*Bioinformatics*, 36(9), 2020, 2907–2908
doi: 10.1093/bioinformatics/btaa031
Advance Access Publication Date: 17 January 2020
Applications Note



Phylogenetics

# Simulating trees with millions of species

**Stilianos Louca** [1,2,*]

[1]Institute of Ecology and Evolution, University of Oregon, Eugene, OR 97403, USA and [2]Department of Biology, University of Oregon, Eugene, OR 97403, USA

*To whom correspondence should be addressed.

Associate Editor: Russell Schwartz

Received on November 2, 2019; revised on January 8, 2020; editorial decision on January 9, 2020; accepted on January 11, 2020

## Abstract

**Motivation:** The birth-death (BD) model constitutes the theoretical backbone of most phylogenetic tools for reconstructing speciation/extinction dynamics over time. Performing simulations of reconstructed trees (linking extant taxa) under the BD model in backward time, conditioned on the number of species sampled at present day and, in some cases, a specific time interval since the most recent common ancestor (MRCA), is needed for assessing the performance of reconstruction tools, for parametric bootstrapping and for detecting data outliers. The few simulation tools that exist scale poorly to large modern phylogenies, which can comprise thousands or even millions of tips (and rising).

**Results:** Here I present efficient software for simulating reconstructed phylogenies under time-dependent BD models in backward time, conditioned on the number of sampled species and (optionally) on the time since the MRCA. On large trees, my software is 1000–10 000 times faster than existing tools.

**Availability and implementation:** The presented software is incorporated into the R package 'castor', which is available on The Comprehensive R Archive Network (CRAN).

**Contact:** louca.research@gmail.com

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 Introduction

Thanks to modern sequencing technology, massive phylogenies covering hundreds of thousands or even millions of extant species are becoming the new norm (Redelings *et al.*, 2017). These phylogenies present exciting opportunities for reconstructing diversification dynamics through geological time and for discovering macroevolutionary patterns across the tree of life. One of the most important theoretical frameworks for analyzing reconstructed phylogenies is the birth-death (BD) model (Morlon, 2014; Nee *et al.*, 1994). In the BD model, lineages split or go extinct randomly and independently over time according to some probabilistic (Poissonian) speciation or extinction rate, respectively. At present day (when the process halts), a random subset of extant species are sampled and the final phylogeny represents the evolutionary relationships between those sampled species.

Efficient simulations of BD models are needed for evaluating the performance of reconstruction tools, for estimating confidence intervals using bootstraps, for estimating the statistical significance of data outliers and for Approximate Bayesian Computations (Janzen *et al.*, 2015). These tasks often require that trees are simulated backward in time (i.e. with time measured as distance from the present), conditioned on the number of sampled species at the present ($N$) and, in some cases, also on the time elapsed since their most recent common ancestor (MRCA) ($T$), to allow direct comparisons with data or to examine BD models fitted to data. To my knowledge only three backward time BD simulation tools can partly or fully accommodate these conditionings, the R packages 'TESS' (Höhna, 2013) (allows conditioning on $N$ and $T$, but not on $N$ alone), 'TreeSim' (Stadler, 2011) (allows conditioning on $N$, only allows conditioning on both $N$ and $T$ when speciation/extinction rates are constant) and 'ape' (Paradis *et al.*, 2004) (only allows conditioning on $N$, but not both $N$ and $T$). Unfortunately these tools become very inefficient at the scales of large modern datasets. For example, to simulate a single tree with 1 million tips under variable speciation rates and no extinction, TreeSim and TESS require several hours and ape requires about 12 min (details below).

Here I present an efficient computational method for simulating large extant timetrees (also known as reconstructed timetrees; Nee *et al.*, 1994; Stadler, 2011) according to the BD model in backward time, conditional on the number of sampled extant species and (optionally) on the age of their crown (MRCA) or stem. My method also allows simulating trees based on a recently proposed model variable, known as 'pulled speciation rate', which alone fully determines the statistical distribution of the generated trees (Louca and Pennell, 2019). My method, called '`generate_tree_hbd_reverse`', has been incorporated into the R package castor (Louca and Doebeli, 2018).

## 2 Methods

My method is based on the same mathematical calculations (Höhna, 2013; Stadler, 2011) that underly TreeSim, TESS and ape

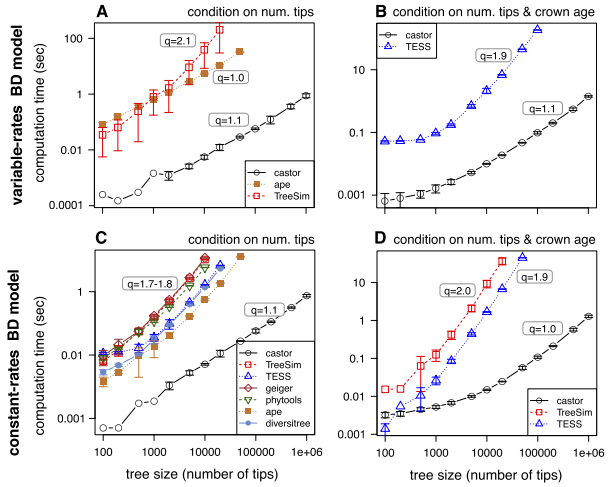


**Fig. 1.** Comparison of time needed to simulate a single timetree, either conditioned on the number of sampled extant species $N$ (left column, sub-figures **A**, **C**), or conditioned on $N$ and the crown age (right column, sub-figures **B**, **D**), under a BD model with time-variable rates (top row) or a BD model with constant rates (bottom row), using castor and other R packages. Curves show mean runtimes across 20 simulations, error bars span 2 standard deviations. Fitted power law exponents (runtime $\propto N^q$) are shown next to every curve. Not all packages support time-variable models or both condition types. Methodological details in Supplementary Material S.1

