

# Package ‘treestats’

January 30, 2024

**Type** Package

**Title** Phylogenetic Tree Statistics

**Version** 1.0.5

**Maintainer** Thijs Janzen <thijsjanzen@gmail.com>

**Description** Collection of phylogenetic tree statistics, collected throughout the literature. All functions have been written to maximize computation speed. The package includes umbrella functions to calculate all statistics, all balance associated statistics, or all branching time related statistics. Furthermore, the 'treestats' package supports summary statistic calculations on Ltables, provides speed-improved coding of branching times, Ltable conversion and includes algorithms to create intermediately balanced trees. Full description can be found in Janzen (2023) <[doi:10.1101/2024.01.24.576848](https://doi.org/10.1101/2024.01.24.576848)>.

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Depends** Rcpp

**LinkingTo** Rcpp, nloptr

**Imports** ape, nloptr, RSpectra, Matrix

**Suggests** phyloTop, testthat, DDD, geiger, nodeSub, nLTT, castor, adephylo, ggplot2, tibble, picante, treebalance, RPANDA, lintr, rmarkdown, knitr

**BugReports** <https://github.com/thijsjanzen/treestats/issues>

**URL** <https://github.com/thijsjanzen/treestats>

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Thijs Janzen [cre, aut] (<<https://orcid.org/0000-0002-4162-1140>>)

**Repository** CRAN

**Date/Publication** 2024-01-30 15:50:02 UTC

**R topics documented:**

|                        |    |
|------------------------|----|
| treestats-package      | 3  |
| area_per_pair          | 4  |
| average_leaf_depth     | 4  |
| avg_ladder             | 5  |
| b1                     | 6  |
| b2                     | 6  |
| beta_statistic         | 7  |
| blum                   | 8  |
| branching_times        | 9  |
| calc_all_stats         | 9  |
| calc_balance_stats     | 11 |
| calc_brts_stats        | 12 |
| cherries               | 13 |
| colless                | 14 |
| crown_age              | 15 |
| diameter               | 15 |
| eigen_vector           | 16 |
| entropy_j              | 16 |
| ew_colless             | 17 |
| gamma_statistic        | 18 |
| ILnumber               | 18 |
| imbalance_steps        | 19 |
| j_one                  | 20 |
| laplacian_spectrum     | 20 |
| list_statistics        | 21 |
| ltable_to_newick       | 22 |
| l_to_phylo             | 22 |
| make_unbalanced_tree   | 23 |
| max_betweenness        | 24 |
| max_closeness          | 24 |
| max_del_width          | 25 |
| max_depth              | 26 |
| max_ladder             | 26 |
| max_width              | 27 |
| mean_branch_length     | 27 |
| mean_branch_length_ext | 28 |
| mean_branch_length_int | 28 |
| mean_i                 | 29 |
| mean_pair_dist         | 29 |
| mntd                   | 30 |
| nLTT                   | 31 |
| nLTT_base              | 32 |
| number_of_lineages     | 32 |
| phylogenetic_diversity | 33 |
| phylo_to_l             | 34 |
| pigot_rho              | 34 |

|                                 |           |
|---------------------------------|-----------|
| pitchforks . . . . .            | 35        |
| prep_lapl_spec . . . . .        | 36        |
| psv . . . . .                   | 36        |
| rebase_ltable . . . . .         | 37        |
| rogers . . . . .                | 37        |
| rquartet . . . . .              | 38        |
| sackin . . . . .                | 39        |
| stairs . . . . .                | 39        |
| stairs2 . . . . .               | 40        |
| sym_nodes . . . . .             | 41        |
| tot_coph . . . . .              | 42        |
| tree_height . . . . .           | 42        |
| var_branch_length . . . . .     | 43        |
| var_branch_length_ext . . . . . | 43        |
| var_branch_length_int . . . . . | 44        |
| var_leaf_depth . . . . .        | 44        |
| var_pair_dist . . . . .         | 45        |
| wiener . . . . .                | 45        |
| <b>Index</b>                    | <b>47</b> |

---

|                   |   |
|-------------------|---|
| treestats-package | <i>Collection of phylogenetic tree statistics</i> |
|-------------------|---|

---

## Description

The 'treestats' package contains a collection of phylogenetic tree statistics, implemented in C++ to ensure high speed.

## Details

Given a phylogenetic tree as a phylo object, the 'treestats' package provides a wide range of individual functions returning the relevant statistic. In addition, there are three functions available that calculate a collection of statistics at once: `calc_all_statistics` (which calculates all currently implemented statistics of treestats), `calc_balance_stats`, which calculates all (im)balance related statistics and `calc_brts_stats`, which calculates all branching times and branch length related statistics. Furthermore, there are a number of additional tools available that allow for phylogenetic tree manipulation: `make_unbalanced_tree`, which creates an imbalanced tree in a stepwise fashion. Then there are two functions related to conversion from and to an `ltable`, an alternative notation method used in some simulations. These are `l_to_phylo` which is a C++ based version of `DDD::L2phylo`, which converts an `ltable` to a phylo object, and `phylo_to_l`, which is a C+ based version of `DDD::phylo2L`, which converts a phylo object to an `ltable`. Lastly, the treestats package also includes a faster, C++ based, implementation of `ape::branching.times` (the function `branching_times`), which yields the same sequence of branching times, but omits the branching names in favour of speed.

## Author(s)

Maintainer: Thijs Janzen <thijsjanzen@gmail.com>

**References**

Phylogenetic tree statistics: a systematic overview using the new R package 'treestats' Thijs Janzen, Rampal S. Etienne bioRxiv 2024.01.24.576848; doi: <https://doi.org/10.1101/2024.01.24.576848>

---

|               |   |
|---------------|---|
| area_per_pair | <i>Fast function using C++ to calculate the area per pair index</i> |
|---------------|---|

---

**Description**

The area per pair index calculates the sum of the number of edges on the path between all two leaves. Instead, the area per pair index (APP) can also be derived from the Sackin (S) and total cophenetic index (TC):  $APP = \frac{2}{n} \cdot S - \frac{4}{n(n-1)} \cdot TC$   $APP = 2/n * S - 4/(n(n-1)) * TC$

**Usage**

```
area_per_pair(phy, normalization = "none")
```

**Arguments**

|               |   |
|---------------|---|
| phy           | phylo object or ltable  |
| normalization | "none" or "yule", in which case the acquired result is divided by the expectation for the Yule model. |

**Value**

Total cophenetic index

**References**

T. Araújo Lima, F. M. D. Marquitti, and M. A. M. de Aguiar. Measuring Tree Balance with Normalized Tree Area. arXiv e-prints, art. arXiv:2008.12867, 2020.

---

|                    |  |
|--------------------|--|
| average_leaf_depth | <i>Calculate the average leaf depth statistic. The average leaf depth statistic is a normalized version of the Sackin index, normalized by the number of tips.</i> |
|--------------------|--|

---

**Description**

Calculate the average leaf depth statistic. The average leaf depth statistic is a normalized version of the Sackin index, normalized by the number of tips.

**Usage**

```
average_leaf_depth(phy, normalization = "none")
```

**Arguments**

phy                    phylo object or ltable  
 normalization      "none" or "yule", in which case the statistic is divided by the expectation under the yule model, following Remark 1 in Coronado et al. 2020.

**Value**

average leaf depth statistic

**References**

M. Coronado, T., Mir, A., Rosselló, F. et al. On Sackin's original proposal: the variance of the leaves' depths as a phylogenetic balance index. *BMC Bioinformatics* 21, 154 (2020). <https://doi.org/10.1186/s12859-020-3405-1>  
 K.-T. Shao and R. R. Sokal. Tree balance. *Systematic Zoology*, 39(3):266, 1990. doi: 10.2307/2992186.

