RESEARCH ARTICLE

Fast Computations for Measures of Phylogenetic Beta Diversity

Constantinos Tsirogiannis*☯, Brody Sandel☯

MADALGO and Department of Bioscience, Aarhus University, Aarhus, Denmark

☯ These authors contributed equally to this work.
* constant@madalgo.au.dk

Abstract

For many applications in ecology, it is important to examine the phylogenetic relations between two communities of species. More formally, let $\mathcal{T}$ be a phylogenetic tree and let A and B be two samples of its tips, representing the examined communities. We want to compute a value that expresses the phylogenetic diversity between A and B in $\mathcal{T}$. There exist several measures that can do this; these are the so-called phylogenetic beta diversity (β-diversity) measures. Two popular measures of this kind are the Community Distance (CD) and the Common Branch Length (CBL). In most applications, it is not sufficient to compute the value of a beta diversity measure for two communities A and B; we also want to know if this value is relatively large or small compared to all possible pairs of communities in $\mathcal{T}$ that have the same size. To decide this, the ideal approach is to compute a standardised index that involves the mean and the standard deviation of this measure among all pairs of species samples that have the same number of elements as A and B. However, no method exists for computing exactly and efficiently this index for CD and CBL. We present analytical expressions for computing the expectation and the standard deviation of CD and CBL. Based on these expressions, we describe efficient algorithms for computing the standardised indices of the two measures. Using standard algorithmic analysis, we provide guarantees on the theoretical efficiency of our algorithms. We implemented our algorithms and measured their efficiency in practice. Our implementations compute the standardised indices of CD and CBL in less than twenty seconds for a hundred pairs of samples on trees with $7 \cdot 10^4$ tips. Our implementations are available through the R package `PhyloMeasures`.

OPEN ACCESS

Citation: Tsirogiannis C, Sandel B (2016) Fast Computations for Measures of Phylogenetic Beta Diversity. PLoS ONE 11(4): e0151167. doi:10.1371/journal.pone.0151167

Editor: M. Sohel Rahman, Bangladesh University of Engineering and Technology, BANGLADESH

Received: April 28, 2015

Accepted: February 24, 2016

Published: April 7, 2016

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by Massive Data Algorithmics (MADALGO) (http://madalgo.au.dk/) and Ecoinformatics and Biodiversity (http://bios.au.dk/en/about-bioscience/organisation/ecoinformatics-and-biodiversity/ecoinformatics-researchprofile/). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Introduction

Ecologists often distinguish three kinds of diversity. Alpha diversity describes the diversity of one sample (such as the number of plant species in a vegetation plot), beta diversity describes the dissimilarity between a pair of samples, and gamma diversity describes the diversity of a large set of samples [1, 2]. These concepts can be applied to a number of diversity measures, including species richness, functional diversity and phylogenetic diversity [3, 4]. Phylogenetic beta diversity describes the phylogenetic distance among pairs of communities [5]. It is an increasingly widely-used concept in ecology, with many interesting recent applications, for example in biogeographical regionalization [6] and understanding broad-scale species distributions [7, 8]. Accordingly, there has also been a recent proliferation of methods to describe beta diversity [9].

Recent work on phylogenetic diversity has made great progress in developing efficient approaches to calculating alpha diversity measures (e.g. the works by Steel, O'Dwyer et al., Tsirogiannis et al., Nipperess and Matsen, Chao et al. [10–15]). On the other hand, so far there are no known algorithms that can efficiently compute phylogenetic beta diversity on large numbers of samples given large phylogenetic trees. This limits their practical applications, and the problem will grow as ecologists analyze larger and larger trees and numbers of samples. To fulfill their potential in a world of growing datasets, it is important that phylogenetic beta diversity metrics can be computed efficiently. Here, we focus on two widely-used metrics related to beta diversity. The first, Community Distance (CD), is the beta diversity analog of the Mean Pairwise Distance (MPD) alpha diversity measure. The second, Common Branch Length (CBL) is not strictly speaking a beta diversity measure (because it describes similarity, rather than difference, between two samples), but it is intimately related to two frequently-used measures, PhyloSor and UniFrac, of which UniFrac is a measure of dissimilarity and PhyloSor can be easily converted to one using 1-PhyloSor [9]. Our primary goal is to develop methods to compute these measures efficiently even on very large trees.

A notable lack in the literature on phylogenetic beta diversity measures is a correction for the species richness of the samples. With respect to phylogenetic alpha diversity measures such as the Mean Pairwise Distance (MPD), the Mean Nearest Taxon Distance (MNTD), and the Phylogenetic Diversity (PD), it is well-known that these measures depend on species richness, and so it is common practice to compute an index that standardizes these values against the species richness of the sample using one of a number of possible null models [16, 17]. This index combines the original value of the measure for a species sample K on a given tree $\mathcal{T}$, with the mean and the variance of this measure's value among all species samples in $\mathcal{T}$ that have the same number of elements as K. However, despite the dependencies of phylogenetic beta diversity measures on species richness, such corrections are not usually applied, perhaps because of the difficulty in standardizing against two species richness values simultaneously. And when such corrections are attempted, they involve slow and imprecise randomization tests [8, 18–20]. Hence, a second goal is to allow such standardization for phylogenetic beta diversity measures by developing algorithms to efficiently and precisely compute the expectation and variance of a beta diversity measure given a tree and pair of species richness values. This expectation and variance depend on a null model which describes the probability of drawing any particular pair of species sets with the specified species richness values. Here, we consider the case that all such pairs are equally probable. It is also possible to define a variety of unequal-probability cases, where the probability can depend on species abundances [21].

Care must be exercised in the interpretation of phylogenetic beta diversity measures, particularly CD and its standardised version [22]. In particular, CD is not a suitable dissimilarity measure, as the CD for a community A and itself is generally not zero, and may be greater than the CD between A and a different community B. As long as this limitation is understood, the measure can still be a useful description of phylogenetic beta diversity in some contexts (e.g. as indicated in the works of Swenson et al. and Pellisier et al. [4, 20]). And, as with measures of standardized alpha diversity, it is crucial to carefully consider the appropriate species pool upon which to base the standardization [5]. When considered carefully, however, the dependence on pool size may actually prove to be a useful feature, allowing more nuanced ecological inference [21, 23], as one allows pool size to vary.

In the present work, we provide results that lead to efficient computations of the the beta diversity measures CD and CBL, and their standardized indices. First, given a phylogenetic tree

$\mathcal{T}$, we show how to derive analytical expressions for computing exactly the mean and the variance of the CD and the CBL among pairs of species samples in $\mathcal{T}$ that have given sizes. Based on these expressions, for each of these two measures we describe efficient algorithms for computing the measure's standardized index for a given pair of samples (and also the measure's original value for this pair). These algorithms are developed based on standard techniques in Algorithms Design, and we provide theoretical guarantees for their performance. We implemented all of the algorithms the we describe, and we conducted experiments that exhibit their efficiency in practice. We measured the performance of our algorithms on trees of several sizes, extracted from a phylogeny that has 71,181 tips and 83,751 nodes in total. As we show later in detail, our implementations ran very fast even for the largest trees that we considered; for each of the two measures, our programs managed to compute in less than a minute the standardized indices for one hundred pairs of species samples on the complete tree with the 71,181 tips. We have made the programs that we developed publicly available through the open source R package `PhyloMeasures` [24]. To exhibit the strength of fast beta diversity computations, we also present an application example; we compute heat maps that illustrate the beta diversity values between a focal point assemblage and assemblages distributed on a world grid. Given our efficient algorithms, it is now possible to produce several such maps in high resolution, something that was infeasible with previously existing software.