(mathematical details in Supplementary Material S.2), but generally achieves far superior efficiency thanks to improved code design (details in Supplementary Material S.4, pseudocode in Supplementary Material S.5). Indeed, benchmarks with various BD scenarios and various tree sizes (100–1 000 000 tips) revealed that castor was faster than existing software in nearly all cases tested, and always orders of magnitude faster for large trees (Fig. 1, details in Supplementary Material S.1). For example, to simulate a single tree with 1 million tips, under a model with exponentially decreasing speciation rate, no extinctions and conditioned on the crown age, castor takes about 1 s on a modern laptop. In comparison, TESS would require about 4 h for the same task (based on extrapolation of a fitted power law, Fig. 1); for a similar task (but without conditioning on crown age) TreeSim would require about 194 h and ape about 12 min. An inspection of runtimes reveals that castor achieves a nearly linear scaling in runtime with tree size (power law exponent $q \sim 1.1$). In contrast, TESS and TreeSim exhibit clearly super-linear scaling ($q \sim 1.8$–$2.1$), which explains why they perform so poorly for larger trees. Interestingly, ape also achieves linear scaling, and so is 'only' roughly 1000 times slower than castor regardless of tree size. In the case of constant-rate models, forward time simulators that either halt after a specific time or after reaching a specific number of tips can be used instead of backward simulators; such simulators are easier to implement and thus more abundant than backward simulators. When compared to existing forward simulators, including 'diversitree' (FitzJohn, 2012), 'geiger' (Pennell *et al.*, 2014) and 'phytools' (Revell, 2012), castor is again orders of magnitude faster. For example, to simulate a single tree with 1 million tips under constant speciation rate and no extinction, TreeSim would require about 13 h, TESS about 2.5 h, ape about 40 min, diversitree about 1.6 h, geiger about 14.2 h and phytools about 3.9 h, while castor takes about 1 s (Fig. 1C).

The increased efficiency of castor does not compromise its accuracy (see detailed discussion in Supplementary Material S.6). Rather, in contrast to TESS and TreeSim, castor allows the user to control the balance between integration accuracy and efficiency through designated function arguments.

## 3 Conclusion

The new method presented here allows simulating general (fully time-dependent) BD models in backward time, conditioned on the number of sampled species and optionally the age of their crown or stem, multiple orders of magnitude faster than existing software. Hence, statistical analyses of a broad class of macroevolutionary scenarios become possible at the scales characteristic of modern large datasets, with substantially reduced computational requirements.

## Acknowledgement

I thank Matt W. Pennell for helpful discussions.

## Funding

S.L. was supported by a startup grant by the Department of Biology, University of Oregon.

*Conflict of Interest*: none declared.

## References

FitzJohn,R.G. (2012) Diversitree: comparative phylogenetic analyses of diversification in R. *Methods Ecol. Evol.*, **3**, 1084–1092.

Höhna,S. (2013) Fast simulation of reconstructed phylogenies under global time-dependent birth–death processes. *Bioinformatics*, **29**, 1367–1374.

Janzen,T. *et al.* (2015) Approximate Bayesian Computation of diversification rates from molecular phylogenies: introducing a new efficient summary statistic, the nLTT. *Methods Ecol. Evol.*, **6**, 566–575.

Louca,S. and Doebeli,M. (2018) Efficient comparative phylogenetics on large trees. *Bioinformatics*, **34**, 1053–1055.

Louca,S. and Pennell,M.W. (2019). Phylogenies of extant species are consistent with an infinite array of diversification histories. bioRxiv, doi: 10.1101/719435.

Morlon,H. (2014) Phylogenetic approaches for studying diversification. *Ecol. Lett.*, **17**, 508–525.

Nee,S. *et al.* (1994) The reconstructed evolutionary process. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, **344**, 305–311.

Paradis,E. *et al.* (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, **20**, 289–290.

Pennell,M.W. *et al.* (2014) geiger v2. 0: an expanded suite of methods for fitting macroevolutionary models to phylogenetic trees. *Bioinformatics*, **30**, 2216–2218.

Redelings,B.D. *et al.* (2017) A supertree pipeline for summarizing phylogenetic and taxonomic information for millions of species. *PeerJ*, **5**, e3058.

Revell,L.J. (2012) phytools: an R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217–223.

Stadler,T. (2011) Simulating trees with a fixed number of extant species. *Syst. Biol.*, **60**, 676–684.

