New Algorithms for Computing Phylogenetic Biodiversity

Constantinos Tsirogiannis[1], Brody Sandel[1], and Adrija Kalvisa[2]

[1] MADALGO* and Department of Bioscience
Aarhus University, Denmark
[2] Faculty of Biology
University of Latvia, Latvia

Abstract. A common problem that appears in many case studies in ecology is the following: given a rooted phylogenetic tree $\mathcal{T}$ and a subset R of its leaf nodes, we want to compute the distance between the elements in R. A very popular distance measure that is used for this reason is the *Phylogenetic Diversity* (PD), which is defined as the cost of the minimum weight Steiner tree in $\mathcal{T}$ that spans the nodes in R. To analyse the value of the PD for a given set R it is important also to calculate the variance of this measure. However, the best algorithm known so far for computing the variance of the PD is inefficient; for any input tree $\mathcal{T}$ that consists of n nodes, this algorithm has $\Theta(n^2)$ running time. Moreover, computing efficiently the variance and higher order statistical moments is a major open problem for several other phylogenetic measures. We provide the following results:

- We describe a new algorithm that computes efficiently in practice the variance of the PD. This algorithm has $O(\mathrm{SI}(\mathcal{T})+\mathrm{DSSI}^2(\mathcal{T}))$ running time; here $\mathrm{SI}(\mathcal{T})$ denotes the Sackin's Index of $\mathcal{T}$, and $\mathrm{DSSI}(\mathcal{T})$ is a new index whose value depends on how balanced $\mathcal{T}$ is.
- We provide for the first time exact formal expressions for computing the mean and the variance of another popular biodiversity measure, the Mean Nearest Taxon Distance (MNTD). We show how we can compute the mean of this measure in $O(n)$ time, and its variance in $O(\mathrm{SI}(\mathcal{T}) + \mathrm{DSSI}^2(\mathcal{T}))$ time.
- We introduce a new measure which we call the *Core Ancestor Cost* (CAC). A major advantage of this measure is that for any integer $k > 0$ we can compute all first k statistical moments of the CAC in $O(\mathrm{SI}(\mathcal{T}) + nk + k^2)$ time in total, using $O(n + k)$ space.

We have implemented the new algorithms for computing the variance of the PD and of the MNTD, and the statistical moments of the CAC. We conducted experiments on large phylogenetic datasets and we show that our algorithms perform efficiently in practice.

1 Introduction

Experts in the field of ecology, but also from other disciplines in biology, are frequently confronted with the following problem: given a set of species, they want to measure if these species are close evolutionary relatives. The most common way to measure this is to use a phylogenetic tree $\mathcal{T}$, where each leaf of the tree corresponds to a species, and the weights of the tree edges represent some concept of distance e.g. time since the last speciation event. From $\mathcal{T}$ we

* Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation.

select a subset of leaves R which correspond to the species that we want to examine. The next step is then to choose a method for computing the distance between the leaves in R based on the structure of $\mathcal{T}$. In the related literature, such methods are refered to as *phylogenetic biodiversity measures*. Two measures of this kind that are widely used are the *Phylogenetic Diversity* (PD) and the *Mean Nearest Taxon Distance* (MNTD). For a given tree $\mathcal{T}$ and a subset R of its leaves, the value of the PD is equal to the cost of the minimum-weight Steiner tree in $\mathcal{T}$ that spans the nodes in R. The value of the MNTD is the average path cost in $\mathcal{T}$ between any node $v \in R$ and its closest neighbour in $R \setminus \{v\}$.

Whichever method we choose for computing the distance between the elements in R, we need to know if the returned distance value is relatively small or large compared to other sets of leaves in $\mathcal{T}$. More specifically, we need to compare the distance value that we got for R with the distance values of all possible subsets of leaves in $\mathcal{T}$ that have exactly the same number of elements. In the related literature, the most common way to do that is to compute the mean and the variance of the distance values among all those subsets of species. In a previous paper we introduced algorithms that compute the mean and the variance values of the PD. For a tree $\mathcal{T}$ that consists of n nodes in total, and for a non-negative integer r, we introduced an algorithm that computes in $O(n)$ time the mean value of the PD among all possible subsets that consist of r leaves. We also introduced an algorithm that computes the variance of the PD in $\Theta(n^2)$ time. The latter algorithm is quite inefficient since it takes $\Theta(n^2)$ time to execute, not only in the worst case but for every input tree. This makes the use of this algorithm infeasible in practice, since in some applications it is required to calculate the variance of some measure for a large number of different trees (for example, constructed algorithmically by slightly changing the structure of a given reference tree).

On the other hand, there are no known algorithms for computing the exact value of the mean and the variance of the MNTD. So far, researchers try to estimate these values using a random sampling technique; for a given subset size r, a few subsets of exactly r leaves in $\mathcal{T}$ are selected at random. Then, the mean and the variance of the MNTD is calculated using the values of this measure only for the selected subsets. The number of the sampled subsets is usually around a thousand. For sufficiently large values of r and n, this is a very small number of samples compared to the number of all possible subsets of r leaves in $\mathcal{T}$. This implies that the sampling approach is inexact, and may yield estimated values for the mean and the variance that are very different from the original ones. Hence, there is need to introduce exact and efficient algorithms for computing these statistics for the MNTD.

Furthermore, in some studies it is required to compute not only the mean and the variance, but also the higher order moments of a given measure [3]. Unfortunately, for the most popular phylogenetic biodiversity measures computing the higher order statistics appears to be a difficult task. For the PD and the MNTD, any preliminary attempts that we made to compute the higher order moments lead to algorithms with running time that scales exponentially as the order of the moment increases. Yet, to this point we have not proven that designing more efficient algorithms is impossible; this is a conjecture. On the other hand, the skewness of another popular measure, the *Mean Pairwise Distance* (MPD), can be computed in $O(n)$ time [5]. However, the analytical expression that yields the value of the MPD skewness is particularly involved. Worse than that, it appears that deriving an expression for the higher order moments of the MPD may be overwhelmingly complicated. Therefore, there is the need for a

non-trivial biodiversity measure for which we can efficiently compute its higher order moments.

Our Results. In this paper we present several results that have to do with the efficient computation of the statistical moments of certain phylogenetic biodiversity measures. Given a phylogenetic tree $\mathcal{T}$ and a positive integer r, we describe an algorithm that computes the variance of the PD among all subsets of r leaves in $O(\mathrm{SI}(\mathcal{T}) + \mathrm{DSSI}^2(\mathcal{T}))$ time, using $O(n)$ space. Here, we use $\mathrm{SI}(\mathcal{T})$ to denote the Sackin's Index of $\mathcal{T}$ which is equal to the sum of the numbers of leaves that appear at the subtree of each node in $\mathcal{T}$ [2]. We use $\mathrm{DSSI}(\mathcal{T})$ to denote a new index that we introduce, which we call the Distinct Subtree Sizes Index. We provide a formal definition of this new index later in this paper. The values of both the $\mathrm{SI}(\mathcal{T})$ and the $\mathrm{DSSI}(\mathcal{T})$ depend on the structure of the tree $\mathcal{T}$. When $\mathcal{T}$ is relatively balanced, the new algorithm has a very good performance, and is much more efficient in practice than the already known $\Theta(n^2)$ algorithm. It is only in the worst case, when $\mathcal{T}$ has $\Omega(n)$ height, that the new algorithm runs in $\Theta(n^2)$ time. Moreover, we present for the first time algorithms for computing the exact value of the mean and the variance of the MNTD for ultrametric trees; a tree is called ultrametric if any simple path from its root to a leaf node has the same cost. Given an ultrametric tree $\mathcal{T}$ of n nodes and a positive integer r, we provide an algorithm that runs in $O(n)$ time, and computes the mean of the MNTD among all subsets of r leaves in $\mathcal{T}$. We also present an algorithm that computes the variance of the MNTD in $O(\mathrm{SI}(\mathcal{T}) + \mathrm{DSSI}^2(\mathcal{T}))$ time, using $O(n)$ space. This algorithm is based on the on the same method as our new algorithm that computes the variance of the PD.

