Package ‘pairheatmap’

February 20, 2015

Title A tool for comparing heatmaps

Version 1.0.1

Date 2012-02-05

Depends R (>= 2.10.0), grid

Author Xiaoyong Sun

Maintainer Xiaoyong Sun `<johnsunx1@gmail.com>`

Description A tool to compare two heatmaps and discover patterns within and across groups. In the context of biology, group can be defined based on gene ontology.

License GPL (>= 2)

LazyData true

Repository CRAN

Date/Publication 2012-02-10 05:54:40

NeedsCompilation no

R topics documented:

pairheatmap-package 1
pairheatmap 2

Index **7**

`pairheatmap-package` *A tool for comparing heatmaps*

Description

A tool to compare two heatmaps and discover patterns within and across groups. In the context of biology, group can be defined based on gene ontology.

Details

Package:	pairheatmap
Type:	Package
Version:	1.0.0
Date:	2011-12-27
License:	What license is it under?

The main interface is pairheatmap() function

Author(s)

Xiaoyong Sun Maintainer: johnsunx1<johnsunx1@gmail.com>

Examples

```
pairheatmap(mtcars, mtcars)
```

`pairheatmap` *Heatmap Comparison*

Description

The main function for comparing heatmaps

Usage

```
pairheatmap (data1, data2,
          scale="none", dendrogram="both",
          matDist=0.5, matrixBorderCol = "grey",
          colorStyle="s1",
          rowGroup=rowGroup,
          orderRowGroup=NULL,
          rowGroupColor=FALSE,
          rowGroupColor.choice,
          groupBorder="line",
          groupBorder.selectList=list(),
          groupBorder.lwd=3,
          groupBorder.col="green",
          rowNameColor="blue",
          colNameColor="blue",
          rowNameFontSize=7,
          colNameFontSize=7,
          rowNameGroupColor=NULL,
          clusterMethod="complete", clusterMembers=NULL,
          clusterRow=TRUE, clusterCol=TRUE, clusterColTogether=FALSE,
          legend.pos="middle", legend.percent=0.5, legend.fontsize=7
          )
```

Arguments

`data1`	numeric matrix 1. It is considered as the standard matrix.
`data2`	numeric matrix 2. Its row order is same as that in data1. Its column order is either same as that in data1 or use separate cluster method.
`scale`	character. It takes four values: "row", "col", "rowsep", "none"(default value). It indicates whether or not the data matrix is scaled in row/column direction. When "rowsep" is selected, the two data matrices are scaled separately in row direction.
`dendrogram`	character. It takes three values: "row", "col", "both"(default value). It indicates whether or not to draw the row/col/both dendrogram.
`matDist`	the separate distance between two data matrices. Default value is 0.5. If its value is 1, the distance between two matrices is exactly one data column.
`matrixBorderCol`	the color of the matrix border. Default value is "grey".
`colorStyle`	the color style for the matrix cell. It takes four values: "s1"(default value), "s2", "s3", "s4". s1 ranges from blue to red; s2 ranges from green to red; s3 uses a default color style from R package, pheatmap; s4 ranges from white to black.
`rowGroup`	Row group variable.
`orderRowGroup`	variable. The default value is "NULL". It is the row levels that should be ordered.
`rowGroupColor`	logical value. It takes two values: "TRUE", "FALSE". It indicates whether or not to draw the row group bar.
`rowGroupColor.choice`	character. It works when rowGroupColor is set as TRUE. The character length must match the unique groups in the rowGroup. If it is not specified, the colorStyle is used as default value.
`groupBorder`	character. It takes two values: "line"(default value), "rect". It controls the shape of the group border.
`groupBorder.selectList`	a list. It controls which group to be selected, which includes four components, "xgroup.start", "xgroup.end", "ygroup.start" and "ygroup.end". The selected groups will utilize the same graphical parameters as groupBorder.
`groupBorder.lwd`	numeric. Default value is 3. It controls the line width of the groupBorder.
`groupBorder.col`	character. Default value is "green". It controls the line color of the groupBorder.
`rowNameColor`	character string. Default value is "blue". It controls the label color of the row name.
`colNameColor`	character string. Default value is "blue". It controls the label color of the column name.
`rowNameFontSize`	numeric scalar. Default value is 7. It controls the font size of the row name.
`colNameFontSize`	numeric scalar. Default value is 7. It controls the font size of the column name.