Analysis

Terminology and Notation

Let $\mathcal{T}$ be a phylogenetic tree. We use V to indicate the set of nodes (representing the species/taxa) in $\mathcal{T}$, and we use E to denote the set of edges (links between nodes) in this tree. For an edge $e \in E$, we denote the (always positive) weight of this edge by w_e. Depending on the context, the edge weights in $\mathcal{T}$ may represent time intervals, molecular distance, or some other notion of difference between taxa. The analysis in this paper does not depend on this notion of difference. We denote the set of leaf nodes in $\mathcal{T}$ (taxa that do not have any child species) by S. We refer to these nodes as the *tips* of $\mathcal{T}$. We indicate the number of the tips in $\mathcal{T}$ by $s(\mathcal{T})$ or simply s, and we indicate the number of all the nodes in $\mathcal{T}$ by n. We consider that $\mathcal{T}$ is a rooted tree.

For any edge e be in $\mathcal{T}$, we use $\mathrm{Ch}(e)$ to denote the edges that are adjacent to the child node of e. We call these edges the *children* of e. Let u be a node in $\mathcal{T}$, and let e be the edge that connects u with its parent node. We use interchangeably $\mathcal{T}(u)$ and $\mathcal{T}(e)$ to denote the subtree of $\mathcal{T}$ whose root is u. We use $S(u)$ and $S(e)$ to indicate the set of tips that appear in $\mathcal{T}(u)$. We denote the number of these tips by $s(u)$ and $s(e)$.

Let u, v be two nodes in $\mathcal{T}$. We call a *simple path* between these nodes the cycle-free sequence of edges that we have to traverse in order to reach u from v. We call the *cost* of this path the sum of the weights of all the edges in the path. We denote this cost by $cost(u, v)$. Since $\mathcal{T}$ is a tree, there exists a unique simple path between any pair of nodes in $\mathcal{T}$. We call the *height* of $\mathcal{T}$ the maximum number of edges that appear on a simple path between the root of $\mathcal{T}$ and any leaf. We represent the height of $\mathcal{T}$ by $h(\mathcal{T})$. Let $R \subseteq S$ be any sample (subset) of the tips in $\mathcal{T}$. We denote the number of tips in this sample by $|R|$. We indicate the set of all paths that connect two elements in R by $\mathrm{Paths}(R)$, that is:

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

We denote the set whose elements are all subsets of S that have cardinality exactly r by $\mathrm{Sub}(S, r)$. 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 $s_R(e)$.

For a given phylogenetic tree $\mathcal{T}$ we call the *total path cost of* $\mathcal{T}$ 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, the toal path cost of a tree $\mathcal{T}$ is equal to:

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

Let e be an edge of $\mathcal{T}$. We call the *total path cost of* e the sum of the costs of all simple paths in Paths(S) that contain e. We denote this quantity by $TC(e)$, thus:

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

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

The Community Distance

Let $\mathcal{T}$ be a phylogenetic tree, and let $A, B \subseteq S$ be two samples of its tips with $|A| = a$, and $|B| = b$. The *Community Distance* (CD) between A and B is equal to the sum of the costs of all paths that connect a tip in A with a tip in B, divided by the total number of these paths. Therefore, the community distance between A and B is equal to:

$$\mathrm{CD}(\mathcal{T}, A, B) = \frac{1}{ab}\sum_{u\in A}\sum_{v\in B} cost(u, v).$$

Samples A and B may not necessarily be of equal size. **The CD is analogous to the *Mean Pairwise Distance* measure for computing the average distance between two samples of species**.

The β Net Relatedness Index. Given a phylogenetic tree $\mathcal{T}$ and two samples of its tips A, B such that $|A| = a$ and $|B| = b$, the standardized index of the CD for these samples is equal to:

$$\mathrm{NRI}_{\beta}(\mathcal{T}, A, B) = \frac{\mathrm{CD}(\mathcal{T}, A, B) - \mathrm{E}_{\mathrm{CD}}(\mathcal{T}, a, b)}{sd_{\mathrm{CD}}(\mathcal{T}, a, b)}, \quad (1)$$

where $\mathrm{E}_{\mathrm{CD}}(\mathcal{T}, a, b)$ and $sd_{\mathrm{CD}}(\mathcal{T}, a, b)$ are respectively the expectation and the standard deviation of the CD for all pairs of tip samples such that one sample contains a tips and the other sample contains b tips. The following theorem provides an analytical expression for the expectation of the CD.

Theorem 1. *Let $\mathcal{T}$ be a phylogenetic tree that has s tips, and let a, b be positive integers with $a, b \leq s$. The expected value of the CD among all pairs of tip samples in $\mathcal{T}$, such that one sample consists of a tips, and the other consists of b tips, is equal to*:

$$\mathrm{E}_{\mathrm{CD}}(\mathcal{T}, a, b) = \frac{2}{s^2} \cdot TC(\mathcal{T})$$

Proof. The expectation of the CD is equal to:

$$\begin{aligned}&\mathrm{E}_{\mathrm{CD}}(\mathcal{T}, a, b) = \\ &\mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}\left[\frac{1}{ab}\sum_{\{u,v\}\in S} cost(u,v)\cdot(AP_A(u)\cdot AP_B(v) + AP_A(v)\cdot AP_B(u))\right] = \\ &\frac{1}{ab}\sum_{\{u,v\}\in S} cost(u,v)\cdot \mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}[AP_A(u)\cdot AP_B(v) + AP_A(v)\cdot AP_B(u)],\end{aligned}$$

where $AP_A(u)$ is a random variable such that $AP_A(u) = 1$ if $u \in A$, otherwise its value is zero. It holds that:

$$\begin{aligned}&\mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}[AP_A(u)\cdot AP_B(v) + AP_A(v)\cdot AP_B(u)] = \\ &2\cdot \mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}[AP_A(u)\cdot AP_B(v)] = \frac{2ab}{s^2},\end{aligned}$$

which yields the analytical expression for the expectation of the CD measure.

Next we present an analytical expression for calculating the standard deviation of the CD measure.

Theorem 2. *Let $\mathcal{T}$ be a phylogenetic tree that has s tips, and let a, b be positive integers with a, b $\leq$ s. The standard deviation of the CD among all pairs of tip samples in $\mathcal{T}$, such that one sample consists of a tips, and the other consists of b tips, is equal to*:

$$\sqrt{k_1\cdot TC^2(\mathcal{T}) + (k_2 - k_1)\sum_{u\in S} TC^2(u) + (k_1 - 2k_2 + k_3)\sum_{e\in E} w_e\cdot TC(e) - \mathrm{E}^2_{\mathrm{CD}}(\mathcal{T}, a, b)},$$

where: $k_1 = \frac{4(a-1)(b-1)}{abs^2(s-1)^2}$, $k_2 = \frac{2(a-1)(b-1)}{abs^2(s-1)^2} + \frac{b-1}{abs^2(s-1)} + \frac{a-1}{abs^2(s-1)}$, $k_3 = \frac{2(a-1)(b-1)}{abs^2(s-1)^2} + \frac{2}{abs^2}$.

Proof. The standard deviation of the CD is equal to:

$$\sqrt{\mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}[\mathrm{CD}^2(\mathcal{T}, A, B)] - \mathrm{E}^2_{\mathrm{CD}}(\mathcal{T}, a, b)}.$$

We already provided an analytical expression for the expectation of the CD; hence, we now focus on deriving an expression for the expectation of the squared value of this measure. We get that:

$$\begin{aligned}&\mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}[\mathrm{CD}^2(\mathcal{T}, A, B)] = \\ &\mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}\left[\frac{1}{a^2b^2}\sum_{\{u,v\}\in S}\sum_{\{x,z\}\in S} cost(u,v)\cdot cost(x,z)\cdot \mathcal{P}(u,v,A,B)\cdot \mathcal{P}(x,z,A,B)\right] = \\ &\frac{1}{a^2b^2}\sum_{\{u,v\}\in S}\sum_{\{x,z\}\in S} cost(u,v)\cdot cost(x,z)\cdot \mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}[\mathcal{P}(u,v,A,B)\cdot \mathcal{P}(x,z,A,B)],\end{aligned} \qquad (2)$$

where $\mathcal{P}(u, v, A, B) = AP_A(u)\cdot AP_B(v) + AP_A(v)\cdot AP_B(u)$.

