Package ‘ggtree’

May 29, 2024

Type Package
Title an R package for visualization of tree and annotation data
Version 3.12.0
Maintainer Guangchuang Yu <guangchuangyu@gmail.com>
Description ‘ggtree’ extends the ‘ggplot2’ plotting system which implemented the grammar of graphics. ‘ggtree’ is designed for visualization and annotation of phylogenetic trees and other tree-like structures with their annotation data.
Depends R (>= 3.5.0)
Imports ape, aplot, dplyr, ggplot2 (> 3.3.6), grid, magrittr, methods, purrr, rlang, ggfun (>= 0.0.9), yulab.utils, tidyr, tidytree (>= 0.4.5), treeio (>= 1.8.0), utils, scales, stats, cli
Suggests emojifont, ggimage, ggplotify, shadowtext, grDevices, knitr, prettydoc, rmarkdown, testthat, tibble, glue
VignetteBuilder knitr
ByteCompile true
Encoding UTF-8
License Artistic-2.0
URL https://www.amazon.com/
   Integration-Manipulation-Visualization-Phylogenetic-Computational-ebook/dp/B0B5NLZR1Z/(book),
BugReports https://github.com/YuLab-SMU/ggtree/issues
biocViews Alignment, Annotation, Clustering, DataImport, MultipleSequenceAlignment, Phylogenetics, ReproducibleResearch, Software, Visualization
RoxygenNote 7.3.1
Roxygen list(markdown = TRUE)
git_url https://git.bioconductor.org/packages/ggtree
Contents

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ggtree-package</td>
<td>4</td>
</tr>
<tr>
<td>add_colorbar</td>
<td>5</td>
</tr>
<tr>
<td>applyLayoutDaylight</td>
<td>6</td>
</tr>
<tr>
<td>as.polytomy</td>
<td>6</td>
</tr>
<tr>
<td>collapse.ggtree</td>
<td>7</td>
</tr>
<tr>
<td>Date2decimal</td>
<td>8</td>
</tr>
<tr>
<td>decimal2Date</td>
<td>8</td>
</tr>
<tr>
<td>expand</td>
<td>9</td>
</tr>
<tr>
<td>facet_data</td>
<td>9</td>
</tr>
<tr>
<td>facet_labeller</td>
<td>10</td>
</tr>
<tr>
<td>facet_plot</td>
<td>11</td>
</tr>
<tr>
<td>facet_widths</td>
<td>12</td>
</tr>
<tr>
<td>flip</td>
<td>12</td>
</tr>
<tr>
<td>geom_aline</td>
<td>13</td>
</tr>
<tr>
<td>geom_balance</td>
<td>14</td>
</tr>
<tr>
<td>geom_cladelab</td>
<td>15</td>
</tr>
<tr>
<td>geom_cladelabel</td>
<td>18</td>
</tr>
<tr>
<td>geom_cladelabel2</td>
<td>20</td>
</tr>
<tr>
<td>geom_hilight</td>
<td>21</td>
</tr>
<tr>
<td>geom_inset</td>
<td>24</td>
</tr>
<tr>
<td>geom_label2</td>
<td>25</td>
</tr>
<tr>
<td>geom_motif</td>
<td>27</td>
</tr>
<tr>
<td>geom_nodelab</td>
<td>28</td>
</tr>
<tr>
<td>geom_nodelab2</td>
<td>29</td>
</tr>
<tr>
<td>geom_nodepoint</td>
<td>30</td>
</tr>
<tr>
<td>geom_point2</td>
<td>31</td>
</tr>
<tr>
<td>geom_range</td>
<td>33</td>
</tr>
<tr>
<td>geom_rootedge</td>
<td>34</td>
</tr>
<tr>
<td>Function</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------</td>
</tr>
<tr>
<td>geom_rootpoint</td>
<td>35</td>
</tr>
<tr>
<td>geom_segment2</td>
<td>37</td>
</tr>
<tr>
<td>geom_strp</td>
<td>38</td>
</tr>
<tr>
<td>geom_striplab</td>
<td>40</td>
</tr>
<tr>
<td>geom_taxalink</td>
<td>43</td>
</tr>
<tr>
<td>geom_text2</td>
<td>44</td>
</tr>
<tr>
<td>geom_tiplab</td>
<td>45</td>
</tr>
<tr>
<td>geom_tiplab2</td>
<td>48</td>
</tr>
<tr>
<td>geom_tippoint</td>
<td>49</td>
</tr>
<tr>
<td>geom_tree</td>
<td>50</td>
</tr>
<tr>
<td>geom_tree2</td>
<td>51</td>
</tr>
<tr>
<td>geom_treescale</td>
<td>52</td>
</tr>
<tr>
<td>geom_zoom_clade</td>
<td>53</td>
</tr>
<tr>
<td>get.path</td>
<td>54</td>
</tr>
<tr>
<td>getNodeAngle.df</td>
<td>54</td>
</tr>
<tr>
<td>getNodesBreadthFirst.df</td>
<td>55</td>
</tr>
<tr>
<td>getSubtree</td>
<td>55</td>
</tr>
<tr>
<td>getSubtree.df</td>
<td>56</td>
</tr>
<tr>
<td>getSubtreeUnrooted</td>
<td>56</td>
</tr>
<tr>
<td>getSubtreeUnrooted.df</td>
<td>57</td>
</tr>
<tr>
<td>getTreeArcAngles</td>
<td>57</td>
</tr>
<tr>
<td>get_clade_position</td>
<td>58</td>
</tr>
<tr>
<td>get_heatmap_column_position</td>
<td>58</td>
</tr>
<tr>
<td>get_taxa_name</td>
<td>59</td>
</tr>
<tr>
<td>ggdensitree</td>
<td>60</td>
</tr>
<tr>
<td>ggtree</td>
<td>62</td>
</tr>
<tr>
<td>gheatmap</td>
<td>64</td>
</tr>
<tr>
<td>gzoom</td>
<td>65</td>
</tr>
<tr>
<td>gzoom.phylo</td>
<td>66</td>
</tr>
<tr>
<td>hexexpand</td>
<td>67</td>
</tr>
<tr>
<td>label_pad</td>
<td>68</td>
</tr>
<tr>
<td>layoutDaylight</td>
<td>69</td>
</tr>
<tr>
<td>layoutEqualAngle</td>
<td>69</td>
</tr>
<tr>
<td>layout_rectangular</td>
<td>70</td>
</tr>
<tr>
<td>mssplot</td>
<td>71</td>
</tr>
<tr>
<td>multplot</td>
<td>72</td>
</tr>
<tr>
<td>nodebar</td>
<td>73</td>
</tr>
<tr>
<td>nodepie</td>
<td>73</td>
</tr>
<tr>
<td>open_tree</td>
<td>74</td>
</tr>
<tr>
<td>range_format</td>
<td>75</td>
</tr>
<tr>
<td>reexports</td>
<td>75</td>
</tr>
<tr>
<td>revts</td>
<td>76</td>
</tr>
<tr>
<td>rotate</td>
<td>76</td>
</tr>
<tr>
<td>rotateTreePoints.df</td>
<td>77</td>
</tr>
<tr>
<td>rotate_tree</td>
<td>78</td>
</tr>
<tr>
<td>scaleClade</td>
<td>78</td>
</tr>
<tr>
<td>scale_color</td>
<td>79</td>
</tr>
<tr>
<td>scale_color_subtree</td>
<td>80</td>
</tr>
</tbody>
</table>
Description

'ggtree' extends the 'ggplot2' plotting system which implemented the grammar of graphics. 'ggtree' is designed for visualization and annotation of phylogenetic trees and other tree-like structures with their annotation data.

Author(s)

Maintainer: Guangchuang Yu <guangchuangyu@gmail.com> (ORCID) [copyright holder]

Authors:
- Tommy Tsan-Yuk Lam <tylam.tommy@gmail.com> [thesis advisor]
- Shuangbin Xu <xshuangbin@163.com> (ORCID)

Other contributors:
- Lin Li <williamlee2220@qq.com> [contributor]
- Bradley Jones <bjones@cfenet.ubc.ca> [contributor]
- Justin Silverman <jsilve24@gmail.com> [contributor]
- Watal M. Iwasaki <heavy.watal@gmail.com> [contributor]
- Yonghe Xia <xiayh17@gmail.com> [contributor]
- Ruizhu Huang <ruizhuRH@gmail.com> [contributor]
add_colorbar

See Also

Useful links:

- Report bugs at [https://github.com/YuLab-SMU/ggtree/issues](https://github.com/YuLab-SMU/ggtree/issues)

Description

add colorbar legend

Usage

add_colorbar(p, color, x = NULL, ymin = NULL, ymax = NULL, font.size = 4)

Arguments

- `p`: tree view
- `color`: output of `scale_color` function
- `x`: x position
- `ymin`: ymin
- `ymax`: ymax
- `font.size`: font size

Value

ggplot2 object

Author(s)

Guangchuang Yu
**applyLayoutDaylight**  

**Description**  
Apply the daylight algorithm to adjust the spacing between the subtrees and tips of the specified node.

**Usage**  
applyLayoutDaylight(df, node_id)

**Arguments**  
- **df**: tree data.frame  
- **node_id**: is id of the node from which daylight is measured to the other subtrees.

**Value**  
list with tree data.frame with updated layout using daylight algorithm and max_change angle.

---

**as.polytomy**  

**Description**  
collapse binary tree to polytomy by applying 'fun' to 'feature'

**Usage**  
as.polytomy(tree, feature, fun)

**Arguments**  
- **tree**: tree object, 'phylo' object only  
- **feature**: selected feature  
- **fun**: function to select nodes to collapse

**Value**  
polytomy tree

**Author(s)**  
Guangchuang Yu
Description

collapse a selected clade, which can later be expanded with the 'expand()' function if necessary

Usage

## S3 method for class 'ggtree'
collapse(x = NULL, node, mode = "none", clade_name = NULL, ...)

Arguments

x tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot will be used.
node internal node number
mode one of 'none' (default), 'max', 'min' and 'mixed'. 'none' would simply collapse the clade as 'tip' and the rest will display a triangle, whose shape is determined by the farthest/closest tip of the collapsed clade to indicate it
clade_name set a name for the collapsed clade. If clade_name = NULL, do nothing
... additional parameters to set the color or transparency of the triangle

Value

tree view

Author(s)

Guangchuang Yu

See Also

expand

Examples

x <- rtree(15)
p <- ggtree(x) + geom_tiplab()
p
pl <- collapse(p, node = 17, mode = "mixed",
               clade_name = "cclade", alpha = 0.8,
               color = "grey", fill = "light blue")
**Date2decimal**

**Description**

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

**Usage**

Date2decimal(x)

**Arguments**

- x : Date

**Value**

numeric

**Author(s)**

Guangchuang Yu

---

**decimal2Date**

**Description**

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

**Usage**

decimal2Date(x)

**Arguments**

- x : numerical number, eg 2014.34

**Value**

Date

**Author(s)**

Guangchuang Yu
Description
expand collapsed clade

Usage
expand(tree_view = NULL, node)

Arguments
- tree_view: tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot object will be used.
- node: internal node number to specify a clade. If NULL, using the whole tree

Value
tree view

Author(s)
Guangchuang Yu

See Also
collapse

Examples
```r
x <- rtree(15)
p <- ggtree(x) + geom_tiplab()
p1 <- collapse(p, 17)
expand(p1, 17)
```

Description
extract data used in facet_plot or geom_facet

Usage
```
facet_data(tree_view, panel)
```
facet_labeller

Arguments

- tree_view: ggtree object
- panel: data plotted in specific panel. If only one dataset used in the panel, return the data frame, else return a list of data frames.

