Supplementary Material for

Tree balance in phylogenetic models

Sophie J. Kersting, Kristina Wicke, and Mareike Fischer

Philosophical Transactions of the Royal Society B: Biological Sciences (submitted)

This is the supplementary material of the manuscript "Tree balance in phylogenetic models" [45]. Here we give extensive information on the tree balance indices and tree models used in the analyses. Additionally, we present further results.

1 Preliminaries

Before we can present our results, we introduce some general definitions and notation, where we mainly follow the terminology of [18]. Throughout this manuscript, X denotes a non-empty finite set, which is often referred to as a *taxon set* or a *species set*. If not stated otherwise, we may assume $X = \{1, 2, \dots, n\}$. Furthermore, we use $\mathcal{I}(x)$ to denote the indicator function.

1.1 Rooted (binary) phylogenetic trees and related concepts

Rooted binary phylogenetic trees. A *rooted phylogenetic X-tree* $\mathcal{T}$ is a tuple $\mathcal{T} = (T, \phi)$, where $T = (V(T), E(T))$ is a rooted tree with root ρ and leaf set $V_L(T)$ such that ϕ is bijection from the leaf set $V_L(T)$ to X. If such a rooted phylogenetic tree has the additional property that each vertex $v \in V(T) \setminus V_L(T)$ has out-degree 2, it is called *binary*. In particular, if $|X| = n$, we also have $|V_L(T)| = n$. Thus, unless stated otherwise, we subsequently assume that n denotes the number of leaves of a phylogenetic tree $\mathcal{T}$ and/or its topology T. T is called a *rooted (binary) tree* and is also often referred to as the *topology* or *tree shape* of $\mathcal{T}$. In the following, unless stated otherwise, we will only consider rooted *binary* (phylogenetic) trees. We use $\mathring{V}(T)$ to denote the set of inner vertices of T (i.e., $\mathring{V}(T) = V(T) \setminus V_L(T)$). Edges incident with leaves are called *pendant* edges, while all other edges are called *inner* edges. If $n = 1$, T consists only of the root ρ and for technical reasons, ρ is in this case at the same time defined to be the root and the only leaf in T. For every $n \in \mathbb{N}_{\geq 1}$, we denote by $\mathcal{BT}^*_n$ the set of (isomorphism classes of) all rooted binary trees with n leaves and by $\mathcal{BT}_n$ the set of (isomorphism classes of) all rooted binary phylogenetic X-trees with n leaves. We have $|\mathcal{BT}_n| = (2n-3)!! = (2n-3) \cdot (2n-5) \cdot \ldots \cdot 1$ and $|\mathcal{BT}^*_n|$ is given by the *Wedderburn-Etherington numbers* $we(n)$ (sequence A001190 in the OEIS [71]), where

$$we(n) = \sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} we(i) \cdot we(n-i) + \frac{1}{2} \cdot we\left(\frac{n}{2}\right) \cdot \left(we\left(\frac{n}{2}\right) + 1\right)$$

with the initial cases $we(1) = we(2) = 1$ and $we(x) = 0$ if $x \notin \mathbb{N}_{\geq 1}$.

Remark 1.1. *The leaves of a tree (sometimes also called "tips") usually represent species or lineages (depending on the context), so in the following we sometimes simplify explanations by saying, e.g., that "a leaf speciates" or "goes extinct", which actually means that the species or lineage represented by that particular leaf undergoes said evolutionary event.*

Ancestors, descendants, and cherries. Whenever there exists a path from a vertex u to a vertex v in T, we call u an *ancestor* of v and v a *descendant* of u. Note that each vertex is at the same time its own ancestor and descendant. If there is an edge $(u, v) \in E(T)$, we also say that u is the *parent* of v and v is a *child* of u. We use $anc(v)$ to denote the set of ancestors of a vertex v excluding v and $child(v)$ to denote the set of children of v. Two leaves $x, y \in V_L(T)$ are called a *cherry*, denoted by $[x, y]$, if they have the same parent. The *lowest common ancestor* $LCA_T(u, v)$ of two vertices $u, v \in V(T)$ is the unique common ancestor of u and v that is a descendant of every other common ancestor of them.

Depth, height, width, and cophenetic values. The *depth* $\delta_T(v)$ of a vertex $v \in V(T)$ is the number of edges on the path from ρ to v, and the height $h(T)$ of T is the maximum depth of any leaf of T, i.e.,

$h(T) = \max\limits_{x \in V_L(T)} \delta_T(x)$. Given two leaves, $x, y \in V_L(T)$, their *cophenetic value* $\varphi_T(x,y)$ is the depth of their lowest common ancestor, i.e., $\varphi_T(x,y) = \delta_T(LCA_T(x,y))$.

The *width* $w_T(i)$ of T at depth i is the number of vertices $v \in V(T)$ with $\delta_T(v) = i$. Given a leaf $x \in V_L(T)$, we denote by $p_x(T)$ (or p_x for short) the probability of reaching x when starting at the root and assuming equiprobable branching at each inner vertex, more precisely $p_x(T) = \prod\limits_{v \in anc(x)} \frac{1}{|child(v)|}$.

Pending subtrees and standard decomposition. Given a rooted binary tree T and an arbitrary vertex $v \in V(T)$, we denote by T_v the pending subtree of T rooted in v and use n_v to denote the number of leaves in T_v, which is also often referred to as *clade size* of v. The *size* of a tree is the clade size of its root. Note that any rooted binary tree T with $n \geq 2$ leaves can be decomposed into its maximal pending subtrees, T_1 and T_2 say, rooted in the children of ρ, and we refer to this as the *standard decomposition of* T, denoted as $T = (T_1, T_2)$.

Symmetry nodes, balance values, and I_v values. Let $v \in \mathring{V}(T)$ be an inner vertex of T with children v_1 and v_2. If not stated otherwise, we assume $n_{v_1} \geq n_{v_2}$. We call v a *symmetry node* or *symmetry vertex* if T_{v_1} and T_{v_2} are isomorphic in the graph theoretical sense, i.e., if there is a bijective map $f : V(T_{v_1}) \to V(T_{v_2})$ which preserves the edges. The number of symmetry nodes of T is denoted by $s(T)$. Moreover, the *balance value* $bal_T(v)$ is defined as $bal_T(v) = |n_{v_1} - n_{v_2}|$. If $bal_T(v) \leq 1$, we call v *balanced*, and if $bal_T(v) = 0$, we call v *perfectly balanced*. If v is such that $n_v \geq 4$, its I_v *value* [24] is defined as $I_v = \frac{n_{v_1} - \lceil \frac{n_v}{2} \rceil}{n_v - 1 - \lceil \frac{n_v}{2} \rceil}$, and its I'_v *value* is defined as $I'_v = \begin{cases} I_v & \text{if } n_v \text{ is odd;} \\ \frac{n_v - 1}{n_v} \cdot I_v & \text{else.} \end{cases}$

Special rooted binary trees and restriction of trees. We now introduce some special trees. For $n \geq 2$, the *caterpillar (or comb) tree*, denoted as T_n^{cat}, is the unique rooted binary tree with n leaves that has precisely one cherry. For technical reasons, the tree consisting only of a single vertex is also considered to be a caterpillar (comb) tree, namely T_1^{cat}. The *fully balanced tree of height* h with $h \in \mathbb{N}_{\geq 0}$, denoted as T_h^{fb}, is the unique rooted binary tree with $n = 2^h$ leaves in which all leaves have depth exactly h. The *maximally balanced tree* [53], denoted as T_n^{mb}, is the unique rooted binary tree with n leaves in which all inner vertices are balanced. Finally, the *greedy from the bottom (GFB) tree*, denoted as T_n^{gfb}, is the unique rooted binary tree with n leaves obtained by greedily clustering trees of minimal size, starting with n single vertices and proceeding until only one tree is left, as described in [13, Algorithm 2]. Finally, given a rooted binary tree T and a subset $Y \subseteq V_L(T)$, the *restriction* $T_{|Y}$ is the tree obtained from the minimal subtree of T connecting the elements in Y by suppressing all vertices with in-degree and out-degree one. If $|Y| = 4$, we call $T_{|Y}$ a *rooted quartet displayed by* T and use $\mathcal{Q}(T)$ to denote the multiset of all rooted quartets displayed by T. Note that any *binary* rooted quartet is either a fully balanced tree of height two or a caterpillar tree on four leaves.

Furnas ranking scheme. Furnas [23] introduced the following (recursive) *left-light rooted ordering* $\prec$ for rooted binary trees, where for two rooted binary trees T' and T we have $T' \prec T$ if and only if

1. $|V_L(T')| < |V_L(T)|$, or
2. $|V_L(T')| = |V_L(T)|$ and $T'_L \prec T_L$, or
3. $|V_L(T')| = |V_L(T)|$ and $T'_L = T_L$ and $T'_R \prec T_R$,

where, provided that T has at least two leaves, T_L and T_R denote the two maximal pending subtrees of T with $T_L \preceq T_R$, i.e., $T_L = T_R$ or $T_L \prec T_R$, and, provided that T' has at least two leaves, T'_L and T'_R denote the two maximal pending subtrees of T' with $T'_L \preceq T'_R$.

For a fixed value of n, the *rank* $r_n(T)$ of a tree T in this ordering of all binary trees with n leaves is one more than the number of other binary trees, T', before T in the ordering.

2 Tree balance indices

A function $t : \mathcal{BT}_n^* \to \mathbb{R}$ is called a *(rooted) binary tree shape statistic (TSS)* if $t(T)$ depends only on the shape of T and not on the labeling of vertices or edge lengths. A binary tree shape statistic t is called a *balance index* if and only if i) the caterpillar tree T_n^{cat} is the unique tree minimizing t on $\mathcal{BT}_n^*$ for all $n \geq 1$ and ii) the fully balanced tree T_h^{fb} is the unique tree maximizing t on $\mathcal{BT}_n^*$ for all $n = 2^h$ with $h \in \mathbb{N}_{\geq 0}$. *Imbalance indices* are defined analogously with the extremal trees swapped.

In Tables 1 and 2 we list all tree (im)balance indices considered in this manuscript with a short description and their formal definitions. We refer the reader to the respective references and [18] for further details.

Table 1: List of tree balance indices (adapted from [18]). The first column contains the name of the index and a brief description. The second column contains its formal definition. Note that although the B_2 index is defined for arbitrary logarithm base it is common to use $\log_2$ when working with binary trees.

Balance indices	
Index	Definition
***B*₁ index** [70] sum of the reciprocals of the heights of the subtrees of T rooted at inner vertices of T (except for ρ)	$B_1(T) \coloneqq \sum\limits_{v \in \mathring{V}(T) \setminus \{\rho\}} h(T_v)^{-1}$
***B*₂ index** [1, 33, 48, 70] Shannon-Wiener information function (measures the equitability of arriving at the leaves of T when starting at the root and assuming equiprobable branching at each inner vertex)	$B_2(T) \coloneqq - \sum\limits_{x \in V_L(T)} p_x \cdot \log(p_x)$
Furnas rank [23, 48] rank of T according to Furnas' left-light rooted ordering on $\mathcal{BT}_n^*$	$F(T) \coloneqq r_n(T)$
Maximum width [9] maximum width at any depth	$mW(T) \coloneqq \max\limits_{i=0,\ldots,h(T)} w_T(i)$
Maximum width over maximum depth [9] ratio of maximum width and maximum depth	$W/D(T) \coloneqq \frac{mW(T)}{mD(T)}$
Modified maximum difference in widths [9, 18] maximum difference in widths of two consecutive depths	$mdelW(T) \coloneqq \max\limits_{i=0,\ldots,h(T)-1} w_T(i+1) - w_T(i)$
Rooted quartet index for binary trees [12] balance index based on the symmetry of the displayed quartets of T	$rQI(T) \coloneqq \lvert\{Q \in \mathcal{Q}(T) : Q \text{ has shape } T_2^{fb}\}\rvert$
stairs2 [9, 57] ratio between the leaf numbers of the smaller and larger pending subtree over all inner vertices	$st2(T) \coloneqq \frac{1}{n-1} \cdot \sum\limits_{v \in \mathring{V}(T)} \frac{\min\{n_{v_1}, n_{v_2}\}}{\max\{n_{v_1}, n_{v_2}\}}$

Table 2: List of tree imbalance indices (adapted from [18]). The first column contains the name of the index and a brief description. The second column contains its formal definition. Note that although the $\widehat{\boldsymbol{s}}$-shape statistic was originally not defined with a specific logarithm base it is common to use $\log_2$ when working with binary trees.

Imbalance indices	
Index	Definition
Average leaf depth [68, 70] mean depth of the leaves of T	$\overline{N}(T) \coloneqq \frac{1}{n} \cdot \sum_{x \in V_L(T)} \delta_T(x)$
Average vertex depth [37]; see also [6, 21, 36] mean depth of the vertices of T	$AVD(T) \coloneqq \frac{1}{\lvert V \rvert} \cdot \sum_{v \in V(T)} \delta_T(v)$
Colijn-Plazzotta rank [10, 67] recursively defined bijective mapping between rooted binary trees and the positive integers	$CP(T) \coloneqq 1$ for $T \in \mathcal{BT}_1^*$ and for $T = (T_1, T_2) \in \mathcal{BT}_n^*$ with $n \geq 2$, $CP(T) \coloneqq \frac{1}{2} CP(T_1)(CP(T_1) - 1) + CP(T_2) + 1$ (with $CP(T_1) \geq CP(T_2)$)
Colless index [11] sum of the balance values of the inner vertices of T	$C(T) \coloneqq \sum_{v \in \mathring{V}(T)} bal_T(v)$
Corrected Colless index [34] $C(T)$ divided by its maximum possible value	$I_C(T) \coloneqq \frac{2 \cdot C(T)}{(n-1)(n-2)}$
Equal weights Colless index [55] modification of $C(T)$ that weighs the imbalance values of all inner vertices of T equally	$I_2(T) \coloneqq \frac{1}{n-2} \cdot \sum_{v \in \mathring{V}(T), n_v > 2} \frac{bal_T(v)}{n_v - 2}$
Total $\boldsymbol{I}$ **and** $\boldsymbol{I'}$ **index** [24, 63] sum/total of the I_v (I'_v) values over all inner vertices v with $n_v \geq 4$	$\Sigma I(T) \coloneqq \sum_{v \in \mathring{V}(T), n_v \geq 4} I_v$ $\Sigma I'(T) \coloneqq \sum_{v \in \mathring{V}(T), n_v \geq 4} I'_v$
Maximum depth [9] maximum depth of any vertex of T; equivalent to $h(T)$	$mD(T) \coloneqq \max_{v \in V(T)} \delta_T(v) = h(T)$
Mean $\boldsymbol{I}$ **and** $\boldsymbol{I'}$ **index** [24, 63] mean of the I_v (I'_v) values over all inner vertices v with $n_v \geq 4$	$\overline{I}(T) \coloneqq \frac{1}{\lvert\{v \in \mathring{V}(T) : n_v \geq 4\}\rvert} \cdot \sum_{v \in \mathring{V}(T), n_v \geq 4} I_v$ $\overline{I'}(T) \coloneqq \frac{1}{\lvert\{v \in \mathring{V}(T) : n_v \geq 4\}\rvert} \cdot \sum_{v \in \mathring{V}(T), n_v \geq 4} I'_v$
Quadratic Colless index [3] sum of the squared balance values of the inner vertices of T	$QC(T) \coloneqq \sum_{v \in \mathring{V}(T)} bal_T(v)^2$
Rogers $\boldsymbol{J}$ **index** [65] number of inner vertices of T which are not perfectly balanced	$J(T) \coloneqq \sum_{v \in \mathring{V}(T)} (1 - \mathcal{I}(bal_T(v) = 0))$
$\widehat{\boldsymbol{s}}$**-shape statistic** [6] sum of $\log(n_v - 1)$ over all inner vertices of T	$\widehat{s}(T) \coloneqq \sum_{v \in \mathring{V}(T)} \log(n_v - 1)$
Sackin index [68, 70] sum of the leaf depths of T	$S(T) \coloneqq \sum_{x \in V_L(T)} \delta_T(x)$
stairs1 [9, 57] proportion of inner vertices that are not perfectly balanced; normalization of Rogers J index	$st1(T) \coloneqq \frac{1}{n-1} \cdot \sum_{v \in \mathring{V}(T)} (1 - \mathcal{I}(bal_T(v) = 0))$
Symmetry nodes index [44] number of inner vertices of T that are not symmetry nodes	$SNI(T) \coloneqq (n-1) - s(T)$
Total cophenetic index [53] sum of the cophenetic values of all distinct pairs of leaves of T	$\Phi(T) \coloneqq \sum_{\substack{\{x,y\} \in V_L(T)^2 \\ x \neq y}} \varphi_T(x, y)$
Total internal path length [49] total of the depths of all inner vertices of T	$TIP(T) \coloneqq \sum_{v \in \mathring{V}(T)} \delta_T(v)$
Total path length [15, 80, 81] total of the depths of all vertices of T	$TPL(T) \coloneqq \sum_{v \in V(T)} \delta_T(v)$
Variance of leaf depths [14, 68, 70] variance of the depths of the leaves of T	$\sigma_N^2(T) \coloneqq \frac{1}{n} \cdot \sum_{x \in V_L(T)} \left(\delta_T(x) - \overline{N}(T)\right)^2$

3 Probabilistic models of binary (phylogenetic) trees

A *probabilistic model of binary (phylogenetic) trees* P_n (short: (phylogenetic) tree model), with $n \geq 1$, is a family of probability mappings $P_n : \mathcal{BT}_n^* \to [0,1]$ or $P_n : \mathcal{BT}_n \to [0,1]$, respectively, associating a tree $T \in \mathcal{BT}_n^*$ or a phylogenetic tree $\mathcal{T} \in \mathcal{BT}_n$ with its probability under the model. For some models, explicit formulas for the probability of a particular (phylogenetic) tree are known, while for others this is not the case (yet). Many models implicitly describe the probability distribution by giving an algorithmic process that constructs a (phylogenetic) tree with n leaves under the corresponding model. In these cases, (phylogenetic) trees are generated with a fixed but not explicitly stated probability.

Any phylogenetic tree model induces a tree model by setting the probability $P_n(T)$ of a rooted binary tree $T \in \mathcal{BT}_n^*$ as the sum of all probabilities $P_n(\mathcal{T})$ over all $\mathcal{T} \in \mathcal{BT}_n$ that have T as their underlying topology. Since our aim is to evaluate tree balance indices that work on trees (and do not depend on leaf labels or edge lengths), we focus purely on the (induced) tree models.

For the remainder of this manuscript, we assume $n \geq 2$. In general, the phylogenetic tree models presented in this manuscript create a phylogenetic tree $\mathcal{T} \in \mathcal{BT}_n$ by generating a tree $T \in \mathcal{BT}_n^*$ with a certain probability and then assigning the leaf labels $1, \ldots, n$ uniformly at random to the leaves. Thus, the probability of a phylogenetic tree $\mathcal{T} \in \mathcal{BT}_n$ can be expressed using the probability $P_n(T)$ of its topology $T \in \mathcal{BT}_n^*$ under the model and the number of possible phylogenetic trees with that topology, $|\mathcal{T}_T| = \frac{n!}{2^{s(T)}}$, where $s(T)$ denotes the number of symmetry nodes (cf. [69, Corollary 2.4.3]), i.e., $P_n(\mathcal{T}) = P_n(T)/\,|\mathcal{T}_T|$.

The remainder of this section summarizes the most established tree models and categorizes them according to their approaches. For several of the models, we additionally refer the reader to Mooers et al. [54] for further information. In this manuscript, we additionally include information on the available R implementations in our package poweRbal, as well as on existing packages like ape [60], diversitree [20], geiger [61], TESS [38], TreeSim [76], or TreeSimGM [26].

Notice that next to listing several models from the literature, we additionally introduce a few new ones. In particular, the maximally balanced (MB) as well as the greedy from the bottom model (GFB), which both produce a type of most balanced trees, have been introduced by us as counterparts to the already established comb model [2], which produces the most imbalanced trees. Furthermore, we introduce the alternative birth-death model, which takes a new approach to extinct leaves: They are no longer removed, but remain as fossils in the tree (and continue to count towards the total number of leaves). This approach is also a possible modification of any tree model using extinction.

Additionally, we remark that some models appear more than once as they can be described in various ways. In particular, the famous Yule model arises as a special case of several tree models in different categories and thus deserves its own paragraph:

The *Yule model* [83], a pure birth model, is a central model in phylogenetics and has been used for many decades. It is known under a multitude of names (Yule model, equal-rates-Markov model (ERM), random branching model, Markovian dichotomous branching model, Yule-Harding model, Yule-Harding-Kingman model (YHK) or simply Markovian model) of which we will only use *Yule model* in the remainder of this manuscript. While it can be described under various approaches, the probabilities to obtain a specific (phylogenetic) tree can be explicitly stated (see, e.g., [79, Proposition 1] or [78, Proposition 3.2]):

$$P_{Y,n}(\mathcal{T}) = \frac{2^{n-1}}{n!} \cdot \prod_{v \in \mathring{V}(T)} \frac{1}{n_v - 1} \qquad \text{and} \qquad P_{Y,n}(T) = 2^{n-1-s(T)} \cdot \prod_{v \in \mathring{V}(T)} \frac{1}{n_v - 1}.$$

The Yule model is one of the oldest and most established tree models in mathematical phylogenetics, even though it is a very simple model. It can be described in various ways and thus occurs in different contexts. For instance, it can be regarded as a bipartitioning tree model (cf. Section 3.1.1) or as a discrete-time tree growing model acting at the leaves (cf. Section 3.2.1). However, it is also a special case of various more sophisticated models like the general birth-death model (cf. Section 3.2.2) or Ford's α-model (cf. Section 3.3). In our simulation studies shown in Section 5, we used the Yule model as the null model. However, our software package poweRbal allows users to choose different null models, too.