Furthermore, we present a new phylogenetic biodiversity measure which we call the *Core Ancestor Cost* (CAC). For a phylogenetic tree $\mathcal{T}$, a subset R of r leaves in $\mathcal{T}$, and a real $\chi \in (0.5, 1]$, the CAC of R is equal to the cost of the simple path that connects the root of $\mathcal{T}$ with the deepest common ancestor node of at least χr of the nodes in R. Unlike several existing measures, we can compute efficiently in practice any of the statistical moments of the CAC. In particular, we prove that for any integer $k > 0$ we can compute all of the first k moments of the CAC in $O(\mathrm{SI}(\mathcal{T}) + nk + k^2)$ time in total, using $O(n + k)$ space.

We have implemented all the algorithms that we introduce in this paper, and we have measured their efficiency using large phylogenetic tree datasets that are publicly available. We show that all of the new algorithms have a very good performance in practice; the new algorithm that computes the variance of the PD appears to clearly outperform its predecessor that runs in $\Theta(n^2)$ time.

2 Computing Efficiently the Variance of Known Biodiversity Measures

Preliminaries. For a phylogenetic tree $\mathcal{T}$ we denote the set of the edges of $\mathcal{T}$ by E. For any edge $e \in E$ we use $w(e)$ to represent the weight of e. We consider that $w(e) > 0$ for every $e \in E$. We use V to denote the set of nodes of $\mathcal{T}$, and we use S to denote the set of leaf nodes of $\mathcal{T}$. We use n to indicate the total number of nodes in $\mathcal{T}$, and we use s to indicate the number of leaves in $\mathcal{T}$. For any node $v \in V$ we use $\mathrm{Ch}(v)$ to indicate the set of the child nodes of v. In this paper we consider only phylogenetic trees that are rooted. We denote the root node of $\mathcal{T}$ by $\mathrm{root}(\mathcal{T})$. Hence, in the rest of this work, whenever we use the term "phylogenetic tree" we mean a rooted tree with edges that have positive weights. We use $h(\mathcal{T})$ to denote the height of the tree, that is the maximum number of

edges that appear on a simple path between the root of $\mathcal{T}$ and a leaf. Since $\mathcal{T}$ is a rooted tree, for any edge $e \in E$ we can distinguish the two nodes adjacent to e into a *parent* node and a *child* node. Here, the child node of e is the one for which the simple path between this node and the root contains e.

Let v be a node in $\mathcal{T}$ and let e be the edge whose child node is v. We use interchangeably $S(e)$ and $S(v)$ to denote the set of leaves that appear in the subtree of v. We denote the number of these leaves by $s(e)$ and $s(v)$. We call this number the *subtree size* of v. For a tree edge $e \in E$, we denote the set of the edges that appear in the subtree of e by $\mathrm{Off}(e)$. For any tree edge e we denote the set of the edges that appear on the simple path between root($\mathcal{T}$) and the child node of e by $\mathrm{Anc}(e)$. From this definition we get that $e \in \mathrm{Anc}(e)$. We also use $\mathrm{Ind}(e)$ to denote the set $E \setminus (\mathrm{Off}(e) \cup \mathrm{Anc}(e))$. For a given node $v \in V$, we use $\mathrm{Anc}(v)$ to represent the set $\mathrm{Anc}(e)$ where e is the edge whose child node is v.

We use $\mathrm{Sub}(S, r)$ to denote the set whose elements are all the subsets of S that have cardinality exactly r. For an edge $e \in E$ and a subset R of the leaves of $\mathcal{T}$, we use $S_R(e)$ to denote the elements of $S(e)$ that are also elements of R, that is $S_R(e) = S(e) \cap R$. We indicate the number of these leaves by $sr(e)$. Let $u, v \in S$ be two leaves in $\mathcal{T}$ and let p be the simple path that connects these leaves. We refer to the sum of the weights of the edges in p as the *cost* of this path. We represent this cost as $\mathrm{cost}(u, v)$.

A tree $\mathcal{T}$ is *ultrametric* if all simple paths between the root and the leaves have the same cost. This means also that for every internal node $x \in \mathcal{T}$ any simple path that connects x with a leaf in $S(x)$ has the same cost.

For a given tree $\mathcal{T}$ the *Sackin's index* of $\mathcal{T}$ is defined as the sum of the number of leaves that appear at the subtree of each node in $\mathcal{T}$. More formally, the Sackin's index of $\mathcal{T}$ is defined as:

$$\mathrm{SI}(\mathcal{T}) = \sum_{v \in V} s(v).$$

Alternatively, in the related literature the Sackin's index is described as the sum of the depths of all leaf nodes in $\mathcal{T}$. Both definitions are equivalent since they lead to exactly the same value. The Sackin's index is mainly used in the literature as a function for measuring how balanced a phylogenetic tree is [2].

Let $\mathcal{T}$ be a phylogenetic tree, and let R be a subset of r leaves in $\mathcal{T}$. Let $f(\mathcal{T}, R)$ be a function that maps the pair $\mathcal{T}, R$ to a non-negative real. Let r be a positive integer such that $r \leq s$. The expected value of f over all subsets that consist of exactly r leaves is equal to:

$$\mu(\mathcal{T}, r) = \mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[f(\mathcal{T}, R) \right] \ .$$

The variance of f over all subsets of r leaves is equal to:

$$var(\mathcal{T}, r) = \mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[(f(\mathcal{T}, R) - \mu(\mathcal{T}, r))^2 \right] \ .$$

We call the expected value and the variance of f the *lower order moments* of f. Let γ be a positive integer such that $\gamma \geq 3$. We define the γ *order moment* of f to be the normalised γ-th central moment of f, which is equal to the following quantity:

$$\frac{\mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[(f(\mathcal{T}, R) - \mu(\mathcal{T}, r))^{\gamma} \right]}{\mathrm{var}^{\gamma/2}(\mathcal{T}, r)} \ .$$

We call the moments that are described by the last expression the *higher order moments* of f. In the present work, whenever we refer to calculating a statistical moment of some measure for a leaf subset size r, we consider a uniform probability distribution for selecting any subset of exactly r leaves in $\mathcal{T}$. In other words, all subsets of exactly r leaves in $\mathcal{T}$ are considered with the same probability when computing a statistical moment of a given measure.

2.1 A New Algorithm for Calculating the Variance of the PD

In a previous paper, we provided a formal expression for the exact value of the standard deviation of the PD [6]. Based on that expression, for a tree $\mathcal{T}$ and a sample size of r leaves, the variance of the PD is equal to:

$$\mathrm{var}_{\mathrm{PD}}(\mathcal{T}, r) = \sum_{e \in E} \sum_{l \in E} w(e) \cdot w(l) \cdot (1 - \mathcal{F}(S, e, l, r)) - \mu^2_{\mathrm{PD}}(\mathcal{T}, r), \qquad (1)$$

where:

$$\mathcal{F}(S,e,l,r) = \begin{cases} \mathcal{F}_{\mathrm{Off}}(S,e,l,r) = \frac{\binom{s(e)}{r} + \binom{s-s(l)}{r} - \binom{s(e)-s(l)}{r}}{\binom{s}{r}} & \text{if } l \in \mathrm{Off}(e). \\ \mathcal{F}_{\mathrm{Off}}(S,l,e,r) = \frac{\binom{s(l)}{r} + \binom{s-s(e)}{r} - \binom{s(l)-s(e)}{r}}{\binom{s}{r}} & \text{if } e \in \mathrm{Off}(l). \\ \mathcal{F}_{\mathrm{Ind}}(S,e,l,r) = \frac{\binom{s-s(e)}{r} + \binom{s-s(l)}{r} - \binom{s-s(e)-s(l)}{r}}{\binom{s}{r}} & \text{otherwise.} \end{cases}$$

and where $\mu_{\mathrm{PD}}(\mathcal{T}, r)$ is the mean value of the PD over all possible subsets of exactly r leaves of $\mathcal{T}$. In our previous paper we showed how we can compute this mean value for a given r in $O(n)$ time. Hence, the bottleneck for calculating the variance of this metric is the computation of the following quantity:

$$\sum_{e \in E} \sum_{l \in E} w(e) \cdot w(l) \cdot (1 - \mathcal{F}(S, e, l, r)) \qquad (2)$$

Given that we can evaluate function $\mathcal{F}$ in constant time[3], the expression in (1) leads to a trivial algorithm that runs in $O(n^2)$ time; for every pair of edges in $e, l \in E$ we calculate explicitly the value of $\mathcal{F}(S, e, l, r)$. However, as we mentioned earlier, the large size of recent phylogenetic datasets makes the use of this algorithm infeasible. Next we show how we can design an algorithm that can be much more efficient in practice, depending on how balanced the input tree $\mathcal{T}$ is. To describe this better, first we introduce a new concept that has to do with the structure of a rooted tree. In particular, let $\mathcal{D}(\mathcal{T})$ denote the set of all subtree sizes that are observed in the tree $\mathcal{T}$, that is $\mathcal{D}(\mathcal{T}) = \{s(e) : e \in E\}$.