Value

data frame or a list of data frames

Author(s)

Guangchuang Yu

References


Description

function to relable selected panels created by `geom_facet` or `facet-plot`

Usage

```
facet_labeller(p, label)
```

Arguments

- p: facet_plot output
- label: new labels of facet panels

Value

ggplot object

Author(s)

Guangchuang Yu
Description

plot tree associated data in an additional panel

Usage

```r
facet_plot(p, mapping = NULL, data, geom, panel, ...)
geom_facet(mapping = NULL, data, geom, panel, ...)
```

Arguments

- `p`: tree view
- `mapping`: aes mapping for 'geom'
- `data`: data to plot by 'geom', first column should be matched with tip label of tree
- `geom`: geom function to plot the data
- `panel`: panel name for plot of input data
- `...`: additional parameters for 'geom'

Details

'facet_plot()' automatically re-arranges the input 'data' according to the tree structure, visualizes the 'data' on specific 'panel' using the 'geom' function with aesthetic 'mapping' and other parameters, and align the graph with the tree 'p' side by side. 'geom_facet' is a 'ggplot2' layer version of 'facet_plot'

Value

ggplot object

Author(s)

Guangchuang Yu

References

Examples

```r
tr <- rtree(10)
dd = data.frame(id=tr$tip.label, value=abs(rnorm(10)))
p <- ggtree(tr)
facet_plot(p, 'Trait', data = dd, geom=geom_point, mapping=aes(x=value))
```

---

**facet_widths**

**facet_widths**

**Description**

set relative widths (for column only) of facet plots

**Usage**

```r
facet_widths(p, widths)
```

**Arguments**

- `p` ggplot or ggtree object
- `widths` relative widths of facet panels

**Value**

ggplot object by redrawing the figure (not a modified version of input object)

**Author(s)**

Guangchuang Yu

---

**flip**

**flip**

**Description**

exchange the position of 2 clades

**Usage**

```r
flip(tree_view = NULL, node1, node2)
```

**Arguments**

- `tree_view` tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot will be used.
- `node1` node number of clade 1. It should share a same parent node with node2
- `node2` node number of clade 2. It should share a same parent node with node1
Value

ggplot object

Author(s)

Guangchuang Yu

Examples

```r
set.seed(123)
x <- rtree(15)
p <- ggtree(x) + geom_tiplab() +
    geom_nodelab(aes(subset=!isTip, label=node), hjust = -.1, color = "red")
flip(p, 23, 24) ## Depends on the condition of your tree
```

Description

add horizontal align lines layer to a tree

Usage

```r
geom_aline(mapping = NULL, linetype = "dotted", linewidth = 1, ...)
```

Arguments

- `mapping`: aes mapping
- `linetype`: set line type of the line, defaults to "dotted"
- `linewidth`: set width of the line, defaults to 1
- `...`: additional parameter

Details

`geom_aline` align all tips to the longest one by adding padding characters to the right side of the tip.

Value

aline layer

Author(s)

Yu Guangchuang
**geom_balance**

**Description**

highlights the two direct descendant clades of an internal node

**Usage**

```r
geom_balance(
    node,
    fill = "steelblue",
    color = "white",
    alpha = 0.5,
    extend = 0,
    extendto = NULL
)
```

**Arguments**

- **node**: selected node (balance) to highlight its two direct descendant
- **fill**: color to fill in the highlight rectangle, default to "steelblue"
- **color**: color to outline highlight rectangle and divide balance, defaults to "white"
- **alpha**: alpha (transparency) for the highlight rectangle, defaults to 0.5
- **extend**: extend xmax of the highlight rectangle by the value of extend
- **extendto**: extend xmax of the highlight rectangle to the value of extendto

**Details**

Particularly useful when studying neighboring clades. Note that balances that correspond to multichotomies will not be displayed.

**Value**

`ggplot2`

**Author(s)**

Justin Silverman and modified by Guangchuang Yu

**References**


Examples
library(ggtree)
set.seed(123)
tr <- rtree(15)
x <- ggtree(tr)
x + geom_balance(17)

Description
annotate a clade with bar and text label or (image)

Usage
geom_cladelab(
  node = NULL,
  label = NULL,
  data = NULL,
  mapping = NULL,
  geom = "text",
  parse = FALSE,
  ...
)

Arguments
node selected node to annotate, when data and mapping is NULL, it is required.
label character, character to be showed, when data and mapping is NULL, it is re-
data data.frame, the data to be displayed in the annotation, defaults to NULL.
mapping Set of aesthetic mappings, defaults to NULL. The detail see the following ex-
geom character, one of 'text', 'label', 'shadowtext', 'image' and 'phylopic', defaults to 'text', and the parameter see the Aesthetics For Specified Geom.
parse logical, whether parse label to emoji font, defaults to FALSE.
... additional parameters, see also following section.

additional parameters can refer the following parameters.
  - offset distance bar and tree, offset of bar and text from the clade, defaults to 0.
  - offset.text distance bar and text, offset of text from bar, defaults to 0.
  - align logical, whether align clade lab, defaults to FALSE.
• extend numeric, extend the length of bar, defaults to 0.
• angle numeric or 'auto', if angle is auto, the angle of text will be calculated automatically, which is useful for the circular etc layout, defaults to 0.
• horizontal logical, whether set label to horizontal, defaults to TRUE.
• barsize the width of line, defaults to 0.5.
• barcolour the colour of line, defaults to 'black'.
• fontsize the size of text, defaults to 3.88.
• textcolour the colour of text, defaults to 'black'.
• imagesize the size of image, defaults to 0.05.
• imagecolor the colour of image, defaults to NULL, when geom="phylopic", it should be required.

The parameters also can be set in mapping, when data is provided. Note: the barsize, barcolour, fontsize, textcolour, imagesize and imagecolor should not be set in mapping (aesthetics). When the color and size are not be set in mapping, user can modify them to adjust the attributes of specified geom.

Aesthetics For Specified Geom

`geom_cladelab()` understands the following aesthetics for geom="text"(required aesthetics are in bold):

• node selected node to high light, it is required.
• label labels to be shown, it is required.
• colour the colour of text, defaults to "black".
• size the size of text, defaults to 3.88.
• angle the angle of text, defaults to 0.
• hjust A numeric vector specifying horizontal justification, defaults to 0.
• vjust A numeric vector specifying vertical justification, defaults to 0.5.
• alpha the transparency of text, defaults to NA.
• family the family of text, defaults to 'sans'.
• face the font face of text, defaults to 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
• lineheight The height of a line as a multiple of the size of text, defaults to 1.2.

when the colour, size are not be set in mapping, and user want to modify the colour of text, they should use textcolour, fontsize to avoid the confusion with bar layer annotation.

`geom_cladelab()` understands the following aesthetics for geom="label" (required aesthetics are in bold):

• node selected node to high light, it is required.
• label labels to be shown, it is required.
• colour the colour of text, defaults to "black".
• fill the background colour of the label, defaults to "white".
• size the size of text, defaults to 3.88.
geom_cladelab

- **angle** the angle of text, defaults to 0.
- **hjust** A numeric vector specifying horizontal justification, defaults to 0.
- **vjust** A numeric vector specifying vertical justification, defaults to 0.5.
- **alpha** the transparency of text, defaults to NA.
- **family** the family of text, defaults to 'sans'.
- **fontface** the font face of text, defaults to 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- **lineheight** The height of a line as a multiple of the size of text, defaults to 1.2.

When the colour, size are not be set in mapping, and user want to modify the colour of text, they should use textcolour, fontsize to avoid the confusion with bar layer annotation.

`geom_cladelab()` understands the following aesthetics for `geom="shadowtext"` (required aesthetics are in bold):

- **node** selected node to highlight, it is required.
- **label** labels to be shown, it is required.
- **colour** the colour of text, defaults to "black".
- **bg.colour** the background colour of text, defaults to 'black'.
- **bg.r** the width of background text, defaults to 0.1.
- **size** the size of text, defaults to 3.88.
- **angle** the angle of text, defaults to 0.
- **hjust** A numeric vector specifying horizontal justification, defaults to 0.
- **vjust** A numeric vector specifying vertical justification, defaults to 0.5.
- **alpha** the transparency of text, defaults to NA.
- **family** the family of text, defaults to 'sans'.
- **fontface** the font face of text, defaults to 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- **lineheight** The height of a line as a multiple of the size of text, defaults to 1.2.

When the colour, size are not be set in mapping, and user want to modify the colour of image, they should use imagecolour, imagesize to avoid the confusion with bar layer annotation.

`geom_cladelab()` understands the following aesthetics for `geom="image"` or `geom="phylopic"` (required aesthetics are in bold):

- **node** selected node to highlight, it is required.
- **label** labels to be shown, it is required.
- **image** the image to be annotated, when `geom="phylopic"`, the uid of phylopic databases, it is required.
- **colour** the color of image, defaults to NULL.
- **size** the size of image, defaults to 0.05.
- **alpha** the alpha of image, defaults to 0.8.

When the colour, size are not be set in mapping, and user want to modify the colour of image, they should use imagecolour, imagesize to avoid the confusion with bar layer annotation.
Examples

```r
set.seed(2015-12-21)
tree <- rtree(30)
data <- data.frame(id=c(34, 56),
                  annote=c("another clade", "long clade names"),
                  image=c("7fb9bea8-e758-4986-afb2-95a2c3bf983d",
                          "0174801d-15a6-4668-bfe0-4c421fbe51e8"),
                  group=c("A", "B"),
                  offset=c(0.1, 0.1),
                  offset.text=c(0.1, 0.2))

p <- ggtree(tree) + xlim(NA, 6)
p + geom_cladelab(node=45, label="test label") +
geom_cladelab(node=34, label="another clade")
p2 <- p + geom_cladelab(data=data,
                         mapping=aes(
                             node=id,
                             label=annote,
                             image=image,
                             color=group,
                             offset=offset
                         ),
                         geom="shadowtext",
                         hjust=0.5,
                         align=TRUE,
                         horizontal=FALSE,
                         angle=90,
                         show.legend = FALSE
)

p2
```

Description

annotate a clade with bar and text label

Usage

```r
geom_cladelabel(
    node,
    label,
    offset = 0,
    offset.text = 0,
    extend = 0,
    align = FALSE,
    barsize = 0.5,
)```
geom_cladelabel

    fontsize = 3.88,
    angle = 0,
    geom = "text",
    hjust = 0,
    color = NULL,
    fill = NA,
    family = "sans",
    parse = FALSE,
    horizontal = TRUE,

    )

Arguments

    node       selected node
    label      clade label
    offset     offset of bar and text from the clade
    offset.text offset of text from bar
    extend     extend bar height
    align      logical
    barsize    size of bar
    fontsize   size of text
    angle      angle of text
    geom       one of 'text' or 'label'
    hjust      justify text horizontally
    color      color for clade & label, of length 1 or 2
    fill       fill label background, only work with geom='label'
    family     sans by default, can be any supported font
    parse      logical, whether parse label
    horizontal logical, whether set label to horizontal, defaults to TRUE.

Value

ggplot layers

Author(s)

Guangchuang Yu

See Also

    geom_cladelabel2
Description

annotate a clade with bar and text label

Usage

```r
gem.m.cladelabel2(
  node,
  label,
  offset = 0,
  offset.text = 0,
  offset.bar = 0,
  align = FALSE,
  barsize = 0.5,
  fontsize = 3.88,
  hjust = 0,
  geom = "text",
  color = NULL,
  family = "sans",
  parse = FALSE,
  horizontal = TRUE,
  ... )
```

Arguments

- **node**: selected node
- **label**: clade label
- **offset**: offset of bar and text from the clade
- **offset.text**: offset of text from bar
- **offset.bar**: offset of bar from text
- **align**: logical
- **barsize**: size of bar
- **fontsize**: size of text
- **hjust**: justify text horizontally
- **geom**: one of 'text' or 'label'
- **color**: color for clade & label, of length 1 or 2
- **family**: sans by default, can be any supported font
- **parse**: logical, whether parse label
- **horizontal**: logical, whether set label to horizontal, defaults to TRUE.
- **...**: additional parameter
Value

ggplot layers

Author(s)

JustGitting

See Also

geom_cladelabel

Description

layer of hilight clade

Usage

geom_hilight(
  data = NULL,
  mapping = NULL,
  node = NULL,
  type = "auto",
  to.bottom = FALSE,
  ...
)

geom_highlight(
  data = NULL,
  mapping = NULL,
  node = NULL,
  type = "auto",
  to.bottom = FALSE,
  ...
)

Arguments

data data.frame, The data to be displayed in this layer, defaults to NULL.
mapping Set of aesthetic mappings, defaults to NULL.
node selected node to hilight, when data and mapping is NULL, it is required.
geom_hilight

- **type** the type of layer, defaults to auto, meaning rectangular, circular, slanted, fan, inward_circular, radial, equal_angle, ape layout tree will use rectangular layer, unrooted and daylight layout tree use will use encircle layer. You can specify this parameter to `rect` (rectangular layer) or `encircle` (encircle layer), 'gradient' (gradient color), 'roundrect' (round rectangular layer).

- **to.bottom** logical, whether set the high light layer to the bottom in all layers of 'ggtree' object, default is FALSE.

... additional parameters, see also the below and Aesthetics section.

- **align** control the align direction of the edge of high light rectangular. Options is 'none' (default), 'left', 'right', 'both'. This argument only work when the 'geom_hilight' is plotting using `geom_hilight(mapping=aes(...))`.

- **gradient.direction** character, the direction of gradient color, defaults to 'rt' meaning the locations of gradient color is from root to tip, options are 'rt' and 'tr'.

- **gradient.length.out** integer, desired length of the sequence of gradient color, defaults to 2.

- **roundrect.r** numeric, the radius of the rounded corners, when `roundrect=TRUE`, defaults to 0.05.

**Details**

`geom_hilight` supports data.frame as input. And aesthetics of layer can be mapped. you can see the Aesthetics section to set parameters.

**Value**

a list object.

**Aesthetics**

`geom_hilight()` understands the following aesthetics for rectangular layer (required aesthetics are in bold):

- **node** selected node to hight light, it is required.
- **colour** the colour of margin, defaults to NA.
- **fill** the colour of fill, defaults to 'steelblue'.
- **alpha** the transparency of fill, defaults to 0.5.
- **extend** extend xmax of the rectangle, defaults to 0.
- **extendto** specify a value, meaning the rectangle extend to, defaults to NULL.
- **linetype** the line type of margin, defaults to 1.
- **linewidth** the width of line of margin, defaults to 0.5.

`geom_hilight()` understands the following aesthetics for encircle layer (required aesthetics are in bold):

- **node** selected node to hight light, it is required.
geom_hilight

- colour the colour of margin, defaults to 'black'.
- fill the colour of fill, defaults to 'steelblue'.
- alpha the transparency of fill, defaults to 0.5.
- expand expands the xspline clade region, defaults to 0.
- spread control the size, when only one point.
- linewidth the width of line of margin, defaults to 0.5.
- linetype the line type of margin, defaults to 1.
- s_shape the shape of the spline relative to the control points, defaults to 0.5.
- s_open whether the spline is a line or a closed shape, defaults to FALSE.

Author(s)

Guangchuang Yu and Shuangbin Xu

References


Examples

