

# Package: treeshapesim (via r-universe)

October 26, 2026

**Title** Simulate Trees with Separately Controlled Imbalance and Branching Times

**Version** 0.2.1

**Description** Simulates rooted, binary, ultrametric trees in which topological imbalance and the distribution of branching times are set by separate parameters. The topology is drawn from Aldous' beta-splitting model and the internode intervals from a pure-birth process in which each of  $k$  lineages splits at rate  $k^c$ . Because the gamma statistic depends only on the internode intervals and the Colless index only on the topology, beta mainly controls imbalance and  $c$  mainly controls gamma.

**License** MIT + file LICENSE

**URL** <https://diogoprov.github.io/treeshapesim/>,  
<https://github.com/diogoprov/treeshapesim>

**BugReports** <https://github.com/diogoprov/treeshapesim/issues>

**Imports** ape, stats

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), treestats

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**X-schema.org-keywords** phylogenetics, phylogenetic-trees, tree-shape, tree-balance, tree-imbalance, gamma-statistic, colless-index, beta-splitting, diversification, simulation, macroevolution, community-phylogenetics

**Repository** <https://diogoprov.r-universe.dev>

**Date/Publication** 2026-10-26 14:00:29 UTC

**RemoteUrl** <https://github.com/diogoprov/treeshapesim>

**RemoteRef** main

**RemoteSha** ac28f2c041d17549d4dce07db0e73a587c0f3f2e

Contents

|                                  |           |
|----------------------------------|-----------|
| assign_ages . . . . .            | 2         |
| calibrate_shape . . . . .        | 3         |
| fit_shape . . . . .              | 4         |
| intervals_lineage_rate . . . . . | 6         |
| rtopology_beta . . . . .         | 6         |
| shape_grid . . . . .             | 7         |
| shape_params . . . . .           | 8         |
| sim_tree_shape . . . . .         | 8         |
| sim_tree_target . . . . .        | 9         |
| tree_shape . . . . .             | 11        |
| <b>Index</b>                     | <b>12</b> |

---

|             |                                                                |
|-------------|----------------------------------------------------------------|
| assign_ages | <i>Assign node ages to a topology from internode intervals</i> |
|-------------|----------------------------------------------------------------|

---

Description

Internal nodes are ranked by a uniform random linear extension of the tree order (the ranking a Yule process implies given the topology): the next node to split is drawn among the available ones with probability proportional to the number of internal nodes in its subtree.

Usage

```
assign_ages(topo, g)
```

Arguments

- topo            Binary, rooted phylo topology.
- g              Internode intervals, as from [intervals\\_lineage\\_rate\(\)](#) (length Ntip(topo) - 1, positive and finite).

Value

topo with branch lengths; ultrametric, with root depth sum(g).

Examples

```
set.seed(1)
assign_ages(rtopology_beta(10), intervals_lineage_rate(10))
```

calibrate\_shape

*Calibrate the shape parameters for a number of tips***Description**

Estimates, for a given number of tips, the mean gamma statistic as a function of  $c$  and the mean normalized Colless index (see `tree_shape()`) as a function of  $\beta$ . Because the topology and the internode intervals are drawn independently, each curve is estimated from its own process alone: gamma from intervals drawn with `intervals_lineage_rate()` and the Colless index from topologies drawn with `rtopology_beta()`, without building trees. The expected values are monotone (gamma increases with  $c$  and the Colless index decreases with  $\beta$ ), but the estimated means need not be, especially for few tips or few draws, so each curve is made monotone by isotonic regression (`stats::isoreg()`) before it is inverted with `shape_params()`.

**Usage**

```
calibrate_shape(
  n,
  beta = c(-1.95, -1.8, -1.6, -1.4, -1.2, -1, -0.75, -0.5, 0, 0.5, 1, 2, 4, 8),
  c = seq(-2, 1, by = 0.25),
  reps = 200
)
```

**Arguments**

|                   |                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <code>n</code>    | Number of tips, a single integer of at least 4 (with 3 tips there is a single topology, so the Colless index cannot be calibrated). |
| <code>beta</code> | Values of $\beta$ at which the Colless index is estimated (at least two).                                                           |
| <code>c</code>    | Values of $c$ at which gamma is estimated (at least two).                                                                           |
| <code>reps</code> | Number of topologies or interval sets per value.                                                                                    |

**Value**

An object of class `shape_calibration`: a list with `n`, `reps` and the data frames `colless` (columns `beta`, `colless` and `colless_mean`) and `gamma` (columns `c`, `gamma` and `gamma_mean`), where the `_mean` columns hold the mean statistic at each value and `colless` and `gamma` their monotone fit, which `shape_params()` inverts.