We call this set the *distinct subtree sizes set* of $\mathcal{T}$. We represent the size of this set by $\mathrm{DSSI}(\mathcal{T})$, that means $\mathrm{DSSI}(\mathcal{T}) = |\mathcal{D}(\mathcal{T})|$. We call this value the *Distinct Subtree Sizes Index* of the tree $\mathcal{T}$. Based on this definition, we provide the following theorem. The proof of the theorem appears in the appendix.

[3] In the definition of $\mathcal{F}$, all the required values that involve binomial coefficients can be precomputed in $O(n)$ time in total in the RAM model. Each of the precomputed values can then be accessed in constant time each time we have to evaluate this expression.

Theorem 1. *Let $\mathcal{T}$ be a phylogenetic tree that consists of n nodes, and let r be a positive integer such that $r \leq n$. The variance of the Phylogenetic Diversity over all subsets of r leaves in $\mathcal{T}$ can be computed in $O(\mathrm{SI}(\mathcal{T}) + \mathrm{DSSI}(\mathcal{T})^2)$ time, using $O(n)$ memory.*

According to Theorem 1, we can compute the variance of the PD using an algorithm whose performance depends on the parameters $\mathrm{SI}(\mathcal{T})$ and $\mathrm{DSSI}(\mathcal{T})$. For every tree $\mathcal{T}$ it holds that $\mathrm{DSSI}(\mathcal{T}) \geq h(\mathcal{T})$ and $\mathrm{DSSI}(\mathcal{T}) \geq \mathrm{SI}(\mathcal{T})/n$. In the best case, when the input tree is balanced and has height $\Theta(\log n)$, the new algorithm runs in $\Theta(n \log n)$ time. But when it comes to the worst case performance, the new approach is not better than the trivial algorithm that was previously known; if $\mathrm{SI}(\mathcal{T}) = \Theta(n^2)$ or $\mathrm{DSSI}(\mathcal{T}) = \Theta(n)$ then the computation of the variance takes $O(n^2)$ time. In Section 4 we present experimental results that indicate that the new approach is much more efficient in practice. For different tree data sets that we use there, the values of $\mathrm{SI}(\mathcal{T})$ and $\mathrm{DSSI}(\mathcal{T})$ are much smaller than in the worst case scenario. In fact, we can prove a non-trivial tight worst case bound for $\mathrm{DSSI}(\mathcal{T})$; this bound depends on the number of nodes and the height of $\mathcal{T}$. The bound that we provide applies to trees that have a height that is at least logarithmic to the number of tree nodes (for example, trees where the nodes have constant maximum degree). The proof of the following lemma appears in the appendix.

Lemma 1. *Let $\mathcal{T}$ be a phylogenetic tree that consists of n nodes and has height $h(\mathcal{T})$. In the worst case, the value of $\mathrm{DSSI}(T)$ can be as large as $\Theta(\sqrt{n \cdot h(\mathcal{T})})$.*

2.2 Computing the Mean Nearest Taxon Distance

Next we show how we can use the main result of the previous section in order to efficiently compute the variance of another popular phylogenetic measure. Let $\mathcal{T}$ be a phylogenetic tree, and let R be a subset of its leaves that consists of $|R| = r$ elements. The Mean Nearest Taxon Distance(MNTD) of the leaves in R is equal to the average distance between an element in R and its closest neighbour in R[7]. More formally, the MNTD is defined as:

$$\mathrm{MNTD}(\mathcal{T}, R) = \frac{1}{r} \sum_{v \in R} \min_{u \in R/\{v\}} cost(u, v) \ . \tag{3}$$

Like with other phylogenetic measures, in order to analyse the value of the MNTD for a set of leaves R it is important to compute the mean and the variance of this measure for all possible subsets of $|R|$ leaves in $\mathcal{T}$. Next we provide for the first time formal expressions that lead to the efficient computation of the exact value of the mean and the variance of the MNTD. The expressions that we provide hold only for ultrametric phylogenetic trees; recall that a tree $\mathcal{T}$ is ultrametric if all simple paths between the root and the leaves of $\mathcal{T}$ have the same cost. Ultrametric tree datasets are very common in phylogenetic research; for instance, ultrametric trees are produced for a given set of taxa when the weights of the tree edges represent specific notions of distance, such as time between speciation events. In the next lemma we show how we can simplify the expression in (3) when we specifically consider ultrametric trees.

Lemma 2. *Let $\mathcal{T}$ be an ultrametric phylogenetic tree and let $R \subseteq S$ be a subset of r leaves. The value of the* MNTD *for this subset is equal to:*

$$\mathrm{MNTD}(\mathcal{T}, R) = \frac{2}{r} \sum_{\substack{e \in E \\ sr(e)=1}} w(e) \ . \tag{4}$$

Proof. Let v be a leaf in R, and let u be the closest leaf to v in $R/\{v\}$; that means $cost(u,v) = \min_{x \in R/\{v\}} cost(v,x)$. Let $p(u,v)$ be the simple path that connects u and v in $\mathcal{T}$. We can partition $p(u,v)$ into two subpaths $p(u,a)$ and $p(v,a)$, where a is the deepest node in $\mathcal{T}$ that is a common ancestor of u and v. Since $\mathcal{T}$ is ultrametric, for every internal node $x \in \mathcal{T}$ any simple path that connects x with a leaf in $S(x)$ has the same cost. Therefore, it holds that $cost(u,a) = cost(v,a) = cost(u,v)/2$. Also, for any edge e that appears in the path $p(v,a)$ we have that $sr(e) = 1$. If that was not the case then there would exist an edge e in $p(v,a)$ and a leaf u' in $S(e)$ such that $u' \notin \{u,v\}$ and $cost(u',v) < cost(u,v)$, which contradicts the assumption that u is the closest leaf to v in R. From the above, we conclude that:

$$\mathrm{MNTD}(\mathcal{T}, R) = \frac{1}{r} \sum_{v \in R} \min_{u \in R/\{v\}} cost(u,v) = \frac{2}{r} \sum_{v \in R} \sum_{\substack{e \in \mathrm{Anc}(v) \\ sr(e)=1}} w(e) = \sum_{\substack{e \in E \\ sr(e)=1}} w(e) \ .$$

□

Next we use the expression in (4) to obtain expressions for efficiently computing the mean and the variance of the MNTD for ultrametric trees. The proofs of the next two theorems appear in the appendix.

Theorem 2. *Let $\mathcal{T}$ be an ultrametric phylogenetic tree that has s leaves and consists of n nodes in total. Let r be a non-negative integer with $r \leq s$. The expected value of the* MNTD *for a subset of exactly r leaves in $\mathcal{T}$ is equal to:*

$$\mu_{\mathrm{MNTD}}(\mathcal{T}, r) = \frac{2}{r} \sum_{e \in E} \frac{w(e) \cdot s(e) \cdot \binom{s-s(e)}{r-1}}{\binom{s}{r}}, \tag{5}$$

and can be computed in $\Theta(n)$ time in the RAM model.

