

Package ‘ggheatmap’

July 25, 2021

Title Plot Heatmap

Version 2.1

Maintainer Baiwei Luo <2734782653@qq.com>

Description The flexibility and excellence of 'ggplot2' is unquestionable, so many drawing tools basically need 'ggplot2' as the operating object. In order to develop a heatmap drawing system based on ggplot2, we developed this tool, mainly to solve the heatmap puzzle problem and the flexible connection between the heatmap and the 'ggplot2' object.

The advantages of this tool are as follows:

1. More flexible label settings;
2. Realize the linkage of heatmap and 'ggplot2' drawing system, which is helpful for operations such as puzzles;
3. Simple and easy to operate;
4. Optimization of clustering tree visualization.

License GPL-3

Encoding UTF-8

LazyData false

RoxygenNote 7.1.1

Depends ggplot2

Imports aplot, factoextra, tibble, ggpubr, tidyr, patchwork

Suggests grDevices, stats, testthat

NeedsCompilation no

Author Baiwei Luo [aut, cre]

Repository CRAN

Date/Publication 2021-07-25 13:20:02 UTC

R topics documented:

ggheatmap	2
ggheatmap_plotlist	6
ggheatmap_theme	6

Index	7
--------------	----------

`ggheatmap`*ggplot2 Version of Heatmap*

Description

The flexibility and excellence of 'ggplot2' is unquestionable, so many drawing tools basically need 'ggplot2' as the operating object. In order to develop a heatmap drawing system based on ggplot2, we developed this tool, mainly to solve the heatmap puzzle problem and the flexible connection between the heatmap and the 'ggplot2' object. The advantages of this tool are as follows: 1. More flexible label settings; 2. Realize the linkage of heatmap and 'ggplot2' drawing system, which is helpful for operations such as puzzles; 3. Simple and easy to operate; 4. Optimization of clustering tree visualization.

Usage

```
ggheatmap(data,
  color=colorRampPalette(c( "#0073c2", "white", "#efc000"))(100),
  legendName="Express",
  scale="none",
  shape=NULL,
  border=NA,
  cluster_rows = F,
  cluster_cols = F,
  dist_method="euclidean",
  hclust_method="complete",
  text_show_rows=waiver(),
  text_show_cols=waiver(),
  text_position_rows="right",
  text_position_cols="bottom",
  annotation_cols=NULL,
  annotation_rows=NULL,
  annotation_color,
  annotation_width=0.03,
  annotation_position_rows="left",
  annotation_position_cols="top",
  show_cluster_cols=T,
  show_cluster_rows=T,
  cluster_num=NULL,
  tree_height_rows=0.1,
  tree_height_cols=0.1,
  tree_color_rows=NULL,
  tree_color_cols=NULL,
  tree_position_rows="left",
  tree_position_cols="top",
  levels_rows=NULL,
  levels_cols=NULL
)
```

Arguments

<code>data</code>	input data(matrix or data.frame)
<code>color</code>	the color of heatmap
<code>legendName</code>	character,the title of heatmap legend
<code>scale</code>	character,the way of scale data("none", "row" or "column")
<code>border</code>	character, the colour of border
<code>shape</code>	character, the shape of cell("square", "circle" and "triangle").Default is NULL
<code>cluster_rows</code>	whether rows should be clustered(TRUE of FALSE)
<code>cluster_cols</code>	whether column should be clustered(TRUE of FALSE)
<code>dist_method</code>	character,the method parameter of dist function. see dist
<code>hclust_method</code>	character,the method parameter of hclust function, see hclust
<code>text_show_rows</code>	a character you want to show for y-axis
<code>text_show_cols</code>	a character you want to show for x-axis
<code>text_position_rows</code>	character,the position of y-axis label("right" or "left")
<code>text_position_cols</code>	character,the position of x-axis label("bottom" or "top")
<code>annotation_cols</code>	a data.frame for column annotation
<code>annotation_rows</code>	a data.frame for row annotation
<code>annotation_color</code>	a list for annotation color
<code>annotation_width</code>	a numeric for annotation width
<code>annotation_position_rows</code>	character,the position of column annotation("right" or "left")
<code>annotation_position_cols</code>	character,the position of row annotation("bottom" or "top")
<code>show_cluster_cols</code>	whether show column cluster tree(TRUE of FALSE)
<code>show_cluster_rows</code>	whether show row cluster tree(TRUE of FALSE)
<code>cluster_num</code>	a numeric for cut cluster tree
<code>tree_height_rows</code>	row cluster tree height
<code>tree_height_cols</code>	column cluster tree height
<code>tree_color_rows</code>	a character for row cluster tree color
<code>tree_color_cols</code>	a character for column cluster tree color

```

tree_position_rows      character,the position of row cluster tree("right" or "left")
tree_position_cols      character,the position of column cluster tree("bottom" or "top")
levels_rows             a character for y-axis label levels
levels_cols             a character for x-axis label levels

```

Value

p

Author(s)