# Simulating trees with millions of species
# - Supplementary Information -

Stilianos Louca[1,2]

[1]Institute of Ecology and Evolution, University of Oregon, Eugene, USA
[2]Department of Biology, University of Oregon, Eugene, USA

## S.1 Benchmark details

The following software packages were included in the benchmarks: `TESS` v2.1.0 (Höhna, 2013), `TreeSim` v2.4 (Stadler, 2011), `ape` v5.3 (Paradis *et al.*, 2004), `diversitree` v0.9.11 (FitzJohn, 2012), `phytools` v0.6.99 (Revell, 2012), `geiger` v2.0.6.2 (Pennell *et al.*, 2014), and `castor` v1.5.0 (Louca and Doebeli, 2017). Various types of tree simulations were considered, depending on whether speciation rates were variable or constant over time, and depending on whether the trees were conditioned on the age of the MRCA (trees were always conditioned on the number of sampled extant species). Because for any BD model there exists a pure-birth BD model (i.e., without extinction) with complete species sampling (i.e., sampling fraction 1) that would generate the same probability distribution of trees as the original model (Louca and Pennell, 2019), in our reported benchmarks we focus on pure-birth models with sampling fraction 1 (the computations are similarly complex in the presence of extinctions). For any specific model and for any given tree size, we simulated 20 independent trees with each of the tested tools, and counted the amount of time required separately for each tree; runtimes were then averaged across the 20 repeats. All benchmarks were performed on a MacBook Pro laptop (2.9 GHz Intel Core i5) using a single core.

For the first benchmark (Fig. 1A in the main article) we simulated trees conditioned on the number of sampled extant species ($N$) under a time-variable BD model, in which the speciation rate varied approximately as $\lambda(\tau) = 0.1 + e^{-\tau/10}$, where $\tau$ is "age" (time before present). Because `TreeSim` requires that rates be specified in a piecewise-constant manner (i.e., through rate-shifts at discrete time points), for TreeSim the speciation rate was specified on 10 equidistant time points spanning ages 0 to 10. For comparison purposes, the same specification was used for `castor`, however I mention that castor also supports "smoothly" varying speciation/extinction rates. Trees were generated using the `castor` command `generate_tree_hbd_reverse` (with option "`splines_degree=0`", and the argument `PSR` set to the speciation rate), the `TreeSim` command `sim.rateshift.taxa` (with options "`complete=FALSE, K=0`"), and the `ape` command `rphylo` (with options "`T0=10, fossils=FALSE`").

For the second benchmark (Fig. 1B in the main article) we simulated trees conditioned on the number of sampled extant species ($N$) and the age of the MRCA (fixed at $T = \ln(N)/\lambda(0)$), with the speciation rate being similar to the first benchmark. Trees were generated using the `castor` function `generate_tree_hbd_reverse` (with option "`splines_degree=0`", and the argument `PSR` set to the speciation rate), and the `TESS` function `tess.sim.taxa.age` (with option "`MRCA=TRUE`").

For the third benchmark (Fig. 1C in the main article) we simulated trees conditioned on the number of sampled extant species ($N$), with the speciation rate being set to a constant value $\lambda = 1$. Trees were generated using the `castor` function `generate_tree_hbd_reverse` (with the argument `PSR` set to the

speciation rate), the TESS function `tess.sim.taxa` (with option "`MRCA=FALSE`"), the `TreeSim` function `sim.bd.taxa`, the `geiger` function `sim.bdtree` (with options "`stop='taxa', extinct=TRUE`"), the `phytools` function `pbtree` (with options "`type='continuous', extant.only=TRUE, quiet=TRUE`"), the `ape` function `rphylo` (with option "`fossils=FALSE`"), and the `diversitree` function `trees` (with options "`type='bd', include.extinct=FALSE`").

For the fourth benchmark (Fig. 1D in the main article) we simulated trees conditioned on the number of sampled extant species ($N$) and the age of the MRCA (fixed at $T = \ln(N)/\lambda$), with a constant speciation rate $\lambda = 1$. Trees were generated using the `castor` function `generate_tree_hbd_reverse` (with the argument `PSR` set to the speciation rate), the `TESS` function `tess.sim.taxa.age` (with option "`MRCA=TRUE`"), and the `TreeSim` function `sim.bd.taxa.age` (with option "`mrca=TRUE`").

## S.2 Mathematical background

In the following I provide the mathematical formulas used by `castor` for generating extant timetrees in reverse time, conditioned on the number of extant species $N$ and, optionally, the stem age or crown age (defined here as the root-node's age). These formulas have been largely derived previously by Stadler (2011) and Höhna (2013), and are mentioned here merely for convenience. The following formulas are all expressed in terms of the "pulled speciation rate" $\lambda_\mathrm{p}$ (Louca and Pennell, 2019), which fully encodes the influence of the (potentially time-dependent) speciation rate $\lambda$, extinction rate $\mu$ and present-day sampling fraction $\rho$ on the distribution of generated extant timetrees. Briefly, at any age $\tau$ (time before present) $\lambda_\mathrm{p}(\tau)$ is defined as the speciation rate $\lambda(\tau)$ multiplied by the probability that a single lineage extant at age $\tau$ would survive to the present and be included in the timetree. Equivalently, $\lambda_\mathrm{p}$ can also be defined as the relative slope ($\lambda_\mathrm{p} := -M^{-1}dM/d\tau$) of the lineages-through-time curve $M$ in the limit of infinitely large trees. As elaborated in Supplement S.3, any two birth-death models with identical $\lambda_\mathrm{p}$ will generate the same distribution of extant timetrees. Further, for a pure-birth model (i.e., with zero extinction rate) the $\lambda_\mathrm{p}$ is identical to the speciation rate (Louca and Pennell, 2019). Hence, we can simply use mathematical formulas from the literature that were a priori derived for pure-birth models.