Theorem 3. *Let $\mathcal{T}$ be an ultrametric phylogenetic tree that has s leaves and consists of n nodes in total. Let r be a natural number with $r \leq s$. The variance of the* MNTD *for a sample of exactly r leaves in $\mathcal{T}$ is equal to:*

$$\mathrm{var}_{\mathrm{MNTD}}(\mathcal{T}, r) = \frac{4}{r^2} \sum_{e \in E} \sum_{l \in E} w(e) \cdot w(l) \cdot \mathcal{G}(S, e, l, r) - \mu^2_{\mathrm{MNTD}}(\mathcal{T}, r), \tag{6}$$

where:

$$\mathcal{G}(S,e,l,r) = \begin{cases} \mathcal{G}_{\mathrm{Off}}(S,e,l,r) = \frac{s(l) \cdot \binom{s-s(e)}{r-1}}{\binom{s}{r}} & \text{if } l \in \mathrm{Off}(e). \\ \mathcal{G}_{\mathrm{Off}}(S,l,e,r) = \frac{s(e) \cdot \binom{s-s(l)}{r-1}}{\binom{s}{r}} & \text{if } e \in \mathrm{Off}(l). \\ \mathcal{G}_{\mathrm{Ind}}(S,e,l,r) = \frac{s(e) \cdot s(l) \cdot \binom{s-s(e)-s(l)}{r-2}}{\binom{s}{r}} & \text{otherwise.} \end{cases}$$

The variance of the MNTD *can be computed in $O(\mathrm{SI}(\mathcal{T}) + \mathrm{DSSI}(\mathcal{T})^2)$ time, using $O(n)$ memory.*

3 A New Biodiversity Measure

Earlier in this paper, we indicated that in several case studies there is the need to compute the higher order moments of a phylogenetic biodiversity measure. Yet, we argued that for a few popular measures this appears to be infeasible. Next we introduce a new non-trivial measure, for which we prove that we can calculate any of its statistical moments efficiently in practice.

Let $\mathcal{T}$ be a phylogenetic tree and let R be a subset of its leaves. Let χ be a any real in the interval $(0.5, 1]$. We use $v_{\text{anc}}(R, \chi)$ to denote the deepest node in the tree that has at least χr elements of R in its subtree. We call this node the *core ancestor of R given χ*. We call the cost of the simple path that connects $v_{\text{anc}}(R, \chi)$ with the root of $\mathcal{T}$ the *Core Ancestor Cost of R given χ* (CAC), and we denote this cost by $\text{CAC}(\mathcal{T}, R, \chi)$.

We consider that the CAC can be a useful tool for phylogenetic analyses; the CAC can be used to measure whether a sample of leaves R consists mostly of a single group of closely related species, or R is made of several small unrelated groups. For example, if $\text{CAC}(\mathcal{T}, R, 0.8)$ is relatively large and comparable to the average path cost between the root and any leaf in $\mathcal{T}$ then about 80% of the species in R have a common ancestor which is deep in the tree, and they are closely related. On the other hand, if $\text{CAC}(\mathcal{T}, R, 0.51)$ is zero then R consists of at least two main unrelated groups of species. In future work we intend to examine the behaviour of this new measure on specific datasets and compare it with other existing biodiversity measures. In the present paper we focus on how we can compute efficiently the CAC and the values of its statistical moments.

For a given sample of leaves R and an integer $\chi \in (0.5, 1]$, value $\text{CAC}(\mathcal{T}, R, \chi)$ can be computed in $O(n)$ time in the following way; first, we compute bottom-up the values $sr(e)$ for every $e \in E$. Then, we start from the root of $\mathcal{T}$ and we compute $\text{CAC}(\mathcal{T}, R, \chi)$ by constructing incrementally the path that connects the root with $v_{\text{anc}}(R, \chi)$.

The major advantage of using the CAC in phylogenetic analysis is that, for a given χ and size of R, we can efficiently compute in practice the value of any statistical moment of this measure. To describe how can do this, we define the following quantity:

$$\mathcal{C}_\chi(\mathcal{T}, r, k) = \text{E}_{R \in \text{Sub}(S,r)} \left[\text{CAC}^k(\mathcal{T}, R, \chi) \right] \ .$$

We can compute any of the moments of CAC by using the values $\mathcal{C}_\chi(\mathcal{T}, r, k)$. In particular, The expectation of CAC for r leaves is equal to $\mathcal{C}_\chi(\mathcal{T}, r, 1)$, and the variance is equal to $\mathcal{C}_\chi(\mathcal{T}, r, 2) - \mathcal{C}^2_\chi(\mathcal{T}, r, 1)$. Using a standard formula from the mathematical literature, for any integer $k > 3$ the k-th order moment of CAC for r leaves can be expressed as:

$$\frac{\sum_{i=0}^{k} \binom{k}{i} (-\mathcal{C}_\chi(\mathcal{T}, r, 1))^i \ \mathcal{C}^{k-i}_\chi(\mathcal{T}, r, i)}{\left(\mathcal{C}_\chi(\mathcal{T}, r, 2) - \mathcal{C}_\chi(\mathcal{T}, r, 1)\right)^{k/2}} \ . \tag{7}$$

Therefore, computing the k-th order moment of CAC boils down to calculating $\mathcal{C}_\chi(\mathcal{T}, r, i)$ for every $i = 1, 2, \ldots, k$. In the next lemma we show that this can be done efficiently in practice. The proof of this lemma is provided in the appendix.

Lemma 3. *Let $\mathcal{T}$ be a phylogenetic tree that has s leaves and consists of n nodes in total. Let $r \leq s$ be a positive integer and let χ be real number such that $\chi \in (0.5, 1]$. For any positive integer k it holds that:*

$$\mathcal{C}_\chi(\mathcal{T}, r, k) = \mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[\mathrm{CAC}^k(\mathcal{T}, R, \chi) \right]$$

$$= \sum_{v \in V} \mathrm{cost}(v, \mathrm{root}(\mathcal{T})) \cdot \frac{\sum_{i=\lceil r\chi \rceil}^{s(v)} \binom{s(v)}{i} \binom{s-s(v)}{r-i} - \sum_{u \in \mathrm{Ch}(v)} \sum_{j=\lceil r\chi \rceil}^{s(u)} \binom{s(u)}{j} \binom{s-s(u)}{r-j}}{\binom{s}{r}} . \tag{8}$$

We can compute the values $\mathcal{C}_\chi(\mathcal{T}, r, t)$ for all $t = 1, 2, \ldots, k$ in $O(\mathrm{SI}(\mathcal{T}) + kn)$ time, using $O(n + k)$ space.

The following theorem follows directly from combining Lemma 3 with Equation (7).

Theorem 4. *Let $\mathcal{T}$ be a phylogenetic tree that consists of n nodes and s leaves. Let r, k be two non-negative integers such that $r \leq s$, and let χ be a real such that $\chi \in (0.5, 1]$. We can compute the k first statistical moments of the Core Ancestor Cost among all possible subsets of exactly r leaf nodes of $\mathcal{T}$ given χ in $O(\mathrm{SI}(\mathcal{T}) + kn + k^2)$ time, using $O(n + k)$ space.*

4 Experiments and Benchmarks

We have implemented all of the algorithms that we introduced in the previous sections, and we have conducted experiments in order to measure their performance. In these experiments we also used an implementation of the old approach for computing the variance of the PD; this is the algorithm that always takes quadratic time to execute with respect to the size of the input tree. We use this implementation as a point of reference for our new algorithm that computes the variance of the PD. All of the implementations were developed in $C++$. The experiments were executed on an Intel i7-3770 eight-core CPU where each core is a 3.40 GHz processor. The main memory of this computer is 16 Gigabytes. The operating system that we used on this computer is Microsoft Windows 7.

In all the experiments that we conducted, we observed that the algorithm that computes the variance of the MNTD had an almost identical performance with the new algorithm that computes the variance of the PD. Therefore, for the sake of brevity, we chose not to illustrate the running times of the MNTD algorithm in this version of the paper.

We performed two sets of experiments; in the first set of experiments we used phylogenetic trees that were produced based on real-world biological data, representing the phylogenetic relations between existing species. We used two datasets of this kind; one dataset is a phylogenetic tree that represents the phylogeny of all mammal species [1]. This tree has 4510 leaf nodes and 6618 nodes in total. We refer to this tree as the `mammals` dataset. The other real-world dataset that we used is a tree that was constructed by Goloboff et. al [4]. This is the largest evolutionary tree of eukaryotic organisms that has been so far constructed from molecular and morphological data. It consists of 71181 leaves and 83751 nodes in total. This tree is unrooted; for the needs of our experiments we picked arbitrarily an internal node and used this as the root. We call this dataset the `eukaryotes` dataset. In the first set of experiments we ran our three

new algorithms plus the old algorithm that computes the PD variance using as input the `mammals` and the `eukaryotes` datasets. We executed each algorithm several times on each dataset and we measured the total running time of the algorithm for all these executions. We did this because for the three algorithms that we introduce in this paper the time taken for a single execution was quite short, and comparable to the time spent by our software to read the input dataset. Hence, we executed each of the algorithms on each of the datasets ninety-nine times, each time using a different value of r, ranging from two to one hundred. Preliminary measurements showed that the value of r does not affect in practice the performance of any of the examined algorithms. This is also the case with the value of the χ parameter and the performance of the CAC algorithm. In the experiments we ran this algorithm with parameter values $\chi = 0.6$ and $k = 3$. We also calculated the values of the SI and the DSSI for each dataset. These results are presented in Table 1.