**Examples**

```
simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
average_leaf_depth(simulated_tree)
```

---

|            |                             |
|------------|-----------------------------|
| avg_ladder | <i>Average ladder index</i> |
|------------|-----------------------------|

---

**Description**

Calculate the avgLadder index, from the phyloTop package. Higher values indicate more unbalanced trees. To calculate the average ladder index, first all potential ladders in the tree are calculated. A ladder is defined as a sequence of nodes where one of the daughter branches is a terminal branch, resulting in a 'ladder' like pattern. The average ladder index then represents the average length across all observed ladders in the tree.

**Usage**

```
avg_ladder(input_obj)
```

**Arguments**

input\_obj            phylo object or ltable

**Value**

average number of ladders

b1

*Fast function using C++ to calculate the B1 metric***Description**

Balance metric (in the case of a binary tree), which measures the sum across all internal nodes of one over the maximum depth of all attached tips to that node.

**Usage**

```
b1(phy, normalization = "none")
```

**Arguments**

|               |  |
|---------------|--|
| phy           | phylo object or ltable   |
| normalization | "none" or "tips", in which case the resulting statistic is divided by the number of tips in the tree, as a crude way of normalization. |

**Value**

B1 statistic

**References**

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

b2

*Fast function using C++ to calculate the B2 metric***Description**

Balance metric that uses the Shannon-Wiener statistic of information content. The b2 measure is given by the sum over the depths of all tips, divided by  $2^{\text{depth}}$ :  $\sum N_i / 2^{\text{depth}}$

**Usage**

```
b2(phy, normalization = "none")
```

**Arguments**

|               |   |
|---------------|---|
| phy           | phylo object or ltable  |
| normalization | "none" or "yule", when "yule" is chosen, the statistic is divided by the Yule expectation, following from theorem 3.7 in Bienvenu 2020. |

**Value**

Maximum depth (in number of edges)

**References**

- K.-T. Shao and R. R. Sokal. Tree Balance. *Systematic Zoology*, 39(3):266, 1990. doi: 10.2307/2992186.
- Biennu, François, Gabriel Cardona, and Celine Scornavacca. "Revisiting Shao and Sokal's  $B_2$  index of phylogenetic balance." *Journal of Mathematical Biology* 83.5 (2021): 1-43.

---

beta\_statistic

*Fast function using C++ to calculate the Aldous beta statistic.*

---

**Description**

The Beta statistic fits a beta splitting model to each node, assuming that the number of extant descendants of each daughter branch is split following a beta distribution, such that the number of extant descendants  $x$  and  $y$  at a node follows  $q(x, y) = s_n(beta)^{-1} \frac{\text{gamma}(x+1+beta)\text{gamma}(y+1+beta)}{\text{gamma}(x+1)\text{gamma}(y+1)}$ , where  $s_n(beta)^{-1}$  is a normalizing constant. When this model is fit to a tree, different values of beta correspond to the expectation following from different diversification models, such that a beta of 0 corresponds to a Yule tree, a beta of  $-3/2$  to a tree following from a PDA model. In general, negative beta values correspond to trees more unbalanced than Yule trees, and beta values larger than zero indicate trees more balanced than Yule trees. The lower bound of the beta splitting parameter is  $-2$ .

**Usage**

```
beta_statistic(
  phy,
  upper_lim = 10,
  algorithm = "COBYLA",
  abs_tol = 1e-04,
  rel_tol = 1e-06
)
```

**Arguments**

|           |  |
|-----------|--|
| phy       | phylogeny or Itable  |
| upper_lim | Upper limit for beta parameter, default = 10.  |
| algorithm | optimization algorithm used, default is "COBYLA" (Constrained Optimization BY Linear Approximations), also available are "subplex" and "simplex". Subplex and Simplex seem to have difficulties with unbalanced trees, e.g. if beta < 0. |
| abs_tol   | absolute stopping criterion of optimization. Default is 1e-4.  |
| rel_tol   | relative stopping criterion of optimization. Default is 1e-6.  |

**Value**

Beta value

**References**

Aldous, David. "Probability distributions on cladograms." Random discrete structures. Springer, New York, NY, 1996. 1-18. Jones, Graham R. "Tree models for macroevolution and phylogenetic analysis." Systematic biology 60.6 (2011): 735-746.

**Examples**

```
simulated_tree <- ape::rphylo(n = 100, birth = 1, death = 0)
if (requireNamespace("nodeSub")) {
  brts <- branching_times(simulated_tree)
  balanced_tree <- nodeSub::create_balanced_tree(brts)
  unbalanced_tree <- nodeSub::create_unbalanced_tree(brts)
  beta_statistic(balanced_tree) # should be approximately 10
  beta_statistic(simulated_tree) # should be near 0
  beta_statistic(unbalanced_tree) # should be approximately -2
}
```

---

blum

*Fast function using C++ to calculate the Blum index of (im)balance.*

---

**Description**

The Blum index of imbalance (also known as the s-shape statistic) calculates the sum of  $\log(N - 1)$  over all internal nodes, where N represents the total number of extant tips connected to that node. An alternative implementation can be found in the Castor R package.

**Usage**

```
blum(phy, normalization = FALSE)
```

**Arguments**

|               |   |
|---------------|---|
| phy           | phylogeny or Itable   |
| normalization | because the Blum index sums over all nodes, the resulting statistic tends to be correlated with the number of extant tips. Normalization can be performed by dividing by the number of extant tips. |

**Value**

Blum index of imbalance

**References**

M. G. B. Blum and O. Francois (2006). Which random processes describe the Tree of Life? A large-scale study of phylogenetic tree imbalance. Systematic Biology. 55:685-691.



**Examples**

```

simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
brts <- branching_times(simulated_tree)
if (requireNamespace("nodeSub")) {
  balanced_tree <- nodeSub::create_balanced_tree(brts)
  unbalanced_tree <- nodeSub::create_unbalanced_tree(brts)
  blum(balanced_tree)
  blum(unbalanced_tree) # should be higher
}

```

---

|                 |  |
|-----------------|--|
| branching_times | <i>Calculates branching times of a tree, using C++</i> |
|-----------------|--|

---

**Description**

C++ based alternative to ‘ape::branching.times’, please note that to maximise speed, ‘treestats::branching\_times’ does not return node names associated to the branching times, in contrast to the ape version.

**Usage**

```
branching_times(phy)
```

**Arguments**

phy                    phylo object or ltable

**Value**

vector of branching times

---

|                |   |
|----------------|---|
| calc_all_stats | <i>function to apply all available tree statistics to a single tree</i> |
|----------------|---|

---

**Description**

this function applies all tree statistics available in this package to a single tree, being:

- gamma
- Sackin
- Colless
- Aldous’ beta statistic
- Blum
- crown age
- tree height

- Pigot's rho
- number of lineages
- nLTT with empty tree
- phylogenetic diversity
- avgLadder index
- cherries
- ILnumber
- pitchforks
- stairs
- stairs2
- laplacian spectrum
- B1
- B2
- area per pair (aPP)
- average leaf depth (aLD)
- I statistic
- ewColless
- max Delta Width (maxDelW)
- maximum of Depth
- variance of Depth
- maximum Width
- Rogers
- total Cophenetic distance
- symmetry Nodes
- mean of pairwise distance (mpd)
- variance of pairwise distance (vpd)
- Phylogenetic Species Variability (psv)
- mean nearest taxon distance (mntd)
- J statistic of entropy
- rquartet index
- Wiener index
- max betweenness
- max closeness
- diameter, without branch lengths
- maximum eigen vector value
- mean branch length
- variance of branch length

- mean external branch length
- variance of external branch length
- mean internal branch length
- variance of internal branch length
- number of imbalancing steps
- j\_one statistic

For the Laplacian spectrum properties, four properties of the eigenvalue distribution are returned: 1) asymmetry, 2) peakedness, 3) log(principal eigenvalue) and 4) eigengap. Please notice that for some very small or very large trees, some of the statistics can not be calculated. The function will report an NA for this statistic, but will not break, to facilitate batch analysis of large numbers of trees.

