

Package ‘heatmap3’

May 8, 2026

Type Package

Title An Improved Heatmap Package

Version 1.1.9

Date 2021-01-05

Author Shilin Zhao, Linlin Yin, Yan Guo, Quanhu Sheng, Yu Shyr

Maintainer Shilin Zhao <shilin.zhao@vanderbilt.edu>

VignetteBuilder knitr

Description An improved heatmap package. Completely compatible with the original R function 'heatmap', and provides more powerful and convenient features.

License GPL (>= 2)

Imports fastcluster

Suggests knitr

LazyLoad yes

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-06 10:10:02 UTC

Contents

colByValue	2
heatmap3	3
showAnn	7
showLegend	8

Index	9
--------------	----------

 colByValue

colByValue

Description

The function `colByValue` convert the values in matrix into colors

Usage

```
colByValue(x, col, range = NA, breaks = NA, cex.axis = 2, las = 1, ...)
```

Arguments

<code>x</code>	matrix with values.
<code>col</code>	a list of colors such as that generated by <code>rainbow</code> , <code>heat.colors</code> , <code>topo.colors</code> , <code>terrain.colors</code> or similar functions.
<code>range</code>	the range of <code>x</code> . values out of the range will be changed to the max or min value of the range.
<code>breaks</code>	either a numeric vector of two or more unique cut points or a single number (greater than or equal to 2) giving the number of intervals into which <code>x</code> is to be cut.
<code>cex.axis</code>	The magnification to be used for axis annotation relative to the current setting of <code>cex</code> .
<code>las</code>	numeric in 0,1,2,3; the style of axis labels.
<code>...</code>	additional arguments passed on to <code>image</code> function to generate the color bar.

Value

A matrix with colors.

Examples

```
temp<-rnorm(1000)
col<-colByValue(temp,col=colorRampPalette(c('chartreuse4','white','firebrick'))(1024),range=c(-2,2))
col<-colByValue(temp,col=colorRampPalette(c('chartreuse4',
'white','firebrick'))(5),breaks=c(-5,-1,-0.1,0.1,1,5),cex.axis=0.8)
```

`heatmap3`*heatmap3*

Description

The function `heatmap3` is completely compatible with the original R function `heatmap`, and provides more new features.

Usage

```
heatmap3(  
  x,  
  Rowv = NULL,  
  Colv = if (symm) "Rowv" else NULL,  
  distfun = function(x) as.dist(1 - cor(t(x), use = "pa")),  
  distfunC,  
  distfunR,  
  balanceColor = F,  
  ColSideLabs,  
  RowSideLabs,  
  showColDendro = T,  
  showRowDendro = T,  
  col = colorRampPalette(c("navy", "white", "firebrick3"))(1024),  
  legendfun,  
  method = "complete",  
  ColAxisColors = 0,  
  RowAxisColors = 0,  
  hclustfun = hclust,  
  reorderfun = function(d, w) reorder(d, w),  
  add.expr,  
  symm = FALSE,  
  revC = identical(Colv, "Rowv"),  
  scale = c("row", "column", "none"),  
  na.rm = TRUE,  
  ColSideFun,  
  ColSideAnn,  
  ColSideWidth = 0.4,  
  ColSideCut,  
  colorCell,  
  highlightCell,  
  file = "heatmap3.pdf",  
  topN = NA,  
  filterFun = sd,  
  returnDistMatrix = FALSE,  
  margins = c(5, 5),  
  ColSideColors,  
  RowSideColors,
```

```

cexRow = 0.2 + 1/log10(nrow(x)),
cexCol = 0.2 + 1/log10(ncol(x)),
lasRow = 2,
lasCol = 2,
labRow = NULL,
labCol = NULL,
main = NULL,
xlab = NULL,
ylab = NULL,
keep.dendro = FALSE,
verbose = getOption("verbose"),
useRaster = if (ncol(x) * nrow(x) >= 50000) TRUE else FALSE,
...
)

```

Arguments

x	numeric matrix of the values to be plotted.
Rowv	determines if and how the <i>row</i> dendrogram should be computed and reordered. Either a dendrogram or a vector of values used to reorder the row dendrogram or <i>NA</i> to suppress any row dendrogram (and reordering) or by default, <i>NULL</i> , see ‘Details’ below.
Colv	determines if and how the <i>column</i> dendrogram should be reordered. Has the same options as the Rowv argument above and <i>additionally</i> when x is a square matrix, Colv = "Rowv" means that columns should be treated identically to the rows (and so if there is to be no row dendrogram there will not be a column one either).
distfun	function used to compute the distance (dissimilarity) between both rows and columns. Defaults to dist .
distfunC	function used to compute the distance (dissimilarity) between and columns. Will be the same as distfun if not specified.
distfunR	function used to compute the distance (dissimilarity) between and rows. Will be the same as distfun if not specified.
balanceColor	logical indicating if the colors need to be balanced so that the median color will represent the 0 value. The default value is F.
ColSideLabs	label for ColSideColors
RowSideLabs	label for RowSideColors
showColDendro	logical indicating if the coloum dendrogram should be plotted (when Colv isn’t <i>NA</i>).
showRowDendro	logical indicating if the row dendrogram should be plotted (when Rowv isn’t <i>NA</i>).
col	specifying the colors, used in image function.
legendfun	function used to generate legend in top left of the figure. If not specified, the color bar will be plotted. The users can use any plot functions to generate their own legend. Or a function showLegend is also provided as a example.