From the last expression we get:

$$\mathrm{E}_{\substack{A \,\in\, \mathrm{Sub}(S,\,a)\\ B \,\in\, \mathrm{Sub}(S,\,b)}}[\mathcal{P}(u,v,A,B)\cdot\mathcal{P}(x,z,A,B)] =$$

$$\begin{cases} \frac{4a(a-1)b(b-1)}{s^2(s-1)^2} & \text{if } \{u,v\}\cap\{x,z\}=\emptyset. \\ \frac{2a(a-1)b(b-1)}{s^2(s-1)^2}+\frac{a(a-1)b}{s^2(s-1)}+\frac{ab(b-1)}{s^2(s-1)} & \text{if } |\{u,v\}\cap\{x,z\}|=1. \\ \frac{2a(a-1)b(b-1)}{s^2(s-1)^2}+\frac{2ab}{s^2} & \text{if } \{u,v\}=\{x,z\}. \end{cases}$$

Thus, we can rewrite Eq (2) as follows:

$$\begin{aligned} & k_1 \sum_{\{u,v\}\in S} \sum_{\substack{\{x,z\}\,\in\, S\\ \{u,v\}\,\cap\,\{x,z\}\,=\,\emptyset}} cost(u,v)\cdot cost(x,z) \\ & + k_2 \sum_{\{u,v\}\in S} \sum_{\substack{\{x,z\}\,\in\, S\\ |\{u,v\}\,\cap\,\{x,z\}|\,=\,1}} cost(u,v)\cdot cost(x,z) + k_3 \sum_{\{u,v\}\in S} cost^2(u,v), \end{aligned} \tag{3}$$

where $k_1 = \frac{4a(a-1)b(b-1)}{s^2(s-1)^2}$, $k_2 = \frac{2a(a-1)b(b-1)}{s^2(s-1)^2}+\frac{a(a-1)b}{s^2(s-1)}+\frac{ab(b-1)}{s^2(s-1)}$, and $k_3 = \frac{2a(a-1)b(b-1)}{s^2(s-1)^2}+\frac{2ab}{s^2}$.

Next, we simplify the expression that appears in Eq (3). The last of the three sums in this expression can be rewritten as:

$$\sum_{\{u,v\}\in S} cost^2(u,v) = \sum_{\{u,v\}\in S}\sum_{e,l\in p(u,v)} w_e\cdot w_l = \sum_{\{u,v\}\in S}\sum_{e,l\in p(u,v)} w_e\cdot w_l = \tag{4}$$

$$\sum_{e\in E} w_e \sum_{\substack{p(a,\,b)\,\in\,\mathrm{Paths}(S)\\ e\,\in\, p(a,\,b)}} \sum_{l\in p(a,b)} w_l = \sum_{e\in E} w_e\cdot TC(e) \tag{5}$$

The second double sum in Eq (3) can be simplified as follows:

$$\begin{aligned} & \sum_{\{u,v\}\in S} \sum_{\substack{\{x,z\}\,\in\, S\\ |\{u,v\}\,\cap\,\{x,z\}|\,=\,1}} cost(u,v)\cdot cost(x,z) = \\ & \sum_{\{u,v\}\in S} cost(u,v)\left[\sum_{k\in S-\{u\}} cost(u,k) + \sum_{m\in S-\{v\}} cost(m,v) - 2\ cost(u,v)\right] = \\ & \sum_{u\in S} TC^2(u) - 2\sum_{\{u,v\}\in S} cost^2(u,v) = \sum_{u\in S} TC^2(u) - 2\sum_{e\in E} w_e\cdot TC(e) \end{aligned} \tag{6}$$

The first double sum in Eq (3) can be rewritten as:

$$\begin{aligned} & \sum_{\{u,v\}\in S} \sum_{\substack{\{x,z\}\,\in\, S\\ \{u,v\}\,\cap\,\{x,z\}\,=\,\emptyset}} cost(u,v)\cdot cost(x,z) = \\ & \sum_{\{u,v\}\in S} cost(u,v)[TC(\mathcal{T}) - TC(u) - TC(v) + cost(u,v)] = \\ & TC^2(\mathcal{T}) - \sum_{u\in S} TC^2(u) + \sum_{e\in E} w_e\cdot TC(e) \end{aligned} \tag{7}$$

Combining Eqs (5), (6) and (7) with Eq (3) we get the expression that appears in the definition of the theorem.

The Common Branch Length

Let $\mathcal{T}$ be a phylogenetic tree and let $A, B \subseteq S$ be two samples of its tips. The *Common Branch Length* (CBL) between A and B is the sum of the weights of all edges that appear both in $\mathcal{T}(A)$ and in $\mathcal{T}(B)$. Recall that $\mathcal{T}(R)$ denotes the smallest subtree in $\mathcal{T}$ that contains all the tips of a sample R. Therefore, the CBL between A and B is equal to:

$$\mathrm{CBL}(\mathcal{T}, A, B) = \sum_{e \in \mathcal{T}(A) \cap \mathcal{T}(B)} w_e.$$

Samples A and B may not have the same number of elements. **The CBL is analogous to the *Phylogenetic Diversity* measure for computing a diversity value between two samples of species.**

The Common Length Index. For a phylogenetic tree $\mathcal{T}$ and two samples of its tips A, B such that $|A| = a$, and $|B| = b$, we denote the standardized index of the CBL by:

$$\mathrm{CLI}(\mathcal{T}, A, B) = \frac{\mathrm{CBL}(\mathcal{T}, A, B) - \mathrm{E}_{\mathrm{CBL}}(\mathcal{T}, a, b)}{sd_{\mathrm{CBL}}(\mathcal{T}, a, b)}, \tag{8}$$

where $\mathrm{E}_{\mathrm{CBL}}(\mathcal{T}, a, b)$ and $sd_{\mathrm{CBL}}(\mathcal{T}, a, b)$ are respectively the expected value and the standard deviation of the CBL for all possible pairs of tip samples where one sample contains exactly a tips and the other contains exactly b tips. We call this index the *Common Length Index* of A and B. The following two theorems provide analytical expressions for the expectation and standard deviation of the CBL.

Theorem 3. *Let $\mathcal{T}$ be a phylogenetic tree that contains s tips, and let a, b be two positive integers such that a, b $\leq$ s. The expected value of the CBL among all pairs of tip samples in $\mathcal{T}$ such that one sample consists of a tips and the other sample consists of b tips, is equal to*:

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

Proof. The expectation of the CBL for two independtly selected samples A and B is equal to:

$$\mathrm{E}_{\mathrm{CBL}}(\mathcal{T}, a, b) = \mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}\left[\sum_{e \in S}(w_e \cdot AP_{A,B}(e))\right] = \sum_{e \in S} w_e \cdot \mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}[AP_{A,B}(e)],$$

where $AP_{A,\,B}(e)$ is equal to one if $e \in \mathcal{T}(A) \cap \mathcal{T}(B)$, otherwise is zero. The expectation of $AP_{A,\,B}(e)$ is equal to:

$$\begin{aligned}
&\mathrm{E}_{\substack{A \in \mathrm{Sub}(S,a) \\ B \in \mathrm{Sub}(S,b)}}[AP_{A,B}(e)] = Pr[e \in \mathcal{T}(A) \cap \mathcal{T}(B)] = Pr[e \in \mathcal{T}(A)] \cdot Pr[e \in \mathcal{T}(B)] = \\
&(1 - Pr[e \notin \mathcal{T}(A)])(1 - Pr[e \notin \mathcal{T}(B)]) = \\
&\left(1 - \frac{\binom{s(e)}{a} + \binom{s-s(e)}{a}}{\binom{s}{a}}\right)\left(1 - - \frac{\binom{s(e)}{b} + \binom{s-s(e)}{b}}{\binom{s}{b}}\right),
\end{aligned}$$

and the expression for the expectation follows.