```r
library(ggplot2)
set.seed(102)
tree <- rtree(60)
p <- ggtree(tree)
p1 <- p + geom_hilight(node=62) + geom_hilight(node=88, fill="red")
p1
dat <- data.frame(id=c(62, 88), type=c("A", "B"))
p2 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type))
p2
p3 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), align="left")
p4 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), align="right")
p5 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), align="both")
# display the high light layer with gradiental color rectangular.
p6 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), type="gradient", alpha=0.68)
p7 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type),
                        type="gradient", gradient.direction="tr", alpha=0.68)
# display the high light layer with round rectangular.
p8 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), type="roundrect", alpha=0.68)
```
Description

gemo_inset can add subplots to tree by accepting a list of ggplot objects that are ancestral stats or data associated with selected nodes in the tree. These ggplot objects can be any kind of charts or hybrid of these charts.

add subplots as insets in a tree

Usage

geom_inset(
  insets,
  width = 0.1,
  height = 0.1,
  hjust = 0,
  vjust = 0,
  x = "node",
  reverse_x = FALSE,
  reverse_y = FALSE
)

inset(
  tree_view,
  insets,
  width,
  height,
  hjust = 0,
  vjust = 0,
  x = "node",
  reverse_x = FALSE,
  reverse_y = FALSE
)

Arguments

insets a list of ggplot objects, named by node number
width width of the inset, relative to the range of x-axis, defaults to .1
height height of the inset, relative to the range of y-axis, defaults to .1
hjust adjust the horizontal position of the charts, charts will go left if hjust > 0
vjust adjust the vertical position of the charts, charts will go down if vjust > 0
x the position where users want to place the charts, one of ’node’ (default) and ’branch’
Details

Users can also use

Value

inset layer
tree view with insets

Author(s)

Guangchuang Yu

References

For demonstration of this function, please refer to chapter 8.3 of Data Integration, Manipulation and Visualization of Phylogenetic Trees http://yulab-smu.top/treedata-book/index.html by Guangchuang Yu.

Description

geom_label2 support aes(subset) via setup_data

Usage

geom_label2(
  mapping = NULL,
  data = NULL,
  ...,
  stat = "identity",
  position = "identity",
  family = "sans",
  parse = FALSE,
  nudge_x = 0,
  nudge_y = 0,
  label.padding = unit(0.25, "lines"),
  label.r = unit(0.15, "lines"),
  label.size = 0.25,
Arguments

- mapping: Set of aesthetic mappings, defaults to NULL.
- data: A layer specific dataset - only needed if you want to override the plot defaults.
- ... other arguments passed on to 'layer'.
- stat: Name of the stat to modify data.
- position: The position adjustment to use for overlapping points on this layer.
- family: "sans" by default, can be any supported font.
- parse: if 'TRUE', the labels will be parsed as expressions, defaults to 'FALSE'.
- nudge_x: adjust the horizontal position of the labels.
- nudge_y: adjust the vertical position of the labels.
- label.padding: Amount of padding around label, defaults to 'unit(0.25, "lines")'.
- label.r: Use to set the radius of rounded corners of the label, defaults to 'unit(0.15, "lines")'.
- label.size: Size of label border, in mm, defaults to 0.25.
- na.rm: If "FALSE" (default), missing values are removed with a warning. If "TRUE", missing values are silently removed, logical.
- show.legend: Whether to show legend, logical, defaults to "NA".
- inherit.aes: Whether to inherit aesthetic mappings, logical, defaults to "TRUE".

Details

'geom_label2' is a modified version of geom_label, with subset aesthetic supported

Value

label layer

Author(s)

Guangchuang Yu

References

For more detailed demonstration of this function, please refer to chapter A.4.5 of Data Integration, Manipulation and Visualization of Phylogenetic Trees http://yulab-smu.top/treedata-book/index.html by Guangchuang Yu.

See Also

geom_label
Examples

```r
library(ggtree)
set.seed(123)
tr <- rtree(15)
x <- ggtree(tr)
x + geom_label2(aes(label = node, subset = isTip == FALSE))
```

Description

geom layer to draw aligned motif

Usage

```r
geom_motif(mapping, data, on, label, align = "centre", ...)
```

Arguments

- `mapping`: aes mapping
- `data`: data
- `on`: gene to center (i.e. set middle position of the `on` gene to 0)
- `label`: specify a column to be used to label genes
- `align`: where to place gene label, default is 'centre' and can be set to 'left' and 'right'
- `...`: additional parameters

Value

geom layer

Author(s)

Guangchuang Yu
Description

add node label layer for a tree

Usage

```r
geom_nodelab(
  mapping = NULL,
  nudge_x = 0,
  nudge_y = 0,
  geom = "text",
  hjust = 0.5,
  node = "internal",
  ...
)
```

Arguments

- **mapping**: aesthetic mappings, defaults to NULL
- **nudge_x**: horizontal adjustment to nudge labels, defaults to 0
- **nudge_y**: vertical adjustment to nudge labels, defaults to 0
- **geom**: one of 'text', 'shadowtext', 'label', 'image' and 'phylopic'
- **hjust**: horizontal alignment, defaults to 0.5
- **node**: a character indicating which node labels will be displayed, it should be one of 'internal', 'external' and 'all'. If it is set to 'internal' will display internal node labels, 'external' will display the tip labels, and 'all' will display internal node and tip labels.
- **...**: additional parameters, see also the additional parameters of `geom_tiplab()`.

Value

gem layer

Author(s)

Guangchuang Yu

References

See Also

geom_tiplab()

Description

add node label for circular layout

Usage

geom_nodelab2(
  mapping = NULL,
  nudge_x = 0,
  nudge_y = 0,
  geom = "text",
  hjust = 0.5,
  ...
)

Arguments

mapping    aesthetic mappings, defaults to NULL
nudge_x    horizontal adjustment to nudge labels, defaults to 0
nudge_y    vertical adjustment to nudge labels, defaults to 0
geom       one of 'text', 'shadowtext', 'label', 'image' and 'phylopic'
hjust      horizontal alignment, defaults to 0.5
...         additional parameters, see also the additional parameters of geom_tiplab().

Value

node label layer

Author(s)

Guangchuang Yu
**Description**

add node point layer to a tree

**Usage**

```r
gem geom_nodepoint(
    mapping = NULL,
    data = NULL,
    position = "identity",
    na.rm = FALSE,
    show.legend = NA,
    inherit.aes = TRUE,
    ...
)
```

**Arguments**

- `mapping` Set of aesthetic mapping created by `aes()`. If `inherit.aes = TRUE`, the mapping can be inherited from the plot mapping as specified in the call to `ggplot()`.
- `data` The data to be displayed in this layer. If 'NULL' (the default), the data is inherited from the plot data as specified in the call to `ggplot()`.
- `position` Position adjustment.
- `na.rm` logical. If 'FALSE' (default), missing values are removed with a warning. If 'TRUE', missing values are silently removed.
- `show.legend` logical. Should this layer be included in the legends? 'NA', the default, includes if any aesthetics are mapped. 'FALSE' never includes, and 'TRUE' always includes.
- `inherit.aes` logical (defaults to 'TRUE'). If 'FALSE', overrides the default aesthetics, rather then combining with them.
- `...` additional parameters that passed on to this layer. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`.

**Value**

node point layer

**Author(s)**

Guangchuang Yu library(ggtree) tr<- rtree(15) x <- ggtree(tr) x + geom_nodepoint()
References


---

**geom_point2**

**Description**

`geom_point2` is a modified version of `geom_point` that supports `aes(subset)`

**Usage**

```r
geom_point2(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

**Arguments**

- `mapping` Set of aesthetic mapping created by `aes()`. If `inherit.aes = TRUE`, the mapping can be inherited from the plot mapping as specified in the call to `ggplot()`.
- `data` The data to be displayed in this layer. If 'NULL' (the default), the data is inherited from the plot data as specified in the call to `ggplot()`.
- `stat` Name of the statistical transformation to be used on the data for this layer.
- `position` Position adjustment.
- `na.rm` logical. If 'FALSE' (default), missing values are removed with a warning. If 'TRUE', missing values are silently removed.
- `show.legend` logical. Should this layer be included in the legends? 'NA', the default, includes if any aesthetics are mapped. 'FALSE' never includes, and 'TRUE' always includes.
- `inherit.aes` logical (defaults to 'TRUE'). If 'FALSE', overrides the default aesthetics, rather then combining with them.
- `...` additional parameters that passed on to this layer. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`.
Details

`geom_point2` creates scatterplots, just similar to `ggplot2::geom_point`. It extends the `ggplot2::geom_point` to support filtering via the subset aesthetic mapping (see Aesthetics section).

Value

point layer

Aesthetics

`geom_point2()` understands the following aesthetics

- subset logical expression indicating elements or rows to keep: missing values are taken as false; should be in aes().
- colour the colour of point, defaults to "black".
- fill the colour of fill, defaults to "black".
- alpha the transparency of fill, defaults to 1.
- size the size of point, defaults to 1.5.
- shape specify a shape, defaults to 19.
- stroke control point border thickness of point, defaults to 0.5.

Author(s)

Guangchuang Yu

References


For more information about tree visualization, please refer to the online book https://yulab-smu.top/treedata-book/

See Also

`geom_point`; `geom_rootpoint` add point of root; `geom_nodepoint` add points of internal nodes; `geom_tippoint` add points of external nodes (also known as tips or leaves).
Examples

library(ggtree)
## add point by aes(subset)
tr <- rtree(10)
# group tip and node
ggtree(tr) + geom_point2(aes(shape=isTip, color=isTip), size=3)
# specify a node to display
ggtree(tr) + geom_point2(aes(subset=(node==15)), shape=21, size=5, fill='green')
# specify a tip to display
ggtree(tr) + geom_point2(aes(subset=(label %in% c("t1", "t3"))), shape=21, size=5, fill='green')

## color point with continuous variables
library(ggtree)
library(treeio)
library(ggplot2)
beast_file <- system.file("examples/MCC_FluA_H3.tree", package="ggtree")
beast_tree <- read.beast(beast_file)
p <- ggtree(beast_tree) +
geom_tiplab(hjust = -.1)+
geom_nodepoint(aes(fill = rate), shape = 21, size = 4) +
scale_fill_continuous(low = 'blue', high = 'red') +
theme_tree2() + theme(legend.position = 'right')
p

geom_range

Description
horizontal bar of range (HPD, range etc) on nodes to present uncertainty of evolutionary inference

Usage
geom_range(range, center = "auto", ...)

