Efficient Computation of Popular Phylogenetic Tree Measures

Constantinos Tsirogiannis[1], Brody Sandel[1], and Dimitris Cheliotis[3]

[1] MADALGO* and Department of Bioscience
Aarhus University, Denmark
[2] Department of Mathematics
University of Athens, Greece

Abstract. Given a phylogenetic tree $\mathcal{T}$ of n nodes, and a sample R of its tips (leaf nodes) a very common problem in ecological and evolutionary research is to evaluate a distance measure for the elements in R. Two of the most common measures of this kind are the Mean Pairwise Distance (MPD) and the Phylogenetic Diversity (PD). In many applications, it is often necessary to compute the expectation and standard deviation of one of these measures over all subsets of tips of $\mathcal{T}$ that have a certain size. Unfortunately, existing methods to calculate the expectation and deviation of these measures are inexact and inefficient.
We present analytical expressions that lead to efficient algorithms for computing the expectation and the standard deviation of the MPD and the PD. More specifically, our main contributions are:

- We present efficient algorithms for computing the expectation and the standard deviation of the MPD exactly, in $\Theta(n)$ time.
- We provide a $\Theta(n)$ time algorithm for computing approximately the expectation of the PD and a $O(n^2)$ time algorithm for computing approximately the standard deviation of the PD. We also describe the major computational obstacles that hinder the exact calculation of these concepts.

We also describe $O(n)$ time algorithms for evaluating the MPD and PD given a single sample of tips. Having implemented all the presented algorithms, we assess their efficiency experimentally using as a point of reference a standard software package for processing phylogenetic trees.

1 Introduction

Background and Motivation Ecologists are increasingly using information on phylogenetic relationships among species to gain new insights into both fundamental and applied questions. This is motivated in part by the observation that closely related species often share similar phenotypic and ecological characteristics. Species with high phylogenetic distinctiveness also represent particularly unique evolutionary histories, and are therefore important conservation targets. Phylogenetic relationships among species can be used to understand biogeographic patterns [9], to infer processes underlying local community assembly [2] and to inform conservation decision making [3].

Given a particular phylogenetic tree $\mathcal{T}$, many measures have been proposed to describe the phylogenetic composition of a set of tips, that is a set of leaf nodes

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

of the tree that represent the finest taxonomic unit in the analysis (for example animal species, languages etcetera). Here, we focus on two widely used concepts. The first is *Phylogenetic Diversity* (PD), which measures the total edge weight of the spanning subtree that connects all tips of the sample. Different variants of this metric have been considered in the related literature [4, 7, 11], mostly as building blocks for the analysis of fundamental combinatorial problems that arise in evolutionary research.

The second is *Mean Pairwise Distance* (MPD), which is equal to the mean cost among all simple paths between pairs of sample tips.

Both PD and MPD depend on the number of tips of the studied sample. Hence, analyses that do not account for this relationship risk conflating patterns of species richness with patterns of phylogenetic community composition. A common solution to this problem is to calculate an index that standardizes PD or MPD based on the expectation and standard deviation over all possible sets of tips of a certain size [12].

In the case of MPD, this index is called the Net Relatedness Index (NRI) [12]. For PD, we call this the phylogenetic diversity index (PDI). Both the expectation and the standard deviation depend on the topology of the tree in potentially complex ways, hence it has been standard to estimate these values using methods that are based on random sampling of tips in $\mathcal{T}$ [13, 8]; for a given sample size r, a large number of samples is extracted (often a thousand, but sometimes fewer), each sample consisting of exactly r distinct tips. In most cases, the tips of each sample are selected using a uniform random distribution, that is each tip has the same probability to be selected in a sample as every other tip. Then, for each of these samples, the value of the considered measure (PD or MPD) is computed and the expectation and standard deviation are calculated among the computed values. This expectation and deviation are then used as an estimate of the actual expectation and standard deviation of the measure.

However, this approach produces inexact estimates; even for a tree $\mathcal{T}$ of a few thousand tips, and for a tip sample size r of a few hundreds, the total number of the subsets of tips in $\mathcal{T}$ that have size exactly r is astronomically large. Therefore, it is debatable if the above random method can provide a reasonable approximation by selecting only a limited, yet computationally feasible, number of tip subsets.

More than that, this approach can be quite slow; even for trees that consist of only a few thousand tips, existing software packages that use this method can take several minutes, or hours, to produce an output.

Thus, it is interesting to ask the question if there can be algorithms that compute the expectation and the standard deviation of either the MPD and PD precisely, and at the same time efficiently, without having to tediously check each possible subset of tips of a certain size.

Our results In this paper we prove computationally feasible analytical expressions for the expectation and standard deviation of the MPD and PD. Based on these expressions, we describe efficient algorithms that speed up the computation of the NRI and PDI. We also provide efficient algorithms that calculate the basic MPD and PD measures given an individual sample of tips.