3.1 Tree models based on partitioning or clustering the leaves

The first class of models we consider produces trees by clustering the set of n leaves either by iteratively bi-partitioning the set of leaves (yielding the descendant sister clades n_1 and n_2 of an inner vertex) or by starting with single leaf clusters and iteratively merging two leaf clusters at a time.

3.1.1 Bipartitioning tree models

Bipartitioning tree models iteratively bipartition sets of m leaves into two subsets of i and $m-i$ leaves, respectively. Starting with the complete set of n leaves, they continue bipartitioning the resulting subsets in each step. Each bipartition thereby induces an inner node v (with $n_v = m$) with two pendant subtrees of size i and $m-i$. The models are, thus, defined by how evenly or unevenly they split the sets, i.e., by their probability distribution $q_m = (q_m(i); i = 1, ..., m-1)$ of splitting a given set of size m into two subsets of sizes i and $m-i$. Note that q_m has to be symmetric, i.e., $q_m(i) = q_m(m-i)$, as we only consider unordered trees, in which the two pendant subtrees of size i and $m-i$ are interchangeable. A list of these models can be found in Table 3.

Table 3: List of bipartitioning tree models. Since many models have multiple names, the name used throughout this manuscript is marked in bold.

Model (Source and R-function)	Probability $q_m(i)$ for $i \in \{1, 2, \ldots, m-1\}$	Description and examples
Yule model [29, 83] poweRbal::genYuleTree ape::rtree	$\frac{1}{m-1}$	Given m odd, then splitting it into sizes $1\vert(m-1)$, $2\vert(m-2)$,..., or $\lfloor \frac{m}{2} \rfloor \vert \lceil \frac{m}{2} \rceil$ is equally probable. If m is even, this also holds true, except for the case where $i = m-1 = \frac{m}{2}$, in which case the probability is half of that of the other possibilities. See [72, Equations (7) & (8)].
Aldous' β-splitting model, $\beta \geq -2$ [2] poweRbal::genAldousBeta Tree	$\frac{1}{a_m(\beta)} \frac{\Gamma(\beta+i+1)\Gamma(\beta+m-i+1)}{\Gamma(i+1)\Gamma(m-i+1)}$ with gamma function $\Gamma(t) = \int\limits_0^\infty x^{t-1}e^{-x}dt$	Balance increases with β. Here, $a_m(\beta)$ is a normalizing constant to ensure $\sum_{i=1}^{m-1} q_m(i) = 1$. Equals Yule for $\beta = 0$, PDA for $\beta = -3/2$, and comb for $\beta = -2$.
Comb model, Caterpillar model [2] poweRbal::genCombTree	$\begin{cases} 0.5 & \text{if } i = 1, \\ 0.5 & \text{if } i = m-1, \\ 0 & \text{else} \end{cases}$	Yields the caterpillar (comb) tree with n leaves, i.e., the most imbalanced tree.
Maximally balanced (**MB**) poweRbal::genMBTree	$\begin{cases} 1 & \text{if } m \text{ even and } i = \frac{m}{2}, \\ 0.5 & \text{if } m \text{ odd and } i = \lfloor \frac{m}{2} \rfloor, \\ 0.5 & \text{if } m \text{ odd and } i = \lceil \frac{m}{2} \rceil, \\ 0 & \text{else.} \end{cases}$	Yields the maximally balanced tree T_n^{mb} [53] with n leaves, a tree regarded as most balanced by several tree balance indices [18].

3.1.2 Coalescent or clustering-based tree models

We next consider tree models that iteratively merge clusters of leaves. In detail, for a desired number of leaves n, these models go backward in time by starting with a set of clusters $C_0 = \{\{1\}, \{2\}, ..., \{n\}\}$ and all n single labeled leaves (each represented by the cluster containing their label), and then incrementally merging two clusters at a time yielding the parent node of the nodes represented by the merged clusters. In detail, these models do the following for steps $k = 0, 1, ..., (n-2)$:

(1) Pick two clusters c_i and $c_j \in C_k$, one after the other, according to a probability distribution $q_{|C_k|} = (q_{|C_k|}(i); i = 1, ..., |C_k|)$ specifying the probability of picking the i^{th} cluster among all clusters in C_k.

(2) Create the new cluster set $C_{k+1} = (C_k \setminus \{c_i, c_j\}) \cup \{c_i \cup c_j\}$.

(3) Tree construction: Insert a new parent node $p_{i,j}$ that is represented by cluster $c_i \cup c_j$ and insert the edges to its children, i.e., the nodes represented by c_i and c_j.

In the last step, $k = n-2$, the two last remaining subsets are merged, yielding the root of the tree. A list of these models can be found in Table 4.

Remark 3.1. *Note that tree models do not only occur in phylogenetics. For instance, there are various models in population genetics which can be used to describe the genealogies of samples of species. Some of them can be thought of as "forwards models", as the Moran model or Galton-Watson branching processes, whereas others, as the Hey model, can be thought of as "backwards models" [54]. In particular, the Moran model describes a lineage splitting process within a population (similar to the Yule model describing speciation events), the Galton-Watson branching describes a process in which individuals/species reproduce according to a reproduction law [30], whereas the Hey model describes lineage coalescences within a population (similar to the coalescent process explained above). However, as the Moran model, the Galton-Watson branching processes (restricted to maximally two offspring to produce binary trees), and the Hey model are closely related to the Yule or general birth-death model and in fact produce topologies that are distributed identically to the Yule model [54], we omit any further details on these models in the present manuscript.*

Table 4: List of clustering tree models. Since many models have multiple names, the name used throughout this manuscript is marked in bold.

Model (Source and R-function)	Probability $q_{\lvert C_k \rvert}(i)$ for $i \in \{1, 2, \dots, \lvert C_k \rvert\}$	Description and examples
Kingman-**Coalescent** [47] ape::rcoal	$\frac{1}{\lvert C_k \rvert}$	Distribution on $\mathcal{BT}_n^*$ is the same as for Yule (see, e.g., [78, p. 45])
Comb model, Caterpillar model [2] poweRbal::genCombTree	$\begin{cases} 1 & \text{if } \lvert c_i \rvert = \max_{c_j \in C_k} \lvert c_j \rvert, \\ 0 & \text{else} \end{cases}$	Yields the caterpillar (comb) tree with n leaves, i.e., the most imbalanced tree.
Greedy from the bottom (**GFB**) poweRbal::genGFBTree	$\begin{cases} 1 & \text{if } \lvert c_i \rvert = \min_{c_j \in C_k} \lvert c_j \rvert, \\ 0 & \text{else} \end{cases}$	Yields the greedy from the bottom tree T_n^{gfb} [13] with n leaves, a tree regarded as most balanced by several tree balance indices [18].

3.2 Tree growing models

We next consider the class of tree growing models, which produce trees through a tree growing procedure. A naive approach to generate trees under such a tree growing model is to start with a single species (or pair of species), evolve a tree until n species are reached, and stop the procedure with the next event. We remark that this "simple sampling approach" (SSA) [31] has been shown to be problematic when the model includes processes such as extinction, in which case it may result in trees with systematically biased shapes and/or branch lengths. Hartmann et al. [31] introduced an alternative sampling approach (called "general sampling approach" (GSA)) to overcome these issues (see [31] for details; see also the R package TreeSim [76]). Nevertheless, we employ the simple sampling approach in this study, and note that it is an interesting question for future research to investigate if the power of different tree shape statistics to detect different evolutionary models is affected by this choice.

Within the class of tree growing models, several subclasses can be distinguished, which for instance differ in whether they only act at the leaves of a tree or everywhere, whether they act in discrete or continuous time, whether they include extinction or not, and whether they use constant speciation (resp. extinction) rates or rates that depend on additional parameters such as age. In the following, we briefly summarize the different subclasses, before giving several examples in the respective subsections.

Tree growing models acting at the leaves. The models presented in Sections 3.2.1 and 3.2.2 purely act at the current leaves and evolve the tree through speciation events, i.e., by turning a leaf into a cherry, and extinction events, i.e., by either letting a leaf never speciate again or removing a leaf and its pendant edge (suppressing the parent) from the tree altogether. Note that in the literature, extinct lineages usually are not counted as leaves even if they are kept in the tree (c.f. Table 6). Biologically, this is plausible as tree models are often used to generate trees with a specific number n of present-day species. However, as these lineages mathematically indeed do form leaves in the tree and thus may impact its balance, one might also want to count them towards the total number n of leaves. This motivated us to introduce the "alternative birth-death model" presented in Table 6, which treats all leaves equally, regardless of their extinction status.

Concerning the rates of speciation and extinction, in the simplest case, speciation (resp. extinction) rates are constant, but in most cases, they vary through time and/or depend on some trait such as age, body size, number of co-existing species, or the environment. For the latter, traits may change deterministically or randomly, and can be heritable or non-heritable. Non-heritable traits change in the same way in all species, whereas for a heritable trait, the initial value of the trait is correlated with the trait value of the mother species at speciation (cf., [50, 74]). Within the tree growing models acting at the leaves, we distinguish between discrete time (Section 3.2.1) and continuous time (Section 3.2.2) models. Discrete-time models evolve trees and modify speciation (resp. extinction) rates at discrete time steps or events. Continuous-time models, on the other hand, continuously modify speciation (resp. extinction) rates or directly vary waiting times until the next event using some probability distribution.

For rate-varying models, there is often a distinction between *symmetric* and *asymmetric speciation* (also known as one- or two-daughter-change [35]): Symmetric speciation means that after speciation two identical "child" species (of age 0) are created, while asymmetric speciation means that the "parent" species persists (and keeps its current age), and only one new "child" species (of age 0) is created. In terms of rates, given a parent with rate r_p, we get the rates (r_c, r_c) for symmetric, respectively (r_p, r_c) for asymmetric speciation. To keep everything concise, we refrain from listing both options for each model that is defined with both symmetric and asymmetric speciation. Instead, all models that allow the usage of both symmetric and asymmetric speciation are marked with $\circledast$ in Tables 5–9. We remark, however, that symmetric versus asymmetric speciation may lead to different macroevolutionary dynamics and may result in different distributions of phylogenetic trees. For an overview of some of the implications, see for instance the review by Stadler [75], where the tree growing models acting at the leaves are further divided into four classes (namely, *constant rate birth-death*, *species-exchangeable* (given a speciation (resp. extinction) event, each species is equally likely to be the one speciating (resp. going extinct)), *species-speciation-exchangeable* (given a speciation event, each species is equally likely to be the one speciating, and the discrete speciation-extinction pattern after a speciation event is independent of which species was the one speciating), and *species-non-exchangeable models* (models where given a speciation event, some species may be more likely to speciate than others, or models, where all species are equally likely to speciate but the subsequent speciation-extinction pattern depends on a particular species speciating). Whether speciation is symmetric or asymmetric and whether speciation rates only depend on time and on a non-heritable trait (e.g., age) or also on other factors like to the number of co-existing species also influences other mathematical properties of the model, e.g., whether it has a description as a so-called *coalescent point process* (for further details, see [50]).

Finally, there are also models that postulate that both child species diverge from their parent species in different ways, referred to as *unequal* or *biased* speciation (in terms of rates: r_{c_1}, r_{c_2}, and r_p are pairwise distinct).

Tree growing models acting at edges (and vertices). The models presented in Section 3.2.3 generate trees by (i) attaching new leaves to existing edges, i.e., subdividing an edge with a new parent node s and connecting it to a new leaf l with a new pendant edge (s, l), or (ii) by attaching new leaves to existing inner vertices. Note that attaching a leaf to a pendant edge is equivalent to turning the incident leaf into a cherry. However, Operation (ii), i.e., attaching leaves to existing inner vertices, allows for the generation of *non-binary* trees.

3.2.1 Discrete-time tree growing models acting at the leaves

In this section, we describe tree growing models acting at the leaves in discrete time more in-depth. These models act at the current leaves and modify their speciation (and extinction) rates. The process starts with a single leaf (the root) with speciation rate λ_0 (and extinction rate μ_0) and at each time step, i.e., after waiting for the next event, one of the currently existing leaves is chosen for a speciation (or extinction) event. A list of these models can be found in Table 5 (pure speciation models) and Table 6 (speciation and extinction models). In Table 5, we include concrete examples for the "rate distribution" across the leaves of small trees. For this purpose, we use $\Lambda = (\lambda_1, \ldots, \lambda_k)$ to denote the speciation rates of the current k leaves $\ell_1, \ldots, \ell_k$. We embed these rates in the Newick format (cf. [17]) to indicate the position of the respective leaves in the tree. The Newick format uses nested parentheses such that closely related taxa/leaves are grouped closely together. Moreover, a rooted binary tree has two comma-separated entries, corresponding to the two maximal pending subtrees of an inner vertex, inside each pair of parentheses. As an example, $((\lambda_1, \lambda_2), (\lambda_3, \lambda_4))$ describes a fully balanced and $(((\lambda_1, \lambda_2), \lambda_3), \lambda_4)$ a caterpillar tree with 4 leaves.

Remarks on the implementation of discrete-time tree growing models. Given a rate r, the waiting time until the next event is $\exp(r)$-distributed, with an expected waiting time of $\frac{1}{r}$. In other words, we expect r events per time unit. As long as the rates remain constant between events, we may simplify the tree generation process for the purpose of this manuscript, as we are not considering edge lengths (determined by the waiting times) but only in which leaf is picked for which next event. In this case, it is sufficient to focus on the proportions of the rates with respect to each other [16, p. 55]:

Let $r_1, \ldots, r_k$ be some rates, e.g., speciation or extinction rates. Then, the waiting time until the next (any) event is the minimum of the waiting times for the single events (each $\exp(r_i)$-distributed). This waiting time is also exponentially distributed with rate $\sum\limits_{1 \leq j \leq k} r_j$. The event is triggered by a particular r_i with probability $p_i = \frac{r_i}{\sum\limits_{1 \leq j \leq k} r_j}$. Note that for $c \in \mathbb{R}_{\geq 0}$, we have $\frac{r_i}{\sum\limits_{1 \leq j \leq k} r_j} = \frac{c\ r_i}{\sum\limits_{1 \leq j \leq k} c\ r_j}$, i.e., the probability p_i does not change if all rates are scaled by a factor c.

Table 5: List of *discrete-time tree growing models without extinction.* Since many models have multiple names, the name used throughout this manuscript is marked in bold. $\lambda_0 > 0$ denotes the initial rate of the starting node, λ_p the parent's rate. The Λ-examples use $\lambda_0 = 1$, the probabilities for $n = 4$ hold for arbitrary λ_0. For IF-diff and biased speciation, we assume that the leaf with the first rate in Λ_2 speciated.
Several models are *trait-inspired* or *-based*, i.e., a leaf's speciation rate (in-)directly depends on its quantitative trait x. The trait can be age, time, a (partly) inheritable trait (e.g., body size) affecting speciation, environment, number of co-existing species etc. $x_0 > 0$ is the initial trait value of the starting node, x_p the parent's value.

Model (Source and R-function)	Children $(\lambda_{c_1}, \lambda_{c_2})$	Other λ_i	Description and examples
Yule model [29, 83] poweRbal::genYuleTree ape::rtree TESS::tess.sim.taxa / TreeSim::sim.bd.taxa(..., mu=0,...)	(λ_0, λ_0)	λ_0	Pure birth process, constant and equal rates. $\Lambda_1 = (1),\ \Lambda_2 = (1,1),\ \Lambda_3 = ((1,1),1)$ $P_4\left(T_2^{fb}\right) = 1/3,\ P_4\left(T_4^{cat}\right) = 2/3$
Direct-children-only (**DCO**), $\zeta > 0$ [43, p. 6] ⊛ poweRbal::genGrowTree(..., "DCO_sym")	$(\zeta\lambda_0, \zeta\lambda_0)$	λ_0	Factor ζ has only short-time effect on children. Equals Yule for $\zeta = 1$. $\Lambda_1 = (1),\ \Lambda_2 = (\zeta, \zeta),\ \Lambda_3 = ((\zeta, \zeta), 1)$ $P_4\left(T_2^{fb}\right) = 1/(2\zeta + 1)),\ P_4\left(T_4^{cat}\right) = 2\zeta/(2\zeta + 1)$
Inherited fertility (**IF**), $\zeta > 0$ [43, p. 6] ⊛ poweRbal::genGrowTree(..., "IF_sym")	$(\zeta\lambda_p, \zeta\lambda_p)$	λ_i	Factor ζ has long-time and exponential effect. Equals Yule for $\zeta = 1$. $\Lambda_1 = (1),\ \Lambda_2 = (\zeta, \zeta),\ \Lambda_3 = ((\zeta^2, \zeta^2), \zeta)$ $P_4\left(T_2^{fb}\right) = 1/(2\zeta + 1)),\ P_4\left(T_4^{cat}\right) = 2\zeta/(2\zeta + 1)$
Unequal fert. inheritance (**IF-diff**), simple biased speciation, $\zeta \geq 1$ [48, p. 1177] poweRbal::genGrowTree(..., "IF-diff")	$\left(\frac{2\zeta}{\zeta+1}\lambda_p, \frac{2}{\zeta+1}\lambda_p\right)$	λ_i	Factor ζ has long-time and exponential effect (opposite effect on the two children). Proportion $\lambda_{c_2}/\lambda_{c_1} = 1/\zeta$ with $\lambda_{c_1} + \lambda_{c_2} = 2\lambda_p$. Equals Yule for $\zeta = 1$. $\Lambda_1 = (1),\ \Lambda_2 = \left(\frac{2\zeta}{\zeta+1}, \frac{2}{\zeta+1}\right),\ \Lambda_3 = \left(\left(\frac{4\zeta^2}{(\zeta+1)^2}, \frac{4\zeta}{(\zeta+1)^2}\right), \frac{2}{\zeta+1}\right)$ $P_4\left(T_2^{fb}\right) = \frac{3\zeta}{(2\zeta+1)(\zeta+2)},\ P_4\left(T_4^{cat}\right) = \frac{2(\zeta^2+\zeta+1)}{(2\zeta+1)(\zeta+2)}$
Biased speciation, $\zeta \in [0, 1]$, [5, p. 149] poweRbal::genGrowTree(..., "biased")	$(\zeta\lambda_p, (1 - \zeta)\lambda_p)$	λ_i	The parent's rate is divided between its two children according to the given ratio ζ. Note that $\zeta \in [0, 0.5]$ yields the same model. $\Lambda_1 = (1),\ \Lambda_2 = (\zeta, (1 - \zeta)),\ \Lambda_3 = ((\zeta^2, \zeta - \zeta^2), (1 - \zeta))$ $P_4\left(T_2^{fb}\right) = -2\zeta^2 + 2\zeta,\ P_4\left(T_4^{cat}\right) = 2\zeta^2 - 2\zeta + 1$
Trait-inspired models			
Age-step-based fertility (**ASB**), $\zeta > 0$ [43, p. 6] ⊛ poweRbal::genGrowTree(..., "ASB")	(λ_0, λ_0)	$\zeta\lambda_i$	Exponential increase ($\zeta > 1$) or decrease of rates ($\zeta < 1$) with age (in time steps) of lineage. Equals Yule for $\zeta = 1$. $\Lambda_1 = (1),\ \Lambda_2 = (1, 1),\ \Lambda_3 = ((1, 1), \zeta)$
Simple Brownian, $\sigma \geq 0$, [43, p. 6] ⊛ poweRbal::genGrowTree(..., "simpleBrown_sym")	$(\lambda_p + \varepsilon_1,$ $\lambda_p + \varepsilon_2)$	λ_i	With $\varepsilon_{1,2} \sim N(0, \sigma^2)$ normally distributed with standard deviation σ. Rates < 0 are set to a small value > 0.

Trait-based models	Children $(\lambda_{c_1}, \lambda_{c_2})$	Other λ_i	Children (x_{c_1}, x_{c_2})	Other x_i	Description
Punctuated(-intermittent) **linear-Brownian**, (Speciational) Brownian evolution model, trait-based, $\sigma_x, \sigma_\lambda \geq 0$, [35, p. 2142] ⊛ poweRbal::genGrowTree(..., "lin-Brown_sym")	$\left(10^{\log(x_{c_1})+\varepsilon_3},\right.$ $\left.10^{\log(x_{c_2})+\varepsilon_4}\right)$	λ_i	$(x_p + \varepsilon_1,$ $x_p + \varepsilon_2)$	x_i	Inherited trait influences speciation rate. With $\varepsilon_{1,2} \sim N(0, \sigma_x^2)$ and $\varepsilon_{3,4} \sim N(0, \sigma_\lambda^2)$. Trait values < 0 are set to a small value > 0. Note that $\log = \log_{10}$. There is also a **bounded** version ("lin-Brown-bounded_sym") of this model with an upper limit for the trait values, e.g., $x_i \in (0, 20)$ symmetrical around $x_0 = 10$ ($x_0 = 100$ in [1]). Equals Yule for $\sigma_\lambda = \sigma_x = 0$.
Punctuated(-intermittent) **log-Brownian**, $\sigma_x, \sigma_\lambda \geq 0$, [35, p. 2142] ⊛ poweRbal::genGrowTree(..., "log-Brown_sym")	$\left(10^{\log(x_{c_1})+\varepsilon_3},\right.$ $\left.10^{\log(x_{c_2})+\varepsilon_4}\right)$	λ_i	$\left(10^{\log(x_p)+\varepsilon_1},\right.$ $\left.10^{\log(x_p)+\varepsilon_2}\right)$	x_i	Inherited trait (with proportional changes) influences speciation rate. With $\varepsilon_{1,2} \sim N(0, \sigma_x^2)$ and $\varepsilon_{3,4} \sim N(0, \sigma_\lambda^2)$. Note that $\log = \log_{10}$. Equals Yule for $\sigma_\lambda = \sigma_x = 0$.

