Package 'collapsibleTree'

May 8, 2026

Type Package

Title Interactive Collapsible Tree Diagrams using 'D3.js'

Version 0.1.8

Maintainer Adeel Khan <AdeelK@gwu.edu>

Description Interactive Reingold-Tilford tree diagrams created using 'D3.js', where every node can be expanded and collapsed by clicking on it.
Tooltips and color gradients can be mapped to nodes using a numeric column in the source data frame.
See 'collapsibleTree' website for more information and examples.

License GPL (>= 3)

URL https://github.com/AdeelK93/collapsibleTree,
https://AdeelK93.github.io/collapsibleTree/

BugReports https://github.com/AdeelK93/collapsibleTree/issues

Encoding UTF-8

Depends R (>= 3.0.0)

Imports htmlwidgets, data.tree, stats, methods

Enhances knitr, shiny

RoxygenNote 7.2.3

Suggests colorspace, RColorBrewer, dplyr, testthat, tibble

NeedsCompilation no

Author Adeel Khan [aut, cre],
Dhrumin Shah [ctb],
Mike Bostock [ctb, cph] (D3.js library, http://d3js.org)

Repository CRAN

Date/Publication 2023-11-13 05:53:23 UTC

Contents

`collapsibleTree` *Create Interactive Collapsible Tree Diagrams*

Description

Interactive Reingold-Tilford tree diagram created using D3.js, where every node can be expanded and collapsed by clicking on it.

Usage

```
collapsibleTree(
  df,
  ...,
  inputId = NULL,
  attribute = "leafCount",
  aggFun = sum,
  fill = "lightsteelblue",
  linkLength = NULL,
  fontSize = 10,
  tooltip = FALSE,
  tooltipHtml = NULL,
  nodeSize = NULL,
  collapsed = TRUE,
  zoomable = TRUE,
  width = NULL,
  height = NULL
)

## S3 method for class 'data.frame'
collapsibleTree(
  df,
  hierarchy,
  root = deparse(substitute(df)),
  inputId = NULL,
  attribute = "leafCount",
  aggFun = sum,
  fill = "lightsteelblue",
  fillByLevel = TRUE,
  linkLength = NULL,
```

```
  fontSize = 10,
  tooltip = FALSE,
  nodeSize = NULL,
  collapsed = TRUE,
  zoomable = TRUE,
  width = NULL,
  height = NULL,
  ...
)

## S3 method for class 'Node'
collapsibleTree(
  df,
  hierarchy_attribute = "level",
  root = df$name,
  inputId = NULL,
  attribute = "leafCount",
  aggFun = sum,
  fill = "lightsteelblue",
  linkLength = NULL,
  fontSize = 10,
  tooltip = FALSE,
  tooltipHtml = NULL,
  nodeSize = NULL,
  collapsed = TRUE,
  zoomable = TRUE,
  width = NULL,
  height = NULL,
  ...
)
```

Arguments

df	a `data.frame` from which to construct a nested list (where every row is a leaf) or a preconstructed `data.tree`
...	other arguments to pass onto S3 methods that implement this generic function - `collapsibleTree.data.frame`, `collapsibleTree.Node`
inputId	the input slot that will be used to access the selected node (for Shiny). Will return a named list of the most recently clicked node, along with all of its parents.
attribute	numeric column not listed in hierarchy that will be used for tooltips, if applicable. Defaults to 'leafCount', which is the cumulative count of a node's children
aggFun	aggregation function applied to the attribute column to determine values of parent nodes. Defaults to `sum`, but `mean` also makes sense.
fill	either a single color or a mapping of colors: • For `data.frame` input, a vector of colors the same length as the number of nodes. By default, vector should be ordered by level, such that the root color is described first, then all the children's colors, and then all the grandchildren's colors

- For `data.tree` input, a tree attribute containing the color for each node

`linkLength` length of the horizontal links that connect nodes in pixels. (optional, defaults to automatic sizing)

`fontSize` font size of the label text in pixels

`tooltip` tooltip shows the node's label and attribute value.

`tooltipHtml` column name (possibly containing html) to override default tooltip contents, allowing for more advanced customization. Applicable only for `data.tree` input.

`nodeSize` numeric column that will be used to determine relative node size. Default is to have a constant node size throughout. 'leafCount' can also be used here (cumulative count of a node's children), or 'count' (count of node's immediate children).

`collapsed` the tree's children will start collapsed by default

- For `data.frame` input, can also be a vector of logical values the same length as the number of nodes. Follows the same logic as the fill vector.
- For `data.tree` input, can also be a tree attribute for conditionally collapsing nodes

`zoomable` pan and zoom by dragging and scrolling

`width` width in pixels (optional, defaults to automatic sizing)

`height` height in pixels (optional, defaults to automatic sizing)

`hierarchy` a character vector of column names that define the order and hierarchy of the tree network. Applicable only for `data.frame` input.

`root` label for the root node

`fillByLevel` which order to assign fill values to nodes. `TRUE`: Filling by level; will assign fill values to nodes vertically. `FALSE`: Filling by order; will assign fill values to nodes horizontally.

