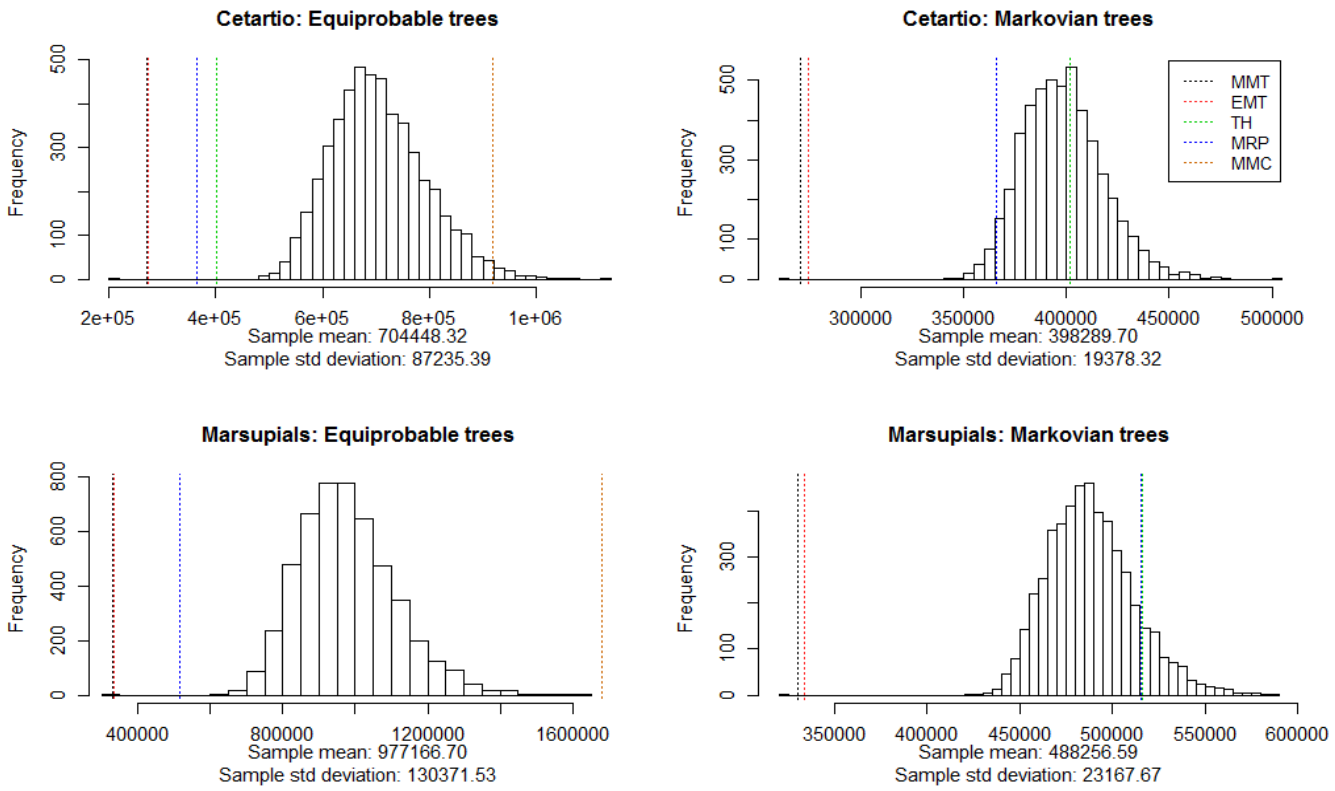

Figure 2: Histograms of the Manhattan PD distances based on the generated tree samples. The dotted lines mark the distances for the supertrees assembled by different methods.