Arguments

- **range**: range(interval) to be displayed, e.g. "height_0.95_HPD"
- **center**: center of the range, mean, median or auto (default, the center of the range)
- **...**: additional parameter, e.g. color, linewidth, alpha

Value
ggplot layer

Author(s)
Guangchuang Yu
References

For demonstration of this function, please refer to chapter 5.2.4 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* [http://yulab-smu.top/treedata-book/index.html](http://yulab-smu.top/treedata-book/index.html) by Guangchuang Yu.

---

**Description**

display root edge layer for a tree

**Usage**

```r
geom_rootedge(rootedge = NULL, ...)
```

**Arguments**

- `rootedge` length of rootedge; use `phylo$root.edge` if `rootedge = NULL` (default).
- `...` additional parameters

Additional parameters can be referred to the following parameters:

- `size` control the width of rootedge, defaults to 0.5.
- `colour` color of rootedge, defaults to black.
- `linetype` the type of line, defaults to 1.
- `alpha` modify colour transparency, defaults to 1.

**Details**

ggtree rootedge is used to create a rootedge as ggtree doesn’t plot the root edge by default.

**Value**

ggtree rootedge layer

**Author(s)**

Guangchuang Yu

**References**

Examples

```r
library(ggtree)
set.seed(123)
## with root edge = 1
tree1 <- read.tree(text='((A:1,B:2):3,C:2):1;')
ggtree(tree1) + geom_tiplab() + geom_rootedge()

## without root edge

## setting root edge
tree2$root.edge <- 2
ggtree(tree2) + geom_tiplab() + geom_rootedge()

## specify length of root edge for just plotting
## this will ignore tree$root.edge
ggtree(tree2) + geom_tiplab() + geom_rootedge(rootedge = 3)
```

## For more detailed demonstration of this function, please refer to chapter A.4.5 of
## *Data Integration, Manipulation and Visualization of Phylogenetic Trees*

---

### geom_rootpoint

geom_rootpoint is used to add root point layer to a tree

#### Usage

```r
gem_rootpoint(
  mapping = NULL,
  data = NULL,
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

#### Arguments

- `mapping`: Set of aesthetic mapping created by `aes()`. If `inherit.aes = TRUE`, the mapping can be inherited from the plot mapping as specified in the call to `ggplot()`.
The data to be displayed in this layer. If 'NULL' (the default), the data is inherited from the plot data as specified in the call to ggplot().

Position adjustment.

logical. If 'FALSE' (default), missing values are removed with a warning. If 'TRUE', missing values are silently removed.

logical. Should this layer be included in the legends? 'NA', the default, includes if any aesthetics are mapped. 'FALSE' never includes, and 'TRUE' always includes.

logical (defaults to 'TRUE'). If 'FALSE', overrides the default aesthetics, rather then combining with them.

additional parameters that passed on to this layer. These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3.

Details

geom_rootpoint inherit from geom_point2, and it is used to display and customize the points on the root

Value

root point layer

Author(s)

Guangchuang Yu

References


For more information, please refer to the online book: Data Integration, Manipulation and Visualization of Phylogenetic Trees. http://yulab-smu.top/treedata-book/

See Also

geom_point; geom_rootpoint add point of root; geom_nodepoint add points of internal nodes; geom_tippoint add points of external nodes (also known as tips or leaves).
Examples

```
library(ggtree)
tr <- rtree(10)
## add root point
ggtree(tr) + geom_rootpoint()
ggtree(tr) + geom_rootpoint(size=2,color="red",shape=2)
```

Description

`geom_segment2` support `aes(subset)` via `setup_data`

Usage

```
geom_segment2(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  lineend = "butt",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  nudge_x = 0,
  arrow = NULL,
  arrow.fill = NULL,
  ...
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mapping</td>
<td>Set of aesthetic mappings, defaults to <code>NULL</code></td>
</tr>
<tr>
<td>data</td>
<td>A layer specific dataset - only needed if you want to override the plot defaults.</td>
</tr>
<tr>
<td>stat</td>
<td>Name of stat to modify data.</td>
</tr>
<tr>
<td>position</td>
<td>The position adjustment to use for overlapping points on this layer.</td>
</tr>
<tr>
<td>lineend</td>
<td>Line end style, one of butt (default), round and square.</td>
</tr>
<tr>
<td>na.rm</td>
<td>If &quot;FALSE&quot; (default), missing values are removed with a warning. If &quot;TRUE&quot;, missing values are silently removed, logical.</td>
</tr>
<tr>
<td>show.legend</td>
<td>Whether to show legend, logical.</td>
</tr>
<tr>
<td>inherit.aes</td>
<td>Whether to inherit aesthetic mappings, logical, defaults to &quot;TRUE&quot;.</td>
</tr>
<tr>
<td>nudge_x</td>
<td>adjust the horizontal position of the segments.</td>
</tr>
<tr>
<td>arrow</td>
<td>specification for arrow heads, as created by arrow().</td>
</tr>
<tr>
<td>arrow.fill</td>
<td>fill color to use for the arrow head (if closed). NULL means use colour aesthetic.</td>
</tr>
<tr>
<td>...</td>
<td>additional parameter</td>
</tr>
</tbody>
</table>
Details

'geom_segment2' is a modified version of geom_segment, with subset aesthetic supported

Value

add segment layer

Author(s)

Guangchuang Yu

See Also

geom_segment

description

annotate associated taxa (from taxa1 to taxa2, can be Monophyletic, Polyphyletic or Paraphyletic Taxa) with bar and (optional) text label

Usage

geom_strip(
  taxa1,  
taxa2,  
label,  
offset = 0,  
offset.text = 0,  
align = TRUE,  
barsize = 0.5,  
extend = 0,  
fontsize = 3.88,  
angle = 0,  
geom = "text",  
hjust = 0,  
color = "black",  
fill = NA,  
family = "sans",  
pseudo = FALSE,  
...  
)
**geom_strip**

**Arguments**

- `taxa1`  
- `taxa2`  
- `label` add label alongside the bar (optional)  
- `offset` offset of bar and text from the clade  
- `offset.text` offset of text from bar  
- `align` logical, whether to align bars to the most distant bar. Defaults to "TRUE". Note that if "FALSE", the bars might cross the tree  
- `barsize` set size of the bar  
- `extend` extend bar length vertically  
- `fontsize` set size of the text  
- `angle` set the angle of text  
- `geom` one of 'text' or 'label'  
- `hjust` adjust the horizontal position of the bar  
- `color` set color for bar and label  
- `fill` set color to fill label background, only work with geom='label'  
- `family` "sans" by default, can be any supported font  
- `parse` logical, whether to parse labels, if "TRUE", the labels will be parsed into expressions, defaults to "FALSE"  
- ... additional parameter

**Value**

ggplot layers

**Author(s)**

Guangchuang Yu

**References**

For more detailed demonstration of this function, please refer to chapter 5.2.1 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* [http://yulab-smu.top/treedata-book/index.html](http://yulab-smu.top/treedata-book/index.html) by Guangchuang Yu.

**Examples**

```r
library(ggtree)
tr <- rtree(15)
x <- ggtree(tr)
x + geom_strip(13, 1, color = "red") + geom_strip(3, 7, color = "blue")
```
Description

annotate associated taxa (from taxa1 to taxa2, can be Monophyletic, Polyphyletic or Paraphyletic Taxa) with bar and (optional) text label or image

Usage

```r
geom_striplab(
  taxa1 = NULL,
  taxa2 = NULL,
  label = NULL,
  data = NULL,
  mapping = NULL,
  geom = "text",
  parse = FALSE,
  ...
)
```

Arguments

taxa1 can be label or node number
taxa2 can be label or node number
label character, character to be showed, when data and mapping is NULL, it is required.
data data.frame, the data to be displayed in the annotation, default is NULL.
mapping Set of aesthetic mappings, default is NULL. The detail see the following explanation.
geom character, one of 'text', 'label', 'shadowtext', 'image' and 'phylopic', default is 'text', and the parameter see the Aesthetics For Specified Geom.
parse logical, whether parse label to emoji font, default is FALSE.
... additional parameters, see also following section.
additional parameters can refer the following parameters. ##'

- offset distance bar and tree, offset of bar and text from the clade, default is 0.
- offset.text distance bar and text, offset of text from bar, default is 0.
- align logical, whether align clade lab, default is FALSE.
- extend numeric, extend the length of bar, default is 0.
- angle numeric or 'auto', if angle is auto, the angle of text will be calculated automatically, which is useful for the circular etc layout, default is 0.
- horizontal logical, whether set label to horizontal, default is TRUE.
• `barsize` the width of line, default is 0.5.
• `barcolour` the colour of line, default is 'black'.
• `fontsize` the size of text, default is 3.88.
• `textcolour` the colour of text, default is 'black'.
• `imagesize` the size of image, default is 0.05.
• `imagecolor` the colour of image, default is NULL, when `geom="phylopic"`, it should be required.

The parameters also can be set in mapping, when data is provided. Note: the `barsize`, `barcolour`, `fontsize`, `textcolour`, `imagesize` and `imagecolor` should not be set in mapping (aesthetics). When the color and size are not be set in mapping, user can modify them to adjust the attributes of specified geom.

**Aesthetics For Specified Geom**

gem_striplab() understands the following aesthetics for `geom="text"` (required aesthetics are in bold):

- `taxa1` selected tip label or tip node, it is required.
- `taxa2` selected another tip label or tip node, it is required.
- `label` labels to be shown, it is required.
- `colour` the colour of text, default is "black".
- `size` the size of text, default is 3.88.
- `angle` the angle of text, default is 0.
- `hjust` A numeric vector specifying horizontal justification, default is 0.
- `vjust` A numeric vector specifying vertical justification, default is 0.5.
- `alpha` the transparency of text, default is NA.
- `family` the family of text, default is 'sans'.
- `fontface` the font face of text, default is 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- `lineheight` The height of a line as a multiple of the size of text, default is 1.2.

When the colour, size are not be set in mapping, and user want to modify the colour of text, they should use `textcolour`, `fontsize` to avoid the confusion with bar layer annotation.

gem_striplab() understands the following aesthetics for `geom="label"` (required aesthetics are in bold):

- `taxa1` selected node to hight light, it is required.
- `taxa2` selected another tip label or tip node, it is required.
- `label` labels to be shown, it is required.
- `colour` the colour of text, default is "black".
- `fill` the background colour of the label, default is "white".
- `size` the size of text, default is 3.88.
- `angle` the angle of text, default is 0.
- `hjust` A numeric vector specifying horizontal justification, default is 0.
geom_striplab

- vjust A numeric vector specifying vertical justification, default is 0.5.
- alpha the transparency of text, default is NA.
- family the family of text, default is 'sans'.
- fontface the font face of text, default is 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- lineheight The height of a line as a multiple of the size of text, default is 1.2.

When the colour, size are not be set in mapping, and user want to modify the colour of text, they should use textcolour, fontsize to avoid the confusion with bar layer annotation.

geom_striplab() understands the following aesthetics for geom="shadowtext" (required aesthetics are in bold):

- taxa1 selected node to highlight, it is required.
- taxa2 selected another tip label or tip node, it is required.
- label labels to be shown, it is required.
- colour the colour of text, default is "black".
- bg.colour the background colour of text, default is 'black'.
- bg.r the width of background text, default is 0.1.
- size the size of text, default is 3.88.
- angle the angle of text, default is 0.
- hjust A numeric vector specifying horizontal justification, default is 0.
- vjust A numeric vector specifying vertical justification, default is 0.5.
- alpha the transparency of text, default is NA.
- family the family of text, default is 'sans'.
- fontface the font face of text, default is 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- lineheight The height of a line as a multiple of the size of text, default is 1.2.

When the colour, size are not be set in mapping, and user want to modify the colour of text, they should use textcolour, fontsize to avoid the confusion with bar layer annotation.

geom_striplab() understands the following aesthetics for geom="image" or geom="phylopic" (required aesthetics are in bold):

- taxa1 selected node to highlight, it is required.
- taxa2 selected another tip label or tip node, it is required.
- label labels to be shown, it is required.
- image the image to be annotated, when geom="phylopic", the uid of phylopic databases, it is required.
- colour the color of image, default is NULL.
- size the size of image, default is 0.05.
- alpha the alpha of image, default is 0.8.