**Conditioning on the stem age:** Suppose that we are given some stem age $T$, a number of sampled extant species $N$, and a pulled speciation rate $\lambda_\mathrm{p}$ as a function of age. We generate random branching ages $0 \leq \tau_1 \leq \tau_2 \leq .. \leq \tau_{N-1} \leq T$, corresponding to splitting events between lineages represented in the tree, as follows. Denote:

$$\Lambda(\tau) := \int_0^\tau \lambda_\mathrm{p}(s)\, ds, \tag{1}$$

and

$$D(\tau) := \lambda_\mathrm{p}(\tau) \cdot e^{-\Lambda(\tau)}. \tag{2}$$

Höhna (2013, Eq. 8 therein) showed that the branching ages are identically and independently distributed (iid) with probability density proportional to $D$ or, equivalently, cumulative distribution function (CDF):

$$F(\tau) := \frac{\int_0^\tau D(s)\, ds}{\int_0^T D(s)\, ds}. \tag{3}$$

Drawing $N-1$ iid branching ages according to the above CDF can be done by first drawing $N-1$ iid numbers $X_1, .., X_{N-1}$ uniformly within $[0,1]$, sorting those numbers in ascending order (where $U_1$,..,$U_{N-1}$ denotes the corresponding order statistic), and assigning $\tau_i := F^{-1}(U_i)$. In `castor`, the $U_i \in [0,1]$ are drawn

directly using an algorithm with time complexity $\mathcal{O}(N)$ to avoid the need for sorting (which has super-linear complexity).

**Conditioning on the crown age:** When conditioning a timetree generated by a pure-birth process on the number of tips $N$ and the crown age $T$, one is essentially sampling two independent pure-birth processes with stem age equal to $T$. For any given combination $(N_1, N_2)$ of final tip counts in the two sub-processes (i.e., any combination satisfying $N_1 \geq 1$, $N_2 \geq 1$ and $N_1 + N_2 = N$), the branching ages in each sub-process are again independently and identically distributed according to the same cumulative distribution function $F$. Hence, one just needs to drawn $N - 2$ iid branching ages according to $F$, and set the last branching age equal to the crown age $T$.

**Not conditioning on stem nor crown age:** When only conditioning on the number of extant tips $N$ but not on the stem or crown age, one must first clarify the probability distribution of the starting age of the birth-death process from which an extant timetree is to be sampled at present-day. Here I take the approach described by Stadler (2011, backward EBDP algorithm): The process is assumed to start at some age $T$ distributed uniformly within some interval $[0, C]$, where $C$ is chosen so large that the probability that an extant timetree older than $C$ will have $N$ tips is negligibly small. Further, since $\lambda_\mathrm{p}$ can be interpreted as the speciation rate of a pure-birth model, we can use the formulas by Stadler (2011) by replacing $\lambda$ with $\lambda_\mathrm{p}$ and $\mu$ with $0$. While Stadler (2011) considers the case where rates shift at discrete time points, her formulas can easily be rewritten to model continuously varying rates by taking the formal limit of infinitely many infinitesimal rate shifts. In that case, Stadler's results imply that the branching ages $0 \leq \tau_1 \leq .. \leq \tau_{N-1}$ can be drawn sequentially using the following algorithm: Upon drawing the first $n-1$ branching ages (where $\tau_0 = 0$), draw $\tau_n \geq \tau_{n-1}$ as the next event time of a non-homogeneous Poisson process with time-dependent rate $(N - n) \cdot \lambda_\mathrm{p}(\tau)$ starting at $\tau = \tau_{n-1}$. According to standard probability theory (Kingman and Frank, 1992), this can be achieved as follows: Draw $\delta_n$ from a rate-1 exponential distribution, and set $\tau_n := \tau_{n-1} + H_M^{-1}(\delta_n)$, where $H_M$ is defined as:

$$H_M(\delta) := \int_0^\delta (N - n) \cdot \lambda_\mathrm{p}(\tau_{n-1} + s)\, ds = (N - n) \cdot [\Lambda(\tau_{n-1} + \delta) - \Lambda(\tau_{n-1})]\,. \tag{4}$$

## S.3 Proof that the distribution of timetrees only depends on $\lambda_\mathrm{p}$

In this section I show that the probability distribution of extant timetrees generated by a birth-death model, conditional upon the number of tips $N$ and, optionally, on the stem or crown age, is fully determined by the pulled speciation rate $\lambda_\mathrm{p}$ (i.e., is the same for all models within a "congruence class"; Louca and Pennell, 2019). It has already been shown previously that this statement holds true for trees generated conditional upon $N$ tips and a specific stem or crown age (Lambert and Stadler, 2013, Louca and Pennell, 2019). I will thus focus on the case where the tree is not conditioned on the stem nor crown age, and instead a prior probability distribution is assumed for the stem or crown age.