`rowNameGroupColor`	character variable. The character length must match the unique groups in the rowGroup. It controls the color of different groups of row names.
`clusterMethod`	character. It takes the follow values: "ward", "single", "complete"(default value), "average", "mcquitty", "median" or "centroid".
`clusterMembers`	NULL or a vector. See function: "hclust" of the package "stats" for details.
`clusterRow`	logical. It takes two values: "TRUE"(default value), "FALSE". It indicates whether or not to cluster rows.
`clusterCol`	logical. It takes two values: "TRUE"(default value), "FALSE". It indicates whether or not to cluster columns.
`clusterColTogether`	logical. It takes two values: "TRUE", "FALSE"(default value). It indicates whether or not the columns of data matrix 2 follows the same clustering order of that in data matrix 1. If the column number of data matrix 2 is different from that of data matrix 1, only the columns matching those of data matrix 1 are reordered.
`legend.pos`	character. It takes three values: "top", "middle"(default value), "bottom". It controls the position of the legend.
`legend.percent`	numeric. It takes value from 0 to 1. Default value is 0.5. If its value is 1, the height of the legend will be equal to the height of the heatmap.
`legend.fontsize`	numeric. Default value is 7. It controls the font size of the legend labels.

Details

This function compares two heatmaps and discovers links and patterns within and across groups. In the context of biology, group can be defined based on gene ontology or selected gene lists.

Value

Invisibly return the row and column index for two matrices.

Author(s)

Xiaoyong Sun

Examples

```
## simple demo
chvalue <- pairheatmap(mtcars, mtcars)
chvalue
pairheatmap(mtcars, mtcars, scale="row")
pairheatmap(mtcars, mtcars[,1:5], scale="row")
pairheatmap(mtcars, mtcars[,1:5], scale="rowsep")
pairheatmap(mtcars, mtcars[,1:5], scale="col")
## row group bar
pairheatmap(mtcars, mtcars,
rowGroup=mtcars$gear,
```

```
rowNameFontSize=6,
colNameFontSize=6,
rowNameGroupColor=rev(c("blue",  "green", "orange")),
rowNameColor="blue",
)
## group options
pairheatmap(mtcars, mtcars,
rowGroup=mtcars$gear,
orderRowGroup=c(5, 4,3)
)
## legend
pairheatmap(mtcars, mtcars,
legend.pos="middle", legend.percent=0.6, legend.fontsize=7)
## cluster analysis
pairheatmap(mtcars, mtcars, clusterMethod="ward", clusterRow=FALSE)
pairheatmap(mtcars,  cbind(mtcars, mtcars), clusterColTogether=TRUE)
## selected groups
pairheatmap(mtcars,  cbind(mtcars, mtcars),
groupBorder.selectList=
      list(xgroup.start=c(2,7), xgroup.end=c(4,9),
           ygroup.start=c(3,30), ygroup.end=c(10,32)
           ))
## more demo
pairheatmap(mtcars, mtcars,  legend.percent=1, legend.pos="middle",
       groupBorder.selectList=list(xgroup.start=c(2,7), xgroup.end=c(4,9), ygroup.start=c(3,30),
         ygroup.end=c(10,32)))
pairheatmap(mtcars, mtcars, scale="row", colorStyle="s3",
         groupBorder.selectList=list(xgroup.start=c(8), xgroup.end=c(11),
         ygroup.start=c(11), ygroup.end=c(15)),
         legend.pos="top")
pairheatmap(mtcars, mtcars,
         colorStyle="s4", rowGroup=mtcars$gear,
         rowGroupColor=TRUE,
         rowGroupColor.choice = rev(c("blue", "orange", "green")),
       groupBorder.selectList=list(xgroup.start=c(8), xgroup.end=c(11), ygroup.start=c(22), ygroup.end=c(25)),
       legend.pos="top", legend.percent=0.6, legend.fontsize=5, orderRowGroup=c(5, 4,3),
         clusterRow=TRUE, clusterCol=TRUE, clusterColTogether=TRUE
         )
mtcars2 <- mtcars[, rev(1:ncol(mtcars))]
pairheatmap(mtcars, mtcars2,
         colorStyle="s1", rowGroup=mtcars$gear,
         rowGroupColor=TRUE,
         rowGroupColor.choice = rev(c("blue", "orange", "green")),
       groupBorder.selectList=list(xgroup.start=c(8), xgroup.end=c(11), ygroup.start=c(22), ygroup.end=c(25)),
       legend.pos="top", legend.percent=0.6, legend.fontsize=5, orderRowGroup=c(5, 4,3),
         clusterRow=TRUE, clusterCol=TRUE, clusterColTogether=FALSE
         )
pairheatmap(mtcars, mtcars,
         dendrogram="row", colorStyle="s1", rowGroup=mtcars$am,
         rowGroupColor=TRUE,
       groupBorder.selectList=list(xgroup.start=c(8), xgroup.end=c(11), ygroup.start=c(22), ygroup.end=c(25)),
         legend.pos="top", legend.percent=0.6, legend.fontsize=5,
         orderRowGroup=c(0,1),
```

```
clusterRow=TRUE, clusterCol=TRUE, clusterColTogether=FALSE
)
```

Index

∗Topic **methods**
    `pairheatmap`, 2
∗Topic **package**
    `pairheatmap-package`, 1

`pairheatmap`, 2
`pairheatmap-package`, 1