In Section 2, we present algorithms for computing analytically the expectation and standard deviation of MPD and PD for a phylogenetic tree $\mathcal{T}$ and for a given tip-sample size, assuming a uniform probability distribution on the selection of the tips of the tree. We show that computing analytically the expectation and the standard deviation of the MPD can be done in $\Theta(n)$ time where n is the

combinatorial size of the input tree. We also indicate the fundamental computational problems that arise in the analytical computation of these concepts for the PD measure. We also provide efficient algorithms that compute the MPD and PD measures given a specific tip sample in $O(n)$ time. The proofs of the theorems that are presented in that section can be found in the full version of the paper.

In Section 3 we test our implementation of all of our algorithms against those that appear in the package `picante` [8] of the software library `R` [10] (one of the standard tools for phylogenetic analyses in ecology). Our tests are designed to assess, first, the improvement in computation time we have achieved, and second, the size of the error induced when estimating the expectation and standard deviation using random sampling.

Significance Our approach drastically reduces computation time for NRI and PDI on large trees. This is significant because the time required to perform an estimation by random sampling can be a limiting factor in large ecological studies. For example, it may be desirable to not only calculate a global NRI for some set of species samples, but to recalculate it under phylogenies of different sizes, different assumptions about divergence times within unresolved clades (often genera [9]), or different subsets of tips. As the number of different comparisons increases, ecological interpretations can become increasingly refined and sophisticated. However, the size of the induced computational problem also grows radically. To date, incremental improvements in the maximum size of the problem that can be considered have come primarily from increases in computing power. We anticipate that the results presented here will allow a breakthrough to considering a much larger parameter spaces, greatly improving resulting ecological insights.

2 Analytical Expressions and Algorithms

Preliminaries. For a phylogenetic tree $\mathcal{T}$ we denote the set of its edges by E. For an edge $e \in E$, we indicate the (always positive) weight of this edge as w_e. We denote the set of leaf nodes of $\mathcal{T}$ by S. From here on we will refer to these nodes also as the *tips* of the tree. We indicate the number of these nodes by s, that is $s = |S|$, and we indicate the number of all the nodes of the tree by n.

A phylogenetic tree is a rooted tree, a specific node in the tree is defined as the root. Hence, for any edge $e \in E$ we can distinguish the two nodes adjacent to e into a *parent* node and a *child* node; the child node of e is the adjacent node for which we have to cross e in order to reach the root of the tree. For a node $u \in \mathcal{T}$, we denote the set of the edges for which u is the parent node by $desc(u)$. We use $\mathcal{T}(e)$ to indicate the subtree of $\mathcal{T}$ whose root is the child node of edge e. We denote the set of tips that appear in $\mathcal{T}(e)$ as $S(e)$, and we use also $s(e)$ to denote the number of these tips.

For an edge $e \in E$, we denote the set of all tree edges that appear in the subtree of e by $\mathrm{Off}(e)$. We denote the set of all edges $e' \in E$ for which e appears in the subtree of e' by $\mathrm{Anc}(e)$. We consider that $e \in \mathrm{Anc}(e)$. We also indicate the set $E - (\mathrm{Off}(e) \cup \mathrm{Anc}(e))$ by $\mathrm{Ind}(e)$.

Given any two nodes u, v of $\mathcal{T}$, we call a *simple path* between these nodes a sequence of edges in E that we have to traverse at most once so as to reach u from v. We indicate this path by $p(u, v)$. We call the *cost* of this path the sum of the weights of all the edges that constitute the path. We denote this cost by $cost(u, v)$. As $\mathcal{T}$ is a tree, there exists a unique simple path for any pair of

nodes in $\mathcal{T}$. Let $R \subseteq S$ be any subset of the tips of a phylogenetic tree $\mathcal{T}$. We denote the set of all pairs of elements in R, that is the set of all combinations that consist of two distinct tips in R, by $\Delta(R)$. We indicate the set of all paths that connect two elements in R by Paths(R), that is:

$$\text{Paths}(R) = \{p(u,v) : \{u,v\} \in \Delta(R)\}$$

We denote the set whose elements are all the subsets of S that have cardinality exactly r by $\text{Sub}(S,r)$. For an edge $e \in E$ and a subset R of the tips of $\mathcal{T}$, we denote the elements of $S(e)$ that are also elements of R by $S_R(e)$, that is $S_R(e) = S(e) \cap R$. We indicate the the number of these tips as $s_R(e)$. Consider the subset of $\mathcal{T}$ that is the union of the edges of all paths in Paths(R). We call this subset the *subtree of* $\mathcal{T}$ *induced by* R. We denote this subset by $\mathcal{T}(R)$ and we indicate the number of edges in $\mathcal{T}(R)$ by $t(R)$.