Table 6: List of *discrete-time tree growing models with extinction.* Since many models have multiple names, the name used throughout this manuscript is marked in bold. λ_0 and μ_0 denote the initial speciation and extinction rates of the starting node, λ_p and μ_p the parent's respective rates, and λ_e and μ_e the rates of the leaf chosen for an extinction event. For the trait-based models, transition rates between states i and j are denoted as τ_{ij}. Note that in all models including extinction, the algorithm might not reach the desired number n of leaves and end prematurely in case all lineages go extinct.

Model (Source and R-function)	Children $(\lambda_{c_1}, \lambda_{c_2})$ (μ_{c_1}, μ_{c_2})	Other λ_i μ_i	Extinct λ_e μ_e	Description and examples
(Simple/constant-rate) **birth-death** model, $\lambda_0 > 0$, $\mu_0 \geq 0$ [64, 83] ape::rphylo TESS::tess.sim.taxa TreeSim::sim.bd.taxa	(λ_0, λ_0) (μ_0, μ_0)	λ_0 μ_0	- -	"Reconstructed process": Birth-death process, where extinct taxa/leaves are removed (and degree-2 vertices are suppressed). Constant and equal speciation and extinction rates. Equals Yule for $\mu_0 = 0$. Distribution on $\mathcal{BT}_n^*$ is the same as for Yule even for $\mu_0 > 0$ [54, p. 5].
Alternative birth-death model, $\lambda_0 > 0$, $\mu_0 \geq 0$ poweRbal::genAltBirthDeathTree	(λ_0, λ_0) (μ_0, μ_0)	λ_i μ_i	0 0	Extinct taxa/leaves remain in the tree as "fossils" with speciation and extinction rate 0. Note that, as opposed to other models employing fossils in which fossils are ignored and only extant species are counted, the fossils in this model count towards the total number n of leaves. Equals Yule for $\mu_0 = 0$.
Trait-based models *(Trait = Age, time, (partly) inheritable trait (e.g., body size) affecting speciation, environment, number of co-existing species etc.)*				
Density-dependent model [32] poweRbal::genDensityTree	(λ_0, λ_0) (μ_N, μ_N)	λ_0 μ_N	- -	With $\mu_N = \begin{cases} \frac{\lambda_0 N}{N^*} & \text{if } N < N^* \\ \lambda_0 & \text{if } N \geq N^*. \end{cases}$, where N denotes the number of currently existing lineages, and N^* is an equilibrium number of lineages. In other words, the speciation rate is constant, but the extinction rate increases linearly with the number of co-existing lineages until the equilibrium N^* is reached at which point $\mu = \lambda_0$. Extinct taxa are removed (reconstructed process).
SSE models state-dependent speciation and extinction models (see, e.g., [19, 20, 22, 25, 52, 82])				Family of models extending birth-death models to account for speciation, extinction, and trait evolution. Each species is associated with a specific trait/state (observed or hidden), and can transition to another state, or speciate (resp. go extinct) with rates depending on its current state.
↳ **BiSSE** [52] diversitree::tree.bisse poweRbal::genBiSSETree				Each species has a binary trait associated with it, and is either in state A or in state B. In any infinitely small time interval $(t, t+dt)$, a species in state A speciates with rate λ_A (resulting in two new species in state A), goes extinct with rate μ_A, or transitions to state B with rate τ_{AB}. Similarly, a species in state B speciates with rate λ_B, goes extinct with rate μ_B, and transitions to state A with rate τ_{AB}. Extinct species are removed. Several SSE models are implemented in the R package diversitree [20]. $\dagger \xleftarrow{\mu_A} \underset{\lambda_A}{\underset{\circlearrowleft}{A}} \underset{\tau_{BA}}{\overset{\tau_{AB}}{\rightleftarrows}} \underset{\lambda_B}{\underset{\circlearrowleft}{B}} \xrightarrow{\mu_B} \dagger$
↳ **Multi-rate (MR) model** [62] ⊛				Given a finite state space S, and a single leaf at time $t = 0$, this model proceeds as follows: In any infinitely small time interval $(t, t+dt)$, any species which was in a state i at time $t \geq 0$, (i) transitions to some other state j with probability $\tau_{ij} dt$, (ii) remains in state i but gives birth to a new species in some state j with probability $\lambda_{ij} dt$, or (iii) does not undergo any change (i.e., remains in state i and does not give birth) with probability $1 - \eta_i$. Here, $\eta_i := \tau_{iS} + \lambda_{iS}$ and $\tau_{iA} := \sum_{j \in A} \tau_{ij}$ and $\lambda_{iA} := \sum_{j \in A} \lambda_{ij}$ for any $A \subseteq S$. We remark that we placed this model under SSE models due to its similarities with the former, even though it is not usually listed as an SSE model in the literature. It may also be interpreted as a continuous-time model. ↳ Is a modification of a multi-type branching process with at most two offspring of one particle at a time; includes as special cases the Yule, PDA, and comb models (for further details, see [62]).

3.2.2 Continuous-time tree growing models acting at the leaves

In this section, we consider continuous-time tree growing models acting at the leaves. These models either modify the leaves' speciation (and extinction) rates, like their discrete-time counterparts (see Section 3.2.1) but in a continuous fashion, or they vary waiting times until events using some probability distribution. In particular, the models in Tables 7 and 8 vary speciation and extinction rates based on time (e.g., based on a species' age) or based on traits. The models in Table 9 on the other hand vary waiting times directly. These models are also known as *Bellman-Harris* (B-H) models [4] and do not use rates and their natural exponentially distributed waiting times like, e.g., the Yule model, but an arbitrary waiting time distribution. However, similar to the idea of Yule, the waiting times of all leaves follow the same distribution and extinction does not occur. The Yule models is thus a special case of the Bellman-Harris models. As a generalization of the B-H models, the birth-death models, and the Yule model, there are the Bellman-Harris models with extinction (B-H-ex). A B-H-ex model consists of an arbitrary waiting time distribution for speciation and another for extinction. For simulation purposes, we remark that the R package TreeSimGM by [26] is devoted to simulating phylogenetic trees under B-H and B-H-ex models (with an additional option of lineage-specific rate changes). As an example, TreeSimGM::sim.taxa(numbsim=1, n=10, waitsp="rweibull(0.5,2)", waitext="rexp(1)") would generate one tree on 10 taxa under a birth-death model with waiting times until speciation following a Weibull(0.5, 2) distribution and waiting times until extinction following an Exponential(1) distribution.

Remarks on the implementation of continuous-time tree growing models: Discrete-time approximation. The continuous process can be simulated with a discrete-time approximation. For this, we update the rates (and other additional trait values) every small time unit δ. Even though we do not have to consider waiting times in other parts of the present manuscript (as they only affect the edge lengths in tree models based on time steps), they are relevant here as they determine how drastically, i.e., after how many time units δ and corresponding updates of the rates, the rates and their relative proportions to each other have changed prior to the current speciation event.

In detail, the procedure looks as follows. Let f be a function that changes some rate (note that we might have several functions changing different rates and/or trait values and the principle described below applies to all of them). Initially, there is only the rate of the initial single node, but at later times, independent of where we are in the procedure, we do the following:

Let t_e be the time of the last event with $d_e \cdot \delta \leq t_e < (d_e+1) \cdot \delta$ for some $d_e \in \mathbb{N}_{\geq 0}$. Now, let there be k rates $\lambda_1, ..., \lambda_k$, e.g., the speciation rates of k species at some point in time $t \geq t_e$ with $(d_e+d) \cdot \delta \leq t < (d_e+d+1) \cdot \delta$ with $d \in \mathbb{N}_{\geq 0}$ after the last event.

We draw a waiting time t_w until the next event at random (it is $\exp\Big(\sum\limits_{1 \leq j \leq k} \lambda_j\Big)$-distributed).

If t_w is larger than the accumulated waiting time $(d_e + d + 1) \cdot \delta - t_e$ until the next point in time $(d_e + d + 1) \cdot \delta$, we update the rates. This results in k updated rates $f(\lambda_1), ..., f(\lambda_k)$ at time $(d_e + d + 1) \cdot \delta$, and we repeat the process.

If $t_w \leq (d_e + d + 1) \cdot \delta - t_e$, an event happens at time

$$t'_e = \begin{cases} t_e + ((d_e + d) \cdot \delta) & \text{if } t_w \leq (d_e + d) \cdot \delta - t_e \\ t_e + t_w & \text{if } (d_e + d) \cdot \delta - t_e < t_w \leq (d_e + d + 1) \cdot \delta - t_e \end{cases}$$

(which event happens is determined by the proportions of the rates). Next, we incorporate the event, for example, a speciation event resulting in rates $\lambda'_1, ..., \lambda'_{k+1}$, before updating the rates. In the example of a speciation event, this would result in $k + 1$ rates $f(\lambda'_1), ..., f(\lambda'_k)$ at time $t = t'_e$ with $(d_e + d) \cdot \delta \leq t'_e < (d_e + d + 1) \cdot \delta$, and we repeat the process.

As an example, Agapow and Purvis [1] implemented such a discrete approximation of a continuous-time model by reassessing the species' ages and rates at least every $\delta = 0.001$ time units and additionally at every speciation event.

Table 7: List of *age-based continuous-time tree growing models based on modifying speciation and extinction rates*. Let a_i be the time since the last speciation of a lineage i, and let $f(\cdot)$ and $g(\cdot)$ denote age-dependent speciation (resp. extinction) rates. Since many models have multiple names, the name used throughout this manuscript is marked in bold. Note that some of the models below are also considered as functions of overall time (rather than species' age) in the literature. In particular, the implementations of these models in TESS and TreeSim depend on time, whereas the implementation in TreeSimGM incorporates the species' age instead.

Model	Living	Extinct	Description and examples
(Source and R-function)	λ_i	λ_e	
	μ_i	μ_e	
General birth-death model [41, 42, 56] ⊛ TESS::tess.sim.taxa TreeSimGM::sim.taxa	$f(a_i)$ $g(a_i)$	0/- 0/-	Varying rates (functions of age). For models with $g \not\equiv 0$, extinct leaves are commonly removed from the tree (rates deleted "-"). A possible modification is to let them remain in the tree as fossils (rates set to 0, but still counting towards the total number of leaves in the tree; see also alternative birth-death model in Table 6). Equals Yule if $f \equiv \lambda_0$ constant and $g \equiv 0$.
↳ Age-based (decreasing) [1, p. 867] ⊛ TreeSimGM::sim.taxa			$f(a_i) = \frac{A}{\sqrt{a_i}} + 0.5$ and $g \equiv 0$ Speciation rate decreasing with lineage's age a_i. $A = 0.3$ for weaker and $A = 0.6$ for stronger dependency on age. Remark: The initial rate for new species remains unclear for this choice of function f as $f(a_i) \xrightarrow[a_i \to 0]{} \infty$ [1, p. 867].
↳ Logistic [58] ⊛ TreeSimGM::sim.taxa TESS::tess.sim.taxa			$f(a_i) = \frac{1}{1+e^{-(\beta_f \cdot a_i + \alpha_f)}}$ and $g(a_i) = \frac{1}{1+e^{-(\beta_g \cdot a_i + \alpha_g)}}$, where $\alpha_f, \alpha_g, \beta_f$, and β_g are parameters. If $=\beta_f$ (resp. β_g) is strictly positive, the corresponding rate is an increasing function with respect to age. If it is strictly negative, it is a decreasing function and if it is is zero, $f(a_i)$ and $g(a_i)$ are constant. Note that we phrased this model in terms of a species age a_i, i.e., time since last speciation, whereas the parameter in [58] is time since the process was started (where the root is associated with time $t = 0$).
↳ **Break-point**, phase-shift, or episodic birth-death model [28, 58, 77] ⊛ TESS::tess.sim.taxa TreeSim::sim.rateshift.taxa TreeSimGM::sim.taxa			$f(a_i) = \begin{cases} \lambda_1 & \text{if } a_i \leq a^* \\ \lambda_2 & \text{if } a_i > a^*. \end{cases}$ Here, a^* is some threshold age, or break point, at which the speciation rate changes (as a generalization, several break points may be incorporated; this is also called an episodic birth-death process, see, e.g., [28, 77]). While [58] did not specify a model for $g(a_i)$, a second break-point model or any other model can be used to model extinction. Note that we state this model in terms of time since the last speciation of a species, but it can also be stated in terms of time since the process started [58].
↳ Explosive radiation [79, p.106] ⊛ TreeSimGM::sim.taxa			$f(a_i) = \begin{cases} \lambda > 0 & \text{if } a_i \leq a^* \\ 0 & \text{else} \end{cases}$ and $g \equiv 0$ Here, a^* is some threshold age, and unless a species has undergone speciation within a time interval of length a^*, it will never do so. Yields the PDA model for $a^* < 1/n$ ([79]).
↳ Asymmetric refractory period[51] TreeSimGM::sim.taxa(..., symmetric=F,...)			$f(a_i) = \begin{cases} 0 & \text{if } a_i \leq a^* \text{and } i \text{ subject to refraction} \\ \lambda > 0 & \text{else} \end{cases}$ and $g \equiv 0$ All lineages have the same speciation rate $\lambda > 0$, except that following speciation, *one* daughter lineage has $\lambda = 0$ during a refractory period of length a^*.
↳ Symmetric refractory period [7] TreeSimGM::sim.taxa(..., symmetric=T,...)			$f(a_i) = \begin{cases} \frac{\lambda}{a^*} \cdot a_i & \text{if } a_i \leq a^* \\ \lambda > 0 & \text{else} \end{cases}$ and $g \equiv 0$ After speciation, *both* daughter lineages undergo a refractory period of length a^*, where their speciation rate increases linearly from zero to $\lambda > 0$, the common constant speciation rate of all lineages.
Mass extinction [77] ⊛ TESS::tess.sym.taxa(..., massExtinctionTimes=c(), massExtinctionSurvival Probabilities=c(),...) TreeSim::sim.rateshift.taxa			Any choice for $f(a_i)$ and $g(a_i)$ (often a break-point model is used, i.e., speciation and extinction rates are constant within a time interval but may shift between time intervals) is paired with events of mass extinctions: at a time point t, a proportion $0 \leq p_t \leq 1$ of the currently existing species survives. The surviving set is chosen uniformly at random.

Table 8: List of *trait-based continuous-time tree growing models (without extinction) based on modifying speciation rates.* Here, $x(i,t)$ is a trait value assigned to species i at time t and $f(\cdot)$ denotes the trait-dependent speciation rate.

Model (Source and R-function)	Description and examples
Gradual linear- and log-Brownian, $\sigma_x, \sigma_\lambda > 0$ [35, p. 2143] ⊛	$x(i,t) = x(i,t-\delta) + \varepsilon_x$ (linear-Brownian) or $\log(x(i,t)) = \log\big(x(i,t-\delta)\big) + \varepsilon_x$ (log-Brownian) and $f(x(i,t)) = 10^{\log(x(i,t))+\varepsilon_\lambda}$. Here, $\delta > 0$ is an infinitely small time step, and $\varepsilon_x \sim N(0,\sigma_x^2)$ and $\varepsilon_\lambda \sim N(0,\sigma_s^2)$ are normally distributed random variables. All trait values $x(i,t)$ and speciation rates $f(\cdot)$ are updated at every time step. Note that $\log = \log_{10}$.
Punctuated-continuous linear- and log-Brownian, $\sigma_x, \sigma_\lambda > 0$ [35, p. 2143] ⊛	$x(i,t) = x(i,t-\delta) + \varepsilon_x$ (linear-Brownian) or $\log(x(i,t)) = \log\big(x(i,t-\delta)\big) + \varepsilon_x$ (log-Brownian) and $f(x(i,t)) = 10^{\log(x(i,t))+\varepsilon_\lambda}$. Here, $\delta > 0$ is an infinitely small time step, and $\varepsilon_x \sim N(0,\sigma_x^2)$ and $\varepsilon_\lambda \sim N(0,\sigma_s^2)$ are normally distributed random variables. Only the trait values $x(i,t)$ of the emerging daughter species are updated at their corresponding speciation events. All speciation rates $f(\cdot)$ are updated at every time step. Note that $\log = \log_{10}$.

Table 9: List of *continuous-time tree growing models based on modifying waiting times.* Since many models have multiple names, the name used throughout this manuscript is marked in bold.

Model (Source and R-function)	Waiting time distribution for speciation [extinction] event at leaf i	Description and examples
Uniform model, proportional-to-distinguishable-arrangements (or -types) model (**PDA**) [66] poweRbal::genPDATree ape::rtopology(..., rooted=T)	Weibull-distributed with shape parameter tending to zero [–]	B-H model that simulates explosive radiation (species can only speciate for a short time after their emergence) [26, p. 755] (using the result of [79, Theorem 6]).
Weibull [27, 39] ⊛ TreeSimGM::sim.taxa	Weibull-distributed [Weibull-distributed] (varying parameters)	Hagen et al. [27] observed a strong signal for the shape parameter of the speciation Weibull distribution to be strictly between 0 and 1 for empirical trees, indicating a decreasing speciation rate with increasing species age. Tree shapes were found to be largely invariant under the parameters of the extinction Weibull distribution, but branch lengths were influenced by the latter. See also [39] for a related study.
Phase-type [73] ⊛ TreeSimGM::sim.taxa	Coxian phase-type (PH) distribution for cont. time Markov chains (CTMC)	PH distributions describe the time to absorption in a CTMC with a single absorbing state (here: speciation or extinction) and a finite number of non-absorbing states (phases). Informally, a species passes through different phases during its lifetime where it can either undergo speciation (extinction) or move to the next phase with certain rates.

3.2.3 Discrete-time tree growing models acting at edges and vertices

The models presented in this section generate trees by either attaching new leaves to existing edges, i.e., subdividing an edge with a new parent node s and connecting it to a new leaf l with a new pendant edge (s,l) (Ford's α-model and PDA model), or by attaching new leaves to existing edges and existing inner vertices (Chen-Ford-Winkel's $\alpha-\gamma$-model). Note that attaching a leaf to a pendant edge is equivalent to turning the incident leaf into a cherry. However, attaching leaves to existing inner vertices, allows for the generation of *non-binary* trees. Furthr details of these models can be found in Table 10.

Table 10: List of further tree growing models that do not use rates. Since many models have multiple names, the name used throughout this manuscript is marked in bold.

Model (Source and R-function)	Description
Ford's α-model, $\alpha \in [0,1]$ [21, 40] poweRbal::genFordsAlphaTree	Start with a cherry and add an additional temporary root edge (ρ_{temp}, ρ). Then, in each step – let m be the current leaf number – choose any edge, where each of the $m-1$ inner edges has probability $p = \frac{\alpha}{m-\alpha}$ and each of the m pendant edges has probability $p = \frac{1-\alpha}{m-\alpha}$, subdivide it, and add a new pendant edge with a leaf in each step. Do this until reaching n leaves. Last, delete the edge incident to ρ_{temp}. Equals Yule for $\alpha = 0$, PDA for $\alpha = 1/2$, comb for $\alpha = 1$.
Chen-Ford-Winkel's α-γ-model [8]	Extension of Ford's α-model to generate non-binary trees: Start with a cherry and add an additional temporary root edge (ρ_{temp}, ρ). Then, in each step – let m be the current leaf number –choose any edge or an (inner) vertex of out-degree $k \geq 2$, where each of the m pendant edges has probability $\frac{1-\alpha}{m-\alpha}$, each other edge has probability $\frac{\gamma}{m-\alpha}$, and each inner vertex of out-degree $k \geq 2$ has probability $\frac{(k-1)\alpha-\gamma}{m-\alpha}$. If an edge is chosen, subdivide it, and add a new pendant edge with a leaf. If an inner vertex is chosen, add a new leaf by attaching a new pendant edge to this vertex. Do this until reaching n leaves. Last, delete the temporary root edge. Equals Ford's α-model for $\gamma = \alpha$.
Uniform model, proportional-to-distinguishable-arrangements (or -types) model (**PDA**) [66] poweRbal::genPDATree ape::rtopology(..., rooted=T)	Start with an unrooted tree with 3 leaves. Then, in each step, choose any edge, subdivide it, and add a new pendant edge with a leaf until reaching n leaves. Last, choose an edge to insert a root [59, p. 8].

3.3 Distribution-based tree models

In this section, we list models that are based on some simple distribution on the space of rooted binary (phylogenetic) trees. A list of these models can be found in Table 11.

Table 11: List of distribution-based tree models. Since many models have multiple names, the name used throughout this manuscript is marked in bold.

Model (Source and R-function)	Probabilities	Description
Uniform model, proportional-to-distinguishable-arrangements (or -types) model (**PDA**) [66] poweRbal::genPDATree ape::rtopology(..., rooted=T)	$P_n(T) = P_n(\mathcal{T}) \cdot \lvert\mathcal{T}_T\rvert$ $= \frac{n!}{2^{s(T)} \cdot (2n-3)!!}$	All phylogenetic trees $\mathcal{T} \in \mathcal{BT}_n$ are equiprobable, i.e., $P_n(\mathcal{T}) = \frac{1}{\lvert\mathcal{BT}_n\rvert} = \frac{1}{(2n-3)!!}$.
Uniform model, equiprobable-types-model (**ETM**) poweRbal::genETMTree ape::rtree(..., equiprob = T)	$P_n(T) = \frac{1}{\lvert\mathcal{BT}^*_n\rvert} = \frac{1}{we(n)}$	All rooted binary trees $T \in \mathcal{BT}^*_n$ are equiprobable.
Comb model, Caterpillar model [2] poweRbal::genCombTree	$P_n(T_n^{cat}) = 1$	Yields the caterpillar tree (comb tree) with n leaves, i.e., the most imbalanced tree.
Greedy from the bottom (**GFB**) poweRbal::genGFBTree	$P_n(T_n^{gfb}) = 1$	Yields the greedy from the bottom tree T_n^{gfb} [13] with n leaves, a tree regarded as most balanced by several tree balance indices [18].
Maximally balanced (**MB**) poweRbal::genMBTree	$P_n(T_n^{mb}) = 1$	Yields the maximally balanced tree T_n^{mb} [53] with n leaves, a tree regarded as most balanced by several tree balance indices [18].