When the colour, size are not be set in mapping, and user want to modify the colour of image, they should use imagecolour, imagesize to avoid the confusion with bar layer annotation.
Examples

```r
set.seed(123)
tr <- rtree(10)
dt <- data.frame(ta1=c("t5", "t1"), ta2=c("t6", "t3"), group=c("A", "B"))
p <- ggplot(tr) + geom_tiplab()
p2 <- p +
  geom_striplab(
    data = dt,
    mapping = aes(taxa1 = ta1, taxa2 = ta2,
                  label = group, color=group),
    align = TRUE,
    show.legend = FALSE
  )
p2
```

Description

`geom_taxalink` supports data.frame as input, the colour, linewidth, linetype and alpha can be mapped. When the data was provided, the mapping should be also provided, which `taxa1` and `taxa2` should be mapped created by `aes`, `aes_` or `aes_string`. In addition, the `hratio`, control the height of curve line, when tree layout is circular, default is 1. `ncp`, the number of control points used to draw the curve, more control points creates a smoother curve, default is 1. They also can be mapped to a column of data.

Usage

```r
geom_taxalink(
  data = NULL,
  mapping = NULL,
  taxa1 = NULL,
  taxa2 = NULL,
  offset = NULL,
  outward = "auto",
  ...
)
```

Arguments

data: data.frame, The data to be displayed in this layer, default is NULL.
mapping: Set of aesthetic mappings, default is NULL.
taxa1: can be label or node number.
taxa2: can be label or node number.
offset: numeric, control the shift of curve line (the ratio of axis value, range is "(0-1)"), default is NULL.
outward logical, control the orientation of curve when the layout of tree is circular, fan or
other layout in polar coordinate, default is "auto", meaning it will automatically.
additional parameter.

Value
a list object.

Aesthetics

geom_taxalink() understands the following aesthetics (required aesthetics are in bold):

• taxa1 label or node number of tree.
• taxa2 label or node number of tree.
• group group category of link.
• colour control the color of line, default is black.
• linetype control the type of line, default is 1 (solid).
• linewidth control the width of line, default is 0.5.
• curvature control the curvature of line, default is 0.5, it will be created automatically in polar
coordinate.
• hratio control the height of curve line, default is 1.
• ncp control the smooth of curve line, default is 1.

Description

geom_text2 support aes(subset) via setup_data

Usage

geom_text2(
  mapping = NULL,
  data = NULL,
  ...,
  stat = "identity",
  position = "identity",
  family = "sans",
  parse = FALSE,
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE,
  nudge_x = 0,
  nudge_y = 0,
  check_overlap = FALSE
)
geom_tiplab

Arguments

- **mapping**: the aesthetic mapping
- **data**: A layer specific dataset - only needed if you want to override the plot defaults.
- **...**: other arguments passed on to `layer`
- **stat**: Name of stat to modify data
- **position**: The position adjustment to use for overlapping points on this layer
- **family**: sans by default, can be any supported font
- **parse**: if TRUE, the labels will be passed into expressions
- **na.rm**: logical
- **show.legend**: logical
- **inherit.aes**: logical
- **nudge_x**: horizontal adjustment
- **nudge_y**: vertical adjustment
- **check_overlap**: if TRUE, text that overlaps previous text in the same layer will not be plotted

Value
text layer

Author(s)
Guangchuang Yu

See Also
geom_text

Description
add tip label layer for a tree

Usage

```r
gem_tiplab(
  mapping = NULL,
  hjust = 0,
  align = FALSE,
  linetype = "dotted",
  linesize = 0.5,
  geom = "text",
  offset = 0,
  as_ylab = FALSE,
  ...
)
```
Arguments

mapping  
aes mapping
hjust  
horizontal adjustment, defaults to 0
align  
if TRUE, align all tip labels to the longest tip by adding padding characters to the left side of tip labels, defaults to "FALSE" with a line connecting each tip and its corresponding label, defaults to "FALSE"
linetype  
set linetype of the line if align = TRUE, defaults to "dotted"
linesize  
set line width if align = TRUE, defaults to 0.5
geom  
one of 'text', 'label', 'shadowtext', 'image' and 'phylopic'
offset  
tiplab offset, horizontal adjustment to nudge tip labels, defaults to 0
as_ylab  
display tip labels as y-axis label, only works for rectangular and dendrogram layouts, defaults to "FALSE"

additional parameter

additional parameters can refer the following parameters.
The following parameters for geom="text".
- size control the size of tip labels, defaults to 3.88.
- colour control the colour of tip labels, defaults to "black".
- angle control the angle of tip labels, defaults to 0.
- vjust A numeric vector specifying vertical justification, defaults to 0.5.
- alpha the transparency of text, defaults to NA.
- family the family of text, defaults to 'sans'.
- fontface the font face of text, defaults to 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- lineHeight The height of a line as a multiple of the size of text, defaults to 1.2.
- nudge_x horizontal adjustment to nudge labels, defaults to 0.
- nudge_y vertical adjustment to nudge labels, defaults to 0.
- check.overlap if TRUE, text that overlaps previous text in the same layer will not be plotted.
- parse if TRUE, the labels will be parsed into expressions, if it is 'emoji', the labels will be parsed into emojifont.

The following parameters for geom="label".
- size the size of tip labels, defaults to 3.88.
- colour the colour of tip labels, defaults to "black".
- fill the colour of rectangular box of labels, defaults to "white".
- vjust numeric vector specifying vertical justification, defaults to 0.5.
- alpha the transparency of labels, defaults to NA.
- family the family of text, defaults to 'sans'.
- fontface the font face of text, defaults to 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- lineHeight The height of a line as a multiple of the size of text, defaults to 1.2.
• nudge_x horizontal adjustment to nudge labels, defaults to 0.
• nudge_y vertical adjustment, defaults to 0.
• check.overlap if TRUE, text that overlaps previous text in the same layer will not be plotted.
• parse if TRUE, the labels will be parsed into expressions, if it is ‘emoji’, the labels will be parsed into emojifont.
• label.padding Amount of padding around label, defaults to 'unit(0.25, "lines")'.
• label.r Radius of rounded corners, defaults to 'unit(0.15, "lines")'.
• label.size Size of label border, in mm, defaults to 0.25.

The following parameters for geom="shadowtext", some parameters are like to geom="text".
• bg.colour the background colour of text, defaults to "black".
• bg.r the width of background of text, defaults to 0.1.

The following parameters for geom="image" or geom="phylopic".
• image the image file path for geom='image', but when geom='phylopic', it should be the uid of phylopic databases.
• size the image size, defaults to 0.05.
• colour the color of image, defaults to NULL.
• alpha the transparency of image, defaults to 0.8.

The following parameters for the line when align = TRUE.
• colour the colour of line, defaults to 'black'.
• alpha the transparency of line, defaults to NA.
• arrow specification for arrow heads, as created by arrow(), defaults to NULL.
• arrow.fill fill color to use for the arrow head (if closed), defaults to 'NULL', meaning use 'colour' aesthetic.

Details
‘geom_tiplab’ not only supports using text or label geom to display tip labels, but also supports image geom to label tip with image files or phylopics.

For adding tip labels to a tree with circular layout, ‘geom_tiplab’ will automatically adjust the angle of the tip labels to the tree by internally calling ‘geom_tiplab2’.

Value
tip label layer

Author(s)
Guangchuang Yu

References
For more detailed demonstration, please refer to chapter 4.3.3 of Data Integration, Manipulation and Visualization of Phylogenetic Trees http://yulab-smu.top/treedata-book/index.html by Guangchuang Yu.
Examples

```r
require(ape)
tr <- rtree(10)
ggtree(tr) + geom_tiplab()
```

Description

add tip label for circular layout

Usage

```r
gem_tiplab2(mapping = NULL, hjust = 0, ...)
```

Arguments

- `mapping` aes mapping
- `hjust` horizontal adjustment, defaults to 0
- `...` additional parameter, see `geom_tiplab`

Details

'geom_tiplab2' will automatically adjust the angle of the tip labels to the tree with circular layout

Value

tip label layer

Author(s)

Guangchuang Yu

References

https://groups.google.com/forum/#!topic/bioc-ggtree/o35PV3iHO-0

See Also

geom_tiplab

Examples

```r
library(ggtree)
set.seed(123)
tr <- rtree(10)
ggtree(tr, layout = "circular") + geom_tiplab2()
```
Description

add tip point layer for a tree

Usage

```r
gem_mtippoint(
mapping = NULL,
data = NULL,
position = "identity",
na.rm = FALSE,
show.legend = NA,
inherit.aes = TRUE,
...
)
```

Arguments

- **mapping**: Set of aesthetic mapping created by `aes()`. If `inherit.aes = TRUE`, the mapping can be inherited from the plot mapping as specified in the call to `ggplot()`.
- **data**: The data to be displayed in this layer. If `NULL` (the default), the data is inherited from the plot data as specified in the call to `ggplot()`.
- **position**: Position adjustment.
- **na.rm**: logical. If `FALSE` (default), missing values are removed with a warning. If `TRUE`, missing values are silently removed.
- **show.legend**: logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes.
- **inherit.aes**: logical (defaults to `TRUE`). If `FALSE`, overrides the default aesthetics, rather than combining with them.
- **...**: additional parameters that passed on to this layer. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`.

Value

Tip point layer

Author(s)

Guangchuang Yu
References


Examples

```r
library(ggtree)
tr <- rtree(15)
x <- ggtree(tr)
x + geom_tippoint()
```

### Description

add tree layer

### Usage

```r
gem_tree(
mapping = NULL,
data = NULL,
layout = "rectangular",
multiPhylo = FALSE,
continuous = "none",
position = "identity",
...
)
```

### Arguments

- **mapping**: aesthetic mapping
- **data**: data of the tree
- **multiPhylo**: logical, whether input data contains multiple phylo class, defaults to ”FALSE”.
- **continuous**: character, continuous transition for selected aesthetic (’size’ or ’color’(’colour’)). It should be one of ’color’ (or ’colour’), ’size’, ’all’ and ’none’, default is ’none’
- **position**: Position adjustment, either as a string, or the result of a call to a position adjustment function, default is “identity”.
- **...**: additional parameter

Some dot arguments:

- **nsplit**: integer, the number of branch blocks divided when ’continuous’ is not ”none”, default is 200.
geom_tree2

Value
tree layer

Aesthetics

geom_tree() understands the following aesthetics:

- color character, control the color of line, default is black (continuous is "none").
- linetype control the type of line, default is 1 (solid).
- linewidth numeric, control the width of line, default is 0.5 (continuous is "none").

Author(s)

Yu Guangchuang

References

For demonstration of this function, please refer to chapter 4.2.1 of Data Integration, Manipulation and Visualization of Phylogenetic Trees http://yulab-smu.top/treedata-book/index.html by Guangchuang Yu.

Examples

tree <- rtree(10)
ggplot(tree) + geom_tree()

Description

add tree layer

Usage

geom_tree2(layout = "rectangular", ...)

Arguments

- layout character, one of 'rectangular', 'slanted', 'circular', 'radial' or 'unrooted'
- ... additional parameter

Value

tree layer

Author(s)

Yu Guangchuang
Description

add tree scale to a tree

Usage

```r
geom_treescale(
  x = NULL,
  y = NULL,
  width = NULL,
  offset = NULL,
  offset.label = NULL,
  label = NULL,
  color = "black",
  linesize = 0.5,
  fontsize = 3.88,
  family = "sans"
)
```

Arguments

- `x` set x position of the scale
- `y` set y position of the scale
- `width` set the length of the tree scale
- `offset` set offset of text to line, defaults to NULL
- `offset.label` set offset of the scale title to line.
- `label` set the title of tree scale, defaults to NULL.
- `color` set color of the scale
- `linesize` set size of line
- `fontsize` set size of text
- `family` 'sans' by default, can be any supported font

Details

'geom_treescale’ automatically adds a scale bar for evolutionary distance

Value

ggplot layers

Author(s)

Guangchuang Yu
### References

For demonstration of this function, please refer to chapter 4.3.1 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* [http://yulab-smu.top/treedata-book/index.html](http://yulab-smu.top/treedata-book/index.html) by Guangchuang Yu.

---

**Description**

zoom selected clade of a tree

**Usage**

```r
gemm_zoom_clade(node, xexpand = NULL)
```

**Arguments**

- `node`: internal node number to zoom in its corresponding clade
- `xexpand`: numeric, extend x, meaning the ratio of range of the xlim of the original tree, defaults to NULL.

**Details**

'geom_zoom_clade' zooms in on a selected clade of a tree, while showing its on the full view of tree as a separated panel for reference

**Value**

updated tree view

**Author(s)**

Guangchuang Yu
get.path

Description
path from start node to end node

Usage
get.path(phylo, from, to)

Arguments
phylo phylo object
from start node
to end node

Value
node vector

Author(s)
Guangchuang Yu

getNodeAngle.df

Description
Get the angle between the two nodes specified.