Table 3: Evaluation of ILP performance against the local search heuristic on small instances of the median tree problem. "LS runs" row shows how many local search runs it took to find an optimal solution by the heuristic, and "LS runtime" records a cumulative runtime of the first LS runs.

ntaxa	n=4				n=6				n=8				n=10
ntrees	k=5	k=10	k=20	k=100	k=5	k=10	k=20	k=100	k=5	k=10	k=20	k=100	k=5
Optimal MPDD	17	34	87	431	66	173	322	1686	181	375	799	4154	350
ILP runtime (s)	0.07	0.16	0.20	0.53	1.21	3.03	9.47	7.10	704	481	402	653	10531
LS runs	1	1	1	2	1	5	1	3	1	2	10	3	8
LS runtime (s)	1.75	1.06	1.13	3.99	2.32	9.25	3.11	17.39	4.67	8.38	65.46	39.23	51.39

sample datasets with raw Manhattan PD distances. These samples allowed us to estimate the distance distributions and map the results from Table 2 on them (see Figure 2).

Once again, we observe how close the results for Manhattan and Euclidean heuristics are. In addition, both triplet heuristic and MRP produced trees that are better in distance terms than any of the random trees drown from the uniform distribution. However, Markovian trees show quite a significant bias in regards to our distance measure (i.e., the sample mean value is almost twice smaller than a sample mean value for the equiprobable trees). Surprisingly, as a result, many Markovian random trees are better than MRP and TH supertrees (this is observed in both datasets). Note that this was not the case for the estimated distributions of Euclidean PD distances [23], where MRP and TH trees were better than any of the generated Markovian trees. On a separate note, MMC supertrees show a negative bias in regards to the Manhattan PD distance, which is also true for the Euclidean PD distance.

6.2 ILP evaluation

Experimental setting. For evaluation of our ILP formulation we artificially generated several instances of the Manhattan median tree problem with random input trees. The size of the generated datasets varied both in terms of the number of taxa and trees. To carry out our experiments we used a Gurobi MILP optimizer [15]. The largest instance of the Manhattan median tree problem that we obtained exact results for consisted of 5 phylogenetic trees over 10 taxa. In addition, we ran the presented local search heuristic on the generated problem instances (20 times each).

Results. Table 3 depcits the results of the performed experiments. It can be seen that the runtime of the ILP solver grows very fast with increase of the number of taxa. We also

observe that the local search algorithm was able to reach a global minimum within the first ten runs on each instance, while being significantly faster than the ILP solver on larger inputs ($n = 8$ and 10).

7. DISCUSSION AND CONCLUSION

Large-scale phylogenetic inference problems are among the central computational problems in evolutionary biology. Computationally efficient median tree and supertree algorithms bring the scale of phylogenetic analysis to unprecedented levels. While MRP is the predominant supertree method in this area [4], different tree assembly methods could be applied to provide an alternative viewpoint on evolutionary events. As the presented experimental studies suggest, phylogenetic trees constructed using two different path-difference distance heuristics (for Manhattan and Euclidean median tree problems) do not strongly correlate with the supertrees constructed using other popular methods. Thus, we can pose these methods as a valuable alternative to MRP. At the same time, the estimated distributions suggest that, while correlated, Euclidean and Manhattan median trees and corresponding search spaces could be quite different; therefore, worth being examined independently.

In this work we generalized the path-difference median tree problem for any L_p vector norm and shown that this problem is NP-hard. While there already exists an efficient local search heuristic for the Euclidean median tree problem [23], here we introduce a similar algorithm for the Manhattan median tree problem with a more complex precomputation step. Interestingly, the described precomputation step with a slight modification could be applied to the Euclidean median tree algorithm, but not vice versa.

Further, we developed an ILP formulation that can be applied to find exact solutions for considerably small Manhattan median tree problem instances (up to 10 taxa as experiments show). The comparison of the ILP performance to the local search performance on the simulated data enabled us to additionally justify the merit of the developed heuristic, since it was able to find *global* minima quite fast.

Encouraged by the promising results of our method, future research will investigate further into the ability and applicability of our method in practice in large-scale comparative studies, as well as studying its theoretical properties.