2.1 The Mean Pairwise Distance Method

Let R be a subset of the tips of a phylogenetic tree $\mathcal{T}$ and let $r = |R|$. The *Mean Pairwise Distance* (MPD) of the tips in R is defined as:

$$\text{MPD}(\mathcal{T}, R) = \frac{2}{r(r-1)} \sum_{\{u,v\} \in \Delta(R)} cost(u,v) \tag{1}$$

More specifically, the mean pairwise distance of a set R of tips is the sum of the costs of all simple paths between two distinct tips in R, divided by the total number of these paths; since R contains r nodes and since there is a unique simple path between any pair of distinct nodes, then the number of all different paths is equal to the number of all different pairs of elements in R, that is $r(r-1)/2$.

Speeding Up the Computation of the MPD Given a tree $\mathcal{T}$ and a subset of its tips R, we can derive a simple algorithm for computing $\text{MPD}(\mathcal{T}, R)$ directly from the expression in (1); for each $\{u,v\} \in \Delta(R)$ compute $cost(u,v)$ by summing the weights of the edges that form the path between u and v, and add this value to the total sum. However, this approach would be quite inefficient in terms of the number of computational steps involved. Recall that there are $r(r-1)/2$ distinct pairs of tips, and therefore as many paths whose costs we need to compute explicitly. In this manner, we need $\Theta(r^2)$ time only to enumerate those paths. Yet, we can compute this sum more efficiently, using the following lemma.

Lemma 1. *Consider a phylogenetic tree* $\mathcal{T}$ *and let* R *be a sample of* $|R| = r$ *tips of* $\mathcal{T}$*. For any edge* e *of this tree, the number of paths in* Paths(R) *that contain* e *is equal to:*

$$|\{p(u,v) : p(u,v) \in \text{Paths}(R) \text{ and } e \in p(u,v)\}| = s_R(e) \cdot (r - s_R(e))$$

Therefore, the MPD *of* R *is equal to:*

$$\text{MPD}(\mathcal{T}, R) = \frac{2}{r(r-1)} \sum_{e \in E} w_e \cdot s_R(e) \cdot (r - s_R(e)) \tag{2}$$

Proof. Let e be an edge of $\mathcal{T}$ and let u, v be two distinct tips in R. Edge e appears in the path between u and v if one of these nodes appears in the subtree of e and the other does not. Thus, for each tip in $S_R(e)$ there are as many as $r - s_R(e)$

paths that contain edge e. That means that exactly $s_R(e)(r - s_R(e))$ paths in Paths(R) contain e, and therefore we prove the first part of this theorem.

Instead of computing the cost of each possible path in R explicitly, we can express the MPD in terms of the weight of the edges of $\mathcal{T}$; the weight of an edge in $\mathcal{T}$ is counted in $\text{MPD}(\mathcal{T}, R)$ as many times as the number of paths in Paths(R) which contain this edge. Therefore, $\text{MPD}(\mathcal{T}, R)$ can be expressed as:

$$\text{MPD}(\mathcal{T}, R) = \frac{2}{r(r-1)} \sum_{e \in E} w_e \cdot occur_R(e)$$

where $occur_R(e)$ is the number of paths in Paths(R) that contain e, and the second part of the theorem follows.

□

The expression in (2) can be computed in $\Theta(t(R))$ time in the following manner; first we extract the set of edges that appear in at least one path in Paths(R), that is the edges of $\mathcal{T}(R)$. This can be easily done in $\Theta(t(R))$ time by tracing $\mathcal{T}(R)$ bottom-up starting from the tips in R. Then, we apply a simple recursive algorithm to compute the value $s_R(e)$ for each edge of $\mathcal{T}(R)$.

The Net Relatedness Index For many applications on phylogenetic trees, given a tree $\mathcal{T}$ and a subset $R \subseteq S$ of $|R| = r$ tips it is important to measure how much the MPD of this set differs from the MPD of any other subset R' of exactly r tips in $\mathcal{T}$. To express this difference, the following quantity is usually calculated:

$$\text{NRI} = \frac{\text{MPD}(\mathcal{T}, R) - \text{E}_{\text{MPD}}(\mathcal{T}, r)}{sd_{\text{MPD}}(\mathcal{T}, r)},$$

$\text{E}_{\text{MPD}}(\mathcal{T}, r), sd_{\text{MPD}}(\mathcal{T}, r)$ are the expected value and the standard deviation respectively of the random variable $\text{MPD}(\mathcal{T}, R)$, where the random set R is picked uniformly out of all subsets of S with r elements.

In what follows, we will compute analytically this expected value and standard deviation. The result for the expected value is stated in the next theorem.

Theorem 1. *Let $\mathcal{T}$ be a phylogenetic tree that contains s tips, and let r be a natural number with $r \leq s$. The expected value of the* MPD *for a sample of exactly r tips of $\mathcal{T}$ is equal to:*

$$\text{E}_{\text{MPD}}(\mathcal{T}, r) = \frac{2}{s(s-1)} \sum_{e \in E} w_e \cdot s(e) \cdot (s - s(e)), \tag{3}$$

and can be computed $\Theta(n)$ time, where n is the total number of nodes of the tree.

Remark 1. From (3) we see that the expected value of the MPD is independent of the size of R, which is an interesting, yet not surprising, result by itself. However, as we show later on in this paper, this is not the case for the standard deviation of the MPD.