**Given a prior distribution for the crown age:** The probability density of a specific tree $\mathcal{T}$, conditional on the crown age $T$ and the number of tips at present $N$, denoted $P(\mathcal{T} \mid T \cap N)$, is invariant within a congruence class (Louca and Pennell, 2019). The same holds for the probability of $N$ conditional upon the crown age, $P(N \mid T)$ (Louca and Pennell, 2019). The probability density of a tree, conditional upon having $N$ tips and under some a priori specified probability distribution of crown ages (with probability density $f(T)$), can be

written as:

$$\begin{aligned} P(\mathcal{T} \mid N) \overset{*}{=} & P(\mathcal{T} \cap T \mid N) \\ = & P(\mathcal{T} \mid T \cap N) P(T \mid N) \\ = & P(\mathcal{T} \mid T \cap N) \frac{P(T \cap N)}{P(N)} \\ = & P(\mathcal{T} \mid T \cap N) \frac{P(N \mid T) f(T)}{P(N)} \\ = & P(\mathcal{T} \mid T \cap N) \frac{P(N \mid T) f(T)}{\int P(N \mid \tau) f(\tau)\, d\tau}. \end{aligned} \tag{5}$$

Note that in step (*) we used the fact that the crown age $T$ is fully determined by (i.e., is part of) the tree random variable $\mathcal{T}$ (since the root-node's age is the crown age). All terms on the right-hand-side of Eq. (5) only depend on the prior probability distribution of crown ages and on quantities that are invariant within a congruence class. Thus, the probability distribution of trees, conditional upon their present-day size and assuming a specific prior probability distribution of crown ages, is invariant within a given congruence class.

**Given a prior distribution for the stem age:** The probability density of a specific tree $\mathcal{T}$, conditional on the stem age $S$ and the number of tips at present $N$, denoted $P(\mathcal{T} \mid S \cap N)$, is invariant within a congruence class (Louca and Pennell, 2019). The same holds for the probability of $N$ conditional upon the stem age, $P(N \mid S)$ (Louca and Pennell, 2019). The probability density of a tree, conditional upon having $N$ tips and under some a priori specified probability density of stem ages (denoted $g(T)$), can be written as:

$$\begin{aligned} P(\mathcal{T} \mid N) \overset{*}{=} & \int P(\mathcal{T} \mid S \cap N) P(S \mid N)\, dS \\ = & \int P(\mathcal{T} \mid S \cap N) \frac{P(S \cap N)}{P(N)}\, dS \\ = & \int P(\mathcal{T} \mid S \cap N) \frac{P(N \mid S) g(S)}{P(N)}\, dS \\ = & \int P(\mathcal{T} \mid S \cap N) \frac{P(N \mid S) g(S)}{\int P(N \mid s) g(s)\, ds}\, dS. \end{aligned} \tag{6}$$

Note that in step (*) we used the "law of total probability" to integrate over all possible stem ages $S$. In contrast to the crown age $T$, the stem age is not fully determined by the tree $\mathcal{T}$. All terms on the right-hand-side of Eq. (6) only depend on the prior probability distribution of stem ages and on quantities that are invariant within a congruence class. Thus, the probability distribution of trees, conditional upon their present-day size and assuming a specific prior probability distribution of stem ages, is invariant within a given congruence class.

## S.4 Keys to improved efficiency

While my method is based on the same formulas (Höhna, 2013, Stadler, 2011) that underly TreeSim, TESS and ape, it achieves its superior efficiency in multiple ways (pseudocode in Supplement S.5). First, to integrate the functions required for the simulation, a symbolic scheme is used instead of a numerical one wherever possible. Specifically, given some function $f$ specified on a discrete age grid and polynomial between grid points, its integral can itself be represented exactly as a piecewise polynomial function with easily computable

coefficients. For example, for a pulled speciation rate $\lambda_{\mathrm{p}}$ specified as a natural splines (polynomial degree 3) on a discrete age grid, the antiderivative $\Lambda(\tau) := \int_0^\tau \lambda_{\mathrm{p}}(s)\ ds$ can again be expressed as a piecewise polynomial of degree 4 whose value can be calculated exactly at any age $\tau$. This permits the use of larger grid steps than normally required for numerical integration for a given targeted accuracy.

Second, I make use of auxiliary data structures that are updated at each coalescence event in constant time, thus permitting nearly linear overall runtimes (i.e., the theoretical optimum), as described in Louca and Doebeli (2017).

Third, when solving the equation $F(\tau_n) = U_n$, where $F$ is a cumulative distribution function, $U_n$ is the $n$-th term in the order statistic of iid uniformly distributed variables in $[0, 1]$ and $\tau_n$ is the $n$-th branching age (Supplement S.2), computation time is reduced by using the fact that $F$ is monotonically non-decreasing in $\tau$ and that $U_1 \leq U_2 \leq ...$ Specifically, the $F$ is only computed once and stored as a piecewise polynomial function on a discrete age grid, and the search for the solution to $F(\tau_n) = U_n$ is restricted to values $\geq \tau_{n-1}$.

Fourth, my method is implemented in C++, a programming language suited for high-performance computing.