While there is a great interest in weighted phylogenetic trees (see TimeTree database [19] for example), none of the widely applied supertree methods is capable of constructing credible weighted trees. The path-difference median tree approache, however, has a clear potential of changing that and boosting the development of large time-annotated trees of life.

8. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable and helpful input. This material is based upon work supported by the National Science Foundation under Grant No. 1617626.

9. REFERENCES

[1] M. S. Bansal, J. G. Burleigh, and O. Eulenstein. Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models. *BMC Bioinformatics*, 11 Suppl 1:S42, 2010.

[2] M. S. Bansal, J. G. Burleigh, O. Eulenstein, and D. Fernández-Baca. Robinson-foulds supertrees. *Algorithms for Molecular Biology*, 5(1):1–12, 2010.

[3] N. G. Bean, N. Kontoleon, and P. G. Taylor. Markovian trees: properties and algorithms. *Annals of Operations Research*, 160(1):31–50, 2007.

[4] O. R. Bininda-Emonds, editor. *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, volume 4 of *Computational Biology*. Springer Verlag, 2004.

[5] J. Bluis and D. Shin. 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*, pages 87–94. IEEE Computer Society, 2003.

[6] D. Bryant. Hunting for trees in binary character sets: efficient algorithms for extraction, enumeration, and optimization. *J Comput Biol*, 3(2):275–288, 1996.

[7] M. Cardillo, O. R. P. Bininda-Emonds, E. Boakes, and A. Purvis. A species-level phylogenetic supertree of marsupials. *Journal of Zoology*, 264:11–31, 2004.

[8] W.-C. Chang, J. G. Burleigh, D. F. Fernández-Baca, and O. Eulenstein. An ILP solution for the gene duplication problem. *BMC Bioinformatics*, 12 Suppl 1:S14, 2011.

[9] R. Chaudhari, G. J. Burleigh, and O. Eulenstein. 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.

[10] R. Chaudhary, M. S. Bansal, A. Wehe, D. Fernández-Baca, and O. Eulenstein. iGTP: a software package for large-scale gene tree parsimony analysis. *BMC Bioinformatics*, 11:574, 2010.

[11] D. Chen, O. Eulenstein, D. Fernández-Baca, and J. Burleigh. Improved heuristics for minimum-flip supertree construction. *Evolutionary Bioinformatics*, 2, 2006.

[12] J. A. Cotton and M. Wilkinson. Majority-rule supertrees. *Syst Biol*, 56(3):445–452, 2007.

[13] J. Farris. A successive approximations approach to character weighting. *Systematic Zoology*, 18:374–385, 1969.

[14] H. Gee. Evolution: ending incongruence. *Nature*, 425(6960):782, Oct 2003.

[15] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2015.

[16] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, USA, 1997.

[17] S. R. Harris, E. J. Cartwright, M. E. Török, M. T. Holden, N. M. Brown, A. L. Ogilvy-Stuart, M. J. Ellington, M. A. Quail, S. D. Bentley, J. Parkhill, and S. J. Peacock. Whole-genome sequencing for analysis of an outbreak of meticillin-resistant staphylococcus aureus: a descriptive study. *Lancet Infect Dis*, 13(2):130–6, 2013.

[18] R. A. Hufbauer, R. A. Marrs, A. K. Jackson, R. Sforza, H. P. Bais, J. M. Vivanco, and S. E.

Carney. 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*, pages 121–126, Morgantown, WV., 2003. USDA Forest Service. Forest Health Technology Enterprise Team.

[19] A. D. Leaché. The timetree of life. S. Blair Hedges and Sudhir Kumar, editors. *Integrative and Comparative Biology*, 50(1):141–142, 2010.

[20] H. T. Lin, J. G. Burleigh, and O. Eulenstein. Triplet supertree heuristics for the tree of life. *BMC Bioinformatics*, 10(Suppl 1), 2009.

[21] H. T. Lin, J. G. Burleigh, and O. Eulenstein. Consensus properties for the deep coalescence problem and their application for scalable tree search. *BMC Bioinformatics*, 13 Suppl 10:S12, 2012.