The Standard Deviation of the MPD Before describing how we can compute analytically the standard deviation of the MPD on a given tree $\mathcal{T}$ and for a given sample size, we introduce a few quantities that relate to groups of paths on $\mathcal{T}$. Our goal is to simplify the computation of the standard deviation of the MPD by expressing the standard deviation in terms of these quantities. We show that we can compute these quantities efficiently by just scanning the tree a constant

number of times and, thus, derive a $\Theta(n)$ algorithm for computing the standard deviation.

For a given phylogenetic tree $\mathcal{T}$ we define the *total path cost of* $\mathcal{T}$ as the sum of the costs of all distinct simple paths that connect tips of $\mathcal{T}$. We denote this quantity by $TC(\mathcal{T})$, thus:

$$TC(\mathcal{T}) = \sum_{\{u,v\} \in \Delta(S)} cost(u, v).$$

According to Lemma 1, we get that:

$$TC(\mathcal{T}) = \sum_{e \in E} w_e \cdot s(e) \cdot (s - s(e)). \tag{4}$$

Let e be an edge of $\mathcal{T}$. We define the *total path cost of* e as the sum of the costs of all those distinct simple paths between tips of $\mathcal{T}$ that contain e. We denote this quantity by $TC(e)$, thus:

$$TC(e) = \sum_{\substack{\{u,v\} \in \Delta(S) \\ e \in p(u,v)}} cost(u, v).$$

It is easy to show that the latter quantity can be expressed as follows:

$$\begin{aligned} TC(e) &= (s - s(e)) \sum_{l \in \mathrm{Off}(e)} w_l \cdot s(l) + s(e) \sum_{l \in \mathrm{Anc}(e)} w_l \cdot (s - s(l)) + s(e) \sum_{l \in \mathrm{Ind}(e)} w_l \cdot s(l) \\ &= (s - s(e)) \sum_{l \in \mathrm{Off}(e)} w_l \cdot s(l) + s(e) \sum_{l \in \mathrm{Anc}(e)} w_l \cdot (s - s(l)) \\ &+ s(e) \left(\sum_{e \in E} w_l \cdot s(l) - \sum_{\mathrm{Off}(e) \cup \mathrm{Anc}(e)} w_l \cdot s(l) \right) \end{aligned} \tag{5}$$

For a node u that is a tip of $\mathcal{T}$, we define the *total path cost of* u as the sum of the costs of all simple paths between u and any other tip of $\mathcal{T}$. We indicate this quantity by $TC(u)$, and it is obvious that $TC(u) = TC(e)$, where e is the unique tree edge that is adjacent to u.

From (4) we can derive directly an algorithm for computing $TC(\mathcal{T})$ in $\Theta(n)$ time; scanning the tree in a recursive manner, we compute for each edge e the value $s(e)$ from the respective values of the edges adjacent to its child node, and from that we calculate the number of occurences of e in a path, multiplied by the edge weight.

Based on (5), we use the combination of the following two simple algorithms for computing $TC(e)$ for all the edges of $\mathcal{T}$:

Algorithm *AllEdgesPathCosts*($\mathcal{T}$)
Input: A phylogenetic tree $\mathcal{T}$.
Output: An array $tc[1 \ldots |E|]$ such that $tc[e] = TC(e), \forall e \in E$.
1. Initialise array $tc[1 \ldots |E|]$ with all values set to zero.
2. Set global variable AllWeights $\leftarrow \sum_{l \in E} w_l \cdot s(l)$
3. **for** every $e \in desc(root(\mathcal{T}))$
4. **do** *SingleEdgeCosts*$(e, w_e(s - s(e)), w_e \cdot s(e), tc)$.
5. **return** $tc[\cdot]$

Algorithm *SingleEdgeCosts*$(e, \text{SumAnc}_1, \text{SumAnc}_2, tc)$
Input: A tree edge e, real numbers SumAnc_1 and SumAnc_2, and (a reference to) the array tc that stores the computed $TC(\cdot)$ values of the tree edges
Output: A real number which is equal to $w_e \cdot s(e) + \sum_{l \in \text{Off}(e)} w_l \cdot s(l)$.
1. $\triangleright$ Precondition 1: $\text{SumAnc}_1 = \sum_{l \in \text{Anc}(e)} w_l(s - s(l))$
2. $\triangleright$ Precondition 2: $\text{SumAnc}_2 = \sum_{l \in \text{Anc}(e)} w_l \cdot s(l)$
3. $\text{SumOff} \leftarrow 0$
4. $u \leftarrow$ child node of e
5. **for** every $l \in \text{desc}(u)$
6. **do** $\text{SumOff} \leftarrow \text{SumOff} +$ *SingleEdgeCost*$(l, \text{SumAnc}_1 + w_l(s - s(l)), \text{SumAnc}_2 + w_l \cdot s(l), tc)$
7. $SO \leftarrow (s - s(e)) \cdot \text{SumOff}$
8. $SA \leftarrow s(e) \cdot \text{SumAnc}_1$
9. $SI \leftarrow s(e) \cdot (\text{AllWeights} - \text{SumAnc}_2 - \text{SumOff})$
10. $tc[e] \leftarrow SO + SA + SI$
11. $\triangleright$ Postcondition: $\text{SumOff} = \sum_{l \in \text{Off}(e)} w_l \cdot s(l)$
12. **return** $\text{SumOff} + w_e \cdot s(e)$