Table 1. The results of the experiments that involve trees which represent relations between species in the real-world. The running time of each algorithm is measured over ninety-nine consecutive executions on the same dataset (PD Old = the old approach for computing the PD variance, PD New = the new algorithm for computing the PD variance, CAC = the algorithm that computes the k first moments of the CAC for $k = 3$ and $\chi = 0.6$). Running times are presented in seconds.

Dataset	n	PD Old	PD New	CAC	SI	DSSI^2
`eukaryotes`	83751	$>$ 3 hours	38.9	14.8	998850	109561
`mammals`	6618	1672	3.6	1.0	79984	26569

According to the results of these experiments, it becomes evident that the new algorithm that computes the variance of the PD outperforms clearly the old approach. For the two datasets that we considered, the new algorithm appears to be hundreds of times faster than the old one. Given that the running times are measured over ninety-nine executions, it appears that the new algorithm for the PD can process a tree of more than $80,000$ nodes in less than half a second. The algorithm that computes the first three moments of the CAC appears to be even faster than that. As it comes to the values of the SI and the DSSI, we see that the Sackin's Index is larger than the square of the DSSI. This may be an indication that, in practice, the SI is the dominating quantity in the analysis of the running time of the new algorithm. For both datasets, the SI appears to be equal to roughly twelve times the size of the input.

In the second set of experiments we used trees of various sizes that we generated algorithmically. These trees were created in the following manner; first we generated twenty trees using a randomised pure birth process. In this process, a tree is grown in a series of steps from a single root node; at each step we choose a leaf node v, and we add two child nodes to v. Node v is chosen uniformly at random among all the leaves of the current tree. Using this process we generated twenty binary trees, each having exactly $4,000$ leaves. From each of these trees we extracted sixteen subtrees; these subtrees have $250k$ leaves with k ranging from one to sixteen. The subtrees were produced by successively pruning chunks of 250 leaves from the original tree of $4,000$ leaves. In this way we produced 320 trees in total. We denote the set of these trees by $\mathcal{U}$.

We ran each of the implemented algorithms using as input the trees in $\mathcal{U}$. As we did in the previous set of experiments, we executed each algorithm

ninety-nine times for each input tree, and we measured the total time taken for these executions. Figure 1 illustrates the running times of the old and the new algorithm that compute the variance of the PD, and the running times of the algorithm that computes the first k moments of the CAC for $\chi = 0.6$ and $k = 3$. Also, for each $\mathcal{T} \in \mathcal{U}$ we measured the values of the SI and the DSSI. Furthermore, we measured the running time of the algorithm that computes the moments of the CAC for a fixed tree of $4,000$ leaves and for different values of k–see Figure 2.

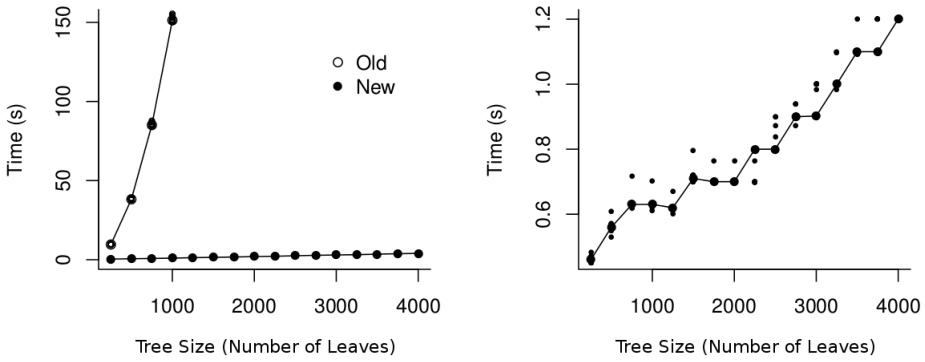

Fig. 1. The running times of three of the implemented algorithms using as input randomly generated trees. For each algorithm, the continuous line segments connect the median values of the measured running times for input trees that have the same number of leaves. Left: The running times of the old and the new algorithms that compute the variance of the PD. For each algorithm, the running times for input trees of the same number of leaves have very small difference in value, and hence they are almost indistinguishable. Right: The running time of the algorithm that computes the first k moments of the CAC for $k = 3$ and $\chi = 0.6$.

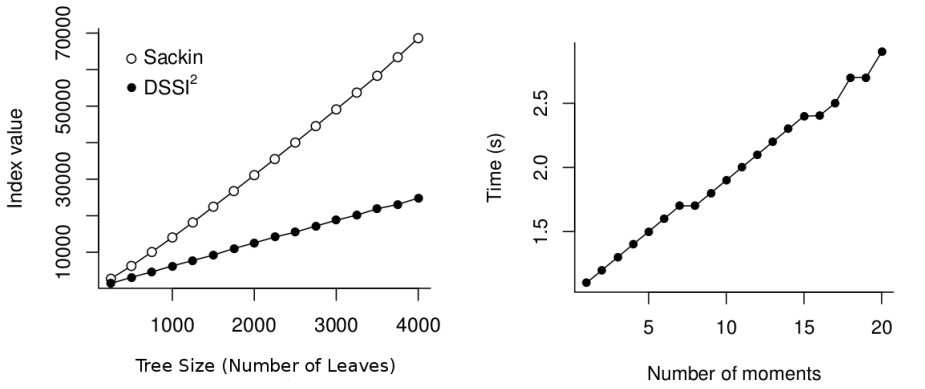

Fig. 2. Left: The values of the SI and of the square of the DSSI for the trees that we generated using a pure birth process. For each number of leaves, we illustrate only the median of these values. The rest of the values are quite close to this median, having at most an absolute difference of roughly two thousand units. Right: The running time of the algorithm that computes the first k moments of the CAC for a single tree of $4,000$ leaves and for k ranging from one to twenty.

Again, as can be seen in Figure 1, the new algorithm for the PD variance has a much better performance than the old one. We see also that the algorithm that computes the moments of the CAC runs very fast, processing almost a hundred trees of a few thousand nodes in less than 1.5 seconds. In Figure 2 we see that the SI is evidently larger the DSSI for the randomly generated trees. Still, the value of the SI is not much larger the size of the input trees; given that the total number of nodes of a binary tree is roughly at most twice the number of its leaves, the SI in this set of experiments is not larger than ten times the size of the input. This possibly explains the very good performance of all the new algorithms that we introduce in this paper. Also, as expected, in Figure 2 we can see that the running time of the algorithm that calculates the moments CAC scales almost linearly as the value of k increases.

References

1. O.R.P. Bininda-Emonds, M. Cardillo, K.E. Jones, R.D.E MacPhee, R.M.D. Beck, R. Grenyer, S.A. Price, R.A. Vos, J.L. Gittleman and A. Purvis. The Delayed Rise of Present-Day Mammals. *Nature* 446: 507–512, 2007.
2. M.G.B. Blum and O. François. On Statistical Tests of Phylogenetic Tree Imbalance: The Sackin and Other Indices Revisited. *Mathematical Biosciences*, 195:14–153, 2005.
3. M. Cadotte, C.H. Albert and S.C. Walker. The Ecology of Differences: Assessing Community Assembly with Trait and Evolutionary Distances. *Ecology Letters*, 16:1234–1244, 2013.
4. P.A. Goloboff, S.A. Catalano, J.M. Mirandeb, C.A. Szumika, J.S. Ariasa, M. Kallersjoc and J.S. Farris. Phylogenetic analysis of 73 060 taxa corroborates major eukaryotic groups. *Cladistics*, 25:211–230, 2009.
5. C. Tsirogiannis and B. Sandel. Computing the Skewness of the Phylogenetic Mean Pairwise Distance in Linear Time. In *Proc. 13th Workshop on Algorithms in Bioinformatics (WABI)*, pages 170 – 184, 2013.
6. C. Tsirogiannis, B. Sandel and D. Cheliotis. Efficient Computation of Popular Phylogenetic Tree Measures. In *Proc. 12th Workshop on Algorithms in Bioinformatics (WABI)*, pages 30 – 43, 2012.
7. C.O Webb, D.D. Ackerly, M.A. McPeek and M.J. Donoghue. Phylogenies and Community Ecology. *Annual review of ecology and systematics*, 33:475–505, 2002.