### Usage

```
calc_all_stats(phylo, normalize = FALSE)
```

### Arguments

|           |  |
|-----------|--|
| phylo     | phylo object   |
| normalize | if set to TRUE, results are normalized (if possible) under either the Yule expectation (if available), or the number of tips |

### Value

List with statistics

---

calc\_balance\_stats     *function to apply all balance statistics to a single tree*

---

### Description

this function applies all tree statistics available in this package to a single tree, being:

- Sackin
- Colless
- Aldous' beta statistic
- Blum
- Average Ladder Size
- cherries
- ILnumber
- pitchforks
- stairs

- stairs2
- B1
- B2
- area per pair (aPP)
- average leaf depth (aLD)
- I statistic
- ewColless
- max Delta Width (maxDelW)
- maximum of Depth
- variance of Depth
- maximum Width
- Rogers
- total Cophenetic distance
- symmetry Nodes
- rquartet index
- j\_one statistic

### Usage

```
calc_balance_stats(phylo, normalize = FALSE)
```

### Arguments

|           |  |
|-----------|--|
| phylo     | phylo object   |
| normalize | if set to TRUE, results are normalized (if possible) under either the Yule expectation (if available), or the number of tips |

### Value

list with statistics

---

|                 |   |
|-----------------|---|
| calc_brts_stats | <i>function to apply all tree statistics related to branching times to a single tree.</i> |
|-----------------|---|

---

**Description**

this function applies all tree statistics based on branching times to a single tree, being:

- gamma
- pigot's rho
- mean branch length
- nLTT with empty tree
- var branch length
- mean internal branch length
- mean external branch length
- var internal branch length
- var external branch length

**Usage**

```
calc_brts_stats(phylo)
```

**Arguments**

phylo            phylo object

**Value**

list with statistics

---

|          |                     |
|----------|---------------------|
| cherries | <i>Cherry index</i> |
|----------|---------------------|

---

**Description**

Calculate the number of cherries, from the phyloTop package. A cherry is a pair of sister tips.

**Usage**

```
cherries(input_obj, normalization = "none")
```

**Arguments**

input\_obj        phylo object or ltable  
normalization    "none", "yule", or "pda", the found number of cherries is divided by the expected number, following McKenzie & Steel 2000.

**Value**

number of cherries

## References

McKenzie, Andy, and Mike Steel. "Distributions of cherries for two models of trees." *Mathematical biosciences* 164.1 (2000): 81-92.

---

colless

*Fast function using C++ to calculate the Colless index of (im)balance.*

---

## Description

The Colless index is calculated as the sum of  $abs(L - R)$  over all nodes, where L (or R) is the number of extant tips associated with the L (or R) daughter branch at that node. Higher values indicate higher imbalance. Two normalizations are available, where a correction is made for tree size, under either a yule expectation, or a pda expectation.

## Usage

```
colless(phy, normalization = "none")
```

## Arguments

|               |   |
|---------------|---|
| phy           | phylo object or ltable  |
| normalization | A character string equals to "none" (default) for no normalization or one of "pda" or "yule". |

## Value

colless index

## References

Colless D H. 1982. Review of: Phylogenetics: The Theory and Practice of Phylogenetic Systematics. *Systematic Zoology* 31:100-104.

## Examples

```
simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
brts <- branching_times(simulated_tree)
if (requireNamespace("nodeSub")) {
  balanced_tree <- nodeSub::create_balanced_tree(brts)
  unbalanced_tree <- nodeSub::create_unbalanced_tree(brts)
  colless(balanced_tree)
  colless(unbalanced_tree) # should be higher
}
```

---

|           |  |
|-----------|--|
| crown_age | <i>Calculates the crown age of a tree using C++.</i> |
|-----------|--|

---

**Description**

In a reconstructed tree, obtaining the crown age is fairly straightforward, and the function `beautier::get_crown_age` does a great job at it. However, in a non-ultrametric tree, that function no longer works. This function provides a functioning alternative

**Usage**

```
crown_age(phy)
```

**Arguments**

|     |                        |
|-----|------------------------|
| phy | phylo object or ltable |
|-----|------------------------|

**Value**

crown age

---

|          |  |
|----------|--|
| diameter | <i>Fast function using C++ to calculate the diameter</i> |
|----------|--|

---

**Description**

The Diameter of a tree is defined as the maximum length of a shortest path. When taking branch lengths into account, this is equal to twice the crown age.

**Usage**

```
diameter(phy, weight = FALSE)
```

**Arguments**

|        |                               |
|--------|-------------------------------|
| phy    | phylo object or ltable        |
| weight | if TRUE, uses branch lengths. |

**Value**

Diameter

**References**

Chindelevitch, Leonid, et al. "Network science inspires novel tree shape statistics." *PloS one* 16.12 (2021): e0259877.

---

|              |   |
|--------------|---|
| eigen_vector | <i>Fast function using C++ to calculate eigen vector centrality</i> |
|--------------|---|

---

**Description**

Eigen vector centrality associates with each node  $v$  the positive value  $e(v)$ , such that:  $\sum_u w(uv) * e(u) = \lambda * e(v)$ . Thus,  $e(v)$  is the Perron-Frobenius eigenvector of the adjacency matrix of the tree.

**Usage**

```
eigen_vector(phy, weight = TRUE, scale = FALSE)
```

**Arguments**

|        |                                      |
|--------|--------------------------------------|
| phy    | phylo object or ltable               |
| weight | if TRUE, uses branch lengths.        |
| scale  | if TRUE, the eigenvector is rescaled |

**Value**

List with the eigen vector and the leading eigen value

**References**

Chindelevitch, Leonid, et al. "Network science inspires novel tree shape statistics." Plos one 16.12 (2021): e0259877.

---

|           |  |
|-----------|--|
| entropy_j | <i>Fast function using C++ to calculate the intensive quadratic entropy statistic J.</i> |
|-----------|--|

---

**Description**

The intensive quadratic entropy statistic  $J$  is given by the average distance between two randomly chosen species, thus given by the sum of all pairwise distances, divided by  $S^2$ , where  $S$  is the number of tips of the tree.

**Usage**

```
entropy_j(phy)
```

**Arguments**

|     |                        |
|-----|------------------------|
| phy | phylo object or ltable |
|-----|------------------------|



**Value**

intensive quadratic entropy statistic J

**References**

Izsák, János, and Laszlo Papp. "A link between ecological diversity indices and measures of biodiversity." *Ecological Modelling* 130.1-3 (2000): 151-156.

---

|            |   |
|------------|---|
| ew_colless | <i>Fast function using C++ to calculate the equal weights Colless index of (im)balance.</i> |
|------------|---|

---

**Description**

The equal weights Colless index is calculated as the sum of  $abs(L - R)/(L + R - 2)$  over all nodes where  $L + R > 2$ , where L (or R) is the number of extant tips associated with the L (or R) daughter branch at that node. Maximal imbalance is associated with a value of 1.0. The ew\_colless index is not sensitive to tree size.

**Usage**

```
ew_colless(phy)
```

**Arguments**

phy                    phylo object or ltable

**Value**

colless index

**References**

A. O. Mooers and S. B. Heard. Inferring Evolutionary Process from Phylogenetic Tree Shape. *The Quarterly Review of Biology*, 72(1), 1997. doi: 10.1086/419657.