4 Software

We now present our software package poweRbal, implemented in the free programming language R, which provides the readers with the means to do similar analyses as shown in the preceding section and adapt them to their needs. The package is published on CRAN. Using this package, the user can freely specify the set of tree shape statistics (all tree balance indices implemented in the package treebalance [18] are already included and can be used, but additional TSS can be added by the user as well), the null model, the set of alternative models (all tree models analyzed in this manuscript have already been included in the package and can be used, but additional models can be added by the user as well), the number of leaves n, and the method parameters like the sample sizes N_d and N_a or the level of significance.

Below, we give a short introduction on how to use the package using illustrative examples. First, the package is loaded with `library("poweRbal")`.

poweRbal provides the means to generate trees under various tree models. For example, if we want to generate a tree specifically under the PDA model, we can use `genPDATree(n)` (information on the function names of the individual tree models is given in the tables above). Another way to access any tree model and produce any number of trees is to use the wrapper function `genTrees(n, Ntrees, tm)`. With `n` we specify the number of leaves, with `Ntrees` the number of trees, and with `tm` the tree model either as a string/name if the tree model does not have or need a parameter or as a list of a string/name and parameter (see `?genTrees` for more information on how to access the individual models). For example, we can use the following commands to generate 5 trees with 10 leaves each, once under the Yule model (default: rate 1), once under the PDA model, and once under Aldous' β splitting model with $\beta = -1$:

```
genTrees(n = 10, Ntrees = 5, tm = "yule")
genTrees(n = 10, Ntrees = 5, tm = "pda")
genTrees(n = 10, Ntrees = 5, tm = list("aldous", -1))
```

To compare the power of pre-implemented TSS under pre-implemented tree models, we can use the function `powerCompare` (see `?powerComp` for detailed information). For this, we only have to specify the TSS, the alternative models, as well as the number of leaves. Optionally, the user can also specify the null model (by default the Yule model) and several other options that define the method (by default: two-tailed test with small-sample correction and a level of significance of 5%; distributions are computed exactly if possible and otherwise approximated through sampling $N_d = 10,000$ trees under the null model; $N_a = 1,000$ trees are sampled under each alternative model to determine the respective power values). For example, to use the Sackin and B_1 index (see `tssInfo` for information on how to access all available TSS) for testing the PDA model and the ETM against the Yule model, we specify:

```
set.seed(1)
pc1 <- powerComp(tss = c("Sackin", "B1I"), alt_models = list("pda", "etm"), n = 10)
pc1$power
# Yields the following matrix:
        pda     etm
Sackin  0.303   0.168
B1I     0.290   0.125
```

The output of the function `powerCompare` is of the class `poweRbal_data`, which consists of a list containing one mandatory element, `power`, and several optional elements:

- `power`: A numeric matrix containing the power values (one row per TSS and one column per alternative model).
- `accept_regions`: A numeric matrix containing information on the region of acceptance (one row per TSS and four columns).
- `CIradius`: A numeric matrix containing the confidence interval radii (one row per TSS and one column per alternative model).
- `actual_sample_sizes`: A numeric vector containing the actual sample sizes under each alternative model, as some models do not always successfully generate trees.
- `alt_model_params`: A numeric vector (one element per alternative model) containing the values of a tree model parameter. This is only suitable if the alternative models all belong to the same tree model family and differ only in one parameter.

- Other input data from the `powerComp` function, such as `tss`, `null_model`, `alt_models`, `n`, `distribs`, `N_null`, `N_alt`, `test_type`, `correction`, and `sig_lvl`.

The S3 methods `summary.poweRbal_data` and `plot.poweRbal_data` can be used to display a detailed overview of a `poweRbal_data` object and to visualize the power data.[1]

```
summary(pc1)
plot(pc1) # Creates a bar plot.
```

As another example, to use the Sackin and B_1 index for Aldous' β splitting model with $\beta = -1.5$, $\beta = -1$, and $\beta = -0.5$, we have:

```
pc2 <- powerComp(tss = c("Sackin", "B1I"), n = 10,
                 alt_models = list(list("aldous", -1.5), list("aldous", -1),
                                   list("aldous", -0.5)))
pc2$power
# Yields the following matrix:
          aldous, -1.5  aldous, -1  aldous, -0.5
Sackin    0.299         0.114       0.072
B1I       0.277         0.125       0.061
```

For power comparisons in which all alternative models belong to the same tree model family and differ only in one parameter, as in `pc2`, `plot.poweRbal_data` provides the option to define the parameter `alt_model_params` to depict the power results in a line plot instead. Access the help files with `?plot.poweRbal_data` for more information on customizing these plots.

```
plot(pc2, alt_model_params = c(-1.5, -1, - 0.5),
     ylim = c(0,0.4)) # Creates a line plot.
```

To compare new TSS (the example below from [43, p.53] combines the Sackin index and the total I' index into a new TSS) with the existing ones, we can create a list object containing the function and a short name.

```
library("treebalance")
new_tss_func <- function(tree){
  n <- length(tree$tip.label)
  return(sackinI(tree)+ 0.14*n*IbasedI(tree, method = "total",
                                       correction = "prime", logs = F))
}
new_tss <- list(func = new_tss_func, short = "SackTotI")
```

Similarly, we can add new tree models by creating a list object containing the function which produces `Ntrees`-many trees with n leaves. With that, it is also possible to use lists of trees, e.g., from a tree model that was implemented in a different programming language, as a basis for a null or alternative model. The `ape` package [60] provides means for importing common file formats like the Newick format. As an example for this, let `treeList` be a list of type `multiphylo` containing trees with the same number of leaves, say $n = 30$. Then we could proceed as follows.

```
set.seed(2)
treeList <- genTrees(n = 30, Ntrees = 1050, tm = "etm") # Exemplary tree list
new_tm_func <- function(n , Ntrees = 1){
    if (n == 30) {
        selection <- sample.int(length(treeList), Ntrees)
        return(treeList[selection])
    } else {
        stop("No such trees available.")
    }
}
new_tm <- list(func = new_tm_func)
```

Now, we can add the name of the list objects as follows. Here, the new tree model is used as an alternative model but it can be also used as the null model by setting `null_model = "new_tm"`.

```
pc3 <- powerComp(tss = c("Sackin", "new_tss", "B1I"),
                 alt_models = list("pda", "new_tm"), n = 30)
plot(pc3)
```

[1] The figures in the main text have been created similarly, but with the help of the `ggplot2` package for aesthetic reasons.

5 Power analysis of balance indices – comprehensive overview of the simulation results

This section contains all figures of the conducted simulation studies (see Sections 3 and 4 in the main document). Where relevant, we used an initial speciation rate of $\lambda_0 = 1$.

All power comparisons can be replicated (or also modified/extended) using our software package poweRbal (see Section 5 in the main document). Below we exemplarily provide the code concerning the symmetric punctuated(-intermittent) log-Brownian model. It can be directly executed after loading the package with `library("poweRbal")`. To replicate any of the other power comparisons, simply exchange the information about n or the tree model and its parameters (marked with `- EDIT -`). Use `?genTrees` for information on how to access the available tree models. The log-Brownian model shown below, for example, requires the user to specify two parameters σ_λ (here varying from 0 to 2) and σ_x (here set to 0.1).

```
# Specify the number of leaves and the sample sizes                       - EDIT -
selected_n <- 30L
selected_N_d <- 100000L
selected_N_a <- 1000L
```

```
# Specify the tree model and the parameter values.                        - EDIT -
selected_params <- c(0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2)
selected_tm <- lapply(X = selected_params,
                      FUN = function(X){list("log-Brown_sym", c(X, 0.1))})
```

```
# Choose all suitable TSS for trees with the given number of leaves
# (except ALD, AVD, TPL, TIP, stairs1, and corrColl because of equivalence).
selected_tss <- getAllTSS(n = selected_n)
selected_tss <- selected_tss[!selected_tss %in% c("ALD", "AVD", "TPL",
                                                  "TIP", "stairs1", "corrColl")]
```

```
# Do the power comparison.
set.seed(42)
power_data <- powerComp(tss = selected_tss, null_model = "yule",
                        alt_models = selected_tm,
                        n = selected_n, N_null = selected_N_d,
                        N_alt = selected_N_a)
```

For a better depiction of the results, the user may specify some further information, e.g., on the name of the tree model, the parameter, or the TSS.

```
# Additional information for the depiction of the name                    - EDIT -
# of the tree model and its parameter.
tm_name <- c(bquote(paste("Symmetric log-Brownian model with parameter ",
                          "\U1D70E", "  ", { }["\U03BB"],"   (", italic(n),
                          " = ", .(selected_n), ", ", italic(N[d]), " = ",
                          .(selected_N_d), ", ", italic(N[a]), " = ",
                          .(selected_N_a), ")")),
             bquote(paste("\U1D70E", "  ", { }["\U03BB"])),
             expression(sigma))
```

```
# Additional information for depiction of the TSS.
selected_tss_names <- getTSSnames(selected_tss)
selected_tss_names[selected_tss=="mWomD"] <- "Max. width over max. depth"
selected_tss_names[selected_tss=="modMaxDelW"] <- "Mod. max. diff. in widths"
selected_tss_names[selected_tss=="CollLike"] <- expression(paste("Colless-l. (",
     italic("f"),"(", italic("n"), ")=", italic("e")^italic("n"), ",
     ",italic("D"),"=var)"))
```

```
# Plot the results.
plot(power_data,
     tss_names = selected_tss_names, tss_colors = getTSScolors(selected_tss),
     sig_lvl = 0.05, legend_pos = "topleft",
     alt_model_params = selected_params, alt_model_family = tm_name,
     ylim = c(0,1), ylab = "Power (null model rejected)")
```

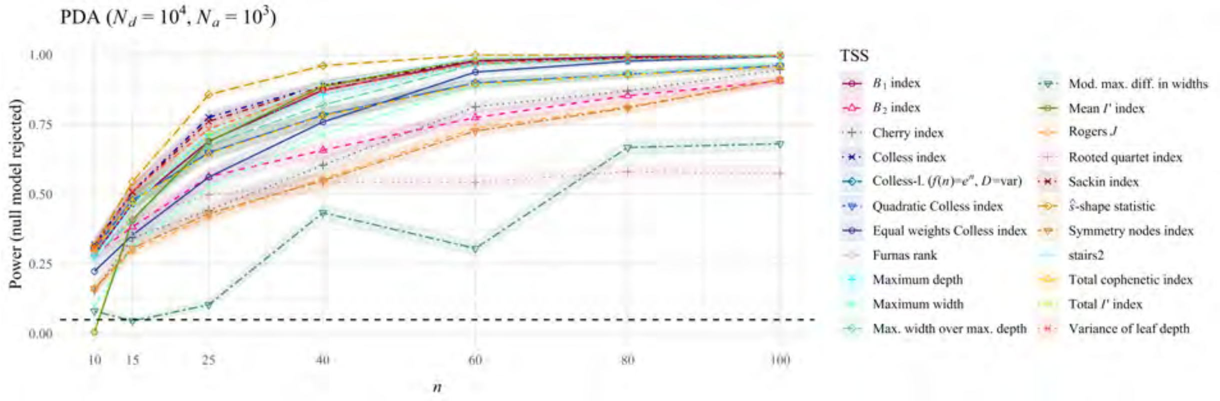


Figure 1: The power of all TSS to correctly identify trees generated under the PDA model as not having been generated under the Yule model as a function of n.

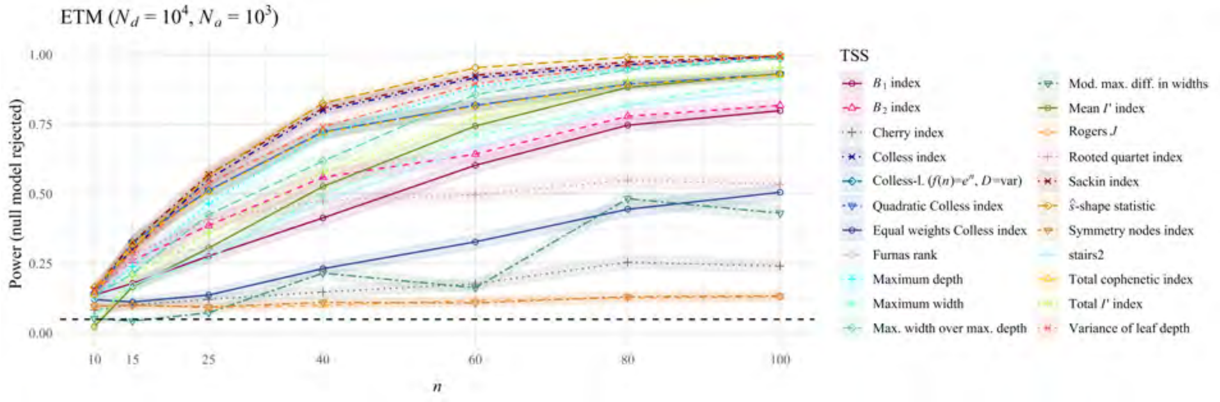


Figure 2: The power of all TSS to correctly identify trees generated under the ETM as not having been generated under the Yule model as a function of n.

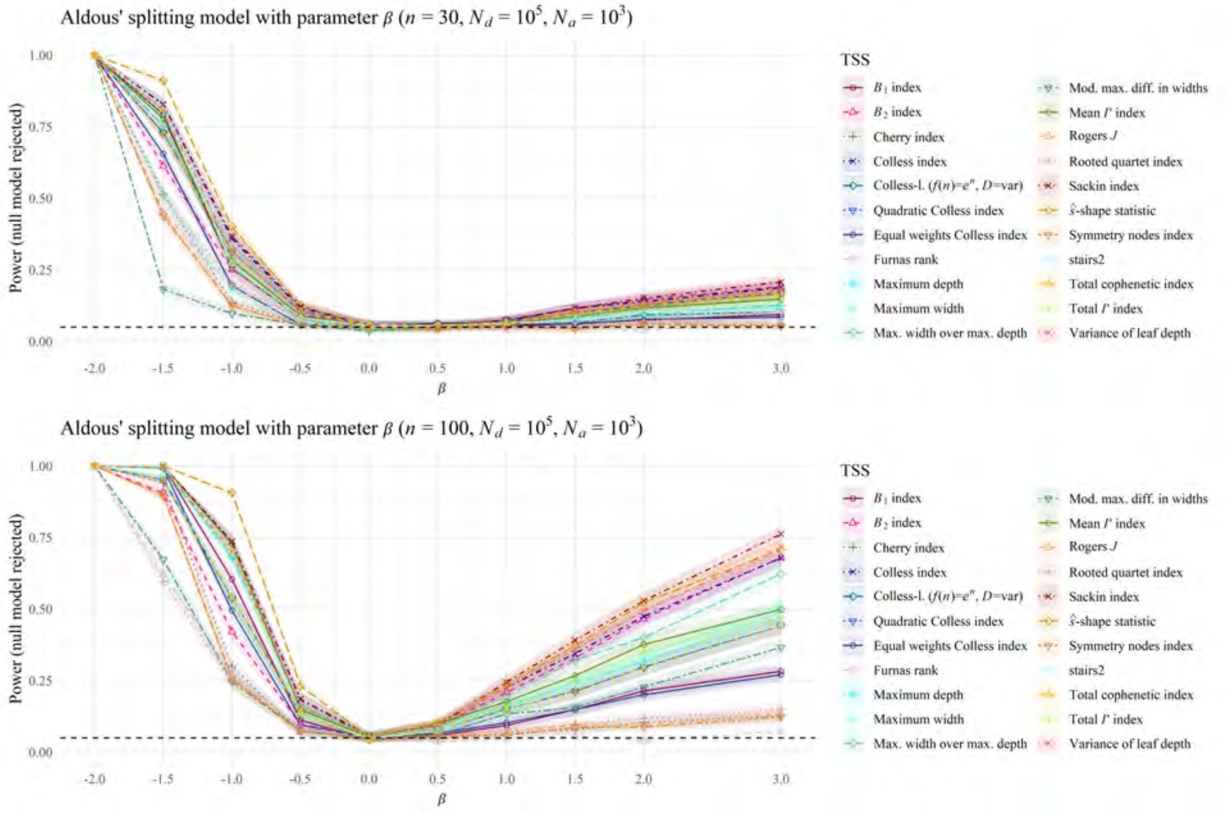


Figure 3: The power of all TSS to correctly identify trees generated under the Aldous' β splitting model as not having been generated under the Yule model as a function of β. Since for $\beta = 0$, Aldous' splitting model corresponds to the Yule model, all TSS rejected ≈ 0.05 % of the trees in that case (as specified with the level of significance).

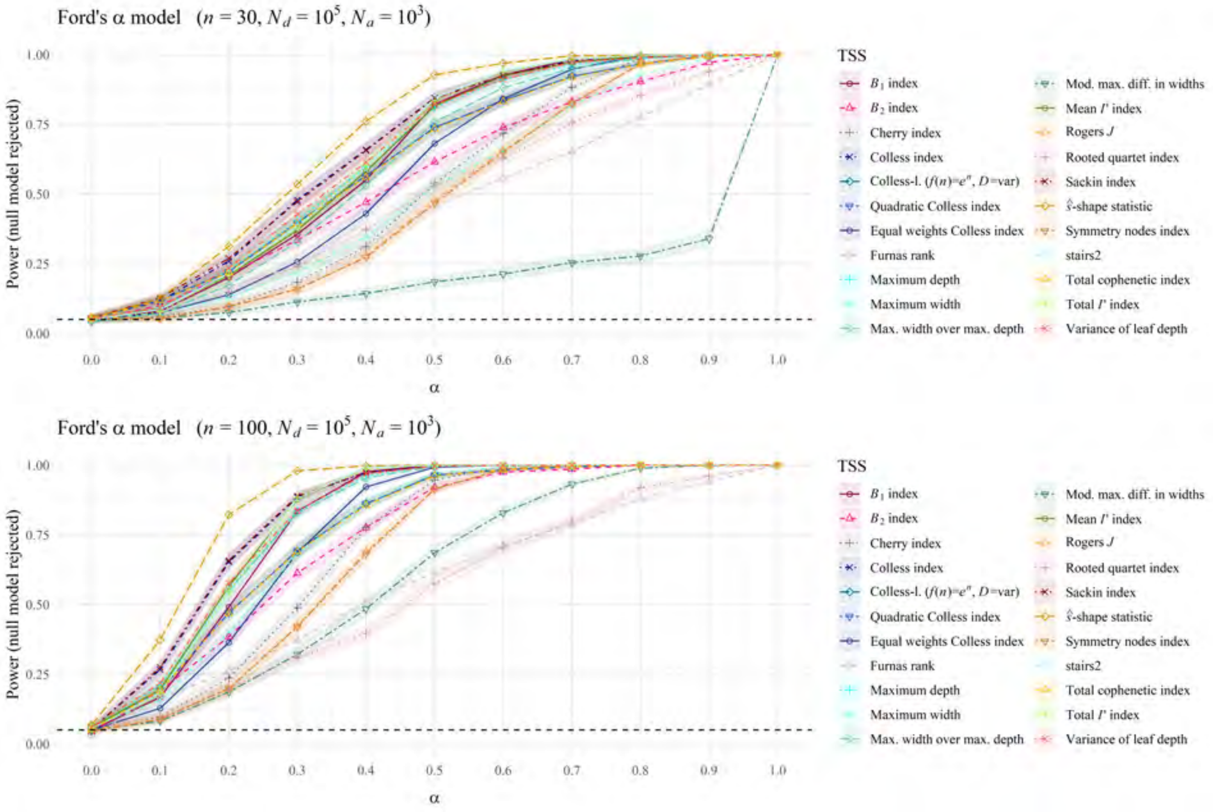


Figure 4: The power of all TSS to correctly identify trees generated under Ford's α model as not having been generated under the Yule model as a function of α. Since for $\alpha = 0$, Ford's α model corresponds to the Yule model, all TSS rejected ≈ 0.05 % of the trees in that case (as specified with the level of significance).