## S.5 Pseudocode

The following pseudocode outlines the method presented in the main article, for simulating timetrees under a birth-death model in backward time, conditioned on the number of tips $N$ and on the stem age $T$. If a conditioning on crown age is desired instead of stem age, the algorithm is only slightly modified (fix the last branching age at the desired crown age, then draw the remaining $N - 2$ branching ages conditioned on a stem age equal to the crown age). If neither a conditioning on crown nor stem age is desired, the algorithm also follows a similar logic (formulas in Supplement S.2).

Here, a model congruence class is specified in terms of the pulled speciation rate $\lambda_{\mathrm{p}}$ as a function of age (time before present), which fully determines the probability distribution of generated trees (Louca and Pennell, 2019). In cases where an alternative model parameterization is preferred (for example in terms of the speciation rate $\lambda$, extinction rate $\mu$ and sampling fraction $\rho$), this can simply be transformed into the corresponding $\lambda_{\mathrm{p}}$ prior to using the pseudocode below.

In the following, "age" refers to time before present, bold characters (e.g., $\mathbf{G}$) denote numerical vectors, and double-faced characters (e.g., $\mathbb{E}$) denote 2D matrices. A "piecewise polynomial" function is any function of age that can be described by a polynomial between adjacent age grid points. No assumption is made about continuity or continuity of derivatives. A "splines" function of degree $D \geq 1$ is a piecewise polynomial function with the additional property that its $(D-1)$'th derivative exists and is continuous at the grid points (where the 0-th derivative is the function itself). If of degree 3, `castor` uses "natural" splines, i.e. such that the 2nd derivative at the endpoints is zero.

**Input:** Stem age $T$

**Input:** Age grid $\mathbf{G} = (t_1, t_2, .., t_M)$, satisfying $t_1 \leq 0$, $t_M \geq T$ and $t_i < t_{i+1} \forall i$.

**Input:** Pulled speciation rates $\lambda_{\mathrm{p}}$, provided on $\mathbf{G}$ and assumed to vary as a spline between grid points (the degree can be chosen by the user among 1,2 and 3).

**Input:** Number of tips $N$

**Output:** Ultrametric timetree with $N$ tips and stem age $T$

*# Calculate the cumulative distribution function (CDF) for branching ages*
*# This only needs to be done once for the whole tree*

1: Determine the polynomial coefficients of the splines $\lambda_{\mathrm{p}}$ on $\mathbf{G}$.

2: If needed (i.e., ensuring time steps and differences of $\lambda_{\mathrm{p}}$ between adjacent grid points are below certain thresholds, controlled by the option `relative_dt`), refine the provided grid $\mathbf{G}$ and re-calculate the

coefficients of $\lambda_{\mathrm{p}}$
Exactly calculate the antiderivative $\Lambda(t) := \int_0^t \lambda_{\mathrm{p}}(s)\, ds$ as a piecewise polynomial on $\mathbf{G}$.
Approximate the function $\exp(-\Lambda(t))$ as a piecewise quadratic-polynomial function on $\mathbf{G}$, denoted $E(t)$.
Exactly calculate $D(t) := \lambda_{\mathrm{p}}(t) \cdot E(t)$ as a piecewise polynomial function on $\mathbf{G}$.
Exactly calculate the antiderivative $L(t) := \int_0^t D(s)\, ds$ as piecewise polynomial on $\mathbf{G}$.
Evaluate $L_o \leftarrow L(T)$.
Exactly calculate $F(t) := L(t)/L_o$ as a piecewise polynomial function on $\mathbf{G}$.

*# Draw $N-1$ ordered branching ages $\tau_1 \leq \tau_2 \leq ..$ between $0$ and $T$, using $F$ as a CDF*
Generate the order statistics of $N-1$ uniformly and independently distributed variables in the interval $[0,1]$, denoted $U_1 \leq U_2 \leq ...$ *# Note that this can be achieved in linear time.*
$g_o \leftarrow 1$
**for** $i = 1 : (N-1)$ **do**
    Find maximum $g \geq g_o$ for which $F(\mathbf{G}[g]) <= U_i$,
    **if** $U_i == 1$ **then**
        Set $\tau_i = T$
    **else**
        Solve $F(\tau_i) = U_i$ for $\tau_i$ within the grid interval $\mathbf{G}[g]$ and $\mathbf{G}[g+1]$, analytically or via bisection
    **end if**
    $g_o \leftarrow g$
**end for**