**Examples**

```
simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
brts <- branching_times(simulated_tree)
if (requireNamespace("nodeSub")) {
  balanced_tree <- nodeSub::create_balanced_tree(brts)
  unbalanced_tree <- nodeSub::create_unbalanced_tree(brts)
  ew_colless(balanced_tree)
  ew_colless(unbalanced_tree) # should be higher
}
```

---

|                 |   |
|-----------------|---|
| gamma_statistic | <i>Calculate the gamma statistic, using a fast implementation in C++.</i> |
|-----------------|---|

---

### Description

The gamma statistic measures the relative position of internal nodes within a reconstructed phylogeny. Under the Yule process, the gamma values of a reconstructed tree follow a standard normal distribution. If  $\gamma > 0$ , the nodes are located more towards the tips of the tree, and if  $\gamma < 0$ , the nodes are located more towards the root of the tree. Only available for ultrametric trees.

### Usage

```
gamma_statistic(phy)
```

### Arguments

|     |                        |
|-----|------------------------|
| phy | phylo object or ltable |
|-----|------------------------|

### Value

gamma statistic

### References

Pybus, O. G. and Harvey, P. H. (2000) Testing macro-evolutionary models using incomplete molecular phylogenies. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 267, 2267–2272.

### Examples

```
simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
gamma_statistic(simulated_tree) # should be around 0.
if (requireNamespace("DDD")) {
  ddd_tree <- DDD::dd_sim(pars = c(1, 0, 10), age = 7)$tes
  gamma_statistic(ddd_tree) # because of diversity dependence, should be < 0
}
```

---

|          |                           |
|----------|---------------------------|
| ILnumber | <i>Calculate ILnumber</i> |
|----------|---------------------------|

---

### Description

The ILnumber is the number of internal nodes with a single tip child. Higher values typically indicate a tree that is more unbalanced.

The ILnumber is the number of internal nodes with a single tip child, as adapted from the phyloTop package.

**Usage**

```
ILnumber(input_obj, normalization = "none")
```

**Arguments**

input\_obj      phylo object or ltable  
normalization   "none" or "tips", in which case the result is normalized by dividing by  $N - 2$ , where  $N$  is the number of tips.

**Value**

ILnumber

---

|                 |                              |
|-----------------|------------------------------|
| imbalance_steps | <i>imbalance steps index</i> |
|-----------------|------------------------------|

---

**Description**

Calculates the number of moves required to transform the focal tree into a fully imbalanced (caterpillar) tree. Higher value indicates a more balanced tree.

**Usage**

```
imbalance_steps(input_obj, normalization = FALSE)
```

**Arguments**

input\_obj      phylo object or ltable  
normalization   if true, the number of steps taken is normalized by tree size, by dividing by the maximum number of moves required to move from a fully balanced to a fully imbalanced tree, which is  $N - \log_2(N) - 1$ , where  $N$  is the number of extant tips.

**Value**

required number of moves

---

j\_one                      *function to calculate the J^1 index.*

---

### Description

The J<sup>1</sup> index calculates the Shannon Entropy of a tree, where at each node with two children, the Shannon Entropy is the sum of  $p_i \log_2(p_i)$  over the two children  $i$ , and  $p_i$  is  $L / (L + R)$ , where  $L$  and  $R$  represent the number of tips connected to the two daughter branches.

### Usage

```
j_one(input_obj)
```

### Arguments

input\_obj            phylo object or ltable

### Value

j<sup>1</sup> index

### References

Jeanne Lemant, Cécile Le Sueur, Veselin Manojlović, Robert Noble, Robust, Universal Tree Balance Indices, *Systematic Biology*, Volume 71, Issue 5, September 2022, Pages 1210–1224, <https://doi.org/10.1093/sysbio/syaa000>

---

laplacian\_spectrum      *function to calculate the laplacian spectrum, from RPANDA*

---

### Description

Computes the distribution of eigenvalues for the modified graph Laplacian of a phylogenetic tree, and several summary statistics of this distribution. The modified graph Laplacian of a phylogeny is given by the difference between its' distance matrix (e.g. all pairwise distances between all nodes), and the degree matrix (e.g. the diagonal matrix where each diagonal element represents the sum of branch lengths to all other nodes). Each row of the modified graph Laplacian sums to zero. For a tree with  $n$  tips, there are  $N = 2n - 1$  nodes, and hence the modified graph Laplacian is represented by a  $N \times N$  matrix. Where RPANDA relies on the package igraph to calculate the modified graph Laplacian, the treestats package uses C++ to directly calculate the different entries in the matrix. This makes the treestats implementation slightly faster, although the bulk of computation occurs in estimating the eigen values, using the function eigen from base.

### Usage

```
laplacian_spectrum(phy)
```

**Arguments**

phy                      phy

**Value**

list with five components: 1) eigenvalues the vector of eigen values, 2) principal\_eigenvalue the largest eigenvalue of the spectral density distribution 3) asymmetry the skewness of the spectral density distribution 4) peak\_height the largest y-axis value of the spectral density distribution and 5) eigengap the position of the largest difference between eigenvalues, giving the number of modalities in the tree.

**References**

Eric Lewitus, Helene Morlon, Characterizing and Comparing Phylogenies from their Laplacian Spectrum, Systematic Biology, Volume 65, Issue 3, May 2016, Pages 495–507, <https://doi.org/10.1093/sysbio/syv116>

---

list\_statistics                      *Provides a list of all available statistics in the package*

---

**Description**

Provides a list of all available statistics in the package

**Usage**

```
list_statistics(only_balance_stats = FALSE)
```

**Arguments**

only\_balance\_stats  
only return those statistics associated with measuring balance of a tree

**Value**

vector with names of summary statistics

---

|                  |  |
|------------------|--|
| ltable_to_newick | <i>Convert an L table to newick string</i> |
|------------------|--|

---

**Description**

Convert an L table to newick string

**Usage**

```
ltable_to_newick(ltab, drop_extinct = TRUE)
```

**Arguments**

|              |   |
|--------------|---|
| ltab         | ltable  |
| drop_extinct | should extinct species be dropped from the phylogeny? |

**Value**

phylo object

---

|            |   |
|------------|---|
| l_to_phylo | <i>Convert an L table to phylo object</i> |
|------------|---|

---

**Description**

Convert an L table to phylo object

**Usage**

```
l_to_phylo(ltab, drop_extinct = TRUE)
```

**Arguments**

|              |   |
|--------------|---|
| ltab         | ltable  |
| drop_extinct | should extinct species be dropped from the phylogeny? |

**Value**

phylo object

---

make\_unbalanced\_tree *this function increasingly increases the imbalance of a tree*

---

### Description

the goal of this function is to increasingly imbalance a tree, by changing the topology, one move at a time. It does so by re-attaching terminal branches to the root lineage, through the ltable. In effect, this causes the tree to become increasingly caterpillarlike. When started with a balanced tree, this allows for exploring the gradient between a fully balanced tree, and a fully unbalanced tree. Please note that the algorithm will try to increase imbalance, until a fully caterpillar like tree is reached, which may occur before unbal\_steps is reached. Three methods are available: "youngest", reattaches branches in order of age, starting with the branch originating from the most recent branching event and working itself through the tree. "Random" picks a random branch to reattach. "Terminal" also picks a random branch, but only from terminal branches (e.g. branches that don't have any daughter lineages, which is maximized in a fully imbalanced tree).