Usage
getNodeAngle.df(df, origin_node_id, node_id)

Arguments
df tree data.frame
origin_node_id origin node id number
node_id end node id number

Value
angle in range \([-1, 1]\), i.e. degrees/180, radians/pi
getNodesBreadthFirst.df

Description
Get the nodes of tree from root in breadth-first order.

Usage
getNodesBreadthFirst.df(df)

Arguments
df tree data.frame

Value
list of node id’s in breadth-first order.

getSubtree

Description
Get all children of node from tree, including start_node.

Usage
gSubtree(tree, node)

Arguments
tree ape phylo tree object
node is the tree node id from which the tree is derived.

Value
list of all child node id’s from starting node.
**getSubtree.df**

**Description**
Get all children of node from df tree using breath-first.

**Usage**

```r
getSubtree.df(df, node)
```

**Arguments**

- `df`: tree data.frame
- `node`: id of starting node.

**Value**

list of all child node id’s from starting node.

---

**getSubtreeUnrooted**

**Description**
Get all subtrees of specified node. This includes all ancestors and relatives of node and return named list of subtrees.

**Usage**

```r
getSubtreeUnrooted(tree, node)
```

**Arguments**

- `tree`: ape phylo tree object
- `node`: is the tree node id from which the subtrees are derived.

**Value**

named list of subtrees with the root id of subtree and list of node id’s making up subtree.
getSubtreeUnrooted.df

**Description**
Get all subtrees of node, as well as remaining branches of parent (ie, rest of tree structure as subtree) return named list of subtrees with list name as starting node id.

**Usage**
```r
getSubtreeUnrooted.df(df, node)
```

**Arguments**
- `df` : tree data.frame
- `node` : is the tree node id from which the subtrees are derived.

**Value**
named list of subtrees with the root id of subtree and list of node id’s making up subtree.

getTreeArcAngles

**Description**
Find the right (clockwise rotation, angle from +ve x-axis to furthest subtree nodes) and left (anti-clockwise angle from +ve x-axis to subtree) Returning arc angle in \([0, 2]\) (0 to 360) domain.

**Usage**
```r
getTreeArcAngles(df, origin_id, subtree)
```

**Arguments**
- `df` : tree data.frame
- `origin_id` : node id from which to calculate left and right hand angles of subtree.
- `subtree` : named list of root id of subtree (node) and list of node ids for given subtree (subtree).

**Value**
named list with right and left angles in range \([0, 2]\) i.e 1 = 180 degrees, 1.5 = 270 degrees.
get_clade_position

Description
get position of clade (xmin, xmax, ymin, ymax)

Usage
get_clade_position(treeview, node)

Arguments

treeview	tree view
node	selected node

Value
data.frame

Author(s)
Guangchuang Yu

get_heatmap_column_position

Description
return a data.frame that contains position information for labeling column names of heatmap produced by gheatmap function

Usage
get_heatmap_column_position(treeview, by = "bottom")

Arguments

treeview	output of gheatmap
by	one of 'bottom' or 'top'

Value
data.frame
**get.taxa.name**

**Author(s)**

Guangchuang Yu

---

**Description**

get taxa name of a selected node (or tree if node=NULL) sorted by their position in plotting

**Usage**

```r
get.taxa.name(tree_view = NULL, node = NULL)
```

**Arguments**

- `tree_view`:
  
  tree view (i.e. the ggtree object). If `tree_view` is NULL, the last ggplot object will be used.

- `node`:
  
  internal node number to specify a clade. If NULL, using the whole tree

**Details**

This function extract an ordered vector of the tips from selected clade or the whole tree based on the ggtree() plot.

**Value**

ordered taxa name vector

**Author(s)**

Guangchuang Yu

**Examples**

```r
tree <- rtree(30)
p <- ggtree(tree)
get.taxa.name(p)
```
Description

drawing phylogenetic trees from list of phylo objects

Usage

ggdensitree(
  data = NULL,
  mapping = NULL,
  layout = "slanted",
  tip.order = "mode",
  align.tips = TRUE,
  jitter = 0,
  ...
)

Arguments

data a list of phylo objects or any object with an as.phylo and fortify method
mapping aesthetic mapping
layout one of 'slanted', 'rectangular', 'fan', 'circular' or 'radial' (default: 'slanted')
tip.order the order of the tips by a character vector of taxa names; or an integer, N, to order the tips by the order of the tips in the Nth tree; 'mode' to order the tips by the most common order; 'mds' to order the tips based on MDS of the path length between the tips; or 'mds_dist' to order the tips based on MDS of the distance between the tips (default: 'mode')
align.tips TRUE (default) to align trees by their tips and FALSE to align trees by their root
jitter deviation to jitter tips
... additional parameters passed to fortify, ggtree and geom_tree

Details

The trees plotted by `ggdensitree()` will be stacked on top of each other and the structures of the trees will be rotated to ensure the consistency of the tip order.

Value

tree layer

Author(s)

Yu Guangchuang, Bradley R. Jones
References

For more detailed demonstration of this function, please refer to chapter 4.4.2 of Data Integration, Manipulation and Visualization of Phylogenetic Trees http://yulab-smu.top/treedata-book/index.html by Guangchuang Yu.

Examples

```r
require(ape)
require(dplyr)
require(tidyr)

# Plot multiple trees with aligned tips
trees <- list(read.tree(text="((a:1,b:1):1.5,c:2.5);"), read.tree(text="((a:1,c:1):1,b:2);"));
ggdensitree(trees) + geom_tiplab()

# Plot multiple trees with aligned tips with tip labels and separate tree colors
trees.fort <- list(trees[[1]] %>% fortify %>% 
mutate(tree="a"), trees[[2]] %>% fortify %>% mutate(tree="b"));
ggdensitree(trees.fort, aes(colour=tree)) + geom_tiplab(colour = 'black')

# Generate example data
set.seed(1)
random.trees <- rmrree(5, 10)
time.trees <- lapply(seq_along(random.trees), function(i) {
  tree <- random.trees[[i]]
  tree$tip.label <- paste0("t", 1:10)
  dates <- estimate.dates(tree, 1:10, mu=1, nsteps=1)
  tree$edge.length <- dates[[tree$edge[, 2]]] - dates[[tree$edge[, 1]]]
  fortify(tree) %>% mutate(tree=factor(i, levels=as.character(1:10)))
})

# Plot multiple trees with aligned tips from multiple time points
ggdensitree(time.trees, aes(colour=tree), tip.order=paste0("t", 1:10)) + geom_tiplab(colour = 'black')

# Read example data
example.trees <- read.tree(system.file("examples", "ggdensitree_example.tree", package="ggtree"))

# Compute OTU
grp <- list(A = c("a.t1", "a.t2", "a.t3", "a.t4"),
           B = c("b.t1", "b.t2", "b.t3", "b.t4"),
           C = c("c.t1", "c.t2", "c.t3", "c.t4"))

otu.trees <- lapply(example.trees, groupOTU, grp)

# Plot multiple trees colored by OTU
ggdensitree(otu.trees, aes(colour=group), alpha=1/6, tip.order='mds') +
  scale_colour_manual(values=c("black", "red", "green", "blue"))
```

```
ggtree

drawing phylogenetic tree from phylo object

Usage

ggtree(
  tr,  
  mapping = NULL,  
  layout = "rectangular",  
  open.angle = 0,  
  mrsd = NULL,  
  as.Date = FALSE,  
  yscale = "none",  
  yscale_mapping = NULL,  
  ladderize = TRUE,  
  right = FALSE,  
  branch.length = "branch.length",  
  root.position = 0,  
  xlim = NULL,  
  layout.params = list(),  
  hang = 0.1,  
  ...  
)

Arguments

tr                  phylo object
mapping             aesthetic mapping
layout              one of 'rectangular', 'dendrogram', 'slanted', 'ellipse', 'roundrect', 'fan', 'circular', 'inward_circular', 'radial', 'equal_angle', 'daylight' or 'ape'
open.angle          open angle, only for 'fan' layout
mrsd                most recent sampling date
as.Date             logical whether using Date class in time tree
yscale               y scale
yscale_mapping      yscale mapping for category variable
ladderize           logical (default TRUE). Should the tree be re-organized to have a 'ladder' aspect?
right               logical. If ladderize = TRUE, should the ladder have the smallest clade on the right-hand side? See ape::ladderize() for more information.
branch.length       variable for scaling branch, if 'none' draw cladogram
root.position       position of the root node (default = 0)
`xlim` x limits, only works for 'inward_circular' layout

`layout.params` list, the parameters of layout, when layout is a function.

`hang` numeric The fraction of the tree plot height by which labels should hang below the rest of the plot. A negative value will cause the labels to hang down from 0. This parameter only work with the 'dendrogram' layout for 'hclust' like class, default is 0.1.

... additional parameter

some dot arguments:

- `nsplit` integer, the number of branch blocks divided when 'continuous' is not "none", default is 200.

Value
tree

Author(s)
Yu Guangchuang

References


See Also

- `geom_tree()`

Examples

```r
c <- rtree(10)
ggtree(c)
```
Description

append a heatmap of a matrix to the right side of a phylogenetic tree

Usage

gheatmap(
p, data, offset = 0, width = 1, low = "green", high = "red", color = "white", colnames = TRUE, colnames_position = "bottom", colnames_angle = 0, colnames_level = NULL, colnames_offset_x = 0, colnames_offset_y = 0, font.size = 4, family = "", hjust = 0.5, legend_title = "value", custom_column_labels = NULL)

Arguments

p tree view
data matrix or data.frame
offset set offset of the heatmap to tree
width total width of heatmap, compare to width of tree, defaults to 1, which means they are of the same length
low set color of the lowest value, defaults to "green"
high set color of the highest value, defaults to "red"
color set color of heatmap cell border, defaults to "white"
colnames logical, whether to add matrix colnames, defaults to "TRUE"
colnames_position set the position of the colnames, one of 'bottom' (default) or 'top'
colnames_angle set the angle of colnames
colnames_level  set levels of colnames
colnames_offset_x  
    set x offset for colnames
colnames_offset_y  
    set y offset for colnames
font.size  set font size of matrix colnames
family  font of matrix colnames, can be any supported font
hjust  adjust horizontal position of column names (0: align left, 0.5: align center (default), 1: align right)
legend_title  title of fill legend

custom_column_labels  
    instead of using the colnames from the input matrix/data.frame, input a custom vector to be set as column labels

Value

tree view

Author(s)

Guangchuang Yu

References

For demonstration of this function, please refer to chapter 7.3 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* [http://yulab-smu.top/treedata-book/index.html](http://yulab-smu.top/treedata-book/index.html) by Guangchuang Yu.

---

gzoom  
gzoom method

description

gzoom method

gzoom method

zoom selected subtree

Usage

```r
.gzoom(object, focus, subtree = FALSE, widths = c(0.3, 0.7), ...)
```

```r
## S4 method for signature 'ggtree'
.gzoom(object, focus, widths = c(0.3, 0.7), xmax_adjust = 0)
```

```r
## S4 method for signature 'treedata'
```
gzoom(object, focus, subtree = FALSE, widths = c(0.3, 0.7))

## S4 method for signature 'phylo'
gzoom(object, focus, subtree = FALSE, widths = c(0.3, 0.7))

### Arguments
- **object**: supported tree objects
- **focus**: selected tips
- **subtree**: logical
- **widths**: widths
- ... additional parameter
- **xmax_adjust**: adjust xmax (xlim[2])

### Value
- **figure**

---

**Description**
plots simultaneously a whole phylogenetic tree and a portion of it.

**Usage**
gzoom.phylo(phy, focus, subtree = FALSE, widths = c(0.3, 0.7))

### Arguments
- **phy**: phylo object
- **focus**: selected tips
- **subtree**: logical
- **widths**: widths

### Value
- a list of ggplot object

**Author(s)**
ygc
Description

hexpand
vexpand
expand xlim (ylim) by ratio of x (y) axis range

Usage

hexpand(ratio, direction = 1)
vexpand(ratio, direction = 1)
ggexpand(ratio, direction = 1, side = "hv")

Arguments

ratio
expand x (y) axis limits by amount of xrange (yrange) * ratio
direction
expand x axis limit at right hand side if direction is 1 (default), or left hand side if direction is -1
side
one of ‘h’ for horizontal and ‘v’ for vertical or ‘hv’ for both (default).

Value

ggexpand object

Author(s)

Guangchuang Yu

Examples

x <- rtree(20)
x$tip.label <- paste0('RRRRREEEEEAAAAALLLLLYYYY_Long_Lable_', x$tip.label)
p1 <- ggtree(x) + geom_tiplab()
p1 + gexpand(1.5, side = "h")
Description

This function adds padding characters to the left side of taxa labels, adjust their length to the longest label.

Usage

`label_pad(label, justify = "right", pad = ".")`

Arguments

- `label`: taxa label
- `justify`: should a character vector be right-justified (default), left-justified, centred or left alone.
- `pad`: padding character (defaults to dots)

Value

Taxa labels with padding characters added

Author(s)

Guangchuang Yu and Yonghe Xia

References

[https://groups.google.com/g/bioc-ggtree/c/INJ0Nfkq3b0/m/lXefnfV5AQAJ](https://groups.google.com/g/bioc-ggtree/c/INJ0Nfkq3b0/m/lXefnfV5AQAJ)

Examples