*# Build tree based on $N-1$ ordered branching ages*
Initialize list of "orphan" tips/nodes: $\mathcal{L} \leftarrow \{1, .., N\}$
Pre-allocate edge matrix: $\mathbb{E} \in \mathbb{R}^{(2N-2)\times 2}$
Pre-allocate edge length vector $\mathbf{L} \in \mathbb{R}^{2N-2}$
Pre-allocate vector of tip/node ages: $\mathbf{A} \leftarrow (0, ..., 0) \in \mathbb{R}^{2N-1}$
$k \leftarrow N$ *# number of tips/nodes added so far*
$e \leftarrow 0$ *# number of edges added so far*
**for** $i = 1 : (N-1)$ **do**
    $k \leftarrow k + 1$
    $e \leftarrow e + 1$
    Choose two random orphan tips/nodes to coalesce: $n, m \in \mathcal{L}$
    Create a new orphan node: $\mathcal{L} \leftarrow \mathcal{L} \cup \{k\}$
    $\mathbf{A}[k] \leftarrow \tau_i$
    Add edge connecting $k$ and $m$: $\mathbf{L}[e] \leftarrow \mathbf{A}[k] - \mathbf{A}[m]$, $\mathbb{E}[e,1] \leftarrow k$, $\mathbb{E}[e,2] \leftarrow m$
    $e \leftarrow e + 1$
    Add edge connecting $k$ and $n$: $\mathbf{L}[e] \leftarrow \mathbf{A}[k] - \mathbf{A}[n]$, $\mathbb{E}[e,1] \leftarrow k$, $\mathbb{E}[e,2] \leftarrow n$
    Remove $m$ and $n$ from list of orphans: $\mathcal{L} \leftarrow \mathcal{L} \backslash \{m, n\}$
**end for**
Wrap timetree into "phylo" format, based on $\mathbb{E}$ and $\mathbf{L}$.

## S.6 Accuracy of the simulation code

The increased efficiency of my code, implemented in `castor`, need not imply a reduction of accuracy compared to other implementations, for multiple reasons. First, the underlying mathematical formulas (Supple-

ment S.2) provide an exact means for sampling extant timetrees under the appropriate statistical model, and are the same formulas as used by other implementations such as TreeSim (Stadler, 2011) and TESS (Höhna, 2013). Second, the only notable approximations that `castor` makes are:

- In the specification of $\lambda$ and $\mu$ as piecewise polynomial curves (of degree 0,1,2 or 3) rather than as an abstract function handle, however this approximation is fully user-controlled and the grid can be made arbitrarily fine to capture $\lambda$ and $\mu$ to any desired accuracy.
- Approximating $e^{-\Lambda(\tau)}$ as a piecewise quadratic-polynomial during integration (see pseudocode above), however the integration step is adaptively refined to ensure that the error from this approximation is below a user-controlled threshold.

Note that TreeSim necessarily requires that $\lambda$ and $\mu$ are approximated by piecewise constant functions, and TESS internally approximates various functions as piecewise linear functions (internal function `tess.ode.piecewise`, as of version 2.1.0). Third, and in contrast to `TreeSim` and `TESS`, `castor` allows the user to control the balance between integration accuracy and efficiency through designated function arguments.

To demonstrate the accuracy of my code, I used `castor` and `TESS` (Höhna, 2013) to generate multiple timetrees conditioned on the number of tips and the crown age, and compared the distributions of various tree statistics generated by the two methods. Specifically, trees with $N = 100$ tips were generated randomly under a model where $\lambda = 1$ and $\mu = 0.5$ as well as under a model where $\lambda(\tau) = 0.1 + e^{-\tau/10}$ and $\mu = 0$, where $\tau$ denotes age (time before present). In all cases the sampling fraction was set to $\rho = 1$ and the crown age was set to $\ln(N)/(\lambda(0) - \mu(0))$, and 100,000 trees were generated for each model and using each method (`castor` and `TESS`). For `castor`, $\lambda$ and $\mu$ were specified as piecewise linear functions on an age grid with time step $0.01$, whereas for TESS $\lambda$ and $\mu$ were provided as function handles. Trees were generated using the `castor` function `generate_tree_hbd_reverse` (with option "`splines_degree=1`") and the `TESS` function `tess.sim.taxa.age` (with option "`MRCA=TRUE`"). All other parameters were kept at their default values. For each generated tree, I calculated the $\gamma$-statistic (Pybus and Harvey, 2000) using the `ape` function `gammaStat` as well as pairwise inter-node distances using the `castor` function `get_all_pairwise_distances`. For any given model and method, the distribution density of $\gamma$-statistics across all generated trees was calculated using the R function `density`, with a Gaussian kernel and a smoothing bandwidth equal to $(\gamma_{99} - \gamma_1)/10$, where $\gamma_1$ and $\gamma_{99}$ are the 1% and 99% quantiles of generated $\gamma$-statistics, respectively. For each generated tree, the distribution density of inter-node distances was calculated using the R function `density`, using a Gaussian kernel and a smoothing bandwidth equal to the inverse present-day speciation rate; inter-node distance distribution densities were then averaged over all trees generated using a given model and method. Distribution densities of $\gamma$-statistics and averaged distribution densities of inter-node distances were visually compared between `castor` and `TESS` (Fig. S1). As can be seen in Fig. S1, for the models considered, trees generated by `castor` seem to be distributed virtually identically to those generated by `TESS`, both based on the $\gamma$-statistic and the inter-node distances.

## References


FitzJohn, R. G. (2012). Diversitree: comparative phylogenetic analyses of diversification in R. *Methods Ecol. Evol.*, **3**(6), 1084–1092.

Höhna, S. (2013). Fast simulation of reconstructed phylogenies under global time-dependent birth–death processes. *Bioinformatics*, **29**(11), 1367–1374.

Kingman, C. and Frank, J. (1992). *Poisson Processes*. Clarendon Press.

Lambert, A. and Stadler, T. (2013). Birth–death models and coalescent point processes: The shape and probability of reconstructed phylogenies. *Theor. Popul. Biol.*, **90**, 113–128.