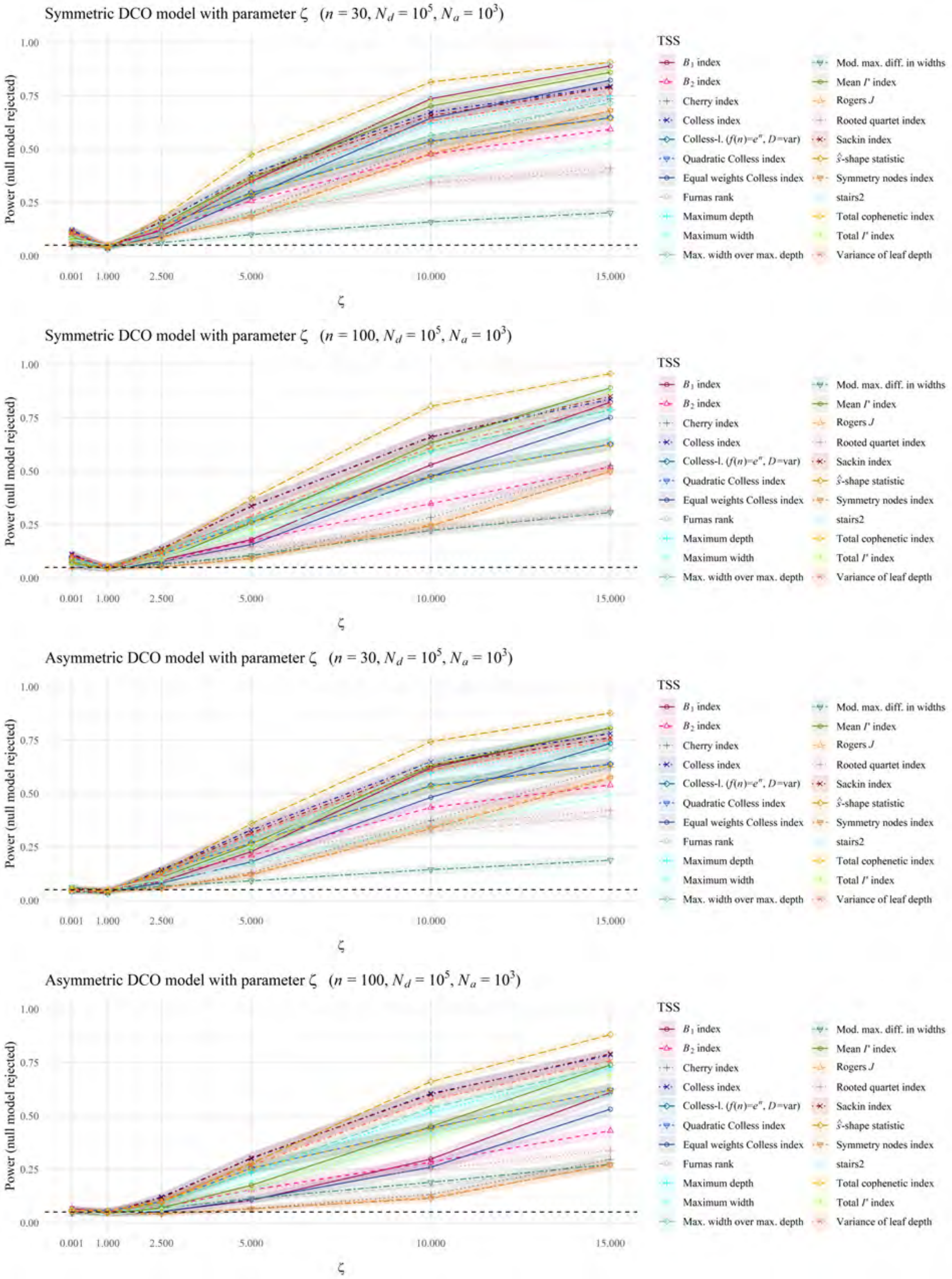


Figure 5: The power of all TSS to correctly identify trees generated under the (a)symmetric DCO model as not having been generated under the Yule model as a function of ζ. Since for $\zeta = 1$, the (a)symmetric DCO model corresponds to the Yule model, all TSS rejected ≈ 0.05 % of the trees in that case (as specified with the level of significance).

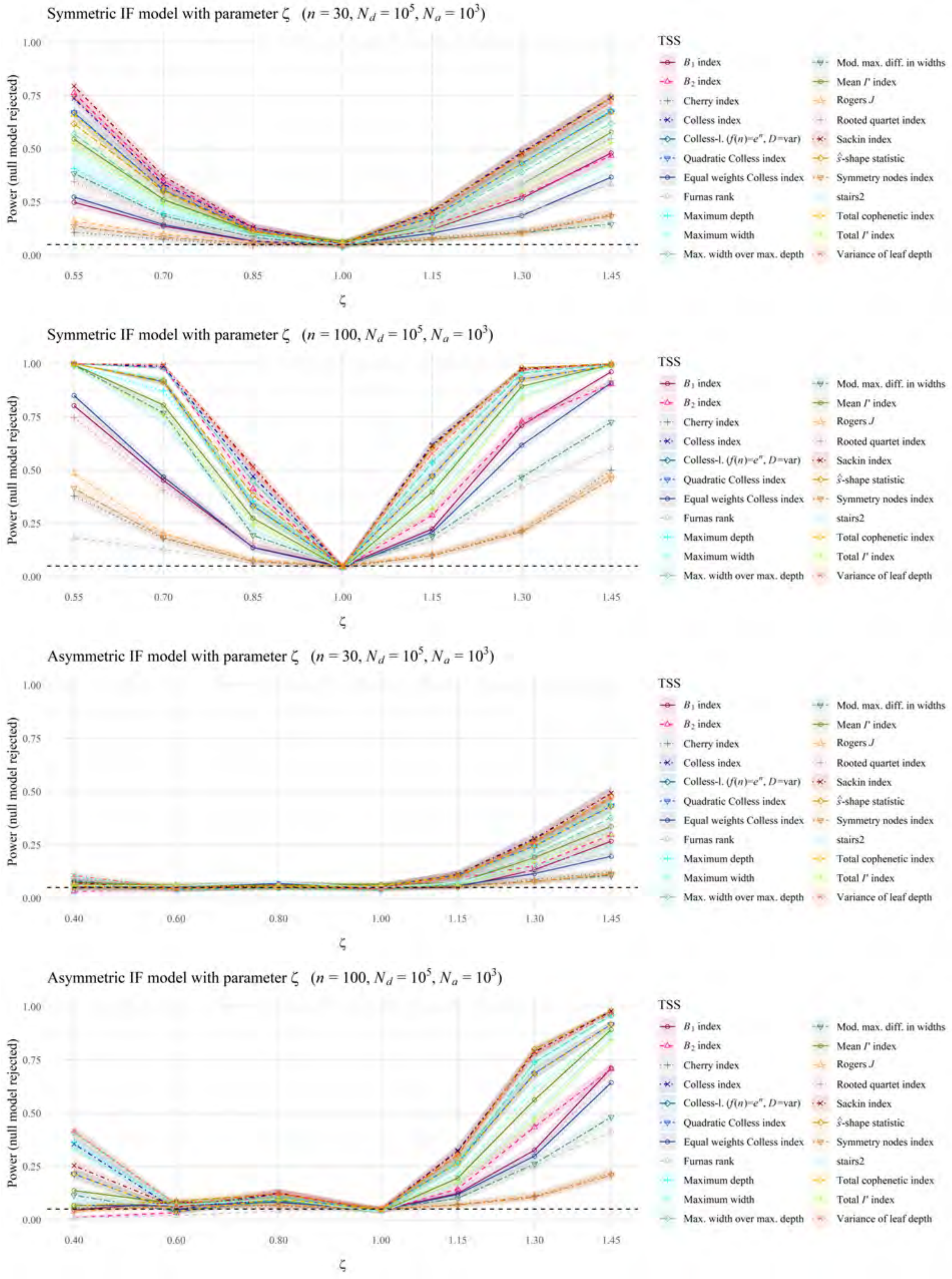


Figure 6: The power of all TSS to correctly identify trees generated under the (a)symmetric IF model as not having been generated under the Yule model as a function of ζ. Since for $\zeta = 1$, the (a)symmetric IF model corresponds to the Yule model, all TSS rejected ≈ 0.05 % of the trees in that case (as specified with the level of significance).

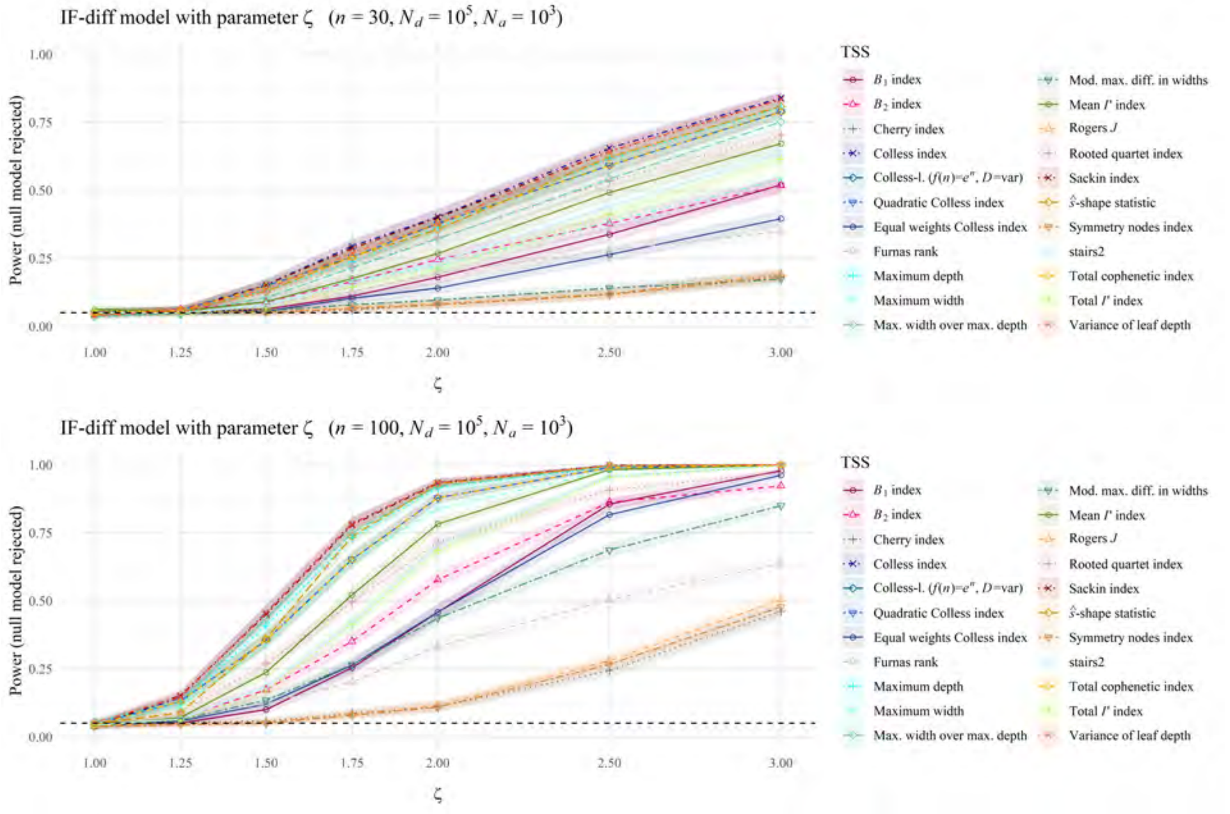


Figure 7: The power of all TSS to correctly identify trees generated under the IF-diff model as not having been generated under the Yule model as a function of ζ. Since for $\zeta = 1$, the IF-diff model corresponds to the Yule model, all TSS rejected ≈ 0.05 % of the trees in that case (as specified with the level of significance).

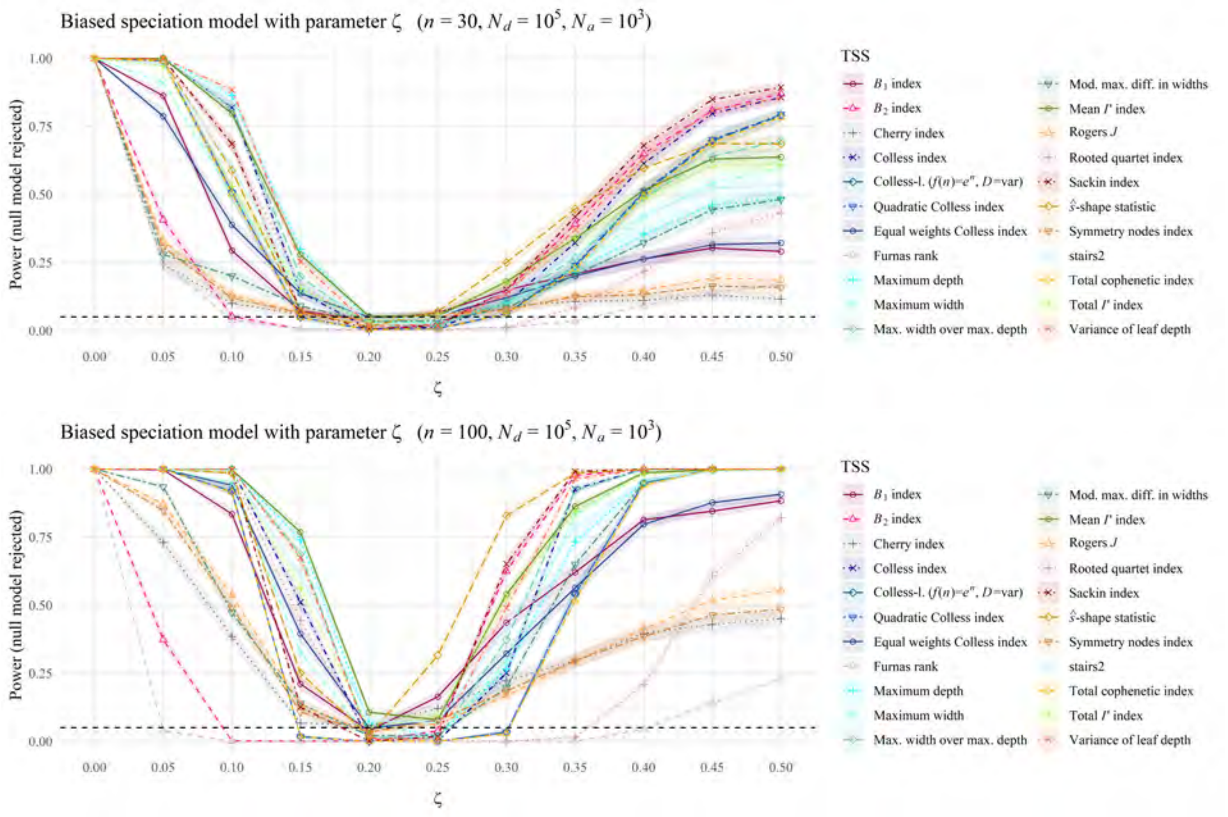


Figure 8: The power of all TSS to correctly identify trees generated under the biased speciation model as not having been generated under the Yule model as a function of ζ.

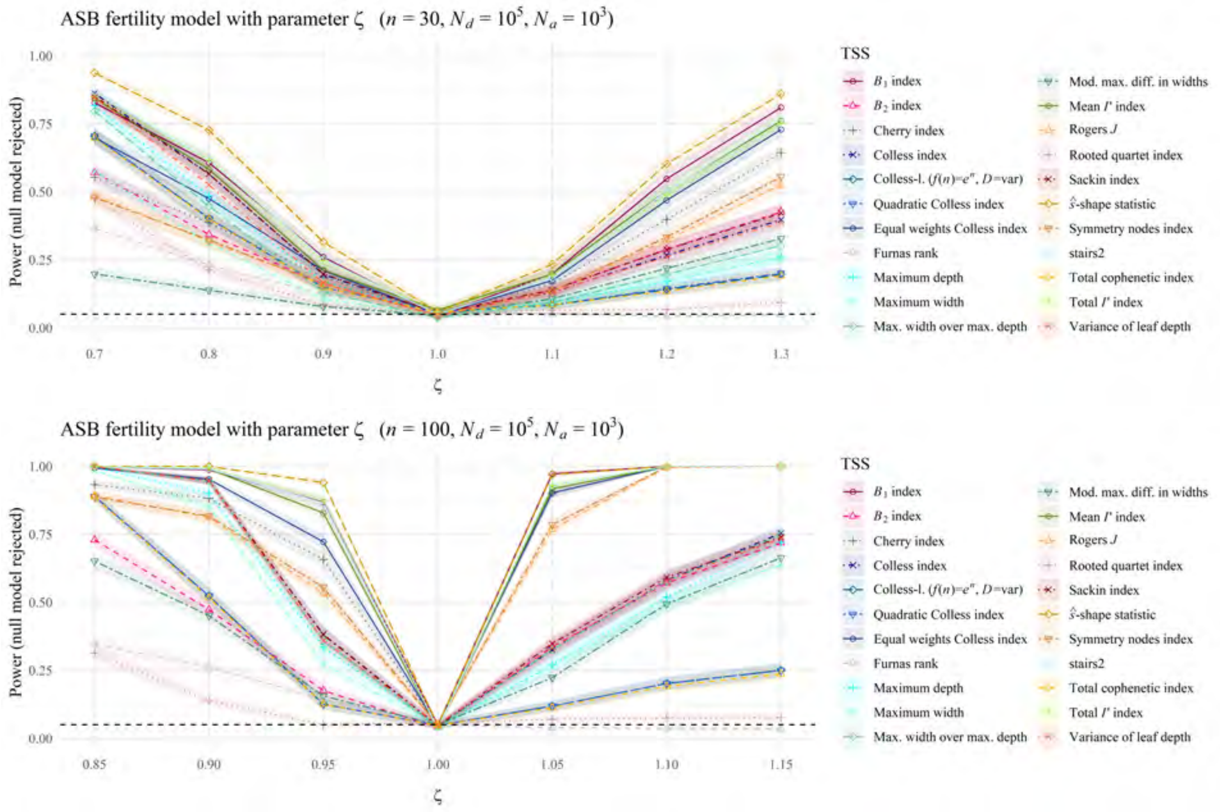


Figure 9: The power of all TSS to correctly identify trees generated under the ASB fertility model as not having been generated under the Yule model as a function of ζ. Since for $\zeta = 1$, the ASB fertility model corresponds to the Yule model, all TSS rejected ≈ 0.05 % of the trees in that case (as specified with the level of significance).

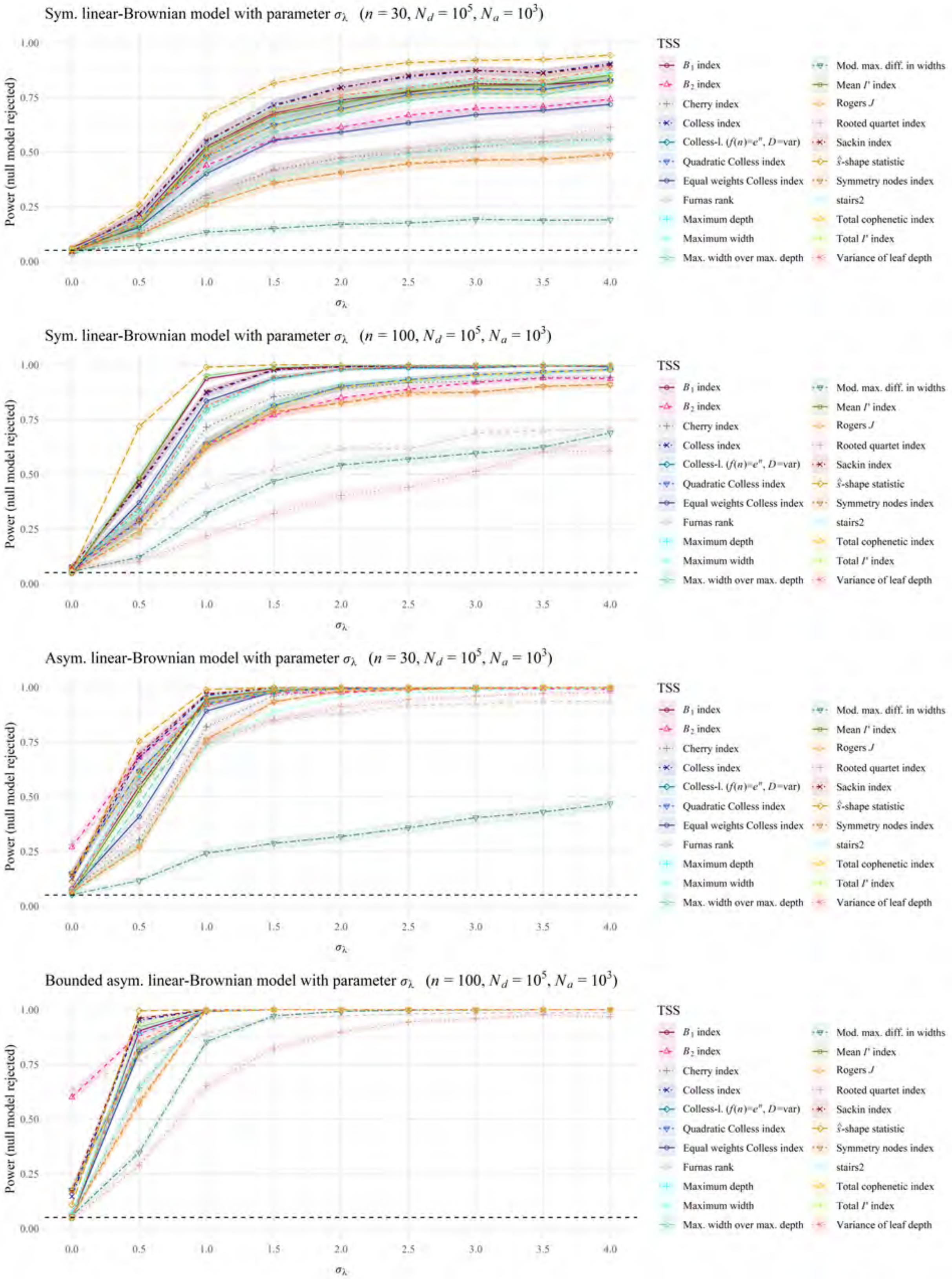
Sym. simple Brownian model with parameter σ ($n = 30$, $N_d = 10^5$, $N_a = 10^3$)

Sym. simple Brownian model with parameter σ ($n = 100$, $N_d = 10^5$, $N_a = 10^3$)

Asym. simple Brownian model with parameter σ ($n = 30$, $N_d = 10^5$, $N_a = 10^3$)

Asym. simple Brownian model with parameter σ ($n = 100$, $N_d = 10^5$, $N_a = 10^3$)

Figure 10: The power of all TSS to correctly identify trees generated under the (a)symmetric simple Brownian model as not having been generated under the Yule model as a function of σ.