In the above algorithm, we consider that values $s(e)$ for every $e \in E$ have been already computed; this can be easily done in $\Theta(n)$ time in total. The recursive routine *SingleEdgeCosts* computes the value $TC(e)$ for the tree edge e for which we call this routine. *SingleEdgeCosts* is called once for each edge $e \in E$, and the time spent for each call is proportional to the number of edges that are adjacent to the child node of e. Given that the preconditions and the postcondition in *SingleEdgeCosts* are maintained with each recursive call of this routine, the correctness of the algorithm follows.

Theorem 2. *Let $\mathcal{T}$ be a phylogenetic tree that consists of n nodes of which s are tips, and let r be a natural number with $r \leq s$. The standard deviation of the* MPD *for a sample of exactly r tips of $\mathcal{T}$ is equal to:*

$$sd_{\text{MPD}}(\mathcal{T}, r) = \sqrt{c_1 \cdot TC^2(\mathcal{T}) + (c_2 - c_1) \sum_{u \in S} TC^2(u) + (c_1 - 2c_2 + c_3) \sum_{e \in E} w_e \cdot TC(e) - \text{E}^2_{\text{MPD}}(\mathcal{T}, r)},$$

where $c_1 = \frac{4(r-2)(r-3)}{r(r-1)s(s-1)(s-2)(s-3)}$, $c_2 = \frac{4(r-2)}{r(r-1)s(s-1)(s-2)}$, *and* $c_3 = \frac{4}{r(r-1)s(s-1)}$.

We can compute $sd_{\text{MPD}}(\mathcal{T}, r)$ in $\Theta(n)$ time, or in $\Theta(1)$ time for each different value of r after running a preprocessing algorithm that runs in $\Theta(n)$ time.

2.2 The Phylogenetic Diversity

Let R be a subset of the tips of a phylogenetic tree $\mathcal{T}$. The *Phylogenetic Diversity* (PD) of the tips in R is defined as:

$$\text{PD}(\mathcal{T}, R) = \sum_{e \in \mathcal{T}(R)} w_e$$

That is, the phylogenetic diversity of a sample R of tips is the sum of the weights of the edges of $\mathcal{T}$ that appear in at least one path in $\text{Paths}(R)$; the weight of each distinct edge $e \in \mathcal{T}(R)$ is counted only once in this sum, even if e appears in more than one path in $\text{Paths}(R)$. In the related literature, the above definition of the PD is known as the *unrooted* PD. The *rooted* version of the PD, instead of just the edge-weights of $\mathcal{T}(R)$, considers the weights of all the edges that have at least one element of R in their subtree. For the analytical expressions of the

expectation and the standard deviation of the rooted PD, the reader may refer to the work of Steel [11].

$\mathrm{PD}(\mathcal{T}, R)$ can be computed in $\Theta(t(R))$ time with the following simple algorihtm; starting from the tips of R, we trace bottom-up $\mathcal{T}(R)$ marking all the edges that appear in $\mathcal{T}(R)$ and adding their weights to the total sum.

The Phylogenetic Diversity Index Given a tree $\mathcal{T}$ and a subset $R \subseteq S$ of $|R| = r$ tips we can express how much the $\mathrm{PD}(\mathcal{T}, R)$ of this set differs from $\mathrm{PD}(\mathcal{T}, R')$ of any other subset $R' \subseteq S$ of exactly r tips by calculating the following index:

$$\mathrm{PDI} = \frac{\mathrm{PD}(\mathcal{T}, R) - \mathrm{E}_{\mathrm{PD}}(\mathcal{T}, r)}{sd_{\mathrm{PD}}(\mathcal{T}, r)},$$

where the $\mathrm{E}_{\mathrm{PD}}(\mathcal{T}, r)$ is the expected value of the PD for all possible subsets that consist of exactly r tips of $\mathcal{T}$, and $sd_{\mathrm{PD}}(\mathcal{T}, r)$ is the corresponding standard deviation for this group of subsets of tips. Assuming again that each tip can be included in a sample of r tips with equal probability, we present next how we can compute the above expected value and standard deviation analytically. In the following theorem, and for the rest of this paper, we consider that $\binom{n}{k} = 0$ if $k > n$.