### Usage

```
make_unbalanced_tree(  
  init_tree,  
  unbal_steps,  
  group_method = "any",  
  selection_method = "random"  
)
```

### Arguments

|                  |   |
|------------------|---|
| init_tree        | starting tree to work with                  |
| unbal_steps      | number of imbalance generating steps        |
| group_method     | choice of "any" and "terminal"              |
| selection_method | choice of "random", "youngest" and "oldest" |

### Value

phylo object

### Examples

```
simulated_tree <- ape::rphylo(n = 16, birth = 1, death = 0)  
brts <- branching_times(simulated_tree)  
if (requireNamespace("nodeSub")) {  
  balanced_tree <- nodeSub::create_balanced_tree(brts)  
  unbalanced_tree <- nodeSub::create_unbalanced_tree(brts)  
  intermediate_tree <- make_unbalanced_tree(balanced_tree, 8)  
  colless(balanced_tree)  
  colless(intermediate_tree) # should be intermediate value
```

```

    colless(unbalanced_tree) # should be highest colless value
}

```

---

|                 |   |
|-----------------|---|
| max_betweenness | <i>Fast function using C++ to calculate maximum betweenness centrality.</i> |
|-----------------|---|

---

### Description

Betweenness centrality associates with each node  $v$ , the two nodes  $u$ ,  $w$ , for which the shortest path between  $u$  and  $w$  runs through  $v$ , if the tree were re-rooted at node  $v$ . Then, we report the node with maximum betweenness centrality.

### Usage

```
max_betweenness(phy, normalization = "none")
```

### Arguments

|               |   |
|---------------|---|
| phy           | phylo object or ltable  |
| normalization | "none" or "tips", if tips is chosen, the obtained maximum betweenness is normalized by the total amount of node pair combinations considered, e.g. $(n-2)*(n-1)$ , where $n$ is the number of tips. |

### Value

Maximum Betweenness

### References

Chindelevitch, Leonid, et al. "Network science inspires novel tree shape statistics." Plos one 16.12 (2021): e0259877.

---

|               |   |
|---------------|---|
| max_closeness | <i>Fast function using C++ to calculate maximum closeness</i> |
|---------------|---|

---

### Description

Closeness is defined as  $1 / \text{Farness}$ , where Farness is the sum of distances from a node to all the other nodes in the tree. Here, we return the node with maximum closeness.

### Usage

```
max_closeness(phy, weight = TRUE, normalization = "none")
```



**Arguments**

|               |  |
|---------------|--|
| phy           | phylo object or ltable   |
| weight        | if TRUE, uses branch lengths.  |
| normalization | "none" or "tips", in which case an arbitrary post-hoc correction is performed by dividing by the expectation of $n \log(n)$ , where $n$ is the number of tips. |

**Value**

Maximum Closeness

**References**

Chindelevitch, Leonid, et al. "Network science inspires novel tree shape statistics." Plos one 16.12 (2021): e0259877. Wang W, Tang CY. Distributed computation of classic and exponential closeness on tree graphs. Proceedings of the American Control Conference. IEEE; 2014. p. 2090–2095.

---

|               |   |
|---------------|---|
| max_del_width | <i>Fast function using C++ to calculate the maximum difference of widths of a phylogenetic tree</i> |
|---------------|---|

---

**Description**

Calculates the maximum difference of widths of a phylogenetic tree. First, the widths are calculated by collecting the depth of each node and tip across the entire tree, where the depth represents the distance (in nodes) to the root. Then, the width represents the number of occurrences of each possible depth. Then, we take the difference between each consecutive width, starting with the first width. The maximum difference is then returned.

**Usage**

```
max_del_width(phy, normalization = "none")
```

**Arguments**

|               |   |
|---------------|---|
| phy           | phylogeny or ltable   |
| normalization | "none" or "tips", in which case the resulting statistic is divided by the number of tips in the tree. |

**Value**

maximum difference of widths

**References**

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..

---

|           |  |
|-----------|--|
| max_depth | <i>Fast function using C++ to calculate maximum depth metric</i> |
|-----------|--|

---

**Description**

The maximum depth metric, measures the maximal path (in edges), between the tips and the root.

**Usage**

```
max_depth(phy, normalization = "none")
```

**Arguments**

|               |   |
|---------------|---|
| phy           | phylo object or ltable  |
| normalization | "none" or "tips", in which case the resulting statistic is divided by the number of tips in the tree. |

**Value**

Maximum depth (in number of edges)

**References**

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.

---

|            |                             |
|------------|-----------------------------|
| max_ladder | <i>maximum ladder index</i> |
|------------|-----------------------------|

---

**Description**

Calculate the maximum ladder index, from the phyloTop package. Higher values indicate more unbalanced trees. To calculate the maximum ladder index, first all potential ladders in the tree are calculated. A ladder is defined as a sequence of nodes where one of the daughter branches is a terminal branch, resulting in a 'ladder' like pattern. The maximum ladder index then represents the longest ladder found amongst all observed ladders in the tree.

**Usage**

```
max_ladder(input_obj)
```

**Arguments**

|           |                        |
|-----------|------------------------|
| input_obj | phylo object or ltable |
|-----------|------------------------|

**Value**

longest ladder in the tree

---

|           |  |
|-----------|--|
| max_width | <i>Fast function using C++ to calculate the maximum width of branch depth.</i> |
|-----------|--|

---

**Description**

Calculates the maximum width, this is calculated by first collecting the depth of each node and tip across the entire tree, where the depth represents the distance (in nodes) to the root. Then, the width represents the number of occurrences of each possible depth. The maximal width then returns the maximum number of such occurrences.

**Usage**

```
max_width(phy, normalization = "none")
```

**Arguments**

|               |   |
|---------------|---|
| phy           | phylogeny or Ltable   |
| normalization | "none" or "tips", in which case the resulting statistic is divided by the number of tips in the tree. |

**Value**

maximum width

**References**

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.

---

|                    |   |
|--------------------|---|
| mean_branch_length | <i>Calculates the mean branch length of a tree, including extinct branches.</i> |
|--------------------|---|

---

**Description**

Calculates the mean branch length of a tree, including extinct branches.

**Usage**

```
mean_branch_length(phy)
```

**Arguments**

|     |                        |
|-----|------------------------|
| phy | phylo object or Ltable |
|-----|------------------------|

**Value**

mean branch length

---

mean\_branch\_length\_ext

*Calculates the mean of external branch lengths of a tree, e.g. of branches leading to a tip.*

---

**Description**

Calculates the mean of external branch lengths of a tree, e.g. of branches leading to a tip.

**Usage**

```
mean_branch_length_ext(phy)
```

**Arguments**

phy                    phylo object or Ltable

**Value**

mean of external branch lengths

---

mean\_branch\_length\_int

*Calculates the variation of internal branch lengths of a tree, e.g. of branches not leading to a tip.*

---

**Description**

Calculates the variation of internal branch lengths of a tree, e.g. of branches not leading to a tip.

**Usage**

```
mean_branch_length_int(phy)
```

**Arguments**

phy                    phylo object or Ltable

**Value**

variation of internal branch lengths

---

 mean\_i

*Fast function using C++ to calculate the mean I value.*


---

**Description**

The mean I value is defined for all nodes with at least 4 tips connected, such that different topologies can be formed. Then, for each node,  $I = (nm - nt/2) / (nt - 1 - nt/2)$ , where  $nt$  is the total number of tips descending from that node,  $nm$  is the daughter branch leading to most tips, and  $nt/2$  is the minimum size of the maximum branch, rounded up. Following Purvis et al 2002, we perform a correction on  $I$ , where we correct  $I$  for odd  $nt$ , such that  $I' = I * (nt - 1) / nt$ . This correction ensures that  $I$  is independent of  $nt$ . We report the mean value across all  $I'$  (again, following Purvis et al. 2002).