[22] W. P. Maddison and L. L. Knowles. Inferring phylogeny despite incomplete lineage sorting. *Syst Biol*, 55(1):21–30, 2006.

[23] A. Markin and O. Eulenstein. *Bioinformatics Research and Applications: 12th International Symposium, ISBRA 2016, Minsk, Belarus, June 5-8, 2016, Proceedings*, chapter Path-Difference Median Trees, pages 211–223. Springer International Publishing, Cham, 2016.

[24] J. O. McInerney, J. A. Cotton, and D. Pisani. The prokaryotic tree of life: past, present... and future? *Trends Ecol Evol*, 23(5):276–81, May 2008.

[25] A. Mir and F. Rosselló. The mean value of the squared path-difference distance for rooted phylogenetic trees. *CoRR*, abs/0906.2470, 2009.

[26] S. Moran, S. Rao, and S. Snir. Using semi-definite programming to enhance supertree resolvability. In *Proceedings of the 5th International Conference on Algorithms in Bioinformatics*, WABI'05, pages 89–103, Berlin, Heidelberg, 2005. Springer-Verlag.

[27] S. Nik-Zainal et al. The life history of 21 breast cancers. *Cell*, 149(5):994–1007, 2012.

[28] R. D. M. Page. Modified mincut supertrees. In *Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, WABI '02, pages 537–552, London, UK, UK, 2002. Springer-Verlag.

[29] H. Philipe and M. J. Telford. Large-scale sequencing and the new animal phylogeny. *TRENDS in Ecology and Evolution*, 21(11):614–620, 2006.

[30] J. B. Phipps. Dendogram topology. *Systematic Zoology*, 20:306–308, 1971.

[31] S. A. Price, O. R. P. Bininda-Emonds, and J. L. Gittleman. A complete phylogeny of the whales, dolphins and even-toed hoofed mammals (cetartiodactyla). *Biological Reviews*, 80(3):445–473, 2005.

[32] P. Puigbò, S. Garcia-Vallvé, and J. O. McInerney. TOPD/FMTS: a new software to compare phylogenetic trees. *Bioinformatics*, 23(12):1556–1558, 2007.

[33] J. J. L. Roux, A. M. Wieczorek, M. M. Ramadan, and C. T. Tran. Resolving the native provenance of invasive fireweed (Senecio madagascariensis Poir.) in the Hawaiian Islands as inferred Poir.) in the Hawaiian Islands as inferred from phylogenetic analysis. *Diversity and Distributions*, 12:694–702, 2006.

[34] C. Semple and M. A. Steel. *Phylogenetics*. University Press, Oxford, 2003.

[35] S. Snir and S. Rao. Quartets maxcut: A divide and conquer quartets algorithm. *IEEE/ACM TCBB*, 7(4):704–718, 2010.

[36] M. Steel, A. W. M. Dress, and S. Bocker. Simple but fundamental limitations on supertree and consensus tree methods. *Systematic Biology*, 49(2):363–368, 2000.

[37] M. A. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9:91–116, 1992.

[38] M. A. Steel and D. Penny. Distributions of tree comparison metrics - some new results. *Systematic Biology*, 42(2):126–141, 1993.

[39] D. L. Swofford. PAUP*. Phylogenetic analysis using parsimony (*and other methods). Version 4. Sinauer Associates, Sunderland, Massachusetts. 2002.

[40] K. Takahashi and M. Nei. Efficiencies of fast algorithms of phylogenetic inference under the criteria of maximum parsimony, minimum evolution, and maximum likelihood when a large number of sequences are used. *Molecular Biology and Evolution*, 17(8):1251–1258, 2000.

[41] C. Than and L. Nakhleh. Species tree inference by minimizing deep coalescences. *PLoS Comput Biol*, 5(9):e1000501, 2009.

[42] W. Williams and H. Clifford. On the Comparison of Two Classifications of the Same Set of Elements. *Taxon*, 20(4):519–522, 1971.