**Examples**

```
set.seed(1)
cal <- calibrate_shape(50, reps = 50)
cal$gamma
```

fit\_shape

*Estimate the shape parameters of a tree***Description**

Estimates, by maximum likelihood, the beta of Aldous' beta-splitting model from the topology and the c of the lineage-dependent pure-birth process from the branching times, which places an empirical tree on the same scale as the trees of `sim_tree_shape()`. Optionally, a parametric bootstrap with the package's own generators gives percentile intervals.

**Usage**

```
fit_shape(
  tree,
  boot = 0,
  level = 0.95,
  beta_range = c(-1.999, 10),
  c_range = c(-5, 5),
  tol = .Machine$double.eps^0.5
)
```

**Arguments**

|                     |                                                                                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tree                | A binary, rooted, ultrametric phylo object with at least four tips.                                                                                                                                                                                                                   |
| boot                | Number of parametric bootstrap replicates (0 for none).                                                                                                                                                                                                                               |
| level               | Coverage of the bootstrap percentile intervals.                                                                                                                                                                                                                                       |
| beta_range, c_range | Search intervals for beta (lower bound above -2) and c.                                                                                                                                                                                                                               |
| tol                 | Tolerance of the ultrametricity check, as in <code>ape::is.ultrametric()</code> (scaled range of root-to-tip distances). Trees with rounded branch lengths may need a larger value, or can be made ultrametric first (for instance with <code>phytools::force.ultrametric()</code> ). |

**Details**

**beta.** Each internal node with a clade of  $m$  tips split into  $i$  and  $m - i$  tips contributes the probability of that split under the beta-splitting model (see `rtopology_beta()`), and the log-likelihood is the sum over nodes (Aldous 1996; this is the estimator applied by Blum & Francois 2006 and implemented in `maxlik.betasplit()` of the archived package `apTreeshape`). It is maximized over `beta_range`. In the package's validation runs (`data-raw/validate_estimators.R` in the source repository), the estimate was nearly unbiased for beta of -1 or less, but had a positive bias and a large variance for balanced trees with few tips (beta of 0 or more with 50 to 100 tips). An estimate at a bound of `beta_range` only says that the topology is extreme for the model.

**c.** With  $k$  lineages, each lineage splits at rate  $rk^c$ , which is the density-dependent exponential (DDX) model of the R package `laser` (Rabosky 2006), whose per-lineage rate is  $rk^{-x}$ , so that  $c = -x$ . As in `laser`, the splits at 2 to  $n - 1$  lineages are events and the last interval, from the most recent split to the present, is not;  $r$  is profiled out (its maximum likelihood value is  $(n - 2) / \sum_k k^{1+c} g_k$ ,

where  $g_k$  is the time with  $k$  lineages). The estimate describes the shape of the lineage-through-time curve, not its cause: branching times alone do not separate diversity-dependent from time-dependent diversification (Pannetier et al. 2021).

**Bootstrap.** With `boot > 0`, topologies are drawn with `rtopology_beta()` at the estimated beta and intervals with `intervals_lineage_rate()` at the estimated c, and both parameters are re-estimated, following the recommendation of a parametric bootstrap for diversification models by Etienne et al. (2016). The spread of the bootstrap estimates around the point estimate includes the small-sample bias of the estimators, and, for c, a small additional bias because `intervals_lineage_rate()` draws the last interval from the same law, whereas the likelihood treats it as censored. Percentile intervals need a few hundred replicates.

## Value

An object of class `shape_fit`: a list with `n`, `beta`, `c`, `r` (speciation rate per lineage at one lineage, in the time units of the tree), `loglik_beta` (log-likelihood of beta up to an additive constant, so not comparable with other software), `loglik_c` (profile log-likelihood of c, equal to the log-likelihood of laser's DDX model at its maximum), `beta_at_bound` and `c_at_bound` (estimate at a bound of the search interval), `beta_range`, `c_range`, `shape` (output of `tree_shape()`) and, with `boot > 0`, `boot` (data frame of bootstrap estimates) and `ci` (matrix of percentile intervals).

