Package 'treelayout'

November 11, 2026

Title Tree Layout Algorithms for Phylogenetic Trees and Other Tree-Like Structures

Version 0.1.0

Description Computes node coordinates for phylogenetic trees and other tree-like structures. Given a tree, returns one row per node with x and y coordinates, ready for plotting by any graphics system. This is the layout engine used by the 'ggtree' package, extracted so that the coordinate computation can be used without a plotting dependency.

License Artistic-2.0

Encoding UTF-8

Depends R (>= 4.2.0)

Imports ape, methods, Rcpp, dplyr, rlang, stats, tibble, tidytree (>= 0.4.5), utils

LinkingTo Rcpp

Suggests igraph, testthat (>= 3.0.0)

Config/testthat/edition 3

ByteCompile true

URL https://github.com/YuLab-SMU/treelayout

BugReports https://github.com/YuLab-SMU/treelayout/issues

Config/roxygen2/version 8.1.1

NeedsCompilation yes

Author Guangchuang Yu [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-6485-8781>)

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Repository CRAN

Date/Publication 2026-11-11 21:30:02 UTC

Contents

calculate_angle . . . 2
calculate_branch_mid . . . 3
clade_geometry . . . 3
Date2decimal . . . 4
decimal2Date . . . 5
getXcoord2 . . . 5
getYcoord . . . 6
get_layout . . . 7
layout_names . . . 7
re_assign_ycoord_df . . . 8
tree_layout . . . 8

Index **10**

`calculate_angle` *Calculate angle coordinates for circular layouts*

Description

Adds an `angle` column holding the label angle for every node, derived from the node's position and its parent.

Usage

```
calculate_angle(data)
```

Arguments

`data`	a node table with `x`, `y`, `node` and `parent` columns.

Value

the node table with an `angle` column added.

Examples

```
d <- tree_layout(ape::rtree(6), layout = "circular")
calculate_angle(d)
```

calculate_branch_mid *Branch mid-point coordinates*

Description

Adds branch (the mid-point x), and for the unrooted layouts also branch.x and branch.y, which place labels on the middle of a branch even when the branch is not axis-aligned.

Usage

```
calculate_branch_mid(res, layout)
```

Arguments

res	a node table with node, parent, x and y columns.
layout	the layout name the coordinates came from; branch.x and branch.y are only computed for the unrooted layouts.

Value

the node table with the branch columns added.

Examples

```
tr <- ape::rtree(6)
d <- tree_layout(tr, layout = "equal_angle")
calculate_branch_mid(d, layout = "equal_angle")
```

clade_geometry *One call instead of three. Returns the left and right boundary angles of the clade subtree, the largest distance from* node *to any node inside the clade, and the largest distance from the root of the whole tree to any node – the two radii the clade label / hilight code needs.*

Description

One call instead of three. Returns the left and right boundary angles of the clade subtree, the largest distance from node to any node inside the clade, and the largest distance from the root of the whole tree to any node – the two radii the clade label / hilight code needs.

Usage

```
clade_geometry(data, node)
```

Arguments

data	a node table with columns node, parent, x, y.
node	the root node id of the clade.

Value

a list with left, right, radius and root_radius.

Examples

```
tr <- ape::rtree(10)
d <- tree_layout(tr, layout = "circular")
clade_geometry(d, node = 12)
```

Date2decimal *convert Date to decimal format, eg "2014-05-05" to "2014.34"*

Description

convert Date to decimal format, eg "2014-05-05" to "2014.34"

Usage

```
Date2decimal(x)
```

Arguments

x	a Date, or a numeric that is passed through unchanged.

Value

numeric year with fraction, e.g. 2014.34.

Author(s)

Guangchuang Yu

Examples

```
Date2decimal(as.Date("2014-05-05"))
```

decimal2Date *convert decimal format to Date, eg "2014.34" to "2014-05-05"*

Description

convert decimal format to Date, eg "2014.34" to "2014-05-05"

Usage

```
decimal2Date(x)
```

Arguments

x	numeric year with fraction, e.g. 2014.34.

Value

a Date.

Author(s)

Guangchuang Yu

Examples

```
decimal2Date(2014.34)
```

getXcoord2 *x coordinates from an edge list*

Description

Computes the x coordinate of every node by walking the edge list from a starting root position. Used by code that recomputes horizontal placement after changing the tree order.