Theorem 3. *Let $\mathcal{T}$ be a phylogenetic tree that contains s tips, and let r be a natural number with $r \leq s$. The expected value of the* PD *for a sample of exactly r tips of $\mathcal{T}$ is equal to:*

$$\mathrm{E}_{\mathrm{PD}}(\mathcal{T}, r) = \sum_{e \in E} w_e \left(1 - \frac{\binom{s(e)}{r} + \binom{s - s(e)}{r}}{\binom{s}{r}} \right)$$

From Theorem 3 we get an expression for $\mathrm{E}_{\mathrm{PD}}(\mathcal{T}, r)$ that involves the hypergeometric probability function, which is a ratio of binomial coefficients. The explicit numerical computation of binomial coefficients is something that must be avoided due to the number of bits that are needed to represent their values. In order to achieve a fast computation, this probability function can be implemented using methods that lead to an approximate evaluation of the function. Although such methods can guarantee a fixed error bound for a single coefficient, this does not imply directly a fixed error bound for computing $\mathrm{E}_{\mathrm{PD}}(\mathcal{T}, r)$; the result of summing n numbers, such that each number contains a bounded error value, is not guaranteed to be of bounded error as well. Therefore, although it is possible to devise an algorithm that computes $\mathrm{E}_{\mathrm{PD}}(\mathcal{T}, r)$ in $\Theta(n)$ time, assuming a constant time approximate evaluation of the hypergeometric probability function, the output of this algorithm is not guaranteed to be a good approximation of the actual expected value of the PD. Hence, it is a challenging open problem to devise a numerical method with which we can compute $\mathrm{E}_{\mathrm{PD}}(\mathcal{T}, r)$ both efficiently and with guaranteed precision. Next we provide an analytical expression for the standard deviation of the PD for all tip samples of a certain size.

Theorem 4. *Let $\mathcal{T}$ be a phylogenetic tree that contains s tips, and let r be a natural number with $r \leq s$. The standard deviation of the* PD *for a sample of exactly r tips of $\mathcal{T}$ is equal to:*

$$sd_{\mathrm{PD}}(\mathcal{T}, r) = \sqrt{\sum_{e \in E} \sum_{l \in E} w_e \cdot w_l \cdot (1 - \mathcal{F}_{\mathrm{PD}}(S, e, l, r)) - \mathrm{E}^2_{\mathrm{PD}}(\mathcal{T}, r)}, \qquad (6)$$

where:

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

Remark 2. The computation of the analytical expression of the standard deviation of the PD in Theorem 4, suffers from the same problems as the computation of the expected value of the PD; we still need to develop an efficient method that can approximate the hypergeometric quantities that appear in the formula of $sd_{\mathrm{PD}}(\mathcal{T}, r)$ and can also guarantee the precision of the final result. However, even if such a method was available, it is still not clear if it is possible to compute the double sum in (6) in subquadratic time with respect to the size of $\mathcal{T}$. This poses one more strong computational constraint on the calculation of the PDI.

3 Experimental Results

We have implemented all the algorithms that we present in Section 2 and we conducted experiments in order to assess their efficiency. The implementation was done in C++, using template programming that allows us to use number types of different precision. All the experiments that appear in the current version of the paper were executed using the `double` built-in C++ type. The experiments were executed on an Intel i5 four-core CPU where each core is a 2.67 GHz processor. The main memory of this computer is 4 Gigabytes.

The trees that we used in the experiments are subtrees of varying size that we extracted from a phylogenetic tree data set that contains 4510 tips and which represents the phylogenetic relations between all mammals [1]. We refer to the complete data set as the `mammals` data set. The input trees are provided in Newick tree format [5] in a `txt` file. As a result, each separate execution of one of our algorithms with a specific input tree took linear time with respect to the input size, in order to read and parse the tree. As a point of reference for the efficiency of our implementation, we did the same experiments using the `picante` software package, which is an extension of the `R` software environment [10].

In the experiments that we conducted, we examined the sensitivity of running time to variation in tree size and sample size by generating random prunings of the full `mammals` tree, and random communities assembled from those subtrees. For each tree and tip sample, we computed MPD, PD, NRI and PDI and recorded the time required to make this computation using our algorithms and `picante`.

In the first set of experiments we measure the running time of our algorithms, versus the `picante` implementation, given trees of different sizes while using a query sample of a fixed number of tips. More precisely, from the complete `mammals` data set we constructed nine tree instances, with the size of each

instance being equal to $n = 200 + 500k$ with k ranging from 0 to 8. For each instance we computed the MPD, PD, NRI and PDI given a sample of exactly 100 tips–see Figure 1, graphs a) and b).