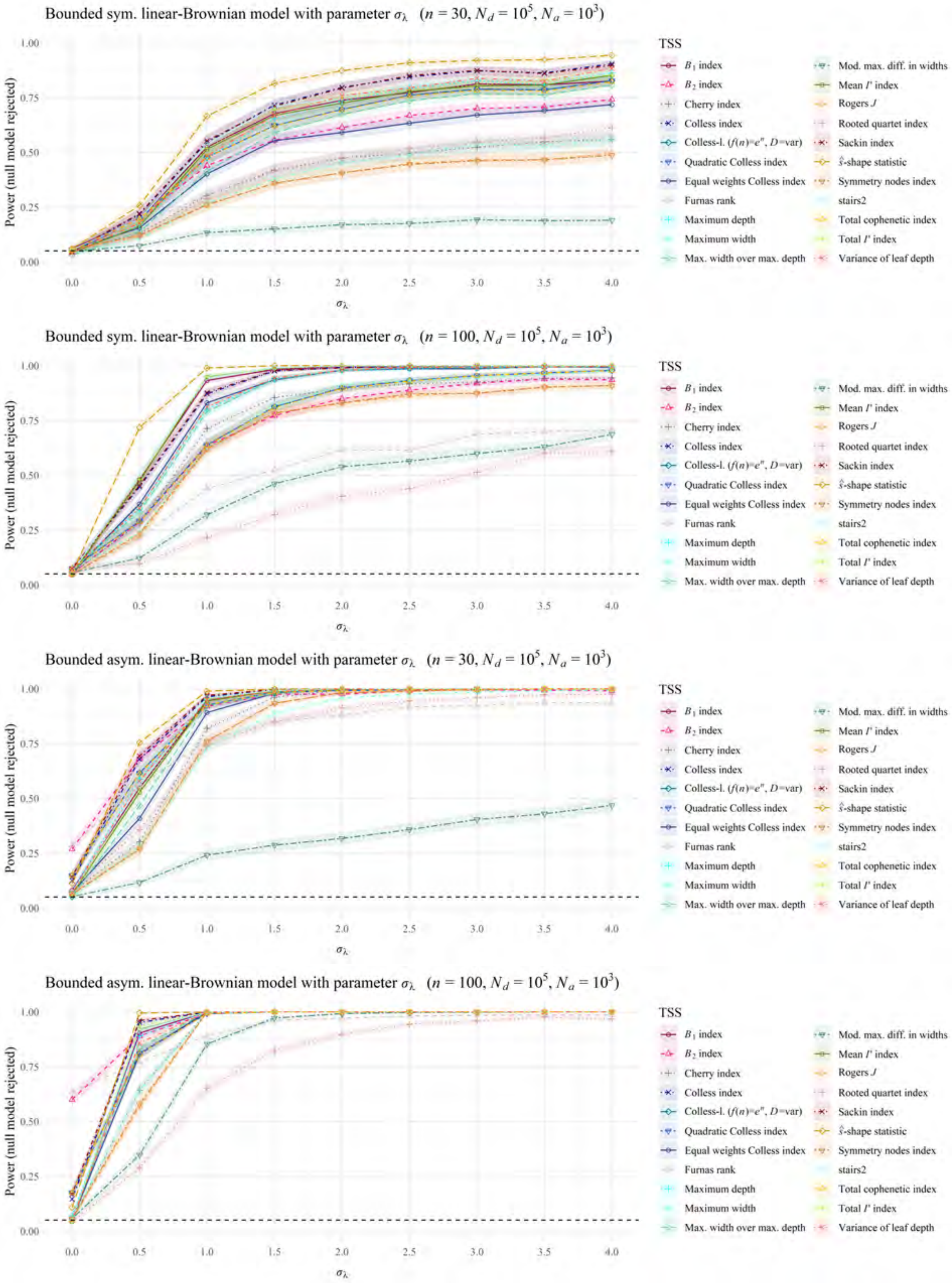
Sym. linear-Brownian model with parameter σ_λ ($n = 30$, $N_d = 10^5$, $N_a = 10^3$)

Sym. linear-Brownian model with parameter σ_λ ($n = 100$, $N_d = 10^5$, $N_a = 10^3$)

Asym. linear-Brownian model with parameter σ_λ ($n = 30$, $N_d = 10^5$, $N_a = 10^3$)

Bounded asym. linear-Brownian model with parameter σ_λ ($n = 100$, $N_d = 10^5$, $N_a = 10^3$)

Figure 11: The power of all TSS to correctly identify trees generated under the (a)symmetric punctuated(-intermittent) linear-Brownian model as not having been generated under the Yule model as a function of σ_λ. The starting trait value is $x_0 = 10$ with constant $\sigma_x = 1$.



Bounded sym. linear-Brownian model with parameter σ_λ ($n = 30$, $N_d = 10^5$, $N_a = 10^3$)

Bounded sym. linear-Brownian model with parameter σ_λ ($n = 100$, $N_d = 10^5$, $N_a = 10^3$)

Bounded asym. linear-Brownian model with parameter σ_λ ($n = 30$, $N_d = 10^5$, $N_a = 10^3$)

Bounded asym. linear-Brownian model with parameter σ_λ ($n = 100$, $N_d = 10^5$, $N_a = 10^3$)

Figure 12: The power of all TSS to correctly identify trees generated under the bounded (a)symmetric punctuated(-intermittent) linear-Brownian model as not having been generated under the Yule model as a function of σ_λ. The starting trait value is $x_0 = 10$ with constant $\sigma_x = 1$.

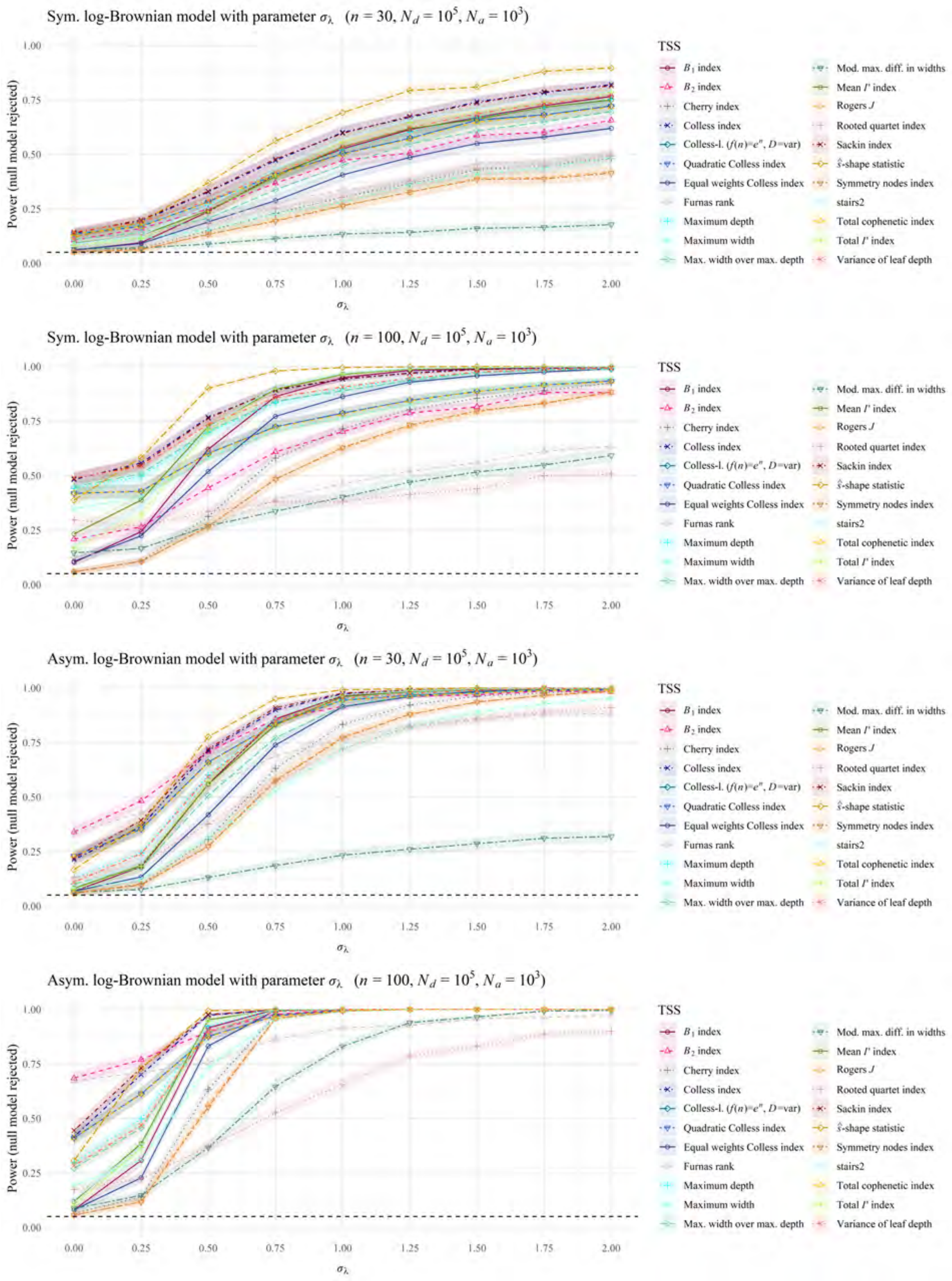


Figure 13: The power of all TSS to correctly identify trees generated under the (a)symmetric punctuated(-intermittent) log-Brownian model as not having been generated under the Yule model as a function of σ_λ. The starting trait value is $x_0 = 10$ with constant $\sigma_x = 0.1$.

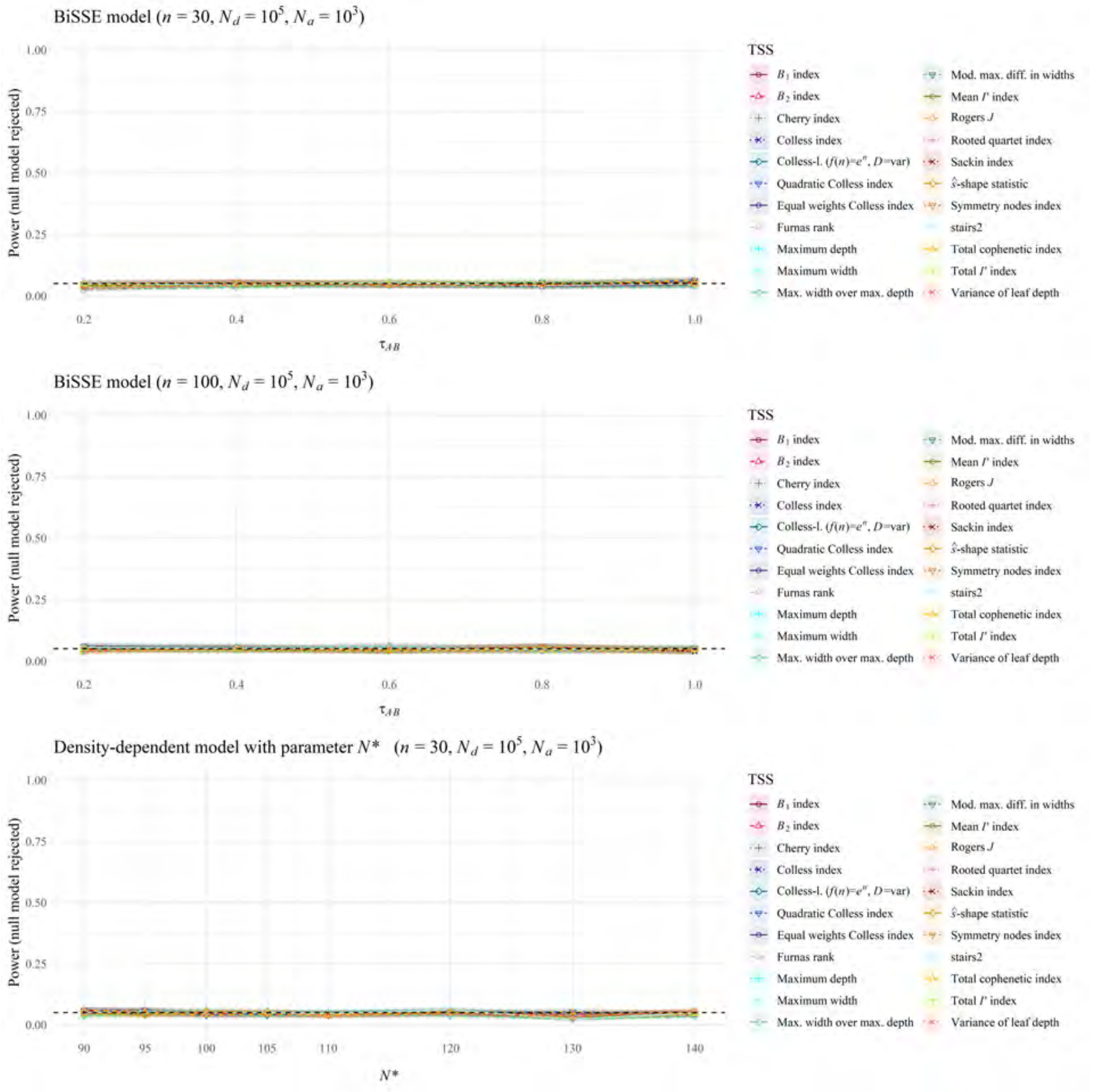


Figure 14: The power of all TSS to correctly identify trees generated under a BiSSE model or density-dependent model as not having been generated under the Yule model as a function of τ_{AB} or N^*, respectively. Note that the power of all TSS in this case is remarkably low; in particular, it is not higher than the significance level of 0.05. The parameters for the BiSSE model are $\lambda_A = \lambda_B = 1$, $\mu_A = 0$, $\mu_B = 0.1$, and $\tau_{BA} = 1$. Other parameter settings yielded similar results, supporting the conjecture that all BiSSE and density-dependent models induce the same distribution on $\mathcal{BT}_n^*$ as the Yule model. This is plausible since they are a mix of two or several simple birth-death models, which already induce the same distribution on $\mathcal{BT}_n^*$ as the Yule model.

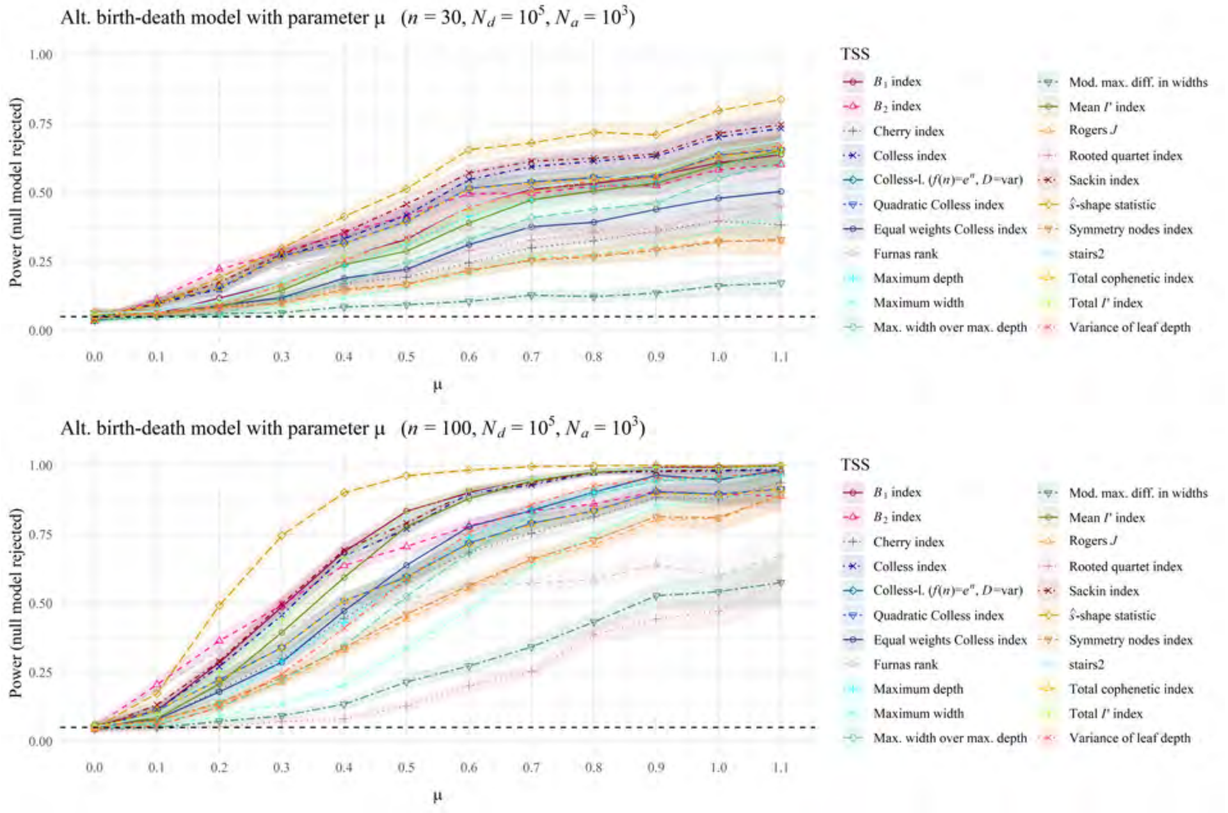


Figure 15: The power of all TSS to correctly identify trees generated under the alternative birth-death model as not having been generated under the Yule model as a function of μ.

6 Experimenting with different null models

6.1 Power comparisons with a different null model

This section contains figures of simulation studies for a different choice of null model. Specifically, we chose the Aldous' β splitting model with $\beta = -1$, which is often discussed as a good candidate model for empirical trees [2]. All power comparisons were conducted analogously to those against the Yule model and can be replicated by using `null_model = list("aldous", -1)` in the code shown at the beginning of Section 5. Where relevant, we again used an initial speciation rate of $\lambda_0 = 1$.

Results In the following, we summarize the most important observations obtained in this simulation study. Many observations are similar to the ones we discussed for the simulation study using the Yule model as the null model (Section 4 in the main text).

First, the power curves look quite different for the various alternative models. Sometimes, they show almost no crossings (e.g., Figure 19), indicating that the ranking of the TSS by their power does not change across the different settings tested, whereas in other cases, there are significant changes (e.g., Figure 18; notice that Figure 18 shows the power curves for the biased speciation model and we observed a very similar pattern for this model (i.e., many crossings) when testing against the Yule model). Moreover, in some cases, the curves are relatively clearly separated (e.g., bottom panels in Figures 16 and 17, as well as Figure 18), indicating observable differences in the power of the TSS, whereas in other cases several curves are on top of each other (e.g., top panels in Figures 16 and 17, first and third panel in Figure 19, and top panel in Figure 20), indicating equal performance.

Figure 20 for the alternative birth-death model is also quite remarkable in another sense: For $n = 30$, we observe a relatively lower power for all TSS, especially at $\mu = 0.4$. For $n = 100$, the power generally increases (as expected), but at $\mu = 0.4$, the power is still close to 0.05 (the significance level) for almost all TSS. The only two exceptions are the B_2 index and the Furnas rank. This is astounding as the latter is primarily used as a tree enumeration method and not often employed in applications. This serves as an excellent example that there are exceptional TSS which should not be overlooked. Without including these two indices, one might have concluded that both models are equivalent. In other instances, the low power of all TSS indeed correctly recognizes the equivalence of models (see $\beta = -1$ in the Aldous' β splitting model (Figure 16)). Additionally, we observe a low power across all TSS for $\alpha = 0.2$ in Ford's α model (Figure 17); this could be an indication that the Ford's $\alpha = 0.2$ and Aldous' $\beta = -1$ models are equivalent, and it would be interesting to investigate this further. It also appears to be the case that the asymmetric log-Brownian model with parameter $\sigma_\lambda = 0.25$ (Figure 19) is very similar (but likely not equivalent) to Aldous' $\beta = -1$ model.

In regard to "good" and "bad" TSS, we can observe that the $\widehat{s}$-shape statistic almost always outperforms the other TSS (the only exception are certain choices of ζ in the biased speciation model; see Figure 18). This is perhaps surprising as the $\widehat{s}$-shape statistic was originally introduced with the purpose to detect deviations from the Yule model [6], not from other null models. Nevertheless, this finding replicates the strong performance of the $\widehat{s}$-shape statistic we already observed for the Yule model. Similarly, recalling that the modified maximum difference in widths and the rooted quartet index showed relatively poor performance when using the Yule model as the null model, we can observe that this is also often the case in the present set of analyses, i.e., when using Aldous' $\beta = -1$ splitting model as the null model. However, there are also TSS which perform quite differently under the two null models. As an example, consider the log-Brownian models depicted in Figure 13 for the Yule model and in Figure 19 for the Aldous' $\beta = -1$ splitting model. Focusing on the Sackin and Colless indices, we can observe that they have a high power of detecting deviation from the Yule model but show a mediocre performance for the Aldous' $\beta = -1$ splitting model.

In summary, we thus reiterate what we already said in the main text, namely that there can be significant performance differences among the different TSS, which depend on factors such as the alternative model, the model parameters, or simply the tree size. Thus, there is no "one-fits-all" TSS, highlighting that the choice of the most powerful TSS for each individual case is important. We thus highly recommend to perform a power analysis such as the one presented here prior to downstream analyses.