`hierarchy_attribute` name of the `data.tree` attribute that contains hierarchy information of the tree network. Applicable only for `data.tree` input.

Source

Christopher Gandrud: http://christophergandrud.github.io/networkD3/.

d3noob: https://bl.ocks.org/d3noob/43a860bc0024792f8803bba8ca0d5ecd.

Examples

```
collapsibleTree(warpbreaks, c("wool", "tension", "breaks"))

# Data from US Forest Service DataMart
species <- read.csv(system.file("extdata/species.csv", package = "collapsibleTree"))
collapsibleTree(df = species, c("REGION", "CLASS", "NAME"), fill = "green")

# Visualizing the order in which the node colors are filled
library(RColorBrewer)
collapsibleTree(
```

```
  warpbreaks, c("wool", "tension"),
  fill = brewer.pal(9, "RdBu"),
  fillByLevel = TRUE
)
collapsibleTree(
  warpbreaks, c("wool", "tension"),
  fill = brewer.pal(9, "RdBu"),
  fillByLevel = FALSE
)

# Tooltip can be mapped to an attribute, or default to leafCount
collapsibleTree(
  warpbreaks, c("wool", "tension", "breaks"),
  tooltip = TRUE,
  attribute = "breaks"
)

# Node size can be mapped to any numeric column, or to leafCount
collapsibleTree(
  warpbreaks, c("wool", "tension", "breaks"),
  nodeSize = "breaks"
)

# collapsibleTree.Node example
data(acme, package="data.tree")
acme$Do(function(node) node$cost <- data.tree::Aggregate(node, attribute = "cost", aggFun = sum))
acme$Do(function(node) node$lessThanMillion <- node$cost < 10^6)
collapsibleTree(
  acme,
  nodeSize  = "cost",
  attribute = "cost",
  tooltip = TRUE,
  collapsed = "lessThanMillion"
)

# Emulating collapsibleTree.data.frame using collapsibleTree.Node
species <- read.csv(system.file("extdata/species.csv", package = "collapsibleTree"))
hierarchy <- c("REGION", "CLASS", "NAME")
species$pathString <- paste(
  "species",
  apply(species[,hierarchy], 1, paste, collapse = "//"),
  sep = "//"
)
df <- data.tree::as.Node(species, pathDelimiter = "//")
collapsibleTree(df)
```

`collapsibleTree-shiny` *Shiny bindings for collapsibleTree*

Description

Output and render functions for using collapsibleTree within Shiny applications and interactive Rmd documents.

Usage

```
collapsibleTreeOutput(outputId, width = "100%", height = "400px")

renderCollapsibleTree(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a collapsibleTree
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

Examples

```
if(interactive()) {

  # Shiny Interaction
  shiny::runApp(system.file("examples/02shiny", package = "collapsibleTree"))

  # Interactive Gradient Mapping
  shiny::runApp(system.file("examples/03shiny", package = "collapsibleTree"))

}
```

---

collapsibleTreeNetwork

Create Network Interactive Collapsible Tree Diagrams

Description

Interactive Reingold-Tilford tree diagram created using D3.js, where every node can be expanded and collapsed by clicking on it. This function serves as a convenience wrapper for network style data frames containing the node's parent in the first column, node parent in the second column, and additional attributes in the rest of the columns. The root node is denoted by having an NA for a parent. There must be exactly 1 root.

Usage

```
collapsibleTreeNetwork(
  df,
  inputId = NULL,
  attribute = "leafCount",
  aggFun = sum,
  fill = "lightsteelblue",
  linkLength = NULL,
  fontSize = 10,
  tooltip = TRUE,
  tooltipHtml = NULL,
  nodeSize = NULL,
  collapsed = TRUE,
  zoomable = TRUE,
  width = NULL,
  height = NULL
)
```

Arguments

df	a network data frame (where every row is a node) from which to construct a nested list • First column must be the parent (NA for root node) • Second column must be the child • Additional columns are passed on as attributes for other parameters • There must be exactly 1 root node
inputId	the input slot that will be used to access the selected node (for Shiny). Will return a named list of the most recently clicked node, along with all of its parents. (For collapsibleTreeNetwork the names of the list are tree depth)
attribute	numeric column not listed in hierarchy that will be used as weighting to define the color gradient across nodes. Defaults to 'leafCount', which colors nodes by the cumulative count of its children
aggFun	aggregation function applied to the attribute column to determine values of parent nodes. Defaults to sum, but mean also makes sense.
fill	either a single color or a column name with the color for each node
linkLength	length of the horizontal links that connect nodes in pixels. (optional, defaults to automatic sizing)
fontSize	font size of the label text in pixels
tooltip	tooltip shows the node's label and attribute value.
tooltipHtml	column name (possibly containing html) to override default tooltip contents, allowing for more advanced customization
nodeSize	numeric column that will be used to determine relative node size. Default is to have a constant node size throughout. 'leafCount' can also be used here (cumulative count of a node's children), or 'count' (count of node's immediate children).

collapsed	the tree's children will start collapsed by default. Can also be a logical value found in the data for conditionally collapsing nodes.
zoomable	pan and zoom by dragging and scrolling
width	width in pixels (optional, defaults to automatic sizing)
height	height in pixels (optional, defaults to automatic sizing)

Source

Christopher Gandrud: http://christophergandrud.github.io/networkD3/.

d3noob: https://bl.ocks.org/d3noob/43a860bc0024792f8803bba8ca0d5ecd.

See Also

FromDataFrameNetwork for underlying function that constructs trees from the network data frame

Examples