Baiwei Luo

Examples

```

#Create data
library(ggheatmap)
library(tidyr)
library(aplot)
set.seed(123)
df <- matrix(runif(600,0,10),ncol = 12)
colnames(df) <- paste("sample",1:12,sep = "")
rownames(df) <- sapply(1:50, function(x)paste(sample(LETTERS,3,replace = FALSE),collapse = ""))
head(df)
#example 1
text_rows <- sample(rownames(df),3)
p <- ggheatmap(df,scale = "row",cluster_rows = TRUE,cluster_cols = TRUE,
  text_show_rows = text_rows)%>%
  ggheatmap_theme(1,theme =list(
    theme(axis.text.x = element_text(angle = 90,face = "bold"),
    axis.text.y = element_text(colour = "red",face = "bold"))
  ))

#example 2
ggheatmap(df,cluster_rows = TRUE,cluster_cols = TRUE,
  border = "grey",
  shape = "circle",
  cluster_num = c(5,4),
  tree_color_rows = c("#3B4992FF", "#EE0000FF", "#008B45FF", "#631879FF", "#008280FF"),
  tree_color_cols = c("#0073C2FF", "#EFC000FF", "#868686FF", "#CD534CFF")
)

#sample 3

row_metaData <- data.frame(exprtype=sample(c("Up", "Down"),50,replace = TRUE),
  genetype=sample(c("Metabolism", "Immune", "None"),50,replace = TRUE))
rownames(row_metaData) <- rownames(df)
col_metaData <- data.frame(tissue=sample(c("Normal", "Tumor"),12,replace = TRUE),

```

```

                                risklevel=sample(c("High","Low"),12,replace = TRUE))
rownames(col_metaData) <- colnames(df)
exprcol <- c("#EE0000FF","#008B45FF" )
names(exprcol) <- c("Up","Down")
genecol <- c("#EE7E30","#5D9AD3","#D0DFE6FF")
names(genecol) <- c("Metabolism","Immune","None")
tissuecol <- c("#98D352","#FF7F0E")
names(tissuecol) <- c("Normal","Tumor")
riskcol <- c("#EEA236FF","#46B8DAFF")
names(riskcol) <- c("High","Low")
col <- list(exprtype=exprcol,genetype=genecol,tissue=tissuecol,risklevel=riskcol)

p<- ggheatmap(df,cluster_rows = TRUE,cluster_cols = TRUE,scale = "row",
              cluster_num = c(5,3),
              tree_color_rows = c("#3B4992FF","#EE0000FF","#008B45FF","#631879FF","#008280FF"),
              tree_color_cols = c("#1F77B4FF","#FF7F0EFF","#2CA02CFF"),
              annotation_rows = row_metaData,
              annotation_cols = col_metaData,
              annotation_color = col
            )
p
ggheatmap_theme(p,2:5,theme = list(
  theme(legend.text = element_text(face = "bold")),
  theme(legend.text = element_text(face = "bold")),
  theme(legend.text = element_text(face = "bold")),
  theme(legend.text = element_text(face = "bold"))
))

#sample 4
ggheatmap(df,cluster_rows = TRUE,cluster_cols = TRUE,scale = "row",
          cluster_num = c(5,3),
          tree_color_rows = c("#3B4992FF","#EE0000FF","#008B45FF","#631879FF","#008280FF"),
          tree_color_cols = c("#1F77B4FF","#FF7F0EFF","#2CA02CFF"),
          annotation_rows = row_metaData,
          annotation_cols = col_metaData,
          annotation_color = col,
          annotation_position_rows="right"
        )

#sample 5
dat <- data.frame(marker=sample(c(1,NA),50,replace = TRUE),
                 gene=rownames(df),
                 shape=sample(c("T","F"),50,replace = TRUE))
p1 <- ggplot(dat,aes(x=1,y=gene,size=marker,color=shape,shape=shape))+
  geom_point()+theme_classic()+
  scale_color_manual(values = c("#D2691E","#1E87D2"))+
  theme(line = element_blank(),axis.text = element_blank(),axis.title = element_blank())+
  guides(size = FALSE)

p%>%insert_right(p1,width = 0.1)

```

ggheatmap_plotlist *Show ggheatmap Plot List*

Description

Display the basic elements of the ggheatmap

Usage

```
ggheatmap_plotlist(ggheatmap)
```

Arguments

ggheatmap heatmap, the result of ggheatmap

Value

plotlist

ggheatmap_theme *The Theme of ggheatmap*

Description

Set the styles of the drawing elements of each component of the ggheatmap except cluster tree.

Usage

```
ggheatmap_theme(ggheatmap, plotlist, theme)
```

Arguments

ggheatmap the result of ggheatmap
plotlist integer, the plotlist in ggheatmap. Use [ggheatmap_plotlist](#)
theme list, the theme of plotlist. More detail can see [theme](#)

Value

ggheatmap

Index

`dist`, 3

`ggheatmap`, 2

`ggheatmap_plotlist`, 6, 6

`ggheatmap_theme`, 6

`hclust`, 3

`theme`, 6