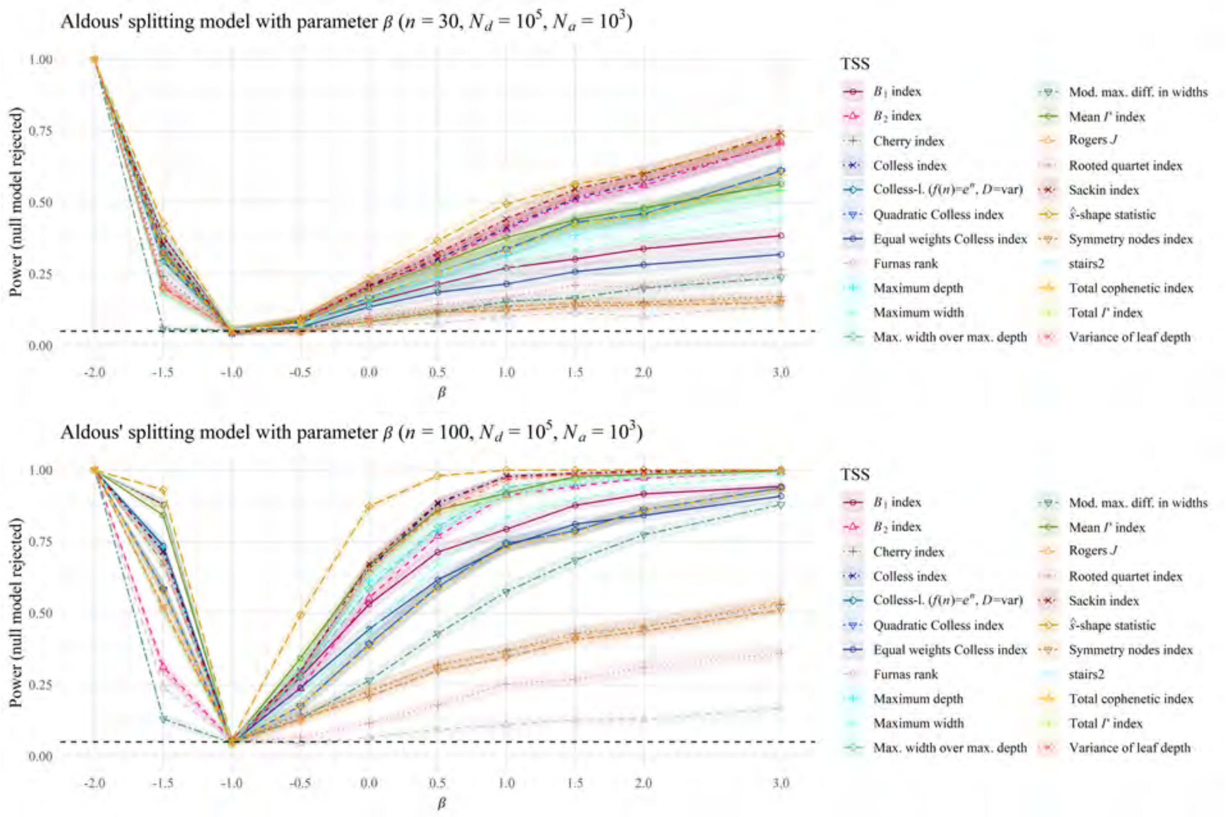


Figure 16: The power of all TSS to correctly identify trees generated under the Aldous' β splitting model as not having been generated under the Aldous' $\beta = -1$ splitting model as a function of β.

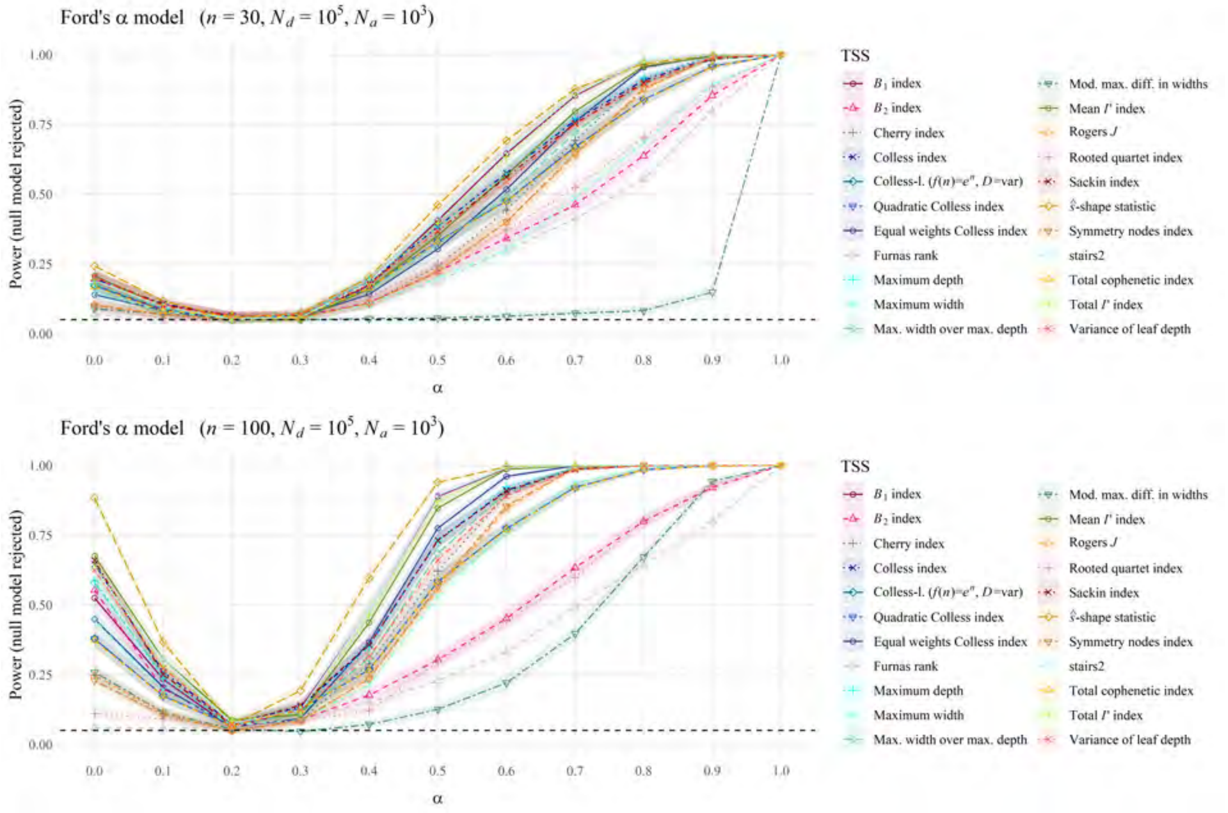


Figure 17: The power of all TSS to correctly identify trees generated under Ford's α model as not having been generated under the Aldous' $\beta = -1$ splitting model as a function of α.

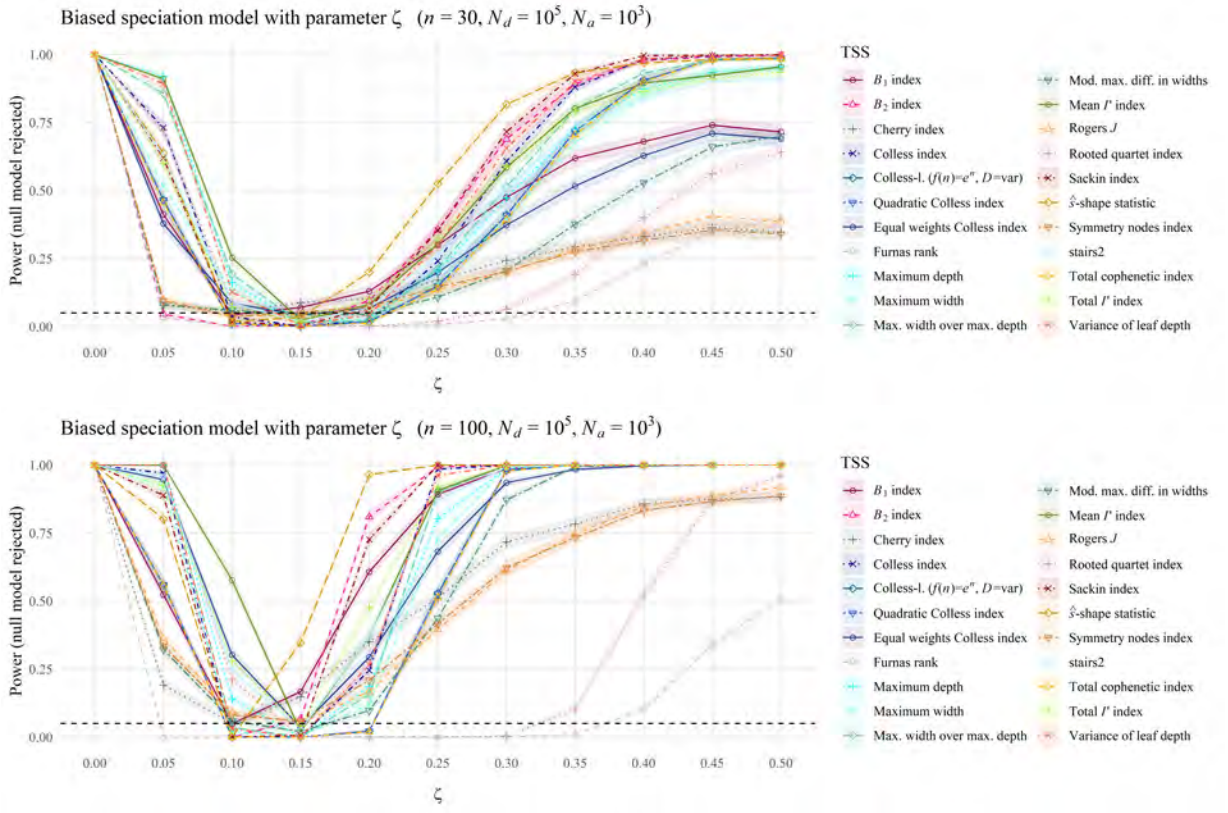


Figure 18: The power of all TSS to correctly identify trees generated under the biased speciation model as not having been generated under the Aldous' $\beta = -1$ splitting model as a function of ζ.

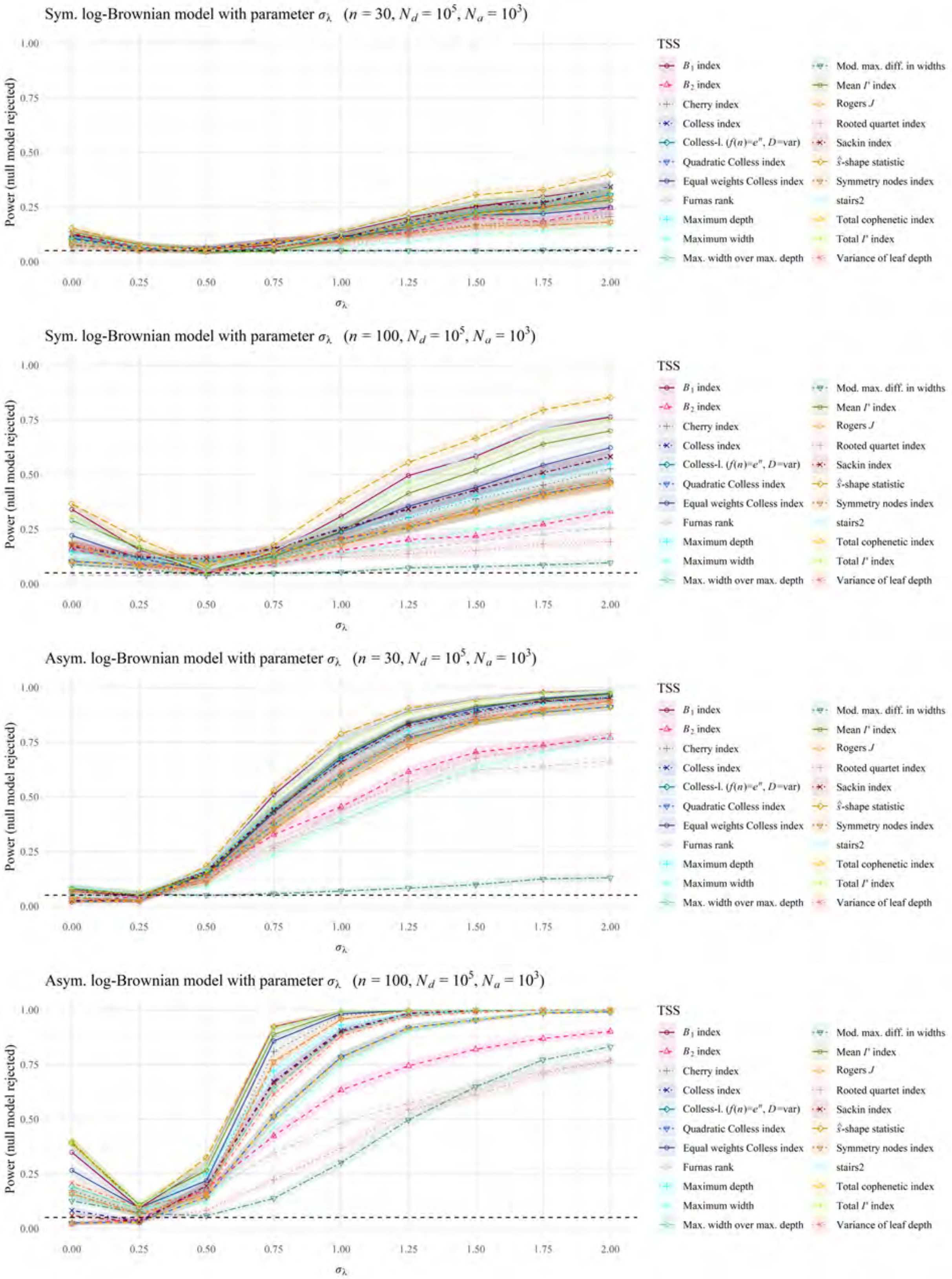


Figure 19: The power of all TSS to correctly identify trees generated under the (a)symmetric punctuated(-intermittent) log-Brownian model as not having been generated under the Aldous' $\beta = -1$ splitting model as a function of σ_λ. The starting trait value is $x_0 = 10$ with constant $\sigma_x = 0.1$.

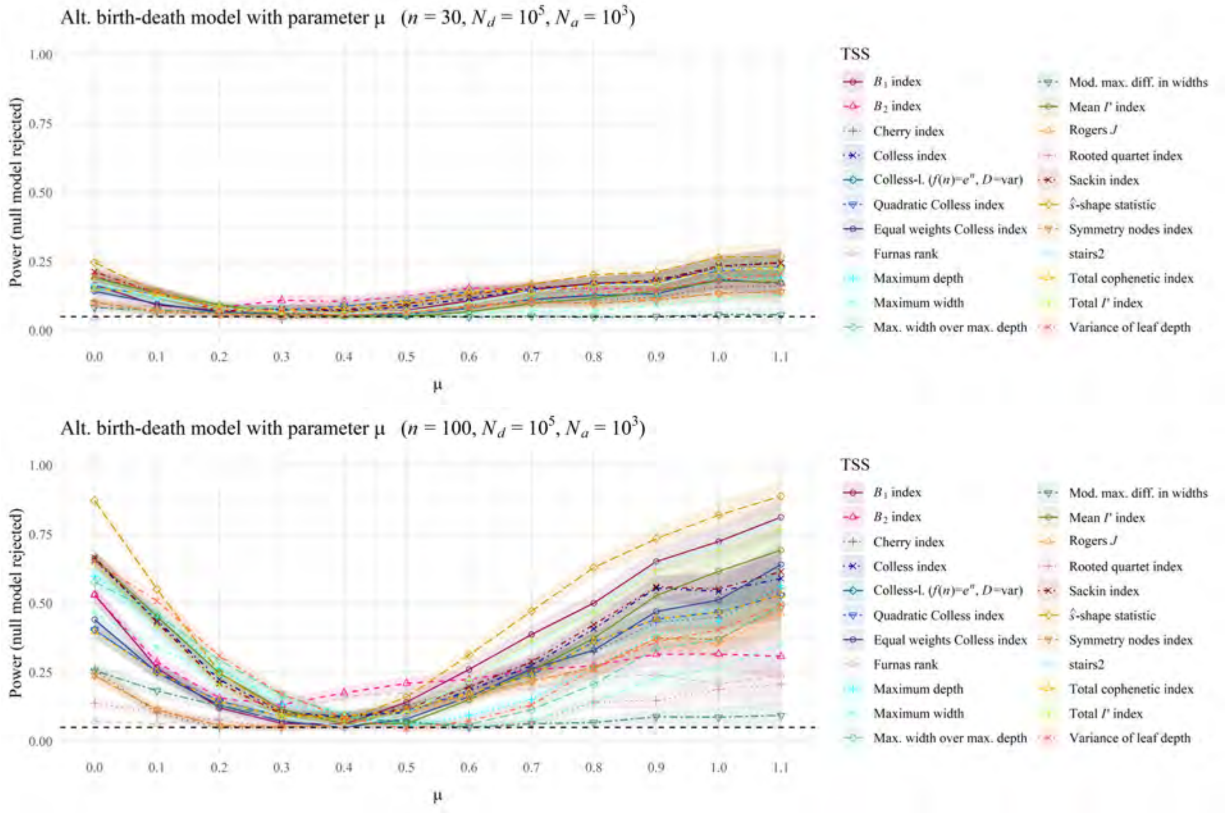


Figure 20: The power of all TSS to correctly identify trees generated under the alternative birth-death model as not having been generated under the Aldous' $\beta = -1$ splitting model as a function of μ.

6.2 Approaching Khurana et al. [46]'s research questions with our methods

Recently, Khurana et al. [46] used a large selection of tree shape statistics in a study aimed at discriminating between empirical trees and trees generated under a constant-rate birth-death model. They used the latter model to simulate trees of the same sizes, i.e., with the same number of leaves n, as the various empirical trees in their data set. Then, they observed the number of standard deviations of the TSS values of the empirical trees and of the simulated trees from the mean of the TSS values across all simulated trees of the same size. The empirical and simulated values were pooled and investigated with regards to systematic differences between both groups. Furthermore, the authors assessed how well different TSS managed to identify differences. Khurana et al. [46] found that overall, the B_2 index, the variance of leaf depth, the I-based indices (specifically mean I and median I), the stairs2 index, and the normalized Colless index were the strongest discriminators, whereas the symmetry nodes index, Roger's J, the Furnas rank, and the B_1 index were poor discriminators.

The advantage of the approach by Khurana et al. [46] is that it can deal with data sets containing trees of different sizes and few representatives of each size. In contrast, our method of comparing the power of different TSS requires a fixed number n of leaves. Nevertheless, we can investigate the problem from a different angle, e.g., by comparing the empirical trees to multiple alternative models.

In an attempt to mimic the study by [46] with our methods using our R package `poweRbal`, we looked at the final list of trees used in Khurana et al. [46]. Unfortunately, the trees had a variety of different leaf numbers, so we used the most common leaf number, namely $n = 11$, with 41 trees of that size. Similar to Khurana et al. [46], who did not remove possible outgroups for their main analysis (and found that removing trees with potential outgroups did not have a big effect on their results), we also retained all potential outgroups and performed two sets of analyses.

First, we used the 41 trees with $n = 11$ leaves as the basis for a null model. Here, we assumed each tree to have a probability of 1/41 to obtain the distribution and computed the regions of acceptance based on this; thus $N_d = 41$. Then, we explored how well the TSS could detect trees generated under several representatives of the Aldous' β splitting model as being distinct from the empirical trees (see Figure 21). Of course, $N_d = 41$ trees is not sufficient to build a statistically sound and robust test, but we point out some observations. First, for most TSS we see a dip in the power for $\beta = -1$, which supports the finding of Aldous [2] that this is a parameter choice that fits empirical data well.

Second, to compare our results to Khurana et al. [46], note that the constant-rate birth-death trees induce the same distribution as the Yule model (and thus as the Aldous' $\beta = 0$ model) on $\mathcal{BT}^*_n$. Thus, we can compare our results with those of [46] by focusing on $\beta = 0$ in Figure 21. Interestingly, our results do not exactly match those of [46]. For instance, we can observe that the B_1 index has the highest power, whereas it was found to be a poor discriminator in [46]. However, we can confirm that the B_2 index, variance of leaf depth, and the stairs2 index perform reasonably well. We can also confirm that the symmetry nodes index, Roger's J, and the Furnas rank perform relatively poorly. Thus, while we do not get exactly the same results in this analysis (which could be due to the small sample size of $N_d = 41$), we can replicate some of the findings of Khurana et al. [46].

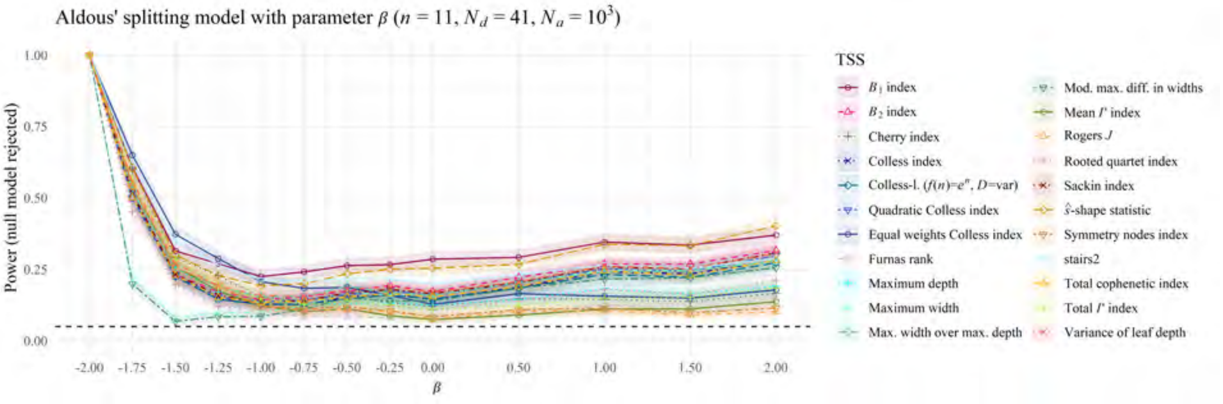


Figure 21: The power of all TSS to correctly identify trees generated under the Aldous' β splitting model as being distinct from the empirical trees with $n = 11$ leaves used in Khurana et al. [46] as a function of β.

In a second set of analyses, we used the empirical trees as the alternative model and tested them against two null models, the Yule model and Aldous' β splitting model with $\beta = -1$. In Figure 22, we plot the fractions of trees for which the TSS rejected the null model, i.e., found the empirical trees to be distinct from the Yule model or Aldous' $\beta = -1$ model, respectively. Again, due to the small sample size of $N_a = 41$, we can make some observations, but not draw any powerful conclusions.

For the first null model, i.e., the Yule model, it is interesting to note that the Mean I' and Total I' indices never reject the null model, while all other TSS show a non-zero fraction of rejection. The fraction of rejected trees is highest for the B_2 index, which resembles the observation by [46] that the B_2 index is a good discriminator. On the other hand, the Furnas rank performs very similarly to the B_2 index in our analysis, whereas it behaved very differently in the study of [46]. Again, these differences could be due to the small sample size.