Appendix

Theorem 1. *Let $\mathcal{T}$ be a phylogenetic tree that consists of n nodes, and let r be a positive integer such that $r \leq n$. The variance of the Phylogenetic Diversity over all subsets of r leaves in $\mathcal{T}$ can be computed in $O(\mathrm{SI}(\mathcal{T}) + \mathrm{DSSI}(\mathcal{T})^2)$ time, using $O(n)$ memory.*

Proof. Based on the description that we provided earlier in this section, to prove the time bound it suffices to describe how we can evaluate efficiently the expression in (2). We can rewrite this expression as follows:

$$\sum_{e\in E}\sum_{l\in E} w(e)\cdot w(l)\cdot(1-\mathcal{F}(S,e,l,r)) \tag{9}$$

$$=\left(\sum_{e\in E} w(e)\right)^2-\sum_{e\in E} w(e)^2\cdot\mathcal{F}_{\text{Off}}(S,e,e,r)-2\sum_{e\in E}\sum_{l\in\text{Off}(e)} w(e)\cdot w(l)\cdot\mathcal{F}_{\text{Off}}(S,e,l,r) \tag{10}$$

$$-2\sum_{e\in E}\sum_{l\in\text{Ind}(e)} w(e)\cdot w(l)\cdot\mathcal{F}_{\text{Ind}}(S,e,l,r)\ . \tag{11}$$

It is easy to show that the first and the second sum in (10) consist of $\Theta(n)$ terms, and therefore they can be computed in $O(n)$ time. The third sum in (10) consists of $\text{SI}(\mathcal{T})$ terms since for every edge $e\in E$ there exist $s(e)$ terms in this sum. Since we can evaluate each of these terms in constant time, the expression in (10) can be evaluated in $O(\text{SI}(\mathcal{T}))$ time in total.

The two nested sums of the quantity in (11) can be analysed as follows:

$$\sum_{e\in E}\sum_{l\in\text{Ind}(e)} w(e)\cdot w(l)\cdot\mathcal{F}_{\text{Ind}}(S,e,l,r)=\sum_{e\in E}\sum_{l\in E} w(e)\cdot w(l)\cdot\mathcal{F}_{\text{Ind}}(S,e,l,r)$$

$$-2\sum_{e\in E}\sum_{l\in\text{Off}(e)} w(e)\cdot w(l)\cdot\mathcal{F}_{\text{Ind}}(S,e,l,r)-\sum_{e\in E} w(e)^2\cdot\mathcal{F}_{\text{Ind}}(S,e,e,r)\ . \tag{12}$$

Based on the same arguments as for the expression in (10), the two last sums in (12) can be evaluated in $O(\text{SI}(\mathcal{T}))$ time in total. Let α

be a positive integer such that $\alpha\in\mathcal{D}(\mathcal{T})$. Recall that $\mathcal{D}(\mathcal{T})$ is the set of all values $s(e)$ that we can observe among the edges of $\mathcal{T}$. Let $\zeta(\alpha)$ denote the sum of the weights of all the edges $e\in E$ for which it holds $s(e)=\alpha$, that means:

$$\zeta(\alpha)=\sum_{\substack{e\in E\\ s(e)=\alpha}} w(e)$$

Using this notation, the first sum in (12) can be written as:

$$\sum_{e\in E}\sum_{l\in E} w(e)\cdot w(l)\cdot\mathcal{F}_{\text{Ind}}(S,e,l,r)=\sum_{\alpha\in\mathcal{D}(\mathcal{T})}\sum_{\beta\in\mathcal{D}(\mathcal{T})}\zeta(\alpha)\cdot\zeta(\beta)\cdot\mathcal{F}_{\text{Ind}}(S,\alpha,\beta,r)\ . \tag{13}$$

In the last expression, we abuse slightly the notation for function $\mathcal{F}_{\text{Ind}}$; for two integers $\alpha,\beta\in\mathcal{D}$ we imply that $\mathcal{F}_{\text{Ind}}(S,\alpha,\beta,r)=\mathcal{F}_{\text{Ind}}(S,e,l,r)$, where $s(e)=\alpha$ and $s(l)=\beta$. The sum in (13) consists of $\Theta(\text{DSSI}^2(\mathcal{T}))$ terms. Each of these terms can be evaluated in constant time given that we have precomputed the values $\zeta(\alpha)$, $\forall\alpha\in\mathcal{D}(\mathcal{T})$. The values $\zeta(\alpha)$ can be precomputed trivially in $\Theta(n)$ time altogether, hence the expression in (13) can be evaluated in $\Theta(\text{DSSI}^2(\mathcal{T}))$ time in total. Given the description that we provided for evaluating the expressions from (10) to (13), we conclude that the variance of the PD can be computed in $O(\text{SI}(\mathcal{T})+\text{DSSI}(\mathcal{T})^2)$ time overall. To do this, we need to store the values of the functions $\mathcal{F}_{\text{Off}}$, and $\mathcal{F}_{\text{Ind}}$, and the values $\zeta(\alpha)$ for every $\alpha\in\mathcal{D}(\mathcal{T})$. These require $O(n)$ memory in total, and the theorem follows. □

Lemma 1. *Let $\mathcal{T}$ be a phylogenetic tree that consists of n nodes and has height $h(\mathcal{T})$. In the worst case, the value of* $\mathrm{DSSI}(T)$ *can be as large as* $\Theta(\sqrt{n \cdot h(\mathcal{T})})$.

Proof. First we prove that $\mathrm{DSSI}(\mathcal{T}) = O\left(\sqrt{n \cdot h(\mathcal{T})}\right)$. To do this, we cut $\mathcal{T}$ into two subsets; the first subset consists of all the subtrees of $\mathcal{T}$ that have less than $\sqrt{n \cdot h(\mathcal{T})}$ leaves each. We indicate this forest of subtrees by $\mathrm{F_{sub}}$. The second subset is the tree $\mathcal{T}'$ that is induced by the nodes in $\mathcal{T} - \mathrm{F_{sub}}$. For every node $v \in \mathrm{F_{sub}}$ it holds that $s(v) < \sqrt{n \cdot h(\mathcal{T})}$, therefore there can be at most $\sqrt{n \cdot h(\mathcal{T})} - 1$ distinct subtree sizes in $\mathrm{F_{sub}}$. The tree $\mathcal{T}'$ has at most $\sqrt{n/h(\mathcal{T})}$ leaf nodes since each of these leaves corresponds to a node in $\mathcal{T}$ which has at least $\sqrt{n \cdot h(\mathcal{T})}$ leaves in its subtree. Since $\mathcal{T}$ has height $h(\mathcal{T})$ then every simple path in $\mathcal{T}'$ between a leaf and the root consists of less than $h(\mathcal{T})$ nodes. Given that every node in $\mathcal{T}'$ appears in such a path, then $\mathcal{T}'$ consists of at most $\sqrt{n \cdot h(\mathcal{T})}$ nodes. Even if all of these nodes have distinct subtree sizes, together with the subtree sizes in $\mathrm{F_{sub}}$ we get that $\mathrm{DSSI}(\mathcal{T}) \leq 2\sqrt{n \cdot h(\mathcal{T})} - 1$. Thus we conclude the claimed upper bound.

Next we present a lower bound construction for the worst case value of $\mathrm{DSSI}(\mathcal{T})$. To describe this construction we use the following definitions and notation. A binary *caterpillar* tree is the binary tree that has the maximum possible height for a given number of leaves; in such a tree each internal node is adjacent to exactly one leaf, except the deepest internal node which is adjacent to two leaves. We use $\mathcal{T}^*(k)$ to denote the star graph that has exactly k leaves. In other words, graph $\mathcal{T}^*(k)$ is a tree that consists of a root node directly connected to k leaves. We define as a *hybrid caterpillar tree* the tree which is constructed by substituting the deepest leaf node of a caterpillar tree with the root of a star graph. We use $\mathcal{T}_{hc}(k, m)$ to represent the hybrid caterpillar tree that is formed by connecting $\mathcal{T}^*(k)$ to the caterpillar of height m–see Fig. 3(a). We call the internal nodes of $\mathcal{T}_{hc}(k, m)$ the *spine* nodes of this tree.