Theorem 4. *Let $\mathcal{T}$ be a phylogenetic tree that contains s tips, and let a, b be two positive integers such that a, b $\leq$ s. The standard deviation of the CBL among all pairs of tip samples in $\mathcal{T}$*

such that one sample consists of a tips and the other sample consists of b tips, is equal to:

$$sd_{\mathrm{CBL}}(\mathcal{T},a,b) = \sqrt{\sum_{e\in E}\sum_{l\in E} w_e \cdot w_l (1-\mathcal{F}(S,e,l,a))(1-\mathcal{F}(S,e,l,b)) - \mathrm{E}^2_{\mathrm{CBL}}(\mathcal{T},a,b)}, \tag{9}$$

where:

$$\mathcal{F}(S,e,l,r) = \begin{cases} \dfrac{\binom{s(e)}{r} + \binom{s-s(l)}{r} - \binom{s(e)-s(l)}{r}}{\binom{s}{r}} & \text{if } l \in \mathcal{T}(e). \\[2ex] \dfrac{\binom{s(l)}{r} + \binom{s-s(e)}{r} - \binom{s(l)-s(e)}{r}}{\binom{s}{r}} & \text{if } e \in \mathcal{T}(l). \\[2ex] \dfrac{\binom{s-s(e)}{r} + \binom{s-s(l)}{r} - \binom{s-s(e)-s(l)}{r}}{\binom{s}{r}} & \text{otherwise.} \end{cases} \tag{10}$$

Proof. The standard deviation of the CBL is equal to:

$$\sqrt{\mathrm{E}_{\substack{A\,\in\,\mathrm{Sub}(S,\,a)\\ B\,\in\,\mathrm{Sub}(S,\,b)}}[\mathrm{CBL}^2(\mathcal{T},A,B)] - \mathrm{E}^2_{\mathrm{CBL}}(\mathcal{T},a,b)}.$$

In Theorem 3 we provided an expression for the expectation of the CD. It remains to derive an expression for the expectation of the squared value of this measure. We get that:

$$\mathrm{E}_{\substack{A\,\in\,\mathrm{Sub}(S,\,a)\\ B\,\in\,\mathrm{Sub}(S,\,b)}}[\mathrm{CBL}^2(\mathcal{T},A,B)] = \sum_{e\in E}\sum_{l\in E} w_e \cdot w_l \cdot \mathrm{E}_{\substack{A\,\in\,\mathrm{Sub}(S,\,a)\\ B\,\in\,\mathrm{Sub}(S,\,b)}}[AP_{A,B}(e)\cdot AP_{A,B}(l)] = \tag{11}$$

$$\sum_{e\in E}\sum_{l\in E} w_e \cdot w_l \cdot Pr[\{e,l\in\mathcal{T}(A)\}\cap\{e,l\in\mathcal{T}(B)\}] \tag{12}$$

Events $\{e,l\in\mathcal{T}(A)\}$ and $\{e,l\in\mathcal{T}(B)\}$ are independent, so the probability value in the last sum can be rewritten as:

$$Pr[\{e,l\in\mathcal{T}(A)\}\cap\{e,l\in\mathcal{T}(B)\}] = Pr[e,l\in\mathcal{T}(A)]\cdot Pr[e,l\in\mathcal{T}(B)]. \tag{13}$$

Value $Pr[e,l\in\mathcal{T}(A)]$ is the probability that both edges e, l appear in $\mathcal{T}(A)$. Recall that $\mathcal{T}(A)$ is the smallest subtree of $\mathcal{T}$ that contains all tips in A. An edge e does not appear in $\mathcal{T}(A)$ if either all, or none of the elements in A are tips in the subtree of e. Given that, we get:

$$\begin{aligned} &Pr[e,l\in\mathcal{T}(A)] = 1 - Pr[(e\notin\mathcal{T}(A))\cup(l\notin\mathcal{T}(A))] = \\ &\quad 1 - Pr[e\notin\mathcal{T}(A)] - Pr[l\notin\mathcal{T}(A)] + Pr[(e\notin\mathcal{T}(A))\cap(l\notin\mathcal{T}(A))] = \\ &\quad 1 - \frac{\binom{s(e)}{a} + \binom{s-s(e)}{a} - \binom{s(l)}{a} + \binom{s-s(l)}{a}}{\binom{s}{a}} + Pr[(e\notin\mathcal{T}(A))\cap(l\notin\mathcal{T}(A))] \end{aligned} \tag{14}$$

For the probability value $Pr[(e\notin\mathcal{T}(A))\cap(l\notin\mathcal{T}(A))]$ we distinguish three cases:

(I). Edge $l\in\mathcal{T}(e)$:

$$Pr[(e\notin\mathcal{T}(A))\cap(l\notin\mathcal{T}(A))] = \frac{\binom{s-s(e)}{a} + \binom{s(l)}{a} + \binom{s(e)-s(l)}{a}}{\binom{s}{a}} \tag{15}$$

(II). Edge $e \in T(l)$. This case is symmetric to case (I):

$$Pr[(e \notin \mathcal{T}(A)) \cap (l \notin \mathcal{T}(A))] = \frac{\binom{s-s(l)}{a} + \binom{s(e)}{a} + \binom{s(l)-s(e)}{a}}{\binom{s}{a}} \tag{16}$$

(III). Subtrees $\mathcal{T}(e)$ and $\mathcal{T}(l)$ do not contain each other:

$$Pr[(e \notin \mathcal{T}(A)) \cap (l \notin \mathcal{T}(A))] = \frac{\binom{s(e)}{a} + \binom{s(l)}{a} + \binom{s-s(e)-s(l)}{a}}{\binom{s}{a}} \tag{17}$$

The probability value $Pr[(e, l \in \mathcal{T}(B))]$ can be expressed in a similar manner. The analytical expression for the standard deviation of the CBL follows by combining Eqs (11)–(17).

Design of Algorithms

Based on the analytical expressions that we presented in the previous sections, we designed efficient algorithms for calculating the value and the standardized indices of the CD and the CBL. Next, we present in short how we designed these algorithms, and we also give a theoretical measure for their efficiency. Before continuing with the description of the algorithms, we explain standard concepts and notation from the field of Algorithms Design that we use to describe the efficiency of our algorithms.

In many applications, it is important to define a simple bound that describes the order of growth for a given function. For example, let G be a set of n real numbers. Suppose that we want to count the number of possible subsets that can be created by picking four elements of G. The exact number of these subsets is equal to $f(n) = \frac{1}{24}(n^4 - 6n^3 + 11n^2 - 6n)$. But, this expression is quite complicated. Suppose that we want to express how fast $f(n)$ grows as the value of n increases. In that case, for large values of n, the term that influences the value of $f(n)$ the most is n^4. The standard way to express this in short is to say that $f(n) = O(n^4)$. The notation $O(n^4)$ is used to indicate that the value of $f(n)$ grows roughly as fast as n^4 when n becomes large. This is known as the *Big-Oh* notation. More formally, let $g(n)$ and $f(n)$ be two functions. We say that $f(n) = O(g(n))$ if there exist two positive constants c and n_0 such that $f(n) \leq c \cdot g(n)$ for every $n \geq n_0$.