For the second null model, i.e., the Aldous $\beta = -1$ splitting model, we observe that the fractions for which it is rejected are very small for most TSS, possibly indicating a good fit of the model.Interestingly, the TSS that show a non-zero fraction of rejected trees for $\beta = -1$ include the symmetry nodes index, Roger's J, the Furnas rank, and the B_1 index, i.e., all indices that were found to be poor discriminators in the study by Khurana et al. [46].

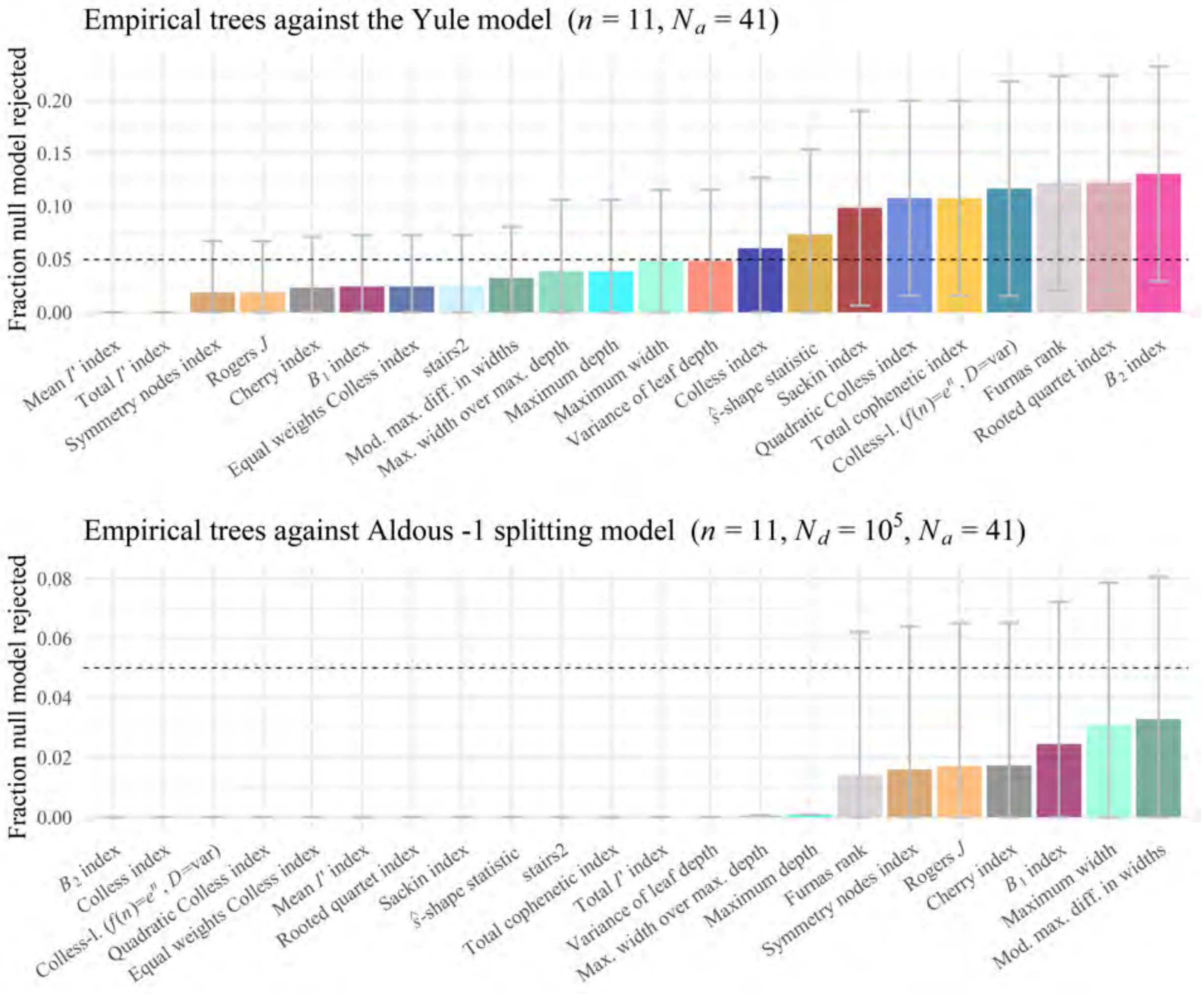


Figure 22: The fractions of trees for which the individual TSS reject the null model, i.e., they assess the empirical trees as being distinct from trees generated under the Yule or Aldous' $\beta = -1$ splitting model, respectively. The results concerning the Yule model are based on the exact distribution for $n = 11$, hence no indication of N_d. The gray error bars indicate the 95%-confidence intervals, which – unsurprisingly for such a small N_a – are very wide.

References

[1] P.-M. Agapow and A. Purvis. Power of eight tree shape statistics to detect nonrandom diversification: a comparison by simulation of two models of cladogenesis. *Systematic Biology*, 51(6):866–872, 2002. doi: 10.1080/10635150290102564.

[2] D. Aldous. Probability Distributions on Cladograms. In *Random Discrete Structures*, pages 1–18. Springer New York, 1996. doi: 10.1007/978-1-4612-0719-1_1.

[3] K. Bartoszek, T. M. Coronado, A. Mir, and F. Rosselló. Squaring within the Colless index yields a better balance index. *Mathematical Biosciences*, 331:108503, 2021. doi: 10.1016/j.mbs.2020.108503.

[4] R. Bellman and T. E. Harris. On the theory of age-dependent stochastic branching processes. *Proceedings of the National Academy of Sciences*, 34(12):601–604, 1948.

[5] M. G. B. Blum and O. François. On statistical tests of phylogenetic tree imbalance: the Sackin and other indices revisited. *Mathematical Biosciences*, 195(2):141–153, 2005. doi: 10.1016/j.mbs.2005.03.003.

[6] M. G. B. Blum and O. François. Which random processes describe the Tree of Life? A large-scale study of phylogenetic tree imbalance. *Systematic Biology*, 55(4):685–691, 2006.

[7] K. M. A. Chan and B. R. Moore. Accounting for mode of speciation increases power and realism of tests of phylogenetic asymmetry. *The American Naturalist*, 153(3):332–346, 1999. doi: 10.1086/303173.

[8] B. Chen, D. Ford, and M. Winkel. A new family of Markov branching trees: the alpha-gamma model. *Electronic Journal of Probability*, 14(0):400–430, 2009. doi: 10.1214/ejp.v14-616.

[9] C. Colijn and J. Gardy. Phylogenetic tree shapes resolve disease transmission patterns. *Evolution, Medicine, and Public Health*, 2014(1):96–108, 2014. ISSN 2050-6201. doi: 10.1093/emph/eou018.

[10] C. Colijn and G. Plazzotta. A metric on phylogenetic tree shapes. *Systematic Biology*, 67(1):113–126, 2018. doi: 10.1093/sysbio/syx046.

[11] D. Colless. Review of "Phylogenetics: the theory and practice of phylogenetic systematics". *Systematic Zoology*, 31(1):100–104, 1982. ISSN 00397989.

[12] T. M. Coronado, A. Mir, F. Rosselló, and G. Valiente. A balance index for phylogenetic trees based on rooted quartets. *Journal of Mathematical Biology*, 79(3):1105–1148, 2019. doi: 10.1007/s00285-019-01377-w.

[13] T. M. Coronado, M. Fischer, L. Herbst, F. Rosselló, and K. Wicke. On the minimum value of the Colless index and the bifurcating trees that achieve it. *Journal of Mathematical Biology*, 80(7):1993–2054, 2020. doi: 10.1007/s00285-020-01488-9.

[14] T. M. Coronado, A. Mir, F. Rosselló, and L. Rotger. On Sackin's original proposal: the variance of the leaves' depths as a phylogenetic balance index. *BMC Bioinformatics*, 21(1), 2020. doi: 10.1186/s12859-020-3405-1.

[15] R. P. Dobrow and J. A. Fill. Total path length for random recursive trees. *Combinatorics, Probability and Computing*, 8(4):317–333, 1999. doi: 10.1017/S0963548399003855.

[16] J. L. Doob. Topics in the theory of Markoff chains. *Transactions of the American Mathematical Society*, 52, 1942. ISSN 0002-9947. doi: 10.1090/S0002-9947-1942-0006633-7.

[17] J. Felsenstein, J. Archie, W. Day, W. Maddison, C. Meacham, F. Rohlf, and D. Swofford. The newick tree format., 2000. URL `http://evolution.genetics.washington.edu/phylip/newicktree.html`.

[18] M. Fischer, L. Herbst, S. J. Kersting, L. Kühn, and K. Wicke. *Tree Balance Indices - A Comprehensive Survey.* Springer, Berlin, 2023. ISBN 978-3-031-39799-8.

[19] R. G. FitzJohn. Quantitative traits and diversification. *Systematic Biology*, 59(6):619–633, 2010. doi: 10.1093/sysbio/syq053.

[20] R. G. FitzJohn. Diversitree: comparative phylogenetic analyses of diversification in R. *Methods in Ecology and Evolution*, 3(6):1084–1092, 2012. doi: 10.1111/j.2041-210x.2012.00234.x.

[21] D. J. Ford. Probabilities on cladograms: introduction to the alpha model, 2005.

[22] W. A. Freyman and S. Höhna. Cladogenetic and anagenetic models of chromosome number evolution: A bayesian model averaging approach. *Systematic Biology*, 67(2):195–215, 2017. doi: 10.1093/sysbio/syx065.

[23] G. W. Furnas. The generation of random, binary unordered trees. *Journal of Classification*, 1(1): 187–233, 1984. doi: 10.1007/bf01890123.

[24] G. Fusco and Q. C. B. Cronk. A new method for evaluating the shape of large phylogenies. *Journal of Theoretical Biology*, 175(2):235–243, 1995. doi: 10.1006/jtbi.1995.0136.

[25] E. E. Goldberg, L. T. Lancaster, and R. H. Ree. Phylogenetic inference of reciprocal effects between geographic range evolution and diversification. *Systematic Biology*, 60(4):451–465, 2011. doi: 10.1093/sysbio/syr046.

[26] O. Hagen and T. Stadler. TreeSim: Simulating phylogenetic trees under general Bellman–Harris models with lineage-specific shifts of speciation and extinction in R. *Methods in Ecology and Evolution*, 9(3): 754–760, 2017. doi: 10.1111/2041-210x.12917.

[27] O. Hagen, K. Hartmann, M. Steel, and T. Stadler. Age-dependent speciation can explain the shape of empirical phylogenies. *Systematic Biology*, 64(3):432–440, 2015. doi: 10.1093/sysbio/syv001.

[28] N. Hallinan. The generalized time variable reconstructed birth–death process. *Journal of Theoretical Biology*, 300:265–276, 2012. doi: 10.1016/j.jtbi.2012.01.041.

[29] E. F. Harding. The probabilities of rooted tree-shapes generated by random bifurcation. *Advances in Applied Probability*, 3(1):44–77, 1971. doi: 10.2307/1426329.

[30] T. E. Harris et al. *The theory of branching processes*, volume 6. Springer Berlin, 1963.

[31] K. Hartmann, D. Wong, and T. Stadler. Sampling trees from evolutionary models. *Systematic Biology*, 59(4):465–476, 2010. doi: 10.1093/sysbio/syq026.

[32] P. H. Harvey, R. M. May, and S. Nee. Phylogenies without fossils. *Evolution*, 48(3):523–529, 1994. doi: 10.2307/2410466.

[33] M. Hayati, B. Shadgar, and L. Chindelevitch. A new resolution function to evaluate tree shape statistics. *PLOS ONE*, 14(11):e0224197, 2019. doi: 10.1371/journal.pone.0224197.

[34] S. B. Heard. Patterns in tree balance among cladistic, phenetic, and randomly generated phylogenetic trees. *Evolution*, 46(6):1818–1826, 1992. ISSN 00143820. doi: 10.1111/j.1558-5646.1992.tb01171.x.

[35] S. B Heard. Patterns in phylogenetic tree balance with variable and evolving speciation rates. *Evolution*, 50(6):2141–2148, 1996.

[36] E. Hernández-García, M. Tuğrul, E. Alejandro Herrada, V. M. Eguíluz, and K. Klemm. Simple models for scaling phylogenetic trees. *International Journal of Bifurcation and Chaos*, 20(03):805–811, March 2010. doi: 10.1142/s0218127410026095.

[37] E. A. Herrada, V. M. Eguíluz, E. Hernández-García, and C. M. Duarte. Scaling properties of protein family phylogenies. *BMC Evolutionary Biology*, 11(1), 2011. doi: 10.1186/1471-2148-11-155.

[38] S. Höhna, M. R. May, and B. R. Moore. TESS: an R package for efficiently simulating phylogenetic trees and performing bayesian inference of lineage diversification rates. *Bioinformatics*, 32(5):789–791, 2015. doi: 10.1093/bioinformatics/btv651.

[39] E. W. Holman. Age-dependent and lineage-dependent speciation and extinction in the imbalance of phylogenetic trees. *Systematic Biology*, 66(6):912–916, 2017. doi: 10.1093/sysbio/syx031.

[40] G. Kaur, K. P. Choi, and T. Wu. Distributions of cherries and pitchforks for the Ford model. *Theoretical Population Biology*, 149:27–38, 2023.

[41] D. G. Kendall. On the generalized “birth-and-death” process. *The Annals of Mathematical Statistics*, 19, 1948. ISSN 0003-4851.

[42] D. G. Kendall. On some modes of population growth leading to R. A. Fisher’s logarithmic series distribution. *Biometrika*, 35(1/2):6–15, 1948. doi: 10.2307/2332624.

[43] S. J. Kersting. Genetic programming as a means for generating improved tree balance indices (Master’s thesis, University of Greifswald), 2020.

[44] S. J. Kersting and M. Fischer. Measuring tree balance using symmetry nodes - a new balance index and its extremal properties. *Mathematical Biosciences*, 341:108690, 2021. doi: 10.1016/j.mbs.2021.108690.

[45] S. J. Kersting, K. Wicke, and M. Fischer. Tree balance in phylogenetic models. *Philosophical Transactions of the Royal Society B (submitted)*, 2024.

[46] M. P. Khurana, N. Scheidwasser-Clow, M. J. Penn, S. Bhatt, and D. A. Duchêne. The limits of the constant-rate birth–death prior for phylogenetic tree topology inference. *Systematic Biology*, 2023. ISSN 1076-836X. doi: 10.1093/sysbio/syad075.

[47] J. F. C. Kingman. On the genealogy of large populations. *Journal of applied probability*, 19(A):27–43, 1982.

[48] M. Kirkpatrick and M. Slatkin. Searching for evolutionary patterns in the shape of a phylogenetic tree. *Evolution*, 47(4):1171–1181, 1993. doi: 10.1111/j.1558-5646.1993.tb02144.x.

[49] D. E. Knuth. *The art of computer programming volume 3: Sorting and searching.* Addison-Wesley Professional, 2nd edition, 1998. ISBN 0201896850.

[50] A. Lambert and T. Stadler. Birth–death models and coalescent point processes: The shape and probability of reconstructed phylogenies. *Theoretical Population Biology*, 90:113–128, 2013. doi: 10.1016/j.tpb.2013.10.002.

[51] J. B. Losos and F. R. Adler. Stumped by trees? A generalized null model for patterns of organismal diversity. *The American Naturalist*, 145(3):329–342, 1995.

[52] W. P. Maddison, P. E. Midford, and S. P. Otto. Estimating a binary character’s effect on speciation and extinction. *Systematic Biology*, 56(5):701–710, 2007. doi: 10.1080/10635150701607033.

[53] A. Mir, F. Rosselló, and L. Rotger. A new balance index for phylogenetic trees. *Mathematical Biosciences*, 241(1):125–136, 2013. doi: 10.1016/j.mbs.2012.10.005.

[54] A. Mooers, L. Harmon, M. Blum, D. Wong, and S. Heard. *Some models of phylogenetic tree shape*, pages –. Oxford University Press, 2007.

[55] A. O. Mooers and S. B. Heard. Inferring evolutionary process from phylogenetic tree shape. *The Quarterly Review of Biology*, 72(1):31–54, 1997. doi: 10.1086/419657.

[56] S. Nee, R. M. May, and P. H. Harvey. The reconstructed evolutionary process. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 344(1309):305–311, 1994. doi: 10.1098/rstb.1994.0068.

[57] M. M. Norström, M. C. F. Prosperi, R. R. Gray, A. C. Karlsson, and M. Salemi. PhyloTempo: a set of R scripts for assessing and visualizing temporal clustering in genealogies inferred from serially sampled viral sequences. *Evolutionary Bioinformatics*, 8:261–269, 2012. doi: 10.4137/EBO.S9738.

[58] E. Paradis. Time-dependent speciation and extinction from phylogenies: A least squares approach: Temporal dynamics of diversification. *Evolution*, 65(3):661–672, 2010. ISSN 0014-3820. doi: 10.1111/j.1558-5646.2010.01179.x.

[59] E. Paradis. *Enumeration and simulation of random tree topologies*, 2023.

[60] E. Paradis and K. Schliep. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics*, 35(3):526–528, 2018. doi: 10.1093/bioinformatics/bty633.

[61] M. W. Pennell, J. M. Eastman, G. J. Slater, J. W. Brown, J. C. Uyeda, R. G. FitzJohn, M. E. Alfaro, and L. J. Harmon. geiger v2.0: an expanded suite of methods for fitting macroevolutionary models to phylogenetic trees. *Bioinformatics*, 30(15):2216–2218, 2014. doi: 10.1093/bioinformatics/btu181.

[62] I. Pinelis. Evolutionary models of phylogenetic trees. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1522):1425–1431, 2003.

[63] A. Purvis, A. Katzourakis, and P.-M. Agapow. Evaluating phylogenetic tree shape: two modifications to Fusco & Cronk's method. *Journal of Theoretical Biology*, 214(1):99–103, 2002. doi: 10.1006/jtbi.2001.2443.

[64] D. M. Raup, S. J. Gould, T. J. M. Schopf, and D. S. Simberloff. Stochastic models of phylogeny and the evolution of diversity. *The Journal of Geology*, 81(5):525–542, 1973. doi: 10.1086/627905.

[65] J. S. Rogers. Central moments and probability distributions of three measures of phylogenetic tree imbalance. *Systematic Biology*, 45(1):99–110, 1996. doi: 10.1093/sysbio/45.1.99.

[66] D. E. Rosen. Vicariant patterns and historical explanation in biogeography. *Systematic Zoology*, 27(2): 159, 1978. doi: 10.2307/2412970.

[67] N. A. Rosenberg. On the Colijn–Plazzotta numbering scheme for unlabeled binary rooted trees. *Discrete Applied Mathematics*, 291:88–98, 2021. doi: 10.1016/j.dam.2020.11.021.

[68] M. J. Sackin. "Good" and "bad" phenograms. *Systematic Biology*, 21(2):225–226, 1972. doi: 10.1093/sysbio/21.2.225.

[69] C. Semple and M. Steel. *Phylogenetics (Oxford lecture series in mathematics and its applications)*. Oxford University Press, 2003. ISBN 0198509421.

[70] K.-T. Shao and R. R. Sokal. Tree balance. *Systematic Zoology*, 39(3):266, 1990. doi: 10.2307/2992186.

[71] N.J.A. Sloane. Wedderburn-Etherington numbers. `https://oeis.org/A001190`.

[72] J. B. Slowinski. Probabilities of n-trees under two models: A demonstration that asymmetrical interior nodes are not improbable. *Systematic Zoology*, 39(1):89, 1990. ISSN 00397989. doi: 10.2307/2992212.

[73] A. C. Soewongsono, B. R. Holland, and M. M. O'Reilly. The shape of phylogenies under phase-type distributed times to speciation and extinction. *Bulletin of Mathematical Biology*, 84(10), 2022. ISSN 1522-9602. doi: 10.1007/s11538-022-01072-w.

[74] T. Stadler. Inferring speciation and extinction processes from extant species data. *Proceedings of the National Academy of Sciences*, 108(39):16145–16146, 2011. doi: 10.1073/pnas.1113242108.

[75] T. Stadler. Recovering speciation and extinction dynamics based on phylogenies. *Journal of Evolutionary Biology*, 26(6):1203–1219, 2013. doi: 10.1111/jeb.12139.

[76] T. Stadler. *TreeSim: Simulating Phylogenetic Trees*, 2019. URL `https://CRAN.R-project.org/package=TreeSim`. R package version 2.4.

[77] Tanja Stadler. Simulating trees with a fixed number of extant species. *Systematic Biology*, 60(5): 676–684, 2011. doi: 10.1093/sysbio/syr029.

[78] M. Steel. *Phylogeny: Discrete and random processes in evolution.* CBMS-NSF regional conference series in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia PA, 2016. ISBN 161197447X.

[79] M. Steel and A. McKenzie. Properties of phylogenetic trees generated by Yule-type speciation models. *Mathematical Biosciences*, 170(1):91–112, 2001.

[80] L. Takacs. On the total heights of random rooted trees. *Journal of Applied Probability*, 29(3):543–556, 1992. doi: 10.2307/3214892.

[81] L. Takacs. On the total heights of random rooted binary trees. *Journal of Combinatorial Theory, Series B*, 61(2):155–166, 1994. ISSN 0095-8956. doi: 10.1006/jctb.1994.1041.

[82] T. Vasconcelos, B. C. O'Meara, and J. M. Beaulieu. A flexible method for estimating tip diversification rates across a range of speciation and extinction scenarios. *Evolution*, 76(7):1420–1433, 2022. doi: 10.1111/evo.14517.

[83] G. U. Yule. A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F. R. S. *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character*, 213(402-410):21–87, 1925. doi: 10.1098/rstb.1925.0002.