method	the agglomeration method to be used by <code>hclust</code> function. This should be (an unambiguous abbreviation of) one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid".
ColAxisColors	integer indicating which column of ColSideColors will be used as colors for labels in column axis. The default value is 0, which means all column labels will be in black color.
RowAxisColors	integer indicating which column of RowSideColors will be used as colors for labels in row axis. The default value is 0, which means all row labels will be in black color.
hclustfun	function used to compute the hierarchical clustering when Rowv or Colv are not dendrograms. Defaults to <code>hclust</code> . Should take as argument a result of <code>distfun</code> and return an object to which <code>as.dendrogram</code> can be applied.
reorderfun	function(d, w) of dendrogram and weights for reordering the row and column dendrograms. The default uses <code>reorder.dendrogram</code> .
add.expr	expression that will be evaluated after the call to <code>image</code> . Can be used to add components to the plot.
symm	logical indicating if x should be treated symmetrically ; can only be true when x is a square matrix.
revC	logical indicating if the column order should be reversed for plotting, such that e.g., for the symmetric case, the symmetry axis is as usual.
scale	character indicating if the values should be centered and scaled in either the row direction or the column direction, or none. The default is "row" if <code>symm</code> false, and "none" otherwise.
na.rm	logical indicating whether NA's should be removed.
ColSideFun	function used to generate annotation and labeling figure in column side. The users can use any plot functions to generate their own figure. And a function <code>showAnn</code> is also provided as a example.
ColSideAnn	data frame with continuous and factor variables as annotation information. This parameter will be sorted by column dendrogram and then passed to ColSideFun.
ColSideWidth	numeric the height of column side area, which can be used by ColSideFun function.
ColSideCut	numeric the value to be used in cutting column dendrogram. The dendrogram and annotation will be divided into different parts and labeled respectively.
colorCell	A data.frame with 3 columns, indicating which cells will be colored by specific colors. The first column is row index, second column is column index, and the third column is color.
highlightCell	A data.frame with 3 or 4 columns, indicating which cells will be highlighted by rectangles with specific colors. The first column is row index, second column is column index, the third column is color for rectangle border, and the optional fourth column is width for rectangle border.
file	pdf file name, only works when topN was used.
topN	vector a list of numbers. topN genes will be used to generate the heatmaps.
filterFun	function used to filter genes, such as sd, mean, sum. It will be used in a apply function to calculate for each row.

<code>returnDistMatrix</code>	logical indicating if the distance matrix will be returned
<code>margins</code>	numeric vector of length 2 containing the margins (see <code>par(mar = *)</code>) for column and row names, respectively.
<code>ColSideColors</code>	(optional) character vector of length <code>ncol(x)</code> containing the color names for a horizontal side bar that may be used to annotate the columns of <code>x</code> .
<code>RowSideColors</code>	(optional) character vector of length <code>nrow(x)</code> containing the color names for a vertical side bar that may be used to annotate the rows of <code>x</code> .
<code>cexRow, cexCol</code>	positive numbers, used as <code>cex.axis</code> in for the row or column axis labeling. The defaults currently only use number of rows or columns, respectively.
<code>lasRow, lasCol</code>	the style of row or column axis labels.
<code>labRow, labCol</code>	character vectors with row and column labels to use; these default to <code>rownames(x)</code> or <code>colnames(x)</code> , respectively.
<code>main, xlab, ylab</code>	main, x- and y-axis titles; defaults to none.
<code>keep.dendro</code>	logical indicating if the dendrogram(s) should be kept as part of the result (when <code>Rowv</code> and/or <code>Colv</code> are not NA).
<code>verbose</code>	logical indicating if information should be printed.
<code>useRaster</code>	logical; if TRUE a bitmap raster is used to plot the image instead of polygons. The grid must be regular in that case, otherwise an error is raised.
<code>...</code>	additional arguments passed on to <code>image</code> .

Value

The same return value as `hclust` function.