```r
library(ggtree)
set.seed(2015-12-21)
tree <- rtree(5)
tree$tip.label[2] <- "long string for test"
label_pad(tree$tip.label)
```
**layoutDaylight**

*Equal daylight layout method for unrooted trees.*

**Description**

# @title

**Usage**

```
layoutDaylight(model, branch.length, MAX_COUNT = 5)
```

**Arguments**

- **model**: tree object, e.g. phylo or treedata
- **branch.length**: set to 'none' for edge length of 1. Otherwise the phylogenetic tree edge length is used.
- **MAX_COUNT**: the maximum number of iterations to run (default 5)

**Value**

tree as data.frame with equal angle layout.

**References**

The following algorithm aims to implement the vague description of the "Equal-daylight Algorithm" in "Inferring Phylogenies" pp 582-584 by Joseph Felsenstein.

*Leafs are subtrees with no children
Initialise tree using equal angle algorithm
```
tree_df = equal_angle(tree)
```

```
nodes = get list of nodes in tree_df breadth-first
nodes = remove tip nodes.
```
**Arguments**

- `model`: tree object, e.g. phylo or treedata
- `branch.length`: set to 'none' for edge length of 1. Otherwise the phylogenetic tree edge length is used.

**Value**

tree as data.frame with equal angle layout.

**References**

"Inferring Phylogenies" by Joseph Felsenstein.

**Description**

- transform circular/fan layout to rectangular layout
- transform rectangular layout to circular layout
- transform rectangular/circular layout to inward circular layout
- transform rectangular/circular layout to fan layout
- transform rectangular layout to dendrogram layout

**Usage**

```r
layout_rectangular()
layout_circular()
layout_inward_circular(xlim = NULL)
layout_fan(angle = 180)
layout_dendrogram()
```

**Arguments**

- `xlim`: setting x limits, which will affect the center space of the tree
- `angle`: open tree at specific angle

**Author(s)**

Guangchuang Yu
Examples

tree <- rtree(20)
p <- ggtree(tree, layout = "circular") + layout_rectangular()
tree <- rtree(20)
p <- ggtree(tree)
p + layout_circular()
tree <- rtree(20)
p <- ggtree(tree)
p + layout_inward_circular(xlim=4) + geom_tiplab(hjust=1)
tree <- rtree(20)
p <- ggtree(tree)
p + layout_fan(angle=90)
tree <- rtree(20)
p <- ggtree(tree)
p + p + layout_dendrogram()

Description

visualize phylogenetic tree with multiple sequence alignment

Usage

msaplot(
p, fasta, offset = 0, width = 1, color = NULL, window = NULL, bg_line = TRUE, height = 0.8
)

Arguments

p tree view
fasta fasta file that contains multiple sequence alignment information
offset set the offset of MSA to tree
width total width of alignment, compare to width of tree, defaults to 1, which means they are of the same length
color set color of the tree
window specific a slice of alignment to display
bg_line whether to add background line in alignment, defaults to “TRUE”
height height ratio of sequence, defaults to 0.8
Value

tree view

Author(s)

Guangchuang Yu

References

For demonstration of this function, please refer to chapter 7.4 of Data Integration, Manipulation and Visualization of Phylogenetic Trees http://yulab-smu.top/treedata-book/index.html by Guangchuang Yu.

Description

plot multiple ggplot objects in one page

Usage

multiplot(
    ..., 
    plotlist = NULL, 
    ncol, 
    widths = rep_len(1, ncol), 
    labels = NULL, 
    label_size = 5
)

Arguments

... plots
plotlist plot list
ncol set the number of column to display the plots
widths the width of each plot
labels set labels for labeling the plots
label_size set font size of the label

Value

plot

Author(s)

Guangchuang Yu
nodebar

Description
generate a list of bar charts for results of ancestral state reconstruction

Usage
nodebar(data, cols, color, alpha = 1, position = "stack")

Arguments
data: a data.frame of stats with an additional column of node number named "node"
cols: columns of the data.frame that store the stats
color: set color of bars
alpha: set transparency of the charts
position: position of bars, if 'stack' (default) make bars stacked atop one another, 'dodge' make them dodged side-to-side

Value
list of ggplot objects

Author(s)
Guangchuang Yu

nodepie

Description
generate a list of pie charts for results of ancestral stat reconstruction

Usage
nodepie(
data, cols, color, alpha = 1, outline.color = "transparent", outline.size = 0
)
**Arguments**

- `data`: a data.frame of stats with an additional column of node number named "node"
- `cols`: columns of the data.frame that store the stats
- `color`: set color of bars
- `alpha`: set transparency of the charts
- `outline.color`: color of outline
- `outline.size`: size of outline

**Value**

list of ggplot objects

**Author(s)**

Guangchuang Yu

---

**Description**

transform a tree in either rectangular or circular layout into the fan layout that opens with a specific angle

**Usage**

`open_tree(treeview, angle)`

**Arguments**

- `treeview`: tree view in rectangular/circular layout
- `angle`: open the tree at a specific angle

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

**Examples**

```r
tree <- rtree(15)
p <- ggtree(tree) + geom_tiplab()
open_tree(p, 180)
```
range_format

---

**range_format**

---

**Description**

format a list of range (HPD, CI, etc that has length of 2)

**Usage**

```r
range_format(x, trans = NULL)
```

**Arguments**

- `x`: input list
- `trans`: transformation function

**Value**

character vector of [lower, upper]

**Author(s)**

Guangchuang Yu

---

**reexports**

*Objects exported from other packages*

---

**Description**

These objects are imported from other packages. Follow the links below to see their documentation.

- **ape** `read.tree, rtree`
- **aplot** `plot_list`
- **dplyr** `collapse`
- **ggfun** `identify`
- **ggplot2** `aes, fortify, geom_label, geom_point, geom_text, ggplot, ggsave, guide_legend, margin, scale_color_manual, scale_colour_manual, scale_fill_manual, scale_x_continuous, theme, xlim`
- **grid** `arrow, unit`
- **magrittr** `%>%`
- **tidytree** `groupClade, groupOTU, MRCA, nodeid, nodelab`
### revts

**Description**
reverse timescale x-axis by setting the most recent tip to 0

**Usage**
```
revts(treeview)
```

**Arguments**
- `treeview`: original tree view

**Details**
'scale_x_continuous(labels=abs)' is required if users want to set the x-axis label to absolute value

**Value**
updated tree view

**Author(s)**
Guangchuang Yu

**Examples**
```
tr <- rtree(10)
p <- ggtree(tr) + theme_tree2()
p2 <- revts(p)
p2 + scale_x_continuous(labels=abs)
```

### rotate

**Description**
rotate selected clade by 180 degree

**Usage**
```
rotate(tree_view = NULL, node)
```
Arguments

tree_view tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot object will be used.

node internal node number to specify a clade. If NULL, using the whole tree

Value
ggplot2 object

Author(s)
Guangchuang Yu

Examples

x <- rtree(15)
p <- ggtree(x) + geom_tiplab()
rotate(p, 17)

Description

Rotate the points in a tree data.frame around a pivot node by the angle specified.

Usage

rotateTreePoints.df(df, pivot_node, nodes, angle)

Arguments

df tree data.frame

pivot_node is the id of the pivot node.

nodes list of node numbers that are to be rotated by angle around the pivot_node

angle in range \([0, 2]\), i.e. degrees/180, radians/pi

Value

updated tree data.frame with points rotated by angle
**rotate_tree**

**Description**

rotate circular tree in a certain angle

**Usage**

`rotate_tree(treeview, angle)`

**Arguments**

- `treeview`: tree view in circular layout
- `angle`: the angle of rotation

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

**Examples**

```r
tree <- rtree(15)
p <- ggtree(tree) + geom_tiplab()
p2 <- open_tree(p, 180)
rotate_tree(p2, 180)
```

**scaleClade**

**Description**

zoom out/in a selected clade to emphasize or de-emphasize it

**Usage**

`scaleClade(tree_view = NULL, node, scale = 1, vertical_only = TRUE)`
Arguments

- **tree_view**: tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot object will be used.
- **node**: internal node number to specify a clade. If NULL, using the whole tree
- **scale**: the scale of the selected clade. The clade will be zoom in when scale > 1, and will be zoom out when scale < 1
- **vertical_only**: logical. If TRUE (default), only vertical will be scaled. If FALSE, the clade will be scaled vertical and horizontally.

Value
tree view

Author(s)
Guangchuang Yu

Examples

```r
x <- rtree(15)
p <- ggtree(x) + geom_tiplab() +
geom_nodelab(aes(subset=!isTip, label=node), hjust = -.1, color = "red")
scaleClade(p, 24, scale = .1)
```

Description

scale_color method

scale color by a numerical tree attribute

Usage

```r
scale_color(object, by, ...)
```

## S4 method for signature 'treedata'
scale_color(object, by, ...)

## S4 method for signature 'phylo'
scale_color(object, by, ...)

Arguments

- **object**: treedata object
- **by**: one of numerical attributes
- **...**: additional parameter
**scale_color_subtree**

**Description**

scale tree color by subtree (e.g., output of cutree, kmeans, or other clustering algorithm)

**Usage**

```r
scale_color_subtree(group)
```

**Arguments**

- `group` : taxa group information

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

---

**scale_x_ggtree**

**Description**

scale x for tree with gheatmap

**Usage**

```r
scale_x_ggtree(breaks = waiver(), labels = waiver())
```

**Arguments**

- `breaks` : set breaks for tree
- `labels` : labels for corresponding breaks
**Details**

Since setting x-axis for tree with gheatmap by using 'theme_tree2()' is quite tricky, 'scale_x_ggtree' can help set the x-axis more reasonably.

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

**References**

For more detailed demonstration of this function, please refer to chapter 7.3 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* [http://yulab-smu.top/treedata-book/index.html](http://yulab-smu.top/treedata-book/index.html) by Guangchuang Yu.

---

**Description**

add second x-axis for geom_range

**Usage**

scale_x_range()

**Details**

notice that the first axis is disabled in the default theme thus users need to enable it first before using scale_x_range

**Value**

ggtree object

**Author(s)**

Guangchuang Yu

**References**

For demonstration of this function, please refer to chapter 5.2.4 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* [http://yulab-smu.top/treedata-book/index.html](http://yulab-smu.top/treedata-book/index.html) by Guangchuang Yu.
Description

set legend for multiple geom_hilight layers

Usage

set_hilight_legend(p, color, label, alpha = 1)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>ggtree object</td>
</tr>
<tr>
<td>color</td>
<td>color vector</td>
</tr>
<tr>
<td>label</td>
<td>label vector</td>
</tr>
<tr>
<td>alpha</td>
<td>transparency of color</td>
</tr>
</tbody>
</table>

Value

updated ggtree object

Author(s)

Guangchuang Yu
### Description
Filter data for tree annotation layer

### Usage

```r
td_filter(..., .f = NULL)
```

### Arguments

- `...` Expressions that return a logical value.
- `.f` a function (if any, defaults to NULL) that pre-operate the data

### Details

The `td_filter()` function returns another function that can be used to subset `ggtree()` plot data. The function can be passed to the `data` parameter of `geom` layer to perform subsetting. All rows that satisfy your conditions will be retained.

### Value

A function to filter `ggtree` plot data using conditions defined by `...`.

### Author(s)

Guangchuang Yu

### References

For more detailed demonstration of this function, please refer to chapter 12.5.1 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* [http://yulab-smu.top/treedata-book/index.html](http://yulab-smu.top/treedata-book/index.html) by Guangchuang Yu.

### See Also

`filter`

### Examples