In the fields of Algorithms Design, the $O(\cdot)$ notation is used extensively for measuring several aspects of an algorithm's efficiency. The standard way of measuring the efficiency of an algorithm is to count the number of basic operations that would take place during its execution. When we refer to basic operations, we mean all simple mathematical operations (such as comparisons, additions, divisions), but also standard operations like initialising a variable, or assigning a value to this variable. Let $\mathcal{A}$ be an algorithm, and suppose we have an input dataset for this algorithm that has size n; for instance a phylogenetic tree $\mathcal{T}$ that consists of n elements. Instead of counting exactly the number of basic operations that take place when $\mathcal{A}$ processes $\mathcal{T}$, we can bound this number using the *Big-Oh* notation. Ideally, we would like to prove that, for some function $g(n)$, algorithm $\mathcal{A}$ takes $O(g(n))$ operations to process *any* input of size n. In that case, we say that the *worst case running time complexity* of $\mathcal{A}$, or simply the *worst case time complexity* of this algorithm is $O(g(n))$.

Next we describe in short the algorithms that we designed for computing the standardized indices of the CD and the CBL. For each of the algorithms that we describe, we also provide a bound for its worst case time complexity. Let $\mathcal{T}$ be a phylogenetic tree, and let A and B be two samples of tips in $\mathcal{T}$ such that A has a tips and B has b tips. As described in Eqs (1) and (8), to compute the standardized index of either CD or CBL we need to calculate three values; we need to calculate the value of one of these measures for samples A and B, and we need to compute the expectation and the standard deviation of this measure among all pairs of samples in $\mathcal{T}$ where one sample has a tips and the other sample has b tips. Next, for each measure, we describe an algorithm for computing each of these three values.

Algorithms for computing the CD and NRI$_\beta$

Computing the value of the CD for a given pair of samples. To compute the value of the CD efficiently, we rewrite the expression of this measure so that it can be evaluated with a small number of basic operations. Let e be an edge in $\mathcal{T}$, and let Num(A, B, e) denote the number of simple paths that connect a tip in sample A with a tip in sample B, and contain edge e. The value of the CD for samples A and B can be rewritten as:

$$\mathrm{CD}(\mathcal{T}, A, B) = \sum_{e \in \mathcal{T}} w_e \cdot \mathrm{Num}(A, B, e). \tag{18}$$

For any edge $e \in \mathcal{T}$, value $Num(A, B, e)$ is equal to:

$$Num(A, B, e) = s_A(e) \cdot (b - s_B(e)) + s_B(e) \cdot (a - s_A(e)). \tag{19}$$

Therefore, computing $\mathrm{CD}(\mathcal{T}, A, B)$ boils down to computing values $s_A(e)$ and $s_B(e)$ for every edge $e \in \mathcal{T}$. We can compute all these values in $O(n)$ operations using a simple recursive algorithm; for each edge e that we process, we first calculate values $s_A(e')$ and $s_B(e')$, for every edge $e' \in Ch(e)$; recall that Ch(e) is the set of the edges that are adjacent to the child node of e. Then, we can calculate values $s_A(e)$ and $s_B(e)$ for e by simply adding the corresponding values of the edges in Ch(e). We can do this with only $O(n)$ operations, by traversing the edges in $\mathcal{T}$ appropriately. We start from the root of $\mathcal{T}$, and traverse the tree top to bottom. For each edge e that we encounter for the first time, we consider two cases; if e is adjacent to a tip, we check if this tip belongs to A or B, and we set values $s_A(e)$ and $s_B(e)$ to one or zero accordingly. Then, we move upwards in the tree and we use $s_A(e)$ and $s_B(e)$ to calculate the corresponding values for the parent edge of e. If e is not adjacent to a tip, then for every edge $l \in Ch(e)$ we first visit the subtree of l and we calculate recursively the values $s_A(\cdot)$ and $s_B(\cdot)$ for every edge in this subtree. After calculating these values for every $l \in Ch(e)$, we can compute $s_A(e)$ and $s_B(e)$ in $O(Ch(e))$ time by evaluating sums $s_A(e) = \Sigma_{l \in Ch(e)} s_A(l)$ and $s_B(e) = \Sigma_{l \in Ch(e)} s_B(l)$. Such a traversal is known as a *post-order* traversal, and requires visiting each edge in $\mathcal{T}$ at most two times. Since $\mathcal{T}$ has $O(n)$ edges, this traversal requires $O(n)$ operations. We also perform $O(Ch(e))$ arithmetic operations to compute $s_A(e)$ and $s_B(e)$ for every edge in $\mathcal{T}$, which sums up to $O(n)$ time operations in total. After calculating values $s_A(e)$ and $s_B(e)$, we can compute $CD(\mathcal{T}, A, B)$ with $O(n)$ arithmetic operations using Eqs (18) and (19). Therefore, the total number of operations that are needed to compute the value of the CD for two tip samples in $\mathcal{T}$ is $O(n)$.

Computing the expectation of the CD. In Theorem 1 we provide an analytical expression for the expectation of the CD. Evaluating this expression boils down to evaluating $TC(\mathcal{T})$. Recall that $TC(\mathcal{T})$ is equal to the sum of the costs of all simple paths that connect pairs of tips in $\mathcal{T}$. We can rewrite this values as:

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

Therefore, to compute the expectation of the CD, it remains to compute $s(e)$ for every edge e in $\mathcal{T}$. This can be done in $O(n)$ operations with a post-order traversal of $\mathcal{T}$, as described also for the algorithm that calculates the value of the CD for two samples. Hence, we can compute the expectation for this measure with only $O(n)$ operations.

Computing the standard deviation of the CD. For the standard deviation of the CD we presented an analytical expression in Theorem 2. This expression contains sums that have $O(n)$ terms in total. Among other terms, these sums contain values $TC(\mathcal{T})$ and $TC(e)$ for every edge e in $\mathcal{T}$. In the algorithm that we described for computing the expectation of the CD, we showed already how we can compute $TC(\mathcal{T})$ with $O(n)$ operations. We can also compute $TC(e)$ for every edge e in the tree, with $O(n)$ operations in total. This can be done as follows. For an edge e, we use $w_{\mathcal{T}}(e)$ to denote the value:

$$w_{\mathcal{T}}(e) = \sum_{l \in \mathcal{T}(e)} s(l) \cdot w_l.$$

For any edge e in the tree, value $TC(e)$ is equal to:

$$TC(e) = \sum_{l \in \mathrm{Ch}(e)} TC(l) + (s(e) - s(l)) \cdot w_{\mathcal{T}}(l) + s(l) \cdot (w_{\mathcal{T}}(e) - w_{\mathcal{T}}(l)). \tag{20}$$

To compute the standard deviation of the CD, we perform two post-order traversals of the tree $\mathcal{T}$. In the first traversal, we compute value $w_{\mathcal{T}}(e)$ for each edge e based on the corresponding values of the edges in $\mathrm{Ch}(e)$. In the second traversal, we use the values $w_{\mathcal{T}}(\cdot)$ that we just calculated to compute $TC(e)$ for every edge in the tree. Then, using these values we can evaluate the expression in Eq (20). It takes $O(n)$ operations to traverse the tree twice and, given values $TC(\mathcal{T})$ and $\mathcal{T}(e)$ for every edge in $\mathcal{T}$, it takes $O(n)$ arithmetic operations to calculate the expression in Eq (20). Hence, we can compute the standard deviation of the CD with $O(n)$ operations in total.

Using the algorithms that we describe above, we can compute the value of the CD for two samples A and B, but also the expectation and the deviation of this measure in $O(n)$ time in the worst case. Therefore, by combining these algorithms, we can derive an algorithm that computes NRI_β in $O(n)$ time in the worst case.