Examples

```
#generate data
set.seed(123456789)
rnormData<-matrix(rnorm(1000), 40, 25)
rnormData[1:15, seq(6, 25, 2)] = rnormData[1:15, seq(6, 25, 2)] + 2
rnormData[16:40, seq(7, 25, 2)] = rnormData[16:40, seq(7, 25, 2)] + 4
colnames(rnormData)<-c(paste("Control", 1:5, sep = ""),
paste(c("TrtA", "TrtB"), rep(1:10,each=2), sep = ""))
rownames(rnormData)<-paste("Probe", 1:40, sep = "")
ColSideColors<-cbind(Group1=c(rep("steelblue2",5), rep(c("brown1",
"mediumpurple2"),10)),Group2=sample(c("steelblue2","brown1",
"mediumpurple2"),25,replace=TRUE))
colorCell<-data.frame(row=c(1,3,5),col=c(2,4,6),color=c("green4",
"black","orange2"),stringsAsFactors=FALSE)
highlightCell<-data.frame(row=c(2,4,6),col=c(1,3,5),color=c("black",
"green4","orange2"),lwd=1:3,stringsAsFactors=FALSE)
#A simple example
heatmap3(rnormData,ColSideColors=ColSideColors,showRowDendro=FALSE,
colorCell=colorCell,highlightCell=highlightCell)
#A more detail example
ColSideAnn<-data.frame(Information=rnorm(25),Group=c(rep("Control",5),
```

```

    rep(c("TrtA", "TrtB"), 10)), stringsAsFactors=TRUE)
row.names(ColSideAnn) <- colnames(rnormData)
RowSideColors <- colorRampPalette(c("chartreuse4", "white",
  "firebrick"))(40)
result <- heatmap3(rnormData, ColSideCut=1.2, ColSideAnn=ColSideAnn,
  ColSideFun=function(x) showAnn(x), ColSideWidth=0.8,
  RowSideColors=RowSideColors, col=colorRampPalette(c("green", "black",
  , "red"))(1024), RowAxisColors=1, legendfun=function()
  showLegend(legend=c("Low", "High"), col=c("chartreuse4", "firebrick"))
  , verbose=TRUE)
# annotations distribution in different clusters and the result
# of statistic tests
result$cutTable

```

 showAnn

showAnn

Description

The function `showAnn` is an example for generating annotation figure in the result of `heatmap3` function. You can use your any plot functions to generate your own annotation figure.

Usage

```
showAnn(annData)
```

Arguments

`annData` a data frame contains the annotation information for samples. It can only contain factor or numeric variables, and each row represent a sample with the same order of the columns in expression matrix.

Examples

```

annData <- data.frame(mtcars[, c("mpg", "am", "wt", "gear")])
annData[, 2] <- as.factor(annData[, 2])
annData[, 4] <- as.factor(annData[, 4])
# Display annotation
## Not run:
showAnn(annData)

## End(Not run)
# Heatmap with annotation
heatmap3(t(mtcars), ColSideAnn=annData, ColSideFun=function(x)
showAnn(x), ColSideWidth=1.2, balanceColor=TRUE)

```

 showLegend

showLegend

Description

The function `showLegend` is an example for generating legend in the figure of `heatmap3` function. You can use your any plot functions to generate your own legend.

Usage

```
showLegend(
  legend = c("Group A", "Group B"),
  lwd = 3,
  cex = 1.1,
  col = c("red", "blue"),
  ...
)
```

Arguments

<code>legend</code>	a character or expression vector of length ≥ 1 to appear in the legend. Other objects will be coerced by as.graphicsAnnot .
<code>lwd</code>	the line widths for lines appearing in the legend.
<code>cex</code>	character expansion factor relative to current <code>par("cex")</code> . Used for text, and provides the default for <code>pt.cex</code> .
<code>col</code>	the color of points or lines appearing in the legend.
<code>...</code>	additional arguments passed on to legend

Examples

```
RowSideColors<-rep("steelblue2",nrow(mtcars))
RowSideColors[c(4:6,15:17,22:26,29)]<-"lightgoldenrod"
RowSideColors[c(1:3,19:21)]<-"brown1"
heatmap3(mtcars,scale="col",margins=c(2,10),RowSideColors=RowSideColors,
  legendfun=function() showLegend(legend=c("European","American",
  "Japanese"),col=c("steelblue2","lightgoldenrod","brown1"),cex=1.5))
```

Index

as.dendrogram, 5
as.graphicsAnnot, 8

colByValue, 2

dendrogram, 4
dist, 4

expression, 8

hclust, 5, 6
heat.colors, 2
heatmap3, 3

image, 2, 4, 6

legend, 8

NA, 4
NULL, 4

par, 6

rainbow, 2
reorder.dendrogram, 5
rev, 5

showAnn, 5, 7
showLegend, 4, 8

terrain.colors, 2
topo.colors, 2