**Usage**

```
mean_i(phy)
```

**Arguments**

```
phy          phylo object or ltable
```

**Value**

average I value across all nodes

**References**

G. Fusco and Q. C. Cronk. A new method for evaluating the shape of large phylogenies. *Journal of Theoretical Biology*, 1995. doi: 10.1006/jtbi.1995.0136. A. Purvis, A. Katzourakis, and P.-M. Agapow. Evaluating Phylogenetic Tree Shape: Two Modifications to Fusco & Cronks Method. *Journal of Theoretical Biology*, 2002. doi: 10.1006/jtbi.2001.2443.

---

 mean\_pair\_dist

*Mean Pairwise distance*


---

**Description**

Fast function using C++ to calculate the mean pairwise distance, using the fast algorithm by Constantinos, Sandel & Cheliotis (2012).

**Usage**

```
mean_pair_dist(phy, normalization = "none")
```

**Arguments**

phy                    phylo object or ltable  
 normalization    "none" or "tips", in which case the obtained mean pairwise distance is normalized by the factor  $2\log(n)$ , where n is the number of tips.

**Value**

Mean pairwise distance

**References**

Webb, C., D. Ackerly, M. McPeck, and M. Donoghue. 2002. Phylogenies and community ecology. *Annual Review of Ecology and Systematics* 33:475-505. Tsirogiannis, Constantinos, Brody Sandel, and Dimitris Cheliotis. "Efficient computation of popular phylogenetic tree measures." *Algorithms in Bioinformatics: 12th International Workshop, WABI 2012, Ljubljana, Slovenia, September 10-12, 2012. Proceedings 12*. Springer Berlin Heidelberg, 2012.

---

 mntd

---

*Mean Nearest Taxon distance*


---

**Description**

Per tip, evaluates the shortest distance to another tip, then takes the average across all tips.

**Usage**

`mntd(phy)`

**Arguments**

phy                    phylo object or ltable

**Value**

Mean Nearest Taxon Distance.

**References**

Webb, C., D. Ackerly, M. McPeck, and M. Donoghue. 2002. Phylogenies and community ecology. *Annual Review of Ecology and Systematics* 33:475-505.

---

|      |                                       |
|------|---------------------------------------|
| nLTT | <i>Calculate the nLTT, using C++.</i> |
|------|---------------------------------------|

---

## Description

The nLTT statistic calculates the sum of absolute differences in the number of lineages over time, where both the number of lineages and the time are normalized. The number of lineages is normalized by the number of extant tips, whereas the time is normalized by the crown age. The nLTT can only be calculated for reconstructed trees. Only use the `treestats` version if you are very certain about the input data, and are certain that performing nLTT is valid (e.g. your tree is ultrametric etc). If you are less certain, use the `nLTT` function from the nLTT package.

## Usage

```
nLTT(phy, ref_tree)
```

## Arguments

|                       |   |
|-----------------------|---|
| <code>phy</code>      | phylo object or <code>ltable</code>                                       |
| <code>ref_tree</code> | reference tree to compare with (should be same type as <code>phy</code> ) |

## Value

number of lineages

## References

Janzen, T., Höhna, S. and Etienne, R.S. (2015), Approximate Bayesian Computation of diversification rates from molecular phylogenies: introducing a new efficient summary statistic, the nLTT. *Methods Ecol Evol*, 6: 566-575. <https://doi.org/10.1111/2041-210X.12350>

## Examples

```
simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
reference_tree <- ape::rphylo(n = 10, birth = 0.2, death = 0)
nLTT(simulated_tree, reference_tree)
nLTT(simulated_tree, simulated_tree) # should be zero.
```

---

|           |   |
|-----------|---|
| nLTT_base | <i>Calculates the nLTT statistic using a reference 'empty' tree with only two lineages.</i> |
|-----------|---|

---

### Description

The base nLTT statistic can be used as a semi stand-alone statistic for phylogenetic trees. However, please note that although this provides a nice way of checking the power of the nLTT statistic without directly comparing two trees, the nLTT\_base statistic is not a substitute for directly comparing two phylogenetic trees. E.g. one would perhaps naively assume that  $nLTT(A, B) = |nLTT(A, base) - nLTT(B, base)|$ . Indeed, in some cases this may hold true (when, for instance, all normalized lineages of A are less than all normalized lineages of B), but once the nLTT curve of A intersects the nLTT curve of B, this no longer applies.

### Usage

```
nLTT_base(phy)
```

### Arguments

phy                    phylo object

### Value

number of lineages

### Examples

```
simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
nLTT_base(simulated_tree)
```

---

|                    |  |
|--------------------|--|
| number_of_lineages | <i>Calculate the number of tips of a tree, including extinct tips.</i> |
|--------------------|--|

---

### Description

Calculate the number of tips of a tree, including extinct tips.

### Usage

```
number_of_lineages(phy)
```

### Arguments

phy                    phylo object



**Value**

number of lineages

---

phylogenetic\_diversity

*Calculates phylogenetic diversity at time point t*

---

**Description**

The phylogenetic diversity at time  $t$  is given by the total branch length of the tree reconstructed up until time point  $t$ . Time is measured increasingly, with the crown age equal to 0. Thus, the time at the present is equal to the crown age.

**Usage**

```
phylogenetic_diversity(input_obj, t = 0, extinct_tol = NULL)
```

**Arguments**

|             |   |
|-------------|---|
| input_obj   | phylo object or Ltable  |
| t           | time point at which to measure phylogenetic diversity, alternatively a vector of time points can also be provided. Time is measured with 0 being the present. |
| extinct_tol | tolerance to determine if a lineage is extinct at time $t$ . Default is $1/100 * \text{smallest branch length of the tree}$ .                                 |

**Value**

phylogenetic diversity, or vector of phylogenetic diversity measures if a vector of time points is used as input.

**References**

Faith, Daniel P. "Conservation evaluation and phylogenetic diversity." *Biological conservation* 61.1 (1992): 1-10.

---

phylo\_to\_l                      *Function to generate an ltable from a phy object.*

---

### Description

This function is a C++ implementation of the function `DDD::phylo2L`. An L table summarises a phylogeny in a table with four columns, being: 1) time at which a species is born, 2) label of the parent of the species, where positive and negative numbers indicate whether the species belongs to the left or right crown lineage, 3) label of the daughter species itself (again positive or negative depending on left or right crown lineage), and the last column 4) indicates the time of extinction of a species, or -1 if the species is extant.

### Usage

```
phylo_to_l(phy)
```

### Arguments

phy                      phylo object

### Value

ltable (see description)

### Examples

```
simulated_tree <- ape::rphylo(n = 4, birth = 1, death = 0)
ltable <- phylo_to_l(simulated_tree)
reconstructed_tree <- DDD::L2phylo(ltable)
old_par <- par()
par(mfrow = c(1, 2))
# trees should be more or less similar, although labels may not match, and
# rotations might cause (initial) visual mismatches
plot(simulated_tree)
plot(reconstructed_tree)
par(old_par)
```

---

pigot\_rho                      *calculate Pigot's rho*

---

### Description

Calculates the change in rate between the first half and the second half of the extant phylogeny.  $\rho = (r_2 - r_1) / (r_1 + r_2)$ , where  $r$  reflects the rate in either the first or second half. The rate within a half is given by  $(\log(n_2) - \log(n_1)) / t$ , where  $n_2$  is the number of lineages at the end of the half, and  $n_1$  the number of lineages at the start of the half.  $\rho$  varies between -1 and 1, with a 0 indicating a constant rate across the phylogeny, a  $\rho < 0$  indicating a slow down and a  $\rho > 0$  indicating a speed up of speciation. In contrast to the Gamma statistic, Pigot's  $\rho$  is not sensitive to tree size.

**Usage**

```
pigot_rho(phy, extant_tree = TRUE)
```

**Arguments**

`phy`                    phylo object

`extant_tree`        Pigot's rho is originally only defined for an extant tree, however we include functionality to calculate Pigot's rho for a complete tree as well.