In the second set of experiments, we studied the performance of the algorithms for various sample sizes; we fixed the input tree to be the full `mammals` tree (4510 tips) and sampled $10 + 50k$ tips from it, where k ranges from 0 to 10. When calculating NRI and PDI in `picante`, we used 100 random draws to estimate the expectation and deviation –see Figure 1, graphs c) and d). For in-

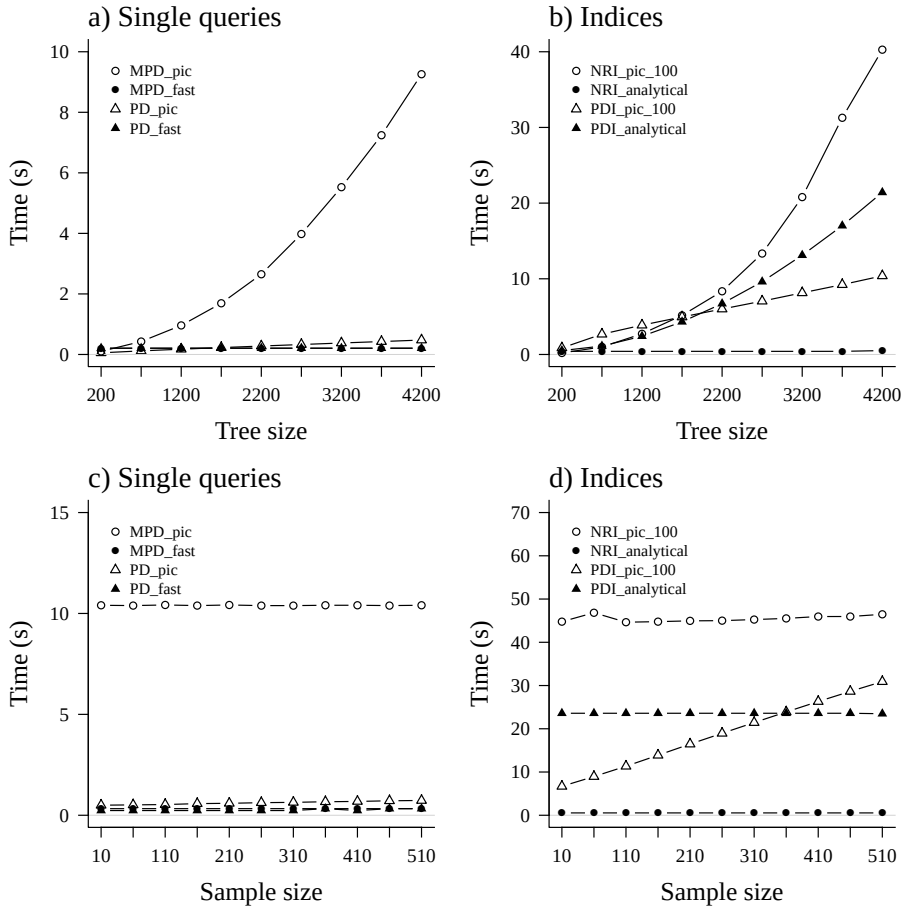

Fig. 1. The running times of the examined algorithms for trees of various size of tips, and a fixed sample of one hundred tips. In the two graphs on the left, MPD_fast, PD_fast are our implementations of the individual MPD and PD queries, while MPD_pic, PD_pic are the respective `picante` routines. In the two graphs on the right, NRI_analytical, and PDI_analytical refer to our implementations for computing the NRI and PDI indices, while NRI_*pic*_100 and PDI_*pic*_100 are the `picante` processes.

dividual MPD and PD queries, our algorithms are faster than those provided by `picante`. This is particularly true for MPD, for which the `picante` computation time scales superlinearly with respect to tree size. This behaviour appears because `picante` asks for the calculation of a pairwise distance matrix among *all* possible pairs of tips, requiring quadratic time with respect to the total

tips size (not sample size) of the tree. Similarly, the analytical solutions for the expectation and deviation of MPD values allowed substantial speed increases, particularly for large trees. In contrast, our algorithm for computing PDI displays better scaling than `picante` with respect to increasing sample size but worse scaling with respect to increasing tree size. This was expected since the algorithm that handles the analytical computation of the standard deviation of the PD runs in quadratic time with respect to the tree size. As mentioned in Remark 2, it seems quite difficult to design, or disprove the existence, of an algorithm with a subquadratic running time for this problem. However, even for large tree sizes and small sample sizes, our algorithm provides better running time, if one uses the standard 1000 random draws rather than 100 as illustrated.

Much of the running time for our algorithms for individual MPD and PD queries is spent reading the input tree. By sending 100 queries at once, using for all the full `mammals` tree, we estimate that the fixed time to perform MPD and PD queries is approximately 0.3 seconds, while the per-query time is between 0.001 and 0.002 seconds and depends weakly on sample size. In contrast, the fixed cost of a MPD query in `picante` is approximately 10.2 seconds, and a PD query is 0.4 seconds. The per-query time for MPD and PD in `picante` increases with tip sample size, from 0.05 to 0.3 seconds for PD and from 0.005 to 0.02 seconds for MPD (for 10 to 510 species).

In the next set of experiments, we simulated a real ecological data set using the full `mammals` tree with 1000 randomly assembled sets of species (e.g. communities), each with a number of species between 2 and 201, and calculated NRI and PDI for this data set. Our algorithms allowed us to calculate NRI for all samples in 2.3 seconds, while the `picante` functions (using 1000 random draws) required 2024.8 seconds. Our algorithms took 4581.4 seconds to calculate PDI, while `picante` functions took 10524 seconds. In this last case, we used just 200 random draws, 1000 would take roughly five times as long.