## References

- Aldous, D. (1996). Probability distributions on cladograms. In D. Aldous & R. Pemantle (Eds.), *Random discrete structures* (pp. 1–18). Springer. doi:10.1007/9781461207191\_1
- Blum, M. G. B., & Francois, O. (2006). Which random processes describe the tree of life? A large-scale study of phylogenetic tree imbalance. *Systematic Biology*, 55, 685–691. doi:10.1080/10635150600889625
- Etienne, R. S., Pigot, A. L., & Phillimore, A. B. (2016). How reliably can we infer diversity-dependent diversification from phylogenies? *Methods in Ecology and Evolution*, 7, 1092–1099. doi:10.1111/2041210X.12565
- Pannetier, T., Martinez, C., Bunnefeld, L., & Etienne, R. S. (2021). Branching patterns in phylogenies cannot distinguish diversity-dependent diversification from time-dependent diversification. *Evolution*, 75, 25–38. doi:10.1111/evo.14124
- Rabosky, D. L. (2006). LASER: a maximum likelihood toolkit for detecting temporal shifts in diversification rates from molecular phylogenies. *Evolutionary Bioinformatics*, 2. doi:10.1177/117693430600200024

## Examples

```
set.seed(1)
tr <- sim_tree_shape(200, beta = -1, c = -0.8)
fit_shape(tr)
fit <- fit_shape(tr, boot = 50)
fit$ci
```

---

intervals\_lineage\_rate

*Internode intervals of a pure-birth process with lineage-dependent rates*


---

### Description

With  $k$  lineages each splits at rate  $k^c$ , so the interval with  $k$  lineages is  $\text{Exp}(k^{1+c})$ .  $c = 0$  is Yule;  $c < 0$  slows diversification as lineages accumulate ( $\gamma < 0$ );  $c > 0$  accelerates it ( $\gamma > 0$ ). The final interval ( $n$  lineages, up to the present) is drawn from the same law.

### Usage

```
intervals_lineage_rate(n, c = 0)
```

### Arguments

$n$                       Number of tips, a single integer of at least 2.  
 $c$                         Lineage dependence of the split rate, a single finite number.

### Value

Intervals  $g_2, \dots, g_n$  (length  $n - 1$ ), scaled to sum 1.

### References

Pybus, O. G., & Harvey, P. H. (2000). Testing macro-evolutionary models using incomplete molecular phylogenies. *Proceedings of the Royal Society B*, 267, 2267–2272. doi:10.1098/rspb.2000.1278

### Examples

```
set.seed(1)
intervals_lineage_rate(10, c = -1)
```

---

rtopology\_beta

*Random topology under Aldous' beta-splitting model*


---

### Description

A clade of  $m$  tips splits into  $i$  and  $m - i$  tips with probability proportional to  $\text{Gamma}(\beta + i + 1) \text{Gamma}(\beta + m - i + 1) / (\text{Gamma}(i + 1) \text{Gamma}(m - i + 1))$ .  $\beta = 0$  is Yule,  $\beta = -1.5$  is PDA,  $\beta$  close to  $-2$  is caterpillar-like and large  $\beta$  is balanced.

### Usage

```
rtopology_beta(n, beta = 0)
```

**Arguments**

|      |                                                                                                |
|------|------------------------------------------------------------------------------------------------|
| n    | Number of tips, a single integer of at least 2.                                                |
| beta | Aldous' beta, a single finite number greater than -2. Lower values give more imbalanced trees. |

**Value**

A phylo object without branch lengths, with tips t1, ..., tn.

**References**

Aldous, D. (1996). Probability distributions on cladograms. In D. Aldous & R. Pemantle (Eds.), *Random discrete structures* (pp. 1–18). Springer. doi:10.1007/9781461207191\_1

**Examples**

```
set.seed(1)
rtopology_beta(10, beta = -1)
```

---

|            |                            |
|------------|----------------------------|
| shape_grid | <i>Grid of tree shapes</i> |
|------------|----------------------------|

---

**Description**

Crosses levels of the gamma statistic with levels of the normalized Colless index and finds the beta and c that give each combination on average, as a design for simulation studies. The levels can be chosen, for instance, from quantiles of `tree_shape()` over empirical trees of similar size.