**Value**

rho

**References**

Alex L. Pigot, Albert B. Phillimore, Ian P. F. Owens, C. David L. Orme, The Shape and Temporal Dynamics of Phylogenetic Trees Arising from Geographic Speciation, *Systematic Biology*, Volume 59, Issue 6, December 2010, Pages 660–673, <https://doi.org/10.1093/sysbio/syq058>

**Examples**

```
simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
pigot_rho(simulated_tree) # should be around 0.
ddd_tree <- DDD::dd_sim(pars = c(1, 0, 10), age = 7)$tes
pigot_rho(ddd_tree) # because of diversity dependence, should be < 0
```

---

|            |   |
|------------|---|
| pitchforks | <i>Calculate pitchforks, from the phyloTop package, a pitchfork is a clade with three tips.</i> |
|------------|---|

---

**Description**

Calculate pitchforks, from the phyloTop package, a pitchfork is a clade with three tips.

**Usage**

```
pitchforks(input_obj, normalization = "none")
```

**Arguments**

`input_obj`            phylo object or ltable

`normalization`    "none" or "tips", in which case the found number of pitchforks is divided by the expected number.

**Value**

number of pitchforks

---

```
prep_lapl_spec      function to create laplacian matrix
```

---

**Description**

function to create laplacian matrix

**Usage**

```
prep_lapl_spec(phy)
```

**Arguments**

```
phy      phy
```

**Value**

numericmatrix

---

```
psv      Fast function using C++ to calculate Phylogenetic Species Variability.
```

---

**Description**

The phylogenetic species variability is bounded in [0, 1]. The psv quantifies how phylogenetic relatedness decrease the variance of a (neutral) trait shared by all species in the tree. As species become more related, the psv tends to 0. Please note that the psv is a special case of the Mean Pair Distance (see appendix of Tucker et al. 2017 for a full derivation), and thus correlates directly.

**Usage**

```
psv(phy, normalization = "none")
```

**Arguments**

```
phy      phylo object or ltable
normalization  "none" or "tips", in which case the obtained mean pairwise distance is normalized by the factor  $2\log(n)$ , where n is the number of tips.
```

**Value**

Phylogenetic Species Variability

**References**

Helmus M.R., Bland T.J., Williams C.K. & Ives A.R. (2007) Phylogenetic measures of biodiversity. *American Naturalist*, 169, E68-E83

Tucker, Caroline M., et al. "A guide to phylogenetic metrics for conservation, community ecology and macroecology." *Biological Reviews* 92.2 (2017): 698-715.

---

|               |  |
|---------------|--|
| rebase_ltable | <i>a function to modify an ltable, such that the longest path in the phylogeny is a crown lineage.</i> |
|---------------|--|

---

**Description**

a function to modify an ltable, such that the longest path in the phylogeny is a crown lineage.

**Usage**

```
rebase_ltable(ltable)
```

**Arguments**

|        |        |
|--------|--------|
| ltable | ltable |
|--------|--------|

**Value**

modified ltable

---

|        |  |
|--------|--|
| rogers | <i>Fast function using C++ to calculate the Rogers J index of (im)balance.</i> |
|--------|--|

---

**Description**

The Rogers index is calculated as the total number of internal nodes that are unbalanced, e.g. for which both daughter nodes lead to a different number of extant tips. in other words, the number of nodes where  $L \neq R$  (where L(R) is the number of extant tips of the Left (Right) daughter node).

**Usage**

```
rogers(phy, normalization = "none")
```

**Arguments**

|               |   |
|---------------|---|
| phy           | phylo object or ltable  |
| normalization | "none" or "tips", in which case the resulting statistic is divided by the number of tips - 2 (e.g. the maximum value of the rogers index for a tree). |

**Value**

Rogers index

**References**

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.

**Examples**

```
simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
brts <- branching_times(simulated_tree)
if (requireNamespace("nodeSub")) {
  balanced_tree <- nodeSub::create_balanced_tree(brts)
  unbalanced_tree <- nodeSub::create_unbalanced_tree(brts)
  rogers(balanced_tree)
  rogers(unbalanced_tree) # should be higher
}
```

---

rquartet

*Fast function using C++ to calculate the rquartet index.*


---

**Description**

The rquartet index counts the number of potential fully balanced rooted subtrees of 4 tips in the tree. The function in treestats assumes a bifurcating tree. For trees with polytomies, we refer the user to treebalance::rqartedI, which can also take polytomies into account.

**Usage**

```
rquartet(phy, normalization = "none")
```

**Arguments**

|               |  |
|---------------|--|
| phy           | phylo object or ltable   |
| normalization | The index can be normalized by the expectation under the Yule ("yule") or PDA model ("pda"). |

**Value**

rquartet index

**References**

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.

---

|        |  |
|--------|--|
| sackin | <i>Fast function using C++ to calculate the sackin index of (im)balance.</i> |
|--------|--|

---

### Description

The Sackin index is calculated as the sum of ancestors for each of the tips. Higher values indicate higher imbalance. Two normalizations are available, where a correction is made for tree size, under either a Yule expectation, or a pda expectation.

### Usage

```
sackin(phy, normalization = "none")
```

### Arguments

phy                    phylogeny or Itable  
normalization        normalization, either 'none' (default), "yule" or "pda".

### Value

Sackin index

### References

M. J. Sackin (1972). "Good" and "Bad" Phenograms. *Systematic Biology*. 21:225-226.

### Examples

```
simulated_tree <- ape::rphylo(n = 10, birth = 1, death = 0)
brts <- branching_times(simulated_tree)
if (requireNamespace("nodeSub")) {
  balanced_tree <- nodeSub::create_balanced_tree(brts)
  unbalanced_tree <- nodeSub::create_unbalanced_tree(brts)
  sackin(balanced_tree)
  sackin(unbalanced_tree) # should be much higher
}
```

---

|        |   |
|--------|---|
| stairs | <i>Calculates the staircase-ness measure, from the phyloTop package. The staircase-ness reflects the number of subtrees that are imbalanced, e.g. subtrees where the left child has more extant tips than the right child, or vice versa.</i> |
|--------|---|

---

**Description**

Calculates the staircase-ness measure, from the phyloTop package. The staircase-ness reflects the number of subtrees that are imbalanced, e.g. subtrees where the left child has more extant tips than the right child, or vice versa.

**Usage**

```
stairs(input_obj)
```

**Arguments**

input\_obj      phylo object or ltable

**Value**

number of stairs

**References**

Norström, Melissa M., et al. "Phylotempo: a set of r scripts for assessing and visualizing temporal clustering in genealogies inferred from serially sampled viral sequences." *Evolutionary Bioinformatics* 8 (2012): EBO-S9738.

---

|         |  |
|---------|--|
| stairs2 | <i>Calculates the stairs2 measure, from the phyloTop package. The stairs2 reflects the imbalance at each node, where it represents the average across measure at each node, the measure being <math>\min(l, r) / \max(l, r)</math>, where <math>l</math> and <math>r</math> reflect the number of tips connected at the left (<math>l</math>) and right (<math>r</math>) daughter.</i> |
|---------|--|

---

**Description**

Calculates the stairs2 measure, from the phyloTop package. The stairs2 reflects the imbalance at each node, where it represents the average across measure at each node, the measure being  $\min(l, r) / \max(l, r)$ , where  $l$  and  $r$  reflect the number of tips connected at the left ( $l$ ) and right ( $r$ ) daughter.

**Usage**

```
stairs2(input_obj)
```

**Arguments**

input\_obj      phylo object or ltable

**Value**

number of stairs



## References

Norström, Melissa M., et al. "Phylotempo: a set of r scripts for assessing and visualizing temporal clustering in genealogies inferred from serially sampled viral sequences." *Evolutionary Bioinformatics* 8 (2012): EBO-S9738.