The analytical approach provides advantages beyond running time improvements. This is particularly true for NRI, as we are able to calculate exactly the expectation and standard deviation of MPD for a tree. We used this to assess the error introduced by randomized estimation of these values, using the full mammal tree and sample sizes of 20, 100 and 500 tips–see Fig. 2. The random method (using 100 draws) produced accurate estimates of the expectation of MPD, but estimates of the standard deviation varied widely around the true value, ranging by as much as 20%. Similar results were found for PDI, but we note that the calculation of the analytical solution for PDI introduces numerical errors, which have an unknown contribution to the final estimate.

4 Concluding Remarks

We described efficient algorithms for the analytical computation of the expectation and the standard deviation of popular measures in phylogeny, namely for MPD and PD. We also provided efficient algorithms for executing individual queries of these meausures on phylogenetic trees. It seems very interesting to extend these results to other kinds of measures that are used in phylogenetic studies; for example computing the expectation and the standard deviation of the mean nearest phylogenetic distance [12]. Also, an important open question that follows from our results is to show if it is possible to derive a method for computing the PDI both efficiently and with arbitrary precision.

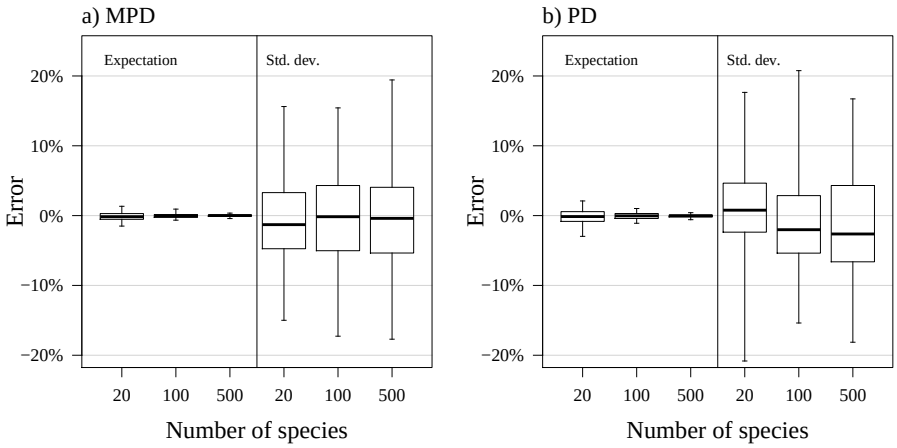

Fig. 2. Errors introduced in the estimation of NRI and PDI using randomization to estimate the expectation and standard deviation. The heavy line indicates the median error, the box shows the interquartile range, and whiskers show the full error range.

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. J. Cavendar-Bares, D.D. Ackerly, D. Baum and F.A. Bazzaz. Phylogenetic overdispersion in the assembly of Floridian oak communities. *American Naturalist* 163: 823–843, 2004.
3. D.P. Faith. Conservation evaluation and phylogenetic diversity. *Biological Conservation* 61: 1–10, 1992.
4. B. Faller, F. Pardi and M. Steel. Distribution of phylogenetic diversity under random extinction. *Journal of Theoretical Biology* 251:286–296, 2008.
5. J. Felsenstein. *PHYLIP: Phylogeny inference package, version 3.57c.* Distributed by the author, Department of Genetics, Univ. of Washington, 1995.
6. C.H. Graham and P.V.A. Fine. Phylogenetic beta diversity: linking ecological and evolutionary processes across space and time. *Ecology Letters* 11: 1265–1277, 2008.
7. K. Hartmann and M. Steel. Phylogenetic diversity: From combinatorics to ecology. O. Gascuel and M. Steel (eds.), *Reconstructing evolution: New mathematical and computational approaches*, Oxford University Press, 2007.
8. S.W. Kembel, D.D. Ackerly, S.P. Blomberg, W.K. Cornwell, P.D. Cowan, M.R. Helmus, H. Morlon and C.O. Webb. *Documentation for picante R package*, 2011.
9. W.D. Kissling, W.L. Eiserhardt, W.J. Baker, F. Borchsenius, T.L.P. Couvreur, H. Balslev and J.-C. Svenning. Cenozoic imprints on the phylogenetic structure of palm species assemblages worldwide. In *Proc. National Academy of Sciences* 109: 7379–7384, 2012.
10. R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, 2010.
11. M. Steel. Tools to construct and study big trees: A mathematical perspective. In Trevor Hodkinson, John Parnell, and Steve Waldren (eds.), *Reconstructing the Tree of Life: Taxonomy and Systematics of Species Rich Taxa.* CRC Press, pages 97–112, 2007.
12. 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.
13. C. Webb, D. Ackerly, S. Kembel. *Phylocom Users Manual, version 4.2*, 2012.