**Usage**

```
shape_grid(n, gamma, colless, calibration = NULL, reps = 200)
```

**Arguments**

|                |                                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------|
| n              | Number of tips, a single integer of at least 2.                                                             |
| gamma, colless | Levels of each statistic.                                                                                   |
| calibration    | Output of <code>calibrate_shape()</code> for the same n; if NULL, it is computed with reps draws per value. |
| reps           | Passed to <code>calibrate_shape()</code> when calibration is NULL.                                          |

**Value**

A data frame with one row per combination of levels (gamma, colless, beta, c), with the calibration in attribute calibration. Each row can be passed to `sim_tree_target()` (targets) or to `sim_tree_shape()` (parameters).

## Examples

```
set.seed(1)
grid <- shape_grid(50, gamma = c(-2, -1, 0), colless = c(0, 0.5, 1), reps = 50)
grid
```

---

|              |                                                                           |
|--------------|---------------------------------------------------------------------------|
| shape_params | <i>Shape parameters that give target mean values of gamma and Colless</i> |
|--------------|---------------------------------------------------------------------------|

---

## Description

Inverts the monotone curves of [calibrate\\_shape\(\)](#) by linear interpolation. Where a curve is flat (tied values of its monotone fit), the parameter is the mean of the tied values.

## Usage

```
shape_params(calibration, gamma, colless)
```

## Arguments

|                |                                                                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| calibration    | Output of <a href="#">calibrate_shape()</a> .                                                                                                                                    |
| gamma, colless | Target mean gamma statistic and normalized Colless index (see <a href="#">tree_shape()</a> ); vectors that are recycled to a common length, as in <a href="#">data.frame()</a> . |

## Value

A data frame with the targets and the corresponding beta and c.

## Examples

```
set.seed(1)
cal <- calibrate_shape(50, reps = 50)
shape_params(cal, gamma = c(0, -2), colless = 0.5)
```

---

|                |                                                                             |
|----------------|-----------------------------------------------------------------------------|
| sim_tree_shape | <i>Ultrametric trees with given imbalance and branching-time parameters</i> |
|----------------|-----------------------------------------------------------------------------|

---

## Description

Draws the topology with [rtopology\\_beta\(\)](#) and the internode intervals with [intervals\\_lineage\\_rate\(\)](#), and combines them with [assign\\_ages\(\)](#). The topology and the intervals are drawn independently, the gamma statistic depends only on the intervals and the Colless index only on the topology, so for a given n the realized gamma depends only on c and the realized Colless index only on beta. To draw trees near target values of both statistics, see [sim\\_tree\\_target\(\)](#).

**Usage**

```
sim_tree_shape(n, beta = 0, c = 0, nsim = 1, root_age = 1)
```

**Arguments**

|          |                                                                                                |
|----------|------------------------------------------------------------------------------------------------|
| n        | Number of tips, a single integer of at least 2.                                                |
| beta     | Aldous' beta, a single finite number greater than -2. Lower values give more imbalanced trees. |
| c        | Lineage dependence of the split rate, a single finite number.                                  |
| nsim     | Number of trees.                                                                               |
| root_age | Age of the root (positive).                                                                    |

**Value**

With `nsim = 1`, a binary, ultrametric phylo object with root depth `root_age`; otherwise a `multiPhylo` object with `nsim` such trees.

**Reproducibility**

The same seed gives the same trees on the same platform. Between platforms (for instance, arm64 and x86-64), differences in the last bit of floating-point functions can change a few draws of the topology, so trees may differ.

**Examples**

```
set.seed(1)
tr <- sim_tree_shape(50, beta = -1, c = -1)
ape::gammaStat(tr)
trees <- sim_tree_shape(50, beta = -1, c = -1, nsim = 10, root_age = 30)
```

---

|                 |                                                               |
|-----------------|---------------------------------------------------------------|
| sim_tree_target | <i>Simulate trees near target values of gamma and Colless</i> |
|-----------------|---------------------------------------------------------------|

---

**Description**

Draws trees whose gamma statistic and normalized Colless index (see [tree\\_shape\(\)](#)) fall within a tolerance of target values. The parameters `beta` and `c` whose mean statistics equal the targets are found with [calibrate\\_shape\(\)](#) and [shape\\_params\(\)](#). Topologies are then drawn with [rtopology\\_beta\(\)](#) and kept if their Colless index is within `tol` of the target, interval sets are drawn with [intervals\\_lineage\\_rate\(\)](#) and kept if their gamma is within `tol` of the target, and the kept topologies and interval sets are combined with [assign\\_ages\(\)](#). Because the topology and the intervals are independent, rejecting them separately gives the same distribution of trees as rejecting whole trees on both statistics, with a much higher acceptance rate. Random numbers are used in a different order than in [sim\\_tree\\_shape\(\)](#), so the same seed does not give the same trees as that function.