Algorithms for computing the CBL and CLI

Computing the value of CBL for a given pair of samples. Recall that, for two tip samples A and B, the value of the CBL is equal to the sum of the weights of all edges e such that e belongs to both subtrees $\mathcal{T}(A)$ and $\mathcal{T}(B)$. Let R be a sample that consists of $|R| = r$ tips in $\mathcal{T}$. If edge e belongs to subtree $\mathcal{T}(R)$, then it holds that $0 < s_R(e) < r$. Hence, the weight of an edge e is counted in the value $CBL(\mathcal{T}, A, B)$ if $0 < s_A(e) < a$ and $0 < s_B(e) < b$. Therefore, to decide for every edge $e \in \mathcal{T}$ whether to include w_e in the CBL value, we simply have to compute values $s_A(e)$ and $s_B(e)$. In the algorithm that computes the value of CD, we showed how we can compute values $s_A(e)$ and $s_B(e)$ with $O(n)$ operations. Thus, the total number of required operations for computing the value of CBL is also $O(n)$.

Computing the expectation of the CBL. Theorem 3 provides an analytical expression for the expectation of the CBL. This expression consists of a sum with $O(n)$ terms, and can be evaluated with $O(n)$ arithmetic operations, given that we have already computed value $s(e)$ for each edge e in the tree. When describing an algorithm that calculates the expectation of the CD, we showed how we can compute these values with $O(n)$ operations. Therefore, we can compute the expectation of the CBL in $O(n)$ time.

Computing the standard deviation of the CBL. The standard deviation of the CBL can be calculated using the expression that we provide in Theorem 4. This expression contains a sum of $O(n^2)$ terms. Evaluating this sum directly takes $O(n^2)$ operations, which can be inefficient in practice.

In a previous paper, we presented a technique for computing similar expressions [13]. We can use this technique for evaluating the sum in Eq (9). As we presented in our previous work, this technique is quite efficient when $\mathcal{T}$ is relatively balanced. More precisely, the worst case time complexity of this algorithm is $O(\mathrm{SI}(\mathcal{T}))$, where $\mathrm{SI}(\mathcal{T})$ denotes the Sackin's Index of a tree $\mathcal{T}$. The Sackin's Index of $\mathcal{T}$ is equal to the sum of the depths of all tips in $\mathcal{T}$. The depth of a tip v is equal to the number of edges that appear on the simple path between v and the root of the tree. For a perfectly balanced tree that consists of n nodes, the depth of each tip is $O(\log (n))$. In this case, the value of of the Sackin's Index is equal to $O(n \log n)$. However, if the tree is skewed, there may exist many tips with depth close to n and therefore the value of the Sackin's Index is $O(n^2)$. Hence, in the worst case, the algorithm that we consider for computing the standard deviation of the CBL takes quadratic time with respect to the size of the input tree. In practice, phylogenetic trees are relatively balanced, and therefore the proposed algorithm is quite efficient. In the next section, we provide evidence for this argument; there we present experiments that we conducted using our algorithms on large tress.

Results and Applications

Experimental Evaluation

We implemented all the algorithms that we present in this paper, and we measured their performance in practice. Our implementations are developed in C++, and are publicly available through the software package `PhyloMeasures` [24]. This package provides functions for computing the value and the standardized indices of several phylogenetic biodiversity measures, and it is available both as an R package and a C++ library.

For the experiments that we conducted, we used a large phylogenetic tree from which we extracted several subtrees of several sizes. The tree that we used was constructed by Goloboff et. al [25]. This is the largest evolutionary tree of eukaryotic organisms that has been so far constructed from molecular and morphological data. It consists of 71,181 tips and 83,751 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.

From the `eukaryotes` we extracted fourteen trees, each tree having $5{,}000k + 1181$ tips with $k \in \{1, 2, \ldots, 14\}$. These subtrees were produced by successively pruning chunks of 5,000 leaves from the `eukaryotes` tree. We represent the set of these trees by EK. Therefore, for any two trees $\mathcal{T}, \mathcal{T}' \in$ EK we have that either $\mathcal{T}$ is a subtree of $\mathcal{T}'$, or vice versa. For each tree $\mathcal{T} \in$ EK we produced one hundred samples of tips with sizes $s(\mathcal{T})/k$ with k ranging from one to a hundred. We denote this set of samples by samples $(\mathcal{T})$. For every $\mathcal{T} \in$ EK we executed the algorithms we implented for computing the values and the standardized indices of the CD and the CBL. For each algorithm and for each tree $\mathcal{T} \in$ EK, we measured the execution time for processing all samples in samples $(\mathcal{T})$. The results of these experiments are illustrated in Fig 1. The experiments were performed using the R version of the `PhyloMeasures` package on a computer with an Intel core i5-2430M processor. This is a four-core CPU with 2.40GHz per core. The main memory of this computer is 7.8 Gigabytes. Our implementations run on a Linux Ubuntu operating system, release 12.04. The experiments were executed using R version 3.1.2 (Pumpkin Helmet).

All of the examined implementations run very fast even for the largest trees in EK. For the complete `eukaryotes` tree that consists of 71,181 tips, the algorithm that computes the value of the CD takes 3.01 seconds to process one hundred samples, the algorithm that computes the CBL value takes 1.61 seconds, and the algorithms that compute the standardized indices of the CD and the CBL take 3.12 and 54.52 seconds respectively. Note that these are the