---

sym\_nodes

*Fast function using C++ to calculate the symmetry nodes metric*

---

## Description

Balance metric that returns the total number of internal nodes that are not-symmetric (confusingly enough). A node is considered symmetric when both daughter trees have the same topology, measured as having the same sum of depths, where depth is measured as the distance from the root to the node/tip.

## Usage

```
sym_nodes(phy, normalization = "none")
```

## Arguments

|               |   |
|---------------|---|
| phy           | phylo object or ltable  |
| normalization | "none" or "tips", in which case the resulting statistic is divided by the number of tips - 2 (e.g. the maximum value of the symmetry nodes index for a tree). |

## Value

Maximum depth (in number of edges)

## References

S. J. Kersting and M. Fischer. Measuring tree balance using symmetry nodes — A new balance index and its extremal properties. *Mathematical Biosciences*, page 108690, 2021. ISSN 0025-5564. doi:<https://doi.org/10.1016/j.mbs.2021.108690>

---

|          |   |
|----------|---|
| tot_coph | <i>Fast function using C++ to calculate the total cophenetic index.</i> |
|----------|---|

---

**Description**

The total cophenetic index is the sum of the depth of the last common ancestor of all pairs of leaves.

**Usage**

```
tot_coph(phy, normalization = "none")
```

**Arguments**

|               |   |
|---------------|---|
| phy           | phylo object or ltable  |
| normalization | "none" or "yule", when "yule" is chosen, the statistic is divided by the Yule expectation |

**Value**

Total cophenetic index

**References**

A. Mir, F. Rosselló, and L. Rotger. A new balance index for phylogenetic trees. *Mathematical Bio-sciences*, 241(1):125-136, 2013. doi: 10.1016/j.mbs.2012.10.005.

---

|             |   |
|-------------|---|
| tree_height | <i>Calculates the height of a tree using C++.</i> |
|-------------|---|

---

**Description**

In a reconstructed tree, obtaining the tree height is fairly straightforward, and the function `beautier::get_crown_age` does a great job at it. However, in a non-ultrametric tree, that function no longer works. Alternatively, taking the maximum value of `adephylo::distRoot` will also yield the tree height (including the root branch), but will typically perform many superfluous calculations and thus be slow.

**Usage**

```
tree_height(phy)
```

**Arguments**

|     |              |
|-----|--------------|
| phy | phylo object |
|-----|--------------|

**Value**

crown age

---

|                   |   |
|-------------------|---|
| var_branch_length | <i>Calculates the variance of branch lengths of a tree, including extinct branches.</i> |
|-------------------|---|

---

**Description**

Calculates the variance of branch lengths of a tree, including extinct branches.

**Usage**

```
var_branch_length(phy)
```

**Arguments**

|     |                        |
|-----|------------------------|
| phy | phylo object or Ltable |
|-----|------------------------|

**Value**

variance of branch lengths

---

|                       |   |
|-----------------------|---|
| var_branch_length_ext | <i>Calculates the variance of external branch lengths of a tree, e.g. of branches leading to a tip.</i> |
|-----------------------|---|

---

**Description**

Calculates the variance of external branch lengths of a tree, e.g. of branches leading to a tip.

**Usage**

```
var_branch_length_ext(phy)
```

**Arguments**

|     |                        |
|-----|------------------------|
| phy | phylo object or Ltable |
|-----|------------------------|

**Value**

variance of external branch lengths

---

var\_branch\_length\_int *Calculates the variance of internal branch lengths of a tree, e.g. of branches not leading to a tip.*

---

**Description**

Calculates the variance of internal branch lengths of a tree, e.g. of branches not leading to a tip.

**Usage**

```
var_branch_length_int(phy)
```

**Arguments**

phy                    phylo object or Ltable

**Value**

variance of internal branch lengths

---

var\_leaf\_depth            *Fast function using C++ to calculate the variance of leaf depth statistic*

---

**Description**

The variance of leaf depth statistic returns the variance of depths across all tips.

**Usage**

```
var_leaf_depth(phy, normalization = "none")
```

**Arguments**

phy                    phylo object or ltable  
normalization    "none" or "yule", when "yule" is chosen, the statistic is divided by the Yule expectation

**Value**

Variance of leaf depths

**References**

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.

---

|               |   |
|---------------|---|
| var_pair_dist | <i>Fast function using C++ to calculate the variance of all pairwise distances.</i> |
|---------------|---|

---

**Description**

After calculating all pairwise distances between all tips, this function takes the variance across these values.

**Usage**

```
var_pair_dist(phy)
```

**Arguments**

|     |                        |
|-----|------------------------|
| phy | phylo object or ltable |
|-----|------------------------|

**Value**

Variance in pairwise distance

**References**

Webb, C., D. Ackerly, M. McPeck, and M. Donoghue. 2002. Phylogenies and community ecology. *Annual Review of Ecology and Systematics* 33:475-505.

---

|        |  |
|--------|--|
| wiener | <i>Fast function using C++ to calculate the Wiener index</i> |
|--------|--|

---

**Description**

The Wiener index is defined as the sum of all shortest path lengths between pairs of nodes in a tree.

**Usage**

```
wiener(phy, normalization = FALSE, weight = TRUE)
```

**Arguments**

|               |   |
|---------------|---|
| phy           | phylo object or ltable  |
| normalization | if TRUE, the Wiener index is normalized by the number of nodes, e.g. by $\text{choose}(n, 2)$ , where $n$ is the number of nodes. |
| weight        | if TRUE, branch lengths are used.   |

**Value**

Wiener index

**References**

Chindelevitch, Leonid, et al. "Network science inspires novel tree shape statistics." Plos one 16.12 (2021): e0259877. Mohar, B., Pisanski, T. How to compute the Wiener index of a graph. J Math Chem 2, 267–277 (1988)

# Index

area\_per\_pair, 4  
average\_leaf\_depth, 4  
avg\_ladder, 5

b1, 6  
b2, 6  
beta\_statistic, 7  
blum, 8  
branching\_times, 9

calc\_all\_stats, 9  
calc\_balance\_stats, 11  
calc\_brts\_stats, 12  
cherries, 13  
colless, 14  
crown\_age, 15

diameter, 15

eigen\_vector, 16  
entropy\_j, 16  
ew\_colless, 17

gamma\_statistic, 18

ILnumber, 18  
imbalance\_steps, 19

j\_one, 20

l\_to\_phylo, 22  
laplacian\_spectrum, 20  
list\_statistics, 21  
ltable\_to\_newick, 22

make\_unbalanced\_tree, 23  
max\_betweenness, 24  
max\_closeness, 24  
max\_del\_width, 25  
max\_depth, 26  
max\_ladder, 26

max\_width, 27  
mean\_branch\_length, 27  
mean\_branch\_length\_ext, 28  
mean\_branch\_length\_int, 28  
mean\_i, 29  
mean\_pair\_dist, 29  
mntd, 30

nLTT, 31  
nLTT\_base, 32  
number\_of\_lineages, 32

phylo\_to\_l, 34  
phylogenetic\_diversity, 33  
pigot\_rho, 34  
pitchforks, 35  
prep\_lapl\_spec, 36  
psv, 36

rebase\_ltable, 37  
rogers, 37  
rquartet, 38

sackin, 39  
stairs, 39  
stairs2, 40  
sym\_nodes, 41

tot\_coph, 42  
tree\_height, 42  
treestats (treestats-package), 3  
treestats-package, 3

var\_branch\_length, 43  
var\_branch\_length\_ext, 43  
var\_branch\_length\_int, 44  
var\_leaf\_depth, 44  
var\_pair\_dist, 45

wiener, 45