```
# Create a simple org chart
org <- data.frame(
  Manager = c(
    NA, "Ana", "Ana", "Bill", "Bill", "Bill", "Claudette", "Claudette", "Danny",
    "Fred", "Fred", "Grace", "Larry", "Larry", "Nicholas", "Nicholas"
  ),
  Employee = c(
    "Ana", "Bill", "Larry", "Claudette", "Danny", "Erika", "Fred", "Grace",
    "Henri", "Ida", "Joaquin", "Kate", "Mindy", "Nicholas", "Odette", "Peter"
  ),
  Title = c(
    "President", "VP Operations", "VP Finance", "Director", "Director", "Scientist",
    "Manager", "Manager", "Jr Scientist", "Operator", "Operator", "Associate",
     "Analyst", "Director", "Accountant", "Accountant"
  )
)
collapsibleTreeNetwork(org, attribute = "Title")

# Add in colors and sizes
org$Color <- org$Title
levels(org$Color) <- colorspace::rainbow_hcl(11)
collapsibleTreeNetwork(
  org,
  attribute = "Title",
  fill = "Color",
  nodeSize = "leafCount",
  collapsed = FALSE
)

# Use unsplash api to add in random photos to tooltip
org$tooltip <- paste0(
  org$Employee,
  "<br>Title: ",
  org$Title,
```

```
  "<br><img src='https://source.unsplash.com/collection/385548/150x100'>"
)

collapsibleTreeNetwork(
  org,
  attribute = "Title",
  fill = "Color",
  nodeSize = "leafCount",
  tooltipHtml = "tooltip",
  collapsed = FALSE
)
```

`collapsibleTreeSummary` *Create Summary Interactive Collapsible Tree Diagrams*

Description

Interactive Reingold-Tilford tree diagram created using D3.js, where every node can be expanded and collapsed by clicking on it. This function serves as a convenience wrapper to add color gradients to nodes either by counting that node's children (default) or specifying another numeric column in the input data frame.

Usage

```
collapsibleTreeSummary(
  df,
  hierarchy,
  root = deparse(substitute(df)),
  inputId = NULL,
  attribute = "leafCount",
  fillFun = colorspace::heat_hcl,
  maxPercent = 25,
  percentOfParent = FALSE,
  linkLength = NULL,
  fontSize = 10,
  tooltip = TRUE,
  nodeSize = NULL,
  collapsed = TRUE,
  zoomable = TRUE,
  width = NULL,
  height = NULL,
  ...
)
```

Arguments

df	a data frame (where every row is a leaf) from which to construct a nested list
hierarchy	a character vector of column names that define the order and hierarchy of the tree network
root	label for the root node
inputId	the input slot that will be used to access the selected node (for Shiny). Will return a named list of the most recently clicked node, along with all of its parents.
attribute	numeric column not listed in hierarchy that will be used as weighting to define the color gradient across nodes. Defaults to 'leafCount', which colors nodes by the cumulative count of its children
fillFun	function that takes its first argument and returns a vector of colors of that length. rainbow_hcl is a good example.
maxPercent	highest weighting percent to use in color scale mapping. All numbers above this value will be treated as the same maximum value for the sake of coloring in the nodes (but not the ordering of nodes). Setting this value too high will make it difficult to tell the difference between nodes with many children.
percentOfParent	toggle attribute tooltip to be percent of parent rather than the actual value of attribute
linkLength	length of the horizontal links that connect nodes in pixels. (optional, defaults to automatic sizing)
fontSize	font size of the label text in pixels
tooltip	tooltip shows the node's label and attribute value.
nodeSize	numeric column that will be used to determine relative node size. Default is to have a constant node size throughout. 'leafCount' can also be used here (cumulative count of a node's children), or 'count' (count of node's immediate children).
collapsed	the tree's children will start collapsed by default (There is no conditional collapsing in this function yet, but it could be implemented if there's sufficient demand)
zoomable	pan and zoom by dragging and scrolling
width	width in pixels (optional, defaults to automatic sizing)
height	height in pixels (optional, defaults to automatic sizing)
...	other arguments passed on to fillFun, such declaring a palette for brewer.pal

Source

Christopher Gandrud: http://christophergandrud.github.io/networkD3/.

d3noob: https://bl.ocks.org/d3noob/43a860bc0024792f8803bba8ca0d5ecd.

Examples

```
# Color in by number of children
collapsibleTreeSummary(warpbreaks, c("wool", "tension", "breaks"), maxPercent = 50)

# Color in by the value of breaks and use the terrain_hcl gradient
collapsibleTreeSummary(
  warpbreaks,
  c("wool", "tension", "breaks"),
  attribute = "breaks",
  fillFun = colorspace::terrain_hcl,
  maxPercent = 50
)
```

Index