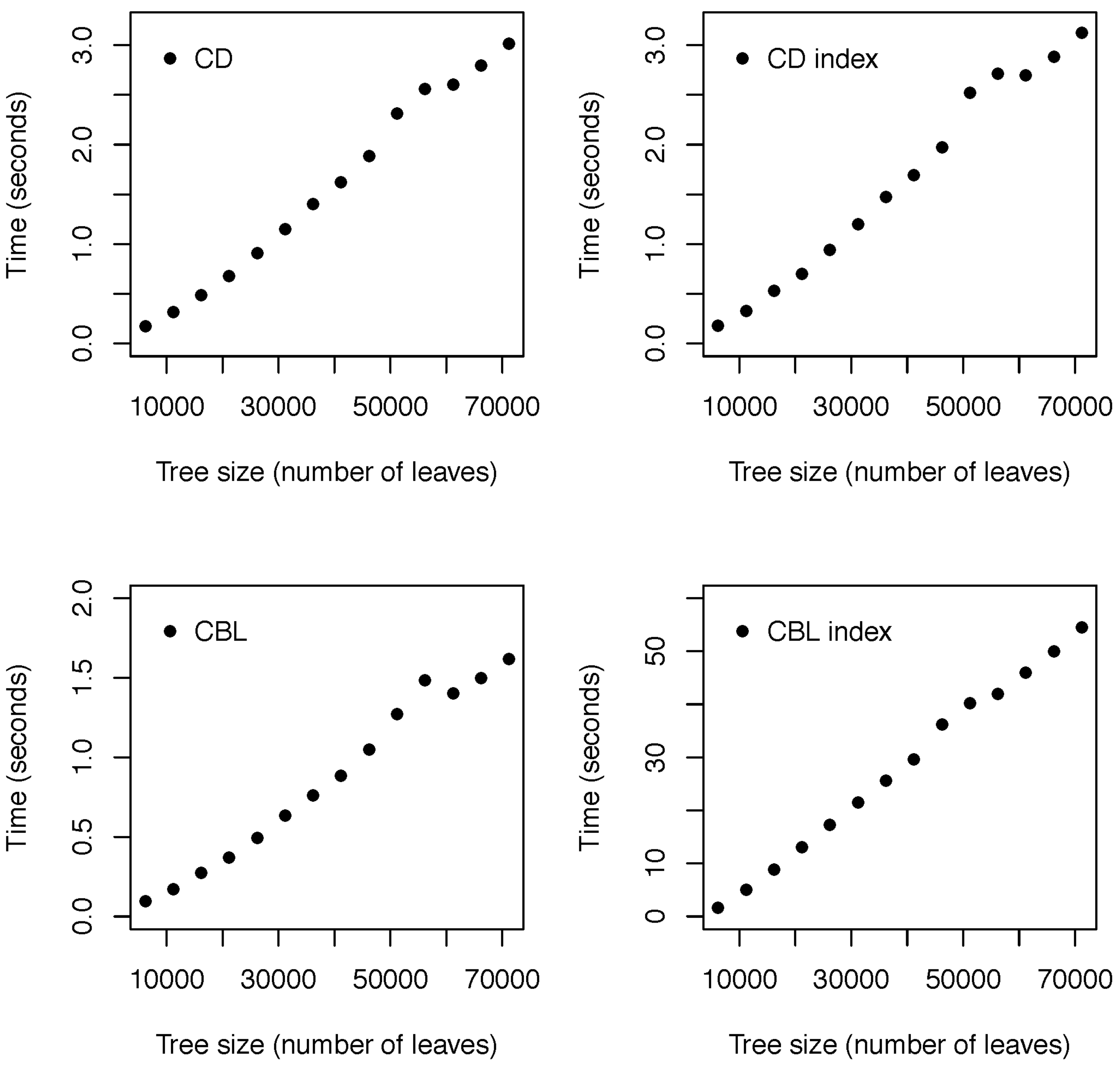

Fig 1. Running times of implemented algorithms computing values and standardized indices for CD and CBL. For each implementation and for each tree size, the figures illustrate the time that it takes for the function to process a set of one hundred samples.

doi:10.1371/journal.pone.0151167.g001

running times that each program takes for computing the results for one hundred samples of species. The program that computes the index of the CBL runs slower than the other programs, yet we see that it is still quite efficient. It seems that, for the datasets that we used, its execution time does not scale as a quadratic function of the tree size.

Applications

To illustrate an application of the algorithms, we used their `PhyloMeasures` implementation to calculate CBL and standardized CBL for mammals globally. We used range maps from

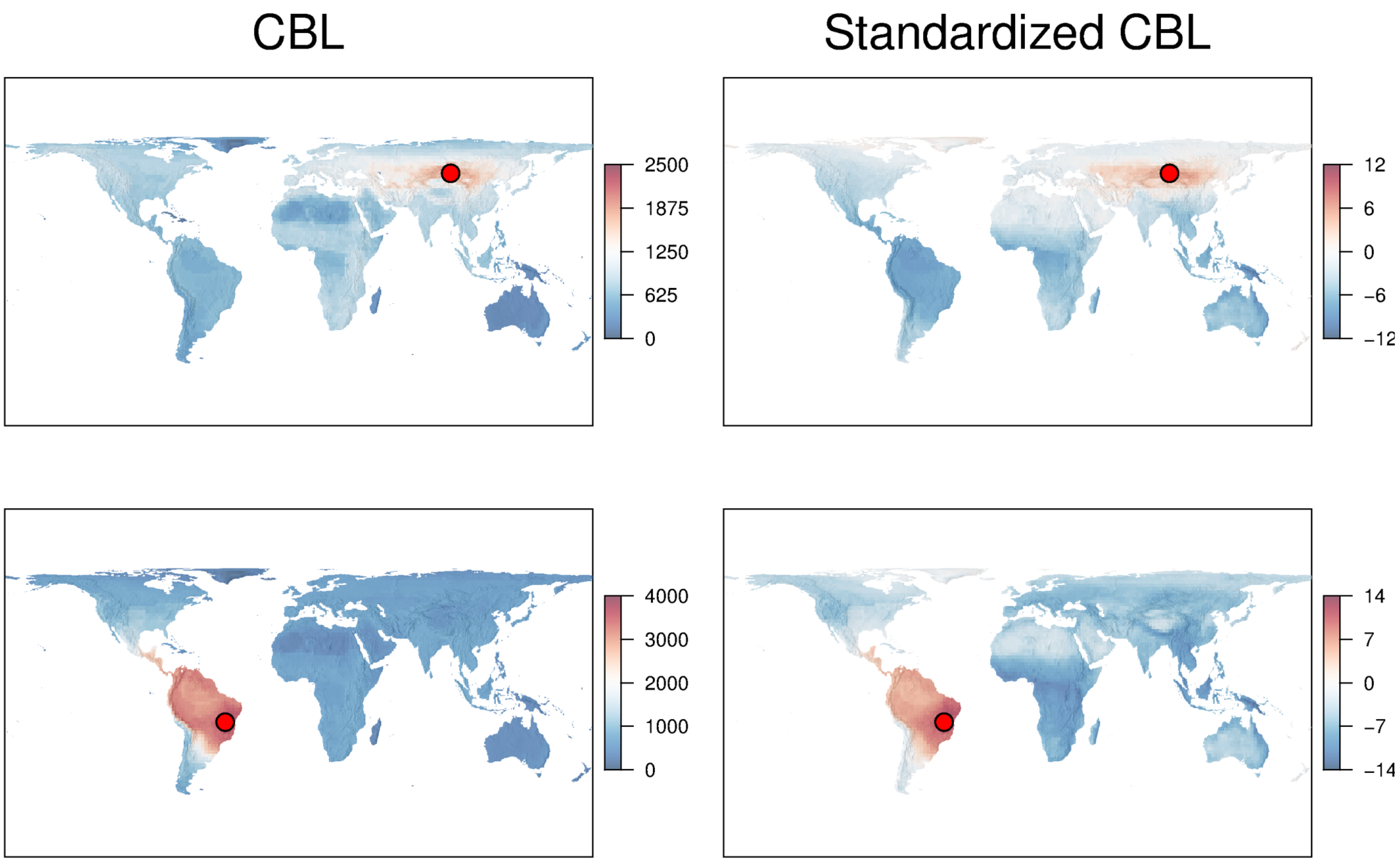

Fig 2. Maps of phylogenetic similarity of mammal assemblages of focal cells (red dot), compared to all other locations in the world. Similarity was calculated as Common Branch Length (CBL), or its richness-standardized version, for two different focal cells.

doi:10.1371/journal.pone.0151167.g002

the IUCN and rasterized them on a Behrmann equal area grid with a resolution of 193km (at 30N or S), corresponding roughly to two degrees. Each of the resulting grids consists of 71 × 180 cells, which means 12780 cells in total for each grid.

We combined these grids with the phylogeny of Bininda-Emonds et al [26]; this is a tree that portrays the phylogenetic relations between all mammal species. The number of leaf nodes in this tree is 4,510. Using this tree, for each grid that we created we then calculated CBL and standardized CBL between a focal cell and all other cells in the world. The results are spatial maps of the phylogenetic similarity between the focal location and all other sites. The constructed spatial maps are illustrated in Fig 2. In Central Asia, for example, there is evidence for a broad longitudinal band of high similarity, with more rapid turnover along the latitudinal gradient, while phylogenetic similarity is fairly high throughout northern South America, but declines rapidly towards the south of the continent.

Such maps can provide a detailed picture of how phylogenetic similarity changes among species assemblages over geographic space. However, constructing high resolution maps of this kind is practically infeasible without efficient algorithms that compute beta-diversity values, and especially the standardised version of these values. Using our implementations, we computed the exact standardized and non-stanndardized CBL values for these maps in one and a half minutes in total. On the other hand, it would take several days to execute these computations with previously existing software, which would also provide these values only approximately.

Conclusion

Phylogenetic beta diversity metrics are widely and increasingly used in ecology, but are often quite slow to compute. In addition, the relationship between species richness of the samples and the beta diversity metrics is often ignored. These problems are related, as computing the dependence of a metric on species richness is even more computationally intensive using traditional approaches, than is computing a single beta diversity value. Here, we propose solutions to these problems, providing 1) efficient algorithms for computing phylogenetic beta diversity metrics, and 2) algorithms to efficiently and exactly calculate their moments, allowing a simple standardization for species richness. We expect that these algorithms will significantly ease the computational burdens of researchers and lead to wider adoption of phylogenetic beta diversity metrics.