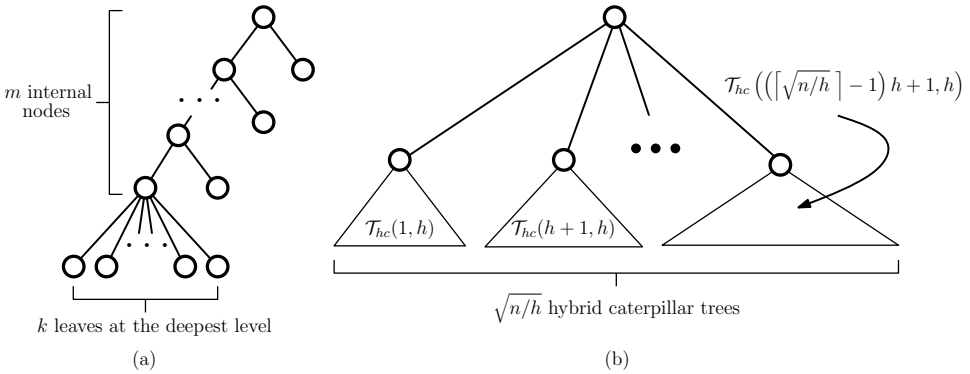

Fig. 3. (a) The structure of the hybrid caterpillar tree $\mathcal{T}_{hc}(k, m)$. (b) The lower bound tree construction $\mathcal{T}'$ with $\mathrm{DSSI}(\mathcal{T}') = \Omega\left(\sqrt{nh}\right)$.

Let $\mathcal{T}$ be a phylogenetic tree that has n nodes and height h. Given any such tree, we present how we can always construct a tree $\mathcal{T}'$ with the following properties: a) tree $\mathcal{T}'$ has $\Theta(n)$ nodes and $\Theta(h)$ height, and b) it holds that $\mathrm{DSSI}(\mathcal{T}') = \Omega\left(\sqrt{nh}\right)$.

The tree construction $\mathcal{T}'$ that we propose is made of two parts, a bottom part and a top part. The bottom part consists of $\sqrt{n/h}$ hybrid caterpillar trees. In particular, this part consists of trees $\mathcal{T}_{hc}(1, h)$, $\mathcal{T}_{hc}(h + 1, h)$, ...,

$\mathcal{T}_{hc}\left(\left(\left\lceil\sqrt{n/h}\,\right\rceil-1\right)h+1,h\right)$. The top part of $\mathcal{T}'$ is simply the star tree $\mathcal{T}_{top}=\mathcal{T}^*\left(\left\lceil\sqrt{n/h}\,\right\rceil\right)$ which is made by connecting the roots of the hybrid caterpillar trees to a single extra node. Let $\mathcal{T}_{hc}(a\cdot h+1,h)$ be one of the hybrid caterpillar trees that appear in the bottom part of $\mathcal{T}'$. According to our description, the set of the subtree sizes of the spine nodes of this tree contains all integers $\{a\cdot h+1, a\cdot h+2,\ldots,(a+1)\cdot h\}$. Therefore, the set of the subtree sizes of the spine nodes of all the hybrid caterpillar trees in $\mathcal{T}'$ contains the set $\left\{1,2,\ldots,\left\lfloor\sqrt{nh}\right\rfloor\right\}$. Therefore, we get that $\mathrm{DSSI}(\mathcal{T}')\geq\sqrt{nh}-1$. The total number of nodes in all the hybrid caterpillar trees of $\mathcal{T}'$, plus the root of $\mathcal{T}'$ is equal to:

$$1+\sum_{i=0}^{\left\lceil\sqrt{n/h}\,\right\rceil-1}(2+i)h\leq n+4\sqrt{n\cdot h}+2h+1\leq 7n+1\ .$$

The height of $\mathcal{T}'$ is equal to $h+1$, and the lemma follows. We can yield a similar result even for trees that have bounded degree d; to do this, we only need to substitute the star trees that we use in $\mathcal{T}'$ with almost complete trees of degree d that have exactly the same number of leaves. The height of the resulting tree $\mathcal{T}'$ will then be at most $3h$, while the number of nodes in the new tree will still be $O(n)$. □

Theorem 2. *Let $\mathcal{T}$ be an ultrametric phylogenetic tree that has s leaves and consists of n nodes in total. Let r be a non-negative integer with $r\leq s$. The expected value of the* MNTD *for a subset of exactly r leaves in $\mathcal{T}$ is equal to:*

$$\mu_{\mathrm{MNTD}}(\mathcal{T},r)=\frac{2}{r}\sum_{e\in E}\frac{w(e)\cdot s(e)\cdot\binom{s-s(e)}{r-1}}{\binom{s}{r}},\tag{14}$$

and can be computed in $\Theta(n)$ time in the RAM model.

Proof. Let R be a subset of r leaves in $\mathcal{T}$, and let e be any edge in $\mathcal{T}$. We use $SP(e,R)$ to denote the function that has value 1 when $sr(e)=1$, otherwise it has value zero. Based on Lemma 2, the expectation of the MNTD for a subset of r leaves in $\mathcal{T}$ is equal to:

$$\mathrm{E}_{\mathrm{MNTD}}(\mathcal{T},r)=\mathrm{E}_{R\in\mathrm{Sub}(S,r)}\left[\frac{2}{r}\sum_{\substack{e\in E\\ sr(e)=1}}w(e)\right]=\mathrm{E}_{R\in\mathrm{Sub}(S,r)}\left[\frac{2}{r}\sum_{e\in E}w(e)\cdot SP(e,R)\right]\tag{15}$$

$$=\frac{2}{r}\sum_{e\in E}w(e)\cdot\mathrm{E}_{R\in\mathrm{Sub}(S,r)}\left[SP(e,R)\right]\ .\tag{16}$$

Considering that every subset R of exactly r leaves is picked with the same probability, the expected value of the function $SP(e,R)$ is equal to:

$$\mathrm{E}_{R\in\mathrm{Sub}(S,r)}\left[SP(e,R)\right]=\frac{s(e)\cdot\binom{s-s(e)}{r-1}}{\binom{s}{r}}\ ,\tag{17}$$

which leads to the expression in (14).

To compute the value of this expression, we first precompute values $\binom{x}{r-1}/\binom{s}{r}$ for every integer $x\in[r-1,s]$. This can be done alltogether in $O(n)$ time in the RAM model. Given these values, the rest of the expression (14) can be straightforwardly evaluated in $O(n)$ time. □

Theorem 3. *Let $\mathcal{T}$ be an ultrametric phylogenetic tree that has s leaves and consists of n nodes in total. Let r be a natural number with $r \leq s$. The variance of the* MNTD *for a sample of exactly r leaves in $\mathcal{T}$ is equal to:*

$$\mathrm{var}_{\mathrm{MNTD}}(\mathcal{T}, r) = \frac{4}{r^2} \sum_{e \in E} \sum_{l \in E} w(e) \cdot w(l) \cdot \mathcal{G}(S, e, l, r) - \mu^2_{\mathrm{MNTD}}(\mathcal{T}, r), \qquad (18)$$

where:

$$\mathcal{G}(S, e, l, r) = \begin{cases} \mathcal{G}_{\mathrm{Off}}(S, e, l, r) = \frac{s(l) \cdot \binom{s - s(e)}{r-1}}{\binom{s}{r}} & \text{if } l \in \mathrm{Off}(e). \\ \mathcal{G}_{\mathrm{Off}}(S, l, e, r) = \frac{s(e) \cdot \binom{s - s(l)}{r-1}}{\binom{s}{r}} & \text{if } e \in \mathrm{Off}(l). \\ \mathcal{G}_{\mathrm{Ind}}(S, e, l, r) = \frac{s(e) \cdot s(l) \cdot \binom{s - s(e) - s(l)}{r-2}}{\binom{s}{r}} & \text{otherwise.} \end{cases}$$

The variance of the MNTD *can be computed in $O(\mathrm{SI}(\mathcal{T}) + \mathrm{DSSI}(\mathcal{T})^2)$ time, using $O(n)$ memory.*