Usage

```
getXcoord2(x, root, parent, child, len, start = 0, rev = FALSE)
```

Arguments

x	numeric vector indexed by node id; the first entry is overwritten with start.
root	node id to start from.
parent	parent node ids of the edge list.
child	child node ids of the edge list.
len	branch length of every edge.
start	x coordinate assigned to the root, default 0.
rev	logical; if TRUE the tree is walked in reverse, so coordinates decrease from the root.

Value

the x vector with one coordinate per node.

Examples

```
tr <- ape::rtree(6)
x <- numeric(max(tr$edge))
getXcoord2(x, root = ape::Ntip(tr) + 1L,
           parent = tr$edge[, 1], child = tr$edge[, 2],
           len = tr$edge.length)
```

---

getYcoord *y coordinates for tips*

Description

Places tips on evenly spaced y positions (or in the order given by tip.order) and fills the internal nodes bottom-up with the mean of their children.

Usage

```
getYcoord(tr, step = 1, tip.order = NULL)
```

Arguments

tr	a phylo object.
step	spacing between adjacent tips, default 1.
tip.order	tip labels in the order they should appear bottom to top; NULL keeps the existing tip order.

Value

numeric vector of y coordinates, indexed like the nodes of tr.

Examples

```
tr <- ape::rtree(6)
getYcoord(tr)
getYcoord(tr, tip.order = rev(tr$tip.label))
```

---

get_layout *Read the layout recorded on a layout result*

Description

Read the layout recorded on a layout result

Usage

```
get_layout(x)
```

Arguments

x	a table returned by tree_layout(), or a plot object whose $data carries the layout attribute.

Value

the layout name, or NULL.

Examples

```
tr <- ape::rtree(10)
get_layout(tree_layout(tr, layout = "circular"))
```

---

layout_names *Layout names accepted by* tree_layout()

Description

Layout names accepted by tree_layout()

Usage

```
layout_names()
```

Value

a character vector of layout names.

Examples

```
layout_names()
```

re_assign_ycoord_df *Re-assign y coordinates*

Description

Fills in missing y values bottom-up: every node whose y is NA gets the mean of its children's y values. Rows with a y value already present are left untouched.

Usage

```
re_assign_ycoord_df(df, currentNode)
```

Arguments

df	a node table with node, parent and y columns.
currentNode	node ids to start the fill from, usually the tips.

Value

the node table with the missing y values replaced.

Examples

```
d <- tree_layout(ape::rtree(6))
d$y[c(2, 3)] <- NA
re_assign_ycoord_df(d, currentNode = d$node[d$isTip])
```

tree_layout *Compute node coordinates for a tree*

Description

Turn a tree into a table of node coordinates, one row per node. This is the layout engine behind 'ggtree', extracted so the coordinates can be computed without any plotting dependency.

Usage

```
tree_layout(
  model,
  data = NULL,
  layout = "rectangular",
  ladderize = TRUE,
  right = FALSE,
  branch.length = "branch.length",
  mrsd = NULL,
  as.Date = FALSE,
```

```
  yscale = "none",
  root.position = 0,
  hang = 0.1,
  ...
)
```

Arguments

model	a tree object: phylo, treedata, multiPhylo, or anything with an as.phylo() method.
data	unused; kept so the signature matches ggplot2::fortify(), which callers such as 'ggtree' pass through.
layout	layout name or a function taking an igraph object and returning an n-by-2 matrix of coordinates.
ladderize	logical, ladderize the tree before layout.
right	logical, ladderize to the right.
branch.length	variable used to scale branch lengths; "none" gives a cladogram.
mrsd	most recent sampling date, for time-scaled trees.
as.Date	logical, whether mrsd should be treated as a Date.
yscale	variable used to scale tips along y; "none" for standard.
root.position	x position of the root.
hang	fraction of the tree height by which labels hang below the rest of the tree, for dendrogram layout of hclust-like input. A negative value makes them hang down from 0.
...	passed on to the layout function.

Value

A tbl_df with one row per node and columns node, parent, x, y, branch, angle, isTip and label, plus any columns carried over from treedata. For a multiPhylo input the results are stacked and an .id column identifies the tree. The attribute "layout" records the layout used ("external" when layout was a function).

Examples

```
tr <- ape::rtree(10)
head(tree_layout(tr))
head(tree_layout(tr, layout = "circular"))
```

Index

calculate_angle, 2
calculate_branch_mid, 3
clade_geometry, 3

Date2decimal, 4
decimal2Date, 5

get_layout, 7
getXcoord2, 5
getYcoord, 6

layout_names, 7

re_assign_ycoord_df, 8

tree_layout, 8
tree_layout(), 7