Acknowledgments

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

Author Contributions

Conceived and designed the experiments: CT BS. Performed the experiments: CT. Analyzed the data: CT BS. Contributed reagents/materials/analysis tools: CT BS. Wrote the paper: CT BS.

References

1. Whittaker RH. Vegetation of the Siskiyou mountains, Oregon and California. Ecological Monographs. 1960; 30: 279–338. doi: 10.2307/1948435
2. Cody ML. Towards a theory of continental species diversities: bird distributions over mediterranean habitat gradients. In: Cody ML, Diamond JM, editors. Ecology and evolution of communities. Harvard University Press; 1975. pp. 214–257.
3. Meynard CN, Devictor V, Mouillot D, Thuiller W, Jiguet F, Mouquet N. Beyond taxonomic diversity patterns: how do α, β and γ components of bird functional and phylogenetic diversity respond to environmental gradients across France? Global Ecology and Biogeography. 2011; 20:893–903. doi: 10.1111/j.1466-8238.2010.00647.x
4. Swenson NG, Erickson DL, Mi X, Bourg NA, Forero-Montaña J, Ge X, Howe R, Lake JK, Liu X, Ma K, Pei N, Thompson J, Uriarte M, Wolf A, Wright SJ, Ye W, Zhang J, Zimmerman JK, Kress WJ. Phylogenetic and functional alpha and beta diversity in temperate and tropical tree communities. Ecology. 2012; 93:S112–S125. doi: 10.1890/11-1180.1
5. Graham CH, Fine PVA. Phylogenetic beta diversity: linking ecological and evolutionary processes across space and time. Ecology Letters. 2008; 11: 1265–1277. doi: 10.1111/j.1461-0248.2008.01256.x PMID: 19046358
6. Holt BG, Lessard JP, Borregaard MK, Fritz SA, Araújo MB, Dimitrov D, Fabre PH, Graham CH, Graves GR, Jønsson KA, Nogués-Bravo D, Wang Z, Whittaker RJ, Fjeldså J, Rahbek C. An update of Wallace's zoogeographic regions of the world. Science. 2013; 339, no. 6115: 74–78. doi: 10.1126/science.1228282 PMID: 23258408
7. Qian H, Swenson NG, Zhang J. Phylogenetic beta diversity of angiosperms in North America. Global Ecology and Biogeography. 2013; 22:1152–1161. doi: 10.1111/geb.12076
8. Peixoto FP, Braga PHP, Cianciaruso MV, Diniz-Filho JAF, Brito D. Global patterns of phylogenetic beta diversity components in bats. Journal of Biogeography. 2014; 41:762–772. doi: 10.1111/jbi.12241

9. Swenson NG. Phylogenetic beta diversity metrics, trait evolution and inferring the functional beta diversity of communities. PLoS ONE. 2011; 6: e21264. doi: 10.1371/journal.pone.0021264 PMID: 21731685

10. Steel M. Tools to construct and study big trees: a mathematical perspective. In: Hodkinson T, Parnell J, Waldren S, editors. Reconstructing the tree of life: taxonomy and systematics of species rich taxa. CRC Press; 2007. pp. 97–112.

11. O'Dwyer JP, Kembel SW, Green JL. Phylogenetic diversity theory sheds light on the structure of microbial communities. PLoS Computational Biology. 2012; 8: e1002832. doi: 10.1371/journal.pcbi.1002832 PMID: 23284280

12. Tsirogiannis C, Sandel B, Cheliotis D. Efficient computation of popular phylogenetic tree measures. Lecture Notes on Computer Science; 2012; 7534: 30–43. doi: 10.1007/978-3-642-33122-0_3

13. Tsirogiannis C, Sandel B, Kalvisa A. New algorithms for computing phylogenetic biodiversity. Algorithms in Bioinformatics. 2014; LNCS 8701:187–203.

14. Nipperess DA, Matsen FA IV. The mean and variance of phylogenetic diversity under rarefaction. Methods in Ecology and Evolution. 2013; 4: 566–572. doi: 10.1111/2041-210X.12042 PMID: 23833701

15. Chao A, Chiu CH, Hsieh TC, Davis T, Nipperess DA, Faith DP. Rarefaction and extrapolation of phylogenetic diversity. Methods in Ecology and Evolution. 2015; 6: 380–388. doi: 10.1111/2041-210X.12247

16. Webb CO, Ackerly DD, McPeek MA, Donoghue MJ. Phylogenies and community ecology. Annual Review of Ecology and Systematics. 2002; 33: 475–505. doi: 10.1146/annurev.ecolsys.33.010802.150448

17. Kembel SW, Cowan PD, Helmus MR, Cornwell WK, Morlon H, Ackerly DD, Blomberg SP, Webb CO. Picante: R tools for integrating phylogenies and ecology. Bioinformatics. 2010; 26:1463–1464. doi: 10.1093/bioinformatics/btq166 PMID: 20395285

18. Graham CH, Parra JL, Rahbek C, McGuire JA. Phylogenetic structure in tropical hummingbird communities. Proceedings of the National Academy of Science. 2009; 106:19673–19678. doi: 10.1073/pnas.0901649106

19. Leprieur F, Albouy C, De Bortoli J, Cowman PF, Bellwood DR, Mouillot D. Correction: Quantifying phylogenetic beta diversity: distinguishing between 'true' turnover of lineages and phylogenetic diversity gradients. PLoS ONE. 2012; 7(10):. doi: 10.1371/journal.pone.0042760 PMID: 22912736

20. Pellissier L, Ndiribe C, Dubuis A, Pradervand JN, Salamin N, Guisan A, Rasmann S. Turnover of plant lineages shapes herbivore phylogenetic beta diversity along ecological gradients. Ecology letters 2013; 16(5), 600–608. doi: 10.1111/ele.12083 PMID: 23448096

21. Feng G, Mi X, Eiserhardt WL, Jin G, Sang W, Lu Z, Wang X, Li X, Li B, Sun I, Ma K, Svenning J-C. Assembly of forest communites across East Asia—insights from phylogenetic community structure and species pool scaling. Scientific Reports. 2015; 5: 9337. doi: 10.1038/srep09337 PMID: 25797420

22. Ricotta C, Bacaro G, Pavoine S. A cautionary note on some phylogenetic dissimilarity measures. Journal of Plant Ecology. 2015; 8(1):12–16. doi: 10.1093/jpe/rtu008

23. Eiserhardt WL, Svenning J-C, Borchsenius F, Kristiansen T, Balslev H. Separating environmental and geographical determinants of phylogenetic community structure in Amazonian palms (Arecaceae). Botanical Journal of the Linnean Society. 2013; 171: 244–259. doi: 10.1111/j.1095-8339.2012.01276.x

24. Tsirogiannis C, Sandel B. PhyloMeasures: a package for computing phylogenetic biodiversity measures and their statistical moments. Ecography. 2015;. doi: 10.1111/ecog.01814

25. Goloboff PA, Catalano SA, Mirandeb JM, Szumika CA, Ariasa JS, Kallersjoc M, Farris JS. Phylogenetic analysis of 73060 taxa corroborates major eukaryotic groups. Cladistics. 2009; 25:211–230. doi: 10.1111/j.1096-0031.2009.00255.x

26. Bininda-Emonds ORP, Cardillo M, Jones KE, MacPhee RDE, Beck RMD, Grenyer R, Price SA, Vos RA, Gittleman JL, Purvis A. The delayed rise of present-day mammals. Nature. 2007; 446:507–512. doi: 10.1038/nature05634 PMID: 17392779