**Usage**

```
sim_tree_target(
  n,
  gamma,
  colless,
  nsim = 1,
  tol = 0.5,
  calibration = NULL,
  reps = 200,
  max_tries = 1000 * nsim,
  root_age = 1
)
```

**Arguments**

|                             |                                                                                                                                                      |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>n</code>              | Number of tips, a single integer of at least 2.                                                                                                      |
| <code>gamma, colless</code> | Target gamma statistic and normalized Colless index (single values).                                                                                 |
| <code>nsim</code>           | Number of trees.                                                                                                                                     |
| <code>tol</code>            | Tolerance around the targets: one value for both statistics, or two values for gamma and Colless, in this order.                                     |
| <code>calibration</code>    | Output of <code>calibrate_shape()</code> for the same <code>n</code> ; if <code>NULL</code> , it is computed with <code>reps</code> draws per value. |
| <code>reps</code>           | Passed to <code>calibrate_shape()</code> when <code>calibration</code> is <code>NULL</code> .                                                        |
| <code>max_tries</code>      | Maximum number of topologies, and of interval sets, drawn.                                                                                           |
| <code>root_age</code>       | Age of the root.                                                                                                                                     |

**Value**

With `nsim = 1`, a phylo object; otherwise a multiPhylo object (also when fewer trees than `nsim` are found). Attributes: `shape` (realized gamma and Colless of each tree), `params` (beta and c used) and `acceptance` (proportion of topologies and of interval sets kept). If fewer than `nsim` trees are found within `max_tries`, the trees found are returned with a warning.

**Examples**

```
set.seed(1)
cal <- calibrate_shape(50, reps = 50)
trees <- sim_tree_target(50, gamma = -1, colless = 0.5, nsim = 5,
  calibration = cal)
attr(trees, "shape")
```

tree\_shape

*Gamma statistic and Colless index of a tree***Description**

Computes Pybus and Harvey's (2000) gamma statistic, with `ape::gammaStat()`, and the Colless index normalized against its expectation under the Yule model,  $(I_C - n \log n - n(\gamma_E - 1 - \log 2))/n$ , where  $n$  is the number of tips and  $\gamma_E$  is Euler's constant (the "yule" normalization of `treestats::colless()`). Both are near zero for Yule trees: gamma is negative when branching times are concentrated near the root, and the normalized Colless index is positive for trees more imbalanced than Yule trees.

**Usage**

```
tree_shape(tree)
```

**Arguments**

**tree**                      A binary, rooted phylo object with at least three tips. The gamma statistic assumes an ultrametric tree; this is not checked.

**Value**

A named vector with gamma and colless.

**References**

Pybus, O. G., & Harvey, P. H. (2000). Testing macro-evolutionary models using incomplete molecular phylogenies. *Proceedings of the Royal Society B*, 267, 2267–2272. doi:10.1098/rspb.2000.1278

**Examples**

```
set.seed(1)
tree_shape(sim_tree_shape(100, beta = -1, c = -1))
```

# Index

`ape::gammaStat()`, [11](#)  
`ape::is.ultrametric()`, [4](#)  
`assign_ages`, [2](#)  
`assign_ages()`, [8](#), [9](#)  
  
`calibrate_shape`, [3](#)  
`calibrate_shape()`, [7–10](#)  
  
`data.frame()`, [8](#)  
  
`fit_shape`, [4](#)  
  
`intervals_lineage_rate`, [6](#)  
`intervals_lineage_rate()`, [2](#), [3](#), [5](#), [8](#), [9](#)  
  
`rtopology_beta`, [6](#)  
`rtopology_beta()`, [3–5](#), [8](#), [9](#)  
  
`shape_grid`, [7](#)  
`shape_params`, [8](#)  
`shape_params()`, [3](#), [9](#)  
`sim_tree_shape`, [8](#)  
`sim_tree_shape()`, [4](#), [7](#), [9](#)  
`sim_tree_target`, [9](#)  
`sim_tree_target()`, [7](#), [8](#)  
`stats::isoreg()`, [3](#)  
  
`tree_shape`, [11](#)  
`tree_shape()`, [3](#), [5](#), [7–9](#)