```r
tree <- rtree(30)
## similar to 'ggtree(tree) + geom_tippoint()'
ggtree(tree) + geom_point(data = td_filter(isTip))
```
td_mutate  td-mutate

Description
mutate data for tree annotation layer

Usage
td_mutate(..., .f = NULL)

Arguments

... additional parameters that pass to dplyr::mutate
.f a function (if any, defaults to NULL) that pre-operate the data

Details
The 'td_mutate()' function returns another function that can be used to mutate ggtree() plot data. The function can be passed to the 'data' parameter of geom layer to perform adding new variables and preserving existing ones.

Value
A function to mutate ggtree plot data

See Also
mutate

---

td_unnest  td-unnest

Description
flattens a list-column of data frame

Usage
td_unnest(cols, ..., .f = NULL)

Arguments

cols columns to unnest
... additional parameters that pass to tidyr::unnest
.f a function (if any, defaults to NULL) that pre-operate the data
Details
The `td_unnest` function returns another function that can be used to unnest ggtree() plot data. The function can be passed to the 'data' parameter of a geom layer to flatten list-column tree data.

Value
A function to unnest ggtree plot data

Author(s)
Guangchuang Yu

References
For demonstration of this function, please refer to chapter 12.5.2 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* [http://yulab-smu.top/treedata-book/index.html](http://yulab-smu.top/treedata-book/index.html) by Guangchuang Yu.

See Also
unnest

---

theme_dendrogram

Description
dendrogram theme

Usage
theme_dendrogram(bgcolor = "white", fgcolor = "black", ...)

Arguments
bgcolor set background color, defaults to "white"
fgcolor set color of axis
... additional parameter

Author(s)
Guangchuang Yu
theme_inset

Description
inset theme

Usage
theme_inset(legend.position = "none", ...)

Arguments
legend.position
set the position of legend
...
additional parameter

Details
theme for inset function

Value
ggplot object

Author(s)
Guangchuang Yu

theme_tree

Description
tree theme

Usage
theme_tree(bgcolor = "white", ...)

Arguments
bgcolor
set background color, defaults to "white"
...
additional parameter
Details

'theme_tree' defines a blank background to display tree

Value

updated ggplot object with new theme

Author(s)

Guangchuang Yu

Examples

```r
require(ape)
tr <- rtree(10)
ggtree(tr) + theme_tree()
```

---

**theme_tree2**

**theme_tree2**

Description

tree2 theme

Usage

```r
theme_tree2(bgcolor = "white", fgcolor = "black", ...)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgcolor</td>
<td>set background color, defaults to &quot;white&quot;</td>
</tr>
<tr>
<td>fgcolor</td>
<td>set foreground color, defaults to &quot;black&quot;</td>
</tr>
<tr>
<td>...</td>
<td>additional parameter</td>
</tr>
</tbody>
</table>

Details

'theme_tree2' supports displaying phylogenetic distance by setting x-axis

Value

updated ggplot object with new theme

Author(s)

Guangchuang Yu
Examples

```r
require(ape)
tr <- rtree(10)
ggtree(tr) + theme_tree2()
```

Description

view a selected clade of tree, clade can be selected by specifying a node number or determined by the most recent common ancestor of selected tips

Usage

```r
viewClade(tree_view = NULL, node, xmax_adjust = 0)
```

Arguments

- **tree_view**: tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot object will be used.
- **node**: internal node number to specify a clade. If NULL, using the whole tree
- **xmax_adjust**: adjust the max range of x axis

Value

clade plot

Author(s)

Guangchuang Yu

Examples

```r
x <- rtree(15)
p <- ggtree(x) + geom_tiplab()
viewClade(p, 18, xmax_adjust = 0.)
```
xlim_expand

Description
expand x axis limits for specific panel

Usage
xlim_expand(xlim, panel)

Arguments
- xlim: x axis limits
- panel: name of the panel to expand

Value
updated tree view

Author(s)
Guangchuang Yu

Examples
x <- rtree(30)
p <- ggtree(x) + geom_tiplab()
d <- data.frame(label = x$tip.label,
value = rnorm(30))
p2 <- p + geom_facet(panel = "Dot", data = d,
geom = geom_point, mapping = aes(x = value))
p2 + xlim_expand(c(-10, 10), 'Dot')

xlim_tree

Description
set x axis limits specially for Tree panel

Usage
xlim_tree(xlim)
Arguments

\texttt{xlim} \quad x \text{ axis limits}

Value

updated tree view

Author(s)

Guangchuang Yu

Examples

\texttt{x <- rtree(30)}
\texttt{p <- ggtree(x) + geom_tiplab()}
\texttt{d <- data.frame(label = x$tip.label,}
\texttt{ value = rnorm(30))}
\texttt{p2 <- p + geom_facet(panel = "Dot", data = d,}
\texttt{ geom = geom_point, mapping = aes(x = value))}
\texttt{p2 + xlim_tree(6)}

Description

zoom in on a selected clade of a tree, while showing its on the full view of tree as a seperated panel for reference

Usage

\texttt{zoomClade(tree\_view = NULL, node, xexpand = NULL)}

Arguments

\texttt{tree\_view} \quad tree view (i.e. the ggtree object). If \texttt{tree\_view} is \texttt{NULL}, the last ggplot object will be used.
\texttt{node} \quad internal node number to specify a clade. If \texttt{NULL}, using the whole tree
\texttt{xexpand} \quad numeric, expend the xlim of the zoom area. default is \texttt{NULL}.

Value

full tree with zoom in clade

Author(s)

Guangchuang Yu
Examples

```r
## Not run:
x <- rtree(15)
p <- ggtree(x) + geom_tiplab() +
    geom_nodelab(aes(subset=!isTip, label=node), hjust = -.1, color = "red")
zoomClade(p, 21, xexpand = .2)
## End(Not run)
```

Description

update data with tree info (y coordination and panel)

Usage

```r
p %+>% .data
```

Arguments

- `p` : tree view
- `data` : data.frame

Details

add tree information to an input data. This function will setup y coordination and panel info for data used in facet_plot and geom_facet

Value

updated data.frame

Author(s)

Guangchuang Yu

References

Description

add annotation data to a tree

Usage

pg %<+% data

Arguments

pg ggplot2 object
data annotation data that contains a column of “node”, or the first column of taxa labels

Details

This operator attaches annotation data to a ggtree graphic object

Value

ggplot object with annotation data added

Author(s)

Guangchuang Yu

References


See Also

geom_facet

Examples

nwk <- system.file("extdata", "sample.nwk", package="treeio")
tree <- read.tree(nwk)
p <- ggtree(tree)
dd <- data.frame(taxa=LETTERS[1:13],
    place=c(rep("GZ", 5), rep("HK", 3), rep("CZ", 4), NA),
    value=round(abs(rnorm(13, mean=70, sd=10)), digits=1))
row.names(dd) <- NULL
p %<+% dd + geom_text(aes(color=place, label=label), hjust=-0.5)
Description
update tree

Usage
pg %<% x

Arguments
pg  ggtree object
x   tree object

Details
This operator apply the visualization directives in ggtree object (lhs) to visualize another tree object (rhs), that is similar to Format Painter.

Value
updated ggplot object

Author(s)
Guangchuang Yu

Examples
library("ggplot2")
nwk <- system.file("extdata", "sample.nwk", package="treeio")
tree <- read.tree(nwk)
p <- ggtree(tree) + geom_tippoint(color="#b5e521", alpha=1/4, size=10)
p %<% rtree(30)
Index

* datasets
  StatBalance, 82
* internal
  ggtree-package, 4
  reexports, 75
  %>% (reexports), 75
  %+%, 91
  %<+, 92
  %<, 93
  %>>, 75
  _PACKAGE (ggtree-package), 4

  add_colorbar, 5
  aes, 75
  aes (reexports), 75
  ape::ladderize(), 62
  applyLayoutDaylight, 6
  arrow, 75
  arrow (reexports), 75
  as.polytomy, 6

  collapse, 75
  collapse (reexports), 75
  collapse.ggtree, 7

  Date2decimal, 8
  decimal2Date, 8

  expand, 9

  facet_data, 9
  facet_labeller, 10
  facet_plot, 11
  facet_widths, 12
  filter, 83
  flip, 12
  fortify, 75
  fortify (reexports), 75

  geom_aline, 13
  geom_balance, 14

  geom_cladelab, 15
  geom_cladellabel, 18, 21
  geom_cladellabel2, 19, 20
  geom_facet (facet_plot), 11
  geom_highlight (geom_hilight), 21
  geom_hilight, 21
  geom_inset, 24
  geom_label, 26, 75
  geom_label (reexports), 75
  geom_label2, 25
  geom_motif, 27
  geom_nodelab, 28
  geom_nodelab2, 29
  geom_nodepoint, 30, 32, 36
  geom_point, 32, 36, 75
  geom_point (reexports), 75
  geom_point2, 31
  geom_range, 33
  geom_rootedge, 34
  geom_rootpoint, 32, 35, 36
  geom_segment, 38
  geom_segment2, 37
  geom_strip, 38
  geom_striplab, 40
  geom_taxalink, 43
  geom_text, 45, 75
  geom_text (reexports), 75
  geom_text2, 44
  geom_tiplab, 45, 48
  geom_tiplab (reexports), 28, 29
  geom_tiplab2, 48
  geom_tippoint, 32, 36, 49
  geom_tree, 50
  geom_tree (reexports), 63
  geom_tree2, 51
  geom_treescale, 52
  geom_zoom_clade, 53
  get.path, 54
  get_clade_position, 58
INDEX

get_heatmap_column_position, 58
get_taxa_name, 59
getNodeAngle.df, 54
getNodesBreadthFirst.df, 55
getSubtree, 55
gSubtree.df, 56
gSubtreeUnrooted, 56
gSubtreeUnrooted.df, 57
gTreeArcAngles, 57
ggdensitree, 60
ggexpand (hexpand), 67
ggplot, 75
ggplot (reexports), 75
ggsave, 75
ggsave (reexports), 75
ggtree, 62
ggtree-package, 4
gheatmap, 64
groupClade, 75
groupClade (reexports), 75
groupOTU, 75
groupOTU (reexports), 75
guide_legend, 75
guide_legend (reexports), 75
gzoom, 65
gzoom, ggtree-method (gzoom), 65
gzoom, phylo-method (gzoom), 65
gzoom, tree-data-method (gzoom), 65
gzoom.phylo, 66

hexpand, 67
identify, 75
identify (reexports), 75
inset (geom_inset), 24

label_pad, 68
layout_circular (layout_rectangular), 70
layout_dendrogram (layout_rectangular), 70
layout_fan (layout_rectangular), 70
layout_inward_circular (layout_rectangular), 70
layout_rectangular, 70
layoutDaylight, 69
layoutEqualAngle, 69

margin, 75
margin (reexports), 75

MRCA, 75
MRCA (reexports), 75
msaplot, 71
multiplot, 72
mutate, 84

nodebar, 73
nodeid, 75
nodeid (reexports), 75
nodelab, 75
nodelab (reexports), 75
nodelpie, 73

open_tree, 74

package-ggtree (ggtree-package), 4
plot_list, 75
plot_list (reexports), 75

range_format, 75
read.tree, 75
read.tree (reexports), 75
reexports, 75
revts, 76
rotate, 76
rotate.tree, 78
rotateTreePoints.df, 77
rtree, 75
rtree (reexports), 75

scale_color, 79
scale_color, phylo-method (scale_color), 79
scale_color, tree-data-method (scale_color), 79
scale_color_manual, 75
scale_color_manual (reexports), 75
scale_colour_manual, 80
scale_colour_manual (reexports), 75
scale_colour_subtree, 80
scale_colour_subtree (reexports), 80
scale_fill_manual, 75
scale_fill_manual (reexports), 75
scale_x_continuous, 75
scale_x_continuous (reexports), 75
scale_x_ggtree, 80
scale_x_range, 81
scaleClade, 78
set_hilight_legend, 82
StatBalance, 82

td_filter, 83
td_mutate, 84
td_unnest, 84
theme, 75
theme (reexports), 75
theme_dendrogram, 85
theme_inset, 86
theme_tree, 86
theme_tree2, 87

unit, 75
unit (reexports), 75
unnest, 85

vexpand (hexpand), 67
viewClade, 88

xlim, 75
xlim (reexports), 75
xlim_expand, 89
xlim_tree, 89

zoomClade, 90