Louca, S. and Doebeli, M. (2017). Efficient comparative phylogenetics on large trees. *Bioinformatics*, **34**(6), 1053–1055.

Louca, S. and Pennell, M. W. (2019). Phylogenies of extant species are consistent with an infinite array of diversification histories. *bioRxiv*. DOI:10.1101/719435.

Paradis, E., Claude, J., and Strimmer, K. (2004). APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, **20**(2), 289–290.

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

Pybus, O. G. and Harvey, P. H. (2000). Testing macro–evolutionary models using incomplete molecular phylogenies. *Proc. R. Soc. Lond. B Biol. Sci.*, **267**(1459), 2267–2272.

Revell, L. J. (2012). phytools: an r package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**(2), 217–223.

Stadler, T. (2011). Simulating trees with a fixed number of extant species. *Syst. Biol.*, **60**(5), 676–684.

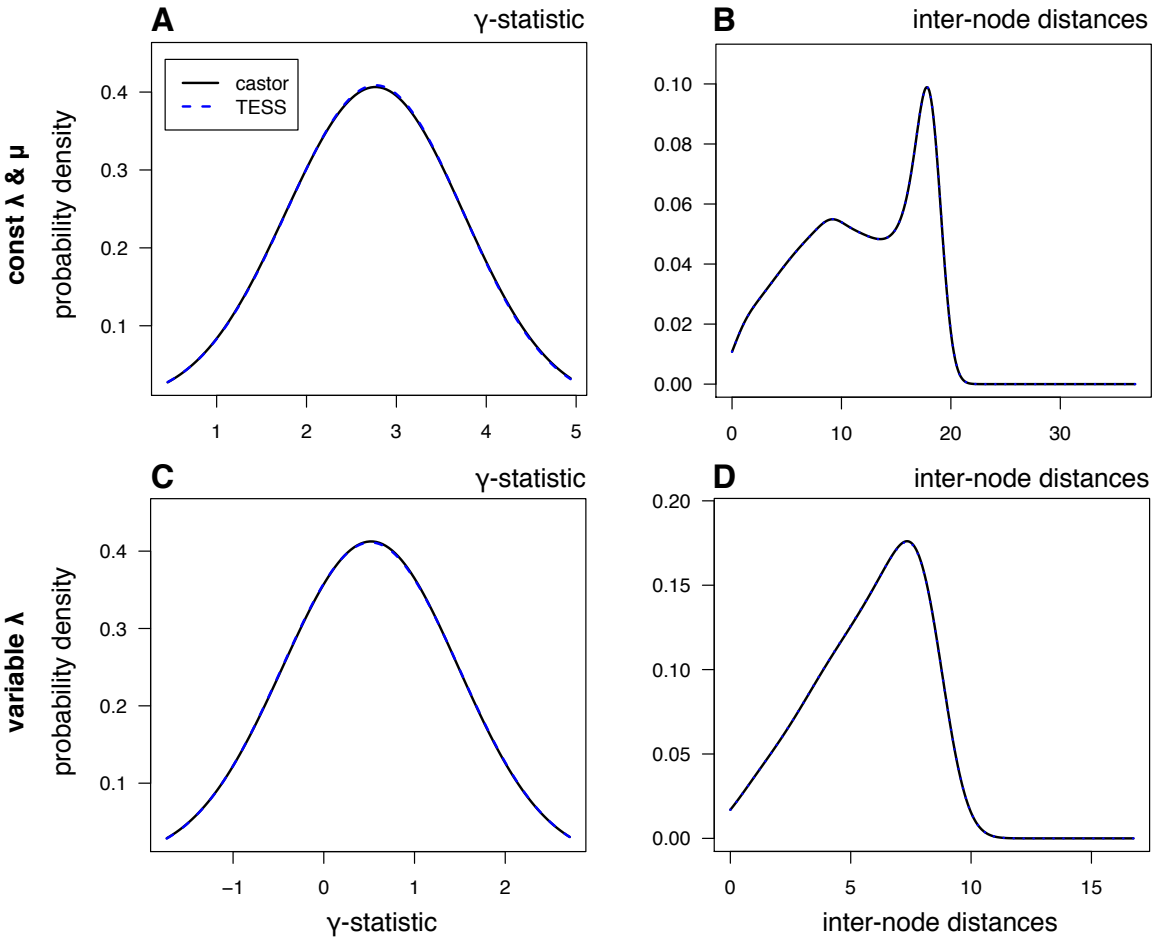


**Figure S1**: **Accuracy of simulated trees.** Comparison of extant timetrees simulated using `castor` and the R package `TESS`, in terms of the distribution of $\gamma$-statistics (Pybus and Harvey, 2000) (left column) and in terms of the distribution of pairwise inter-node distances (right column), for two different birth-death models (top: constant $\lambda$ and $\mu$, bottom: exponentially varying $\lambda$ and zero $\mu$). In A and C each curve represents the probability density of $\gamma$-statistics across all generated trees. In B and D each curve represents the probability density averaged across all generated trees. Note that in all cases the curves corresponding to `castor` and `TESS` largely overlap and are thus nearly indistinguishable. Methodological details are provided in Supplement S.6.