Proof. The variance of the MNTD for all subsets of exactly r leaves in $\mathcal{T}$ is equal to:

$$\mathrm{var}_{\mathrm{MNTD}}(\mathcal{T}, r) = \mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[\mathrm{MNTD}^2(\mathcal{T}, R) \right] - \mu^2_{\mathrm{MNTD}}(\mathcal{T}, r) \ . \qquad (19)$$

In Theorem 2 we showed how we can compute the expectation of the MNTD in $O(n)$ time. Therefore, next we focus on deriving an expression for the expectated value of the square of the MNTD. According to Lemma 2, we get:

$$\mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[\mathrm{MNTD}^2(\mathcal{T}, R) \right] = \mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[\frac{4}{r^2} \left(\sum_{e \in E} w(e) \cdot SP(e, R) \right)^2 \right]$$

$$= \frac{4}{r^2} \sum_{e \in E} \sum_{l \in E} w(e) \cdot w(l) \cdot \mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[SP(e, R) \cdot SP(l, R) \right] \ , \qquad (20)$$

where function $SP(e, R)$ has value 1 if the edge e has exactly one element of R in its subtree, otherwise it is equal to zero. Assuming that the sets of exactly r leaves in $\mathcal{T}$ are selected uniformly at random, for any two edges $e, l \in \mathcal{T}$ we get the following cases:

(a) Edge $l \in \mathrm{Off}(e)$:

$$\mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[SP(e, R) \cdot SP(l, R) \right] = \frac{s(l) \cdot \binom{s - s(e)}{r-1}}{\binom{s}{r}} \ . \qquad (21)$$

(b) Edge $e \in \mathrm{Off}(l)$. This is symmetric to case (a):

$$\mathrm{E}_{R \in \mathrm{Sub}(S,r)} \left[SP(e, R) \cdot SP(l, R) \right] = \frac{s(e) \cdot \binom{s - s(l)}{r-1}}{\binom{s}{r}} \ . \qquad (22)$$

(*c*) Edge $l \in \mathrm{Ind}(e)$:

$$\mathrm{E}_{R\in\mathrm{Sub}(S,r)}\left[SP(e,R)\cdot SP(l,R)\right] = \frac{s(e)\cdot s(l)\cdot \binom{s-s(e)-s(l)}{r-2}}{\binom{s}{r}} \ . \tag{23}$$

The expression in (18) follows from combining expressions (19) to (23). To compute the value of the expression in (18) we apply almost the same analysis as we did for the variance of the PD in Theorem 1. The only difference here is that we substitute in the analysis functions $\mathcal{F}_{\mathrm{Off}}(S,e,l,r)$ and $\mathcal{F}_{\mathrm{Ind}}(S,e,l,r)$ with functions $\mathcal{G}_{\mathrm{Off}}(S,e,l,r)$ and $\mathcal{G}_{\mathrm{Ind}}(S,e,l,r)$ respectively. To evaluate the latter two functions efficiently, we need to precompute values $\binom{x}{r-1}/\binom{s}{r}$ and $\binom{x}{r-2}/\binom{s}{r}$ for every integer $x \in [r-2,s]$. This can be done in $O(n)$ time in total, and the theorem follows. □

Lemma 3. *Let $\mathcal{T}$ be a phylogenetic tree that has s leaves and consists of n nodes in total. Let $r \leq s$ be a positive integer and let χ be real number such that $\chi \in (0.5, 1]$. For any positive integer k it holds that:*

$$\mathcal{C}_\chi(\mathcal{T},r,k) = \mathrm{E}_{R\in\mathrm{Sub}(S,r)}\left[\mathrm{CAC}^k(\mathcal{T},R,\chi)\right]$$

$$= \sum_{v\in V}\mathrm{cost}(v,\mathrm{root}(\mathcal{T}))\cdot\frac{\sum_{i=\lceil r\chi\rceil}^{s(v)}\binom{s(v)}{i}\binom{s-s(v)}{r-i} - \sum_{u\in\mathrm{Ch}(v)}\sum_{j=\lceil r\chi\rceil}^{s(u)}\binom{s(u)}{j}\binom{s-s(u)}{r-j}}{\binom{s}{r}} \ . \tag{24}$$

We can compute the values $\mathcal{C}_\chi(\mathcal{T},r,t)$ for all $t = 1,2,\ldots,k$ in $O(\mathrm{SI}(\mathcal{T})+kn)$ time, using $O(n+k)$ space.

Proof. For any $v \in V$ let $CP(v,R,\chi)$ be the function such that $CP(v,R,\chi) = 1$ if v is the core ancestor of R given χ, otherwise $CP(v,R,\chi) = 0$. The value of $\mathcal{C}_\chi(\mathcal{T},r,k)$ is equal to:

$$\mathcal{C}_\chi(\mathcal{T},r,k) = \mathrm{E}_{R\in\mathrm{Sub}(S,r)}\left[\left(\sum_{v\in V}\mathrm{cost}(v,\mathrm{root}(\mathcal{T}))\cdot CP(v,R,\chi)\right)^k\right]$$

$$= \sum_{v\in V}\mathrm{cost}^k(v,\mathrm{root}(\mathcal{T}))\cdot\mathrm{E}_{R\in\mathrm{Sub}(S,r)}\left[CP(v,R,\chi)\right] \ . \tag{25}$$

The last expression follows from the fact that $CP(v,R,\chi)\cdot CP(u,R,\chi) = 0$ when $u \neq v$. For any $v \in V$ the expected value of $CP(v,R,\chi)$ is equal to the probability that v is the core ancestor of R when R is selected uniformly at random among all possible subsets of r leaves in $\mathcal{T}$. This is equal to the probability that at least $\lceil r\chi\rceil$ of the leaves in R appear in the subtree of v *and* every child of of v has less than $\lceil r\chi\rceil$ elements of R in its subtree. Therefore we get that:

$$\mathrm{E}_{R\in\mathrm{Sub}(S,r)}\left[CP(v,R,\chi)\right] = Pr\left[sr(v)\geq\lceil r\chi\rceil \text{ and } sr(u)\geq\lceil r\chi\rceil,\ \forall u\in\mathrm{Ch}(v)\right]$$

$$= Pr\left[sr(v)\geq\lceil r\chi\rceil\right] - \sum_{u\in\mathrm{Ch}(v)}Pr\left[sr(u)\geq\lceil r\chi\rceil\right] \ . \tag{26}$$

The expression in (26) follows from the fact that the events $sr(u) \geq \lceil r\chi \rceil$, $u \in V$ are mutually exclusive; this is because $\chi > 0.5$. For any node $v \in V$ it holds that:

$$Pr\left[sr(v) \geq \lceil r\chi \rceil\right] = \text{cost}(v, \text{root}(\mathcal{T})) \cdot \frac{\sum_{i=\lceil r\chi \rceil}^{s(v)} \binom{s(v)}{i}\binom{s-s(v)}{r-i}}{\binom{s}{r}} \ . \tag{27}$$

From combining (25), (26) and (27) we yield the expression in (24). We proceed by proving the time bound for evaluating this expression. For every node $v \in V$, we can compute all values $\text{cost}(\text{root}(\mathcal{T}), v)$ in a single top-to-bottom scan of $\mathcal{T}$. This can be performed in $O(n)$ time in total. For each $v \in V$, to evaluate $Pr\left[sr(v) \geq \lceil r\chi \rceil\right]$ we have to calculate the value of as many as $2s(v)+1$ binomial coefficients. This can be done in $O(s(v))$ time for each node v, which leads to $O(\text{SI}(\mathcal{T}))$ time in total for all the nodes in $\mathcal{T}$.

We can calculate the values $\mathcal{C}_\chi(\mathcal{T}, r, i)$ successively in increasing order of i, each time storing the values $\text{cost}^i(\text{root}(\mathcal{T}), v)$ for every $v \in V$. In this way, we can evaluate $\mathcal{C}_\chi(\mathcal{T}, r, i+1)$ by performing $O(n)$ additions and multiplications. This leads to $O(kn)$ more computations for evaluating all values $\mathcal{C}_\chi(\mathcal{T}, r, i)$ for $i = 1, 2, \ldots, k$. According to the above, we need to store at most $O(n)$ values to compute $\mathcal{C}_\chi(\mathcal{T}, r, i)$. We also need $O(k)$ additional space to store the values $\mathcal{C}_\chi(\mathcal{T}, r, i)$ that we are computing, and the lemma follows. □