Package 'wheatmap'

May 8, 2016

Type Package

Title WHeatmap

Version 0.1.0

Author Wanding Zhou

Maintainer Wanding Zhou <zhouwanding@gmail.com>

Description Plot heatmap in a sequential manner.

License MIT license

LazyData TRUE

RoxygenNote 5.0.1

R topics documented:

`Beneath` *Beneath*

Description

Generate dimension beneath another object

Usage

```
Beneath(x = NULL, height = NULL, pad = 0.01, min.ratio = 0.02,
  h.aln = NULL, v.scale = NULL, v.scale.proportional = FALSE)
```

Arguments

`x`	an object with dimension
`height`	the height of the new object (when NULL set proportional to the data)
`pad`	padding between the target and current
`min.ratio`	minimum ratio of dimensions when auto-scale
`h.aln`	object for horizontal alignment (when NULL, set to x)
`v.scale`	object for vertical scaling (when NULL, set to x)
`v.scale.proportional`	when v.scale is provided, whether to make proportional to data

Value

a dimension generator beneath x

both.cluster *row- and column-cluster a matrix*

Description

row- and column-cluster a matrix

Usage

```
both.cluster(mat, hc.method = "ward.D2")
```

Arguments

hc.method	method to use in hclust
at	input matrix

Value

a list of clustered row, column and matrix

CalcTextBounding *Calculate Text Bounding*

Description

Calculate bounding box including texts.

Usage

```
CalcTextBounding(x, ...)
```

Arguments

x	object

Details

W.R.T lower left corner of the view port in the unit of points

---

`CalcTextBounding.WDendrogram` *Calculate Text Ranges*

Description

Calculate Text Ranges

Usage

```
## S3 method for class 'WDendrogram'
CalcTextBounding(dd, group)
```

`CalcTextBounding.WHeatmap` *Calculate Texting Bounding for WHeatmap*

Description

Calculate Texting Bounding for WHeatmap

Usage

```
## S3 method for class 'WHeatmap'
CalcTextBounding(hm, group)
```

Arguments

`hm`	object of class WHeatmap

Value

an object of class WDim in coordinate points

CMPar *Color Map Parameters*

Description

Create color map parameters

Usage

```
CMPar(dmin = NULL, dmax = NULL, brewer.name = "Accent", brewer.n = 3,
  colorspace.name = "rainbow_hcl", colorspace.n = 2, cmap = "jet",
  stop.points = NULL, grey.scale = FALSE)
```

Arguments

dmin	minimum for continuous color map
dmax	maximum for continuous color map
brewer.name	palette name for RColorbrewer
brewer.n	number of stop points in RColorbrewer for continuous color map
colorspace.name	colorspace name
colorspace.n	number of stops in colorspace palettes
cmap	customized colormap name
stop.points	custome stop points
grey.scale	whether to use grey scale
cm	existing color maps

Value

an object of class CMPar

ColorMap *Constructor for ColoMap object*

Description

Create color maps

Usage

```
ColorMap(continuous = TRUE, colors = NULL, dmin = NULL, dmax = NULL,
  scaler = NULL, mapper = NULL)
```

Arguments

colors	colors for each data point
dmin	miminum in continuous color map
dmax	maximum in continuous color map
scaler	scaler function from data range to 0-1
mapper	function that maps data to color
discrete	whether colormap is discrete

Value

an object of class ColorMap

---

column.cluster *column cluster a matrix*

Description

column cluster a matrix

Usage

```
column.cluster(mat, hc.method = "ward.D2")
```

Arguments

mat	input matrix
hc.method	method to use in hclust

Value

a list of clustered row, column and matrix

FromAffine *Convert from affine coordinates*

Description

Convert from affine coordinates

Usage

```
FromAffine(dm.affine, dm.sys)
```

Arguments

dm.sys	the affine system
obj	object on affine coordinate

Value

object on absolute coordinate

grid.dendrogram *Draw dendrogram under grid system*

Description

The dendrogram can be renderred. A viewport is created which contains the dendrogram.

Usage

```
grid.dendrogram(dend, facing = c("bottom", "top", "left", "right"),
  max_height = NULL, order = c("normal", "reverse"), ...)
```

Arguments

dend	a stats::dendrogram object.
facing	facing of the dendrogram.
max_height	maximum height of the dendrogram.

Details

-order should leaves of dendrogram be put in the normal order (1, ..., n) or reverse order (n, ..., 1)? -... pass to ‘grid::viewport‘ which contains the dendrogram.

This function only plots the dendrogram without adding labels. The leaves of the dendrogram locates at unit(c(0.5, 1.5, ...(n-0.5))/n, "npc").

Source

adapted from the ComplexHeatmap package authored by Zuguang Gu <z.gu@dkfz.de>

`LeftOf` *LeftOf*

Description

Generate dimension to the left of another object

Usage

```
LeftOf(x = NULL, width = NULL, pad = 0.01, min.ratio = 0.02,
  v.aln = NULL, h.scale = NULL, h.scale.proportional = FALSE)
```

Arguments

`x`	an object with dimension
`width`	the width of the new object (when NULL, set proportional to data)
`pad`	padding between the target and current
`min.ratio`	minimum ratio of dimensions when auto-scale
`v.aln`	object for vertical alignment (when NULL, set to x)
`h.scale`	object for horizontal scaling (when NULL, set to x)
`h.scale.proportional`	when h.scale is provided, whether to make proportional to data

Value

a dimension to the left of x

`ly` *show layout*

Description

show layout

Usage

```
ly(x)
```

MapToContinuousColors *map data to continuous color*

Description

map data to continuous color

Usage

```
MapToContinuousColors(data, cmp = CMPar(), given.cm = NULL)
```

Arguments

data	numeric vector
cmp	an color map parameter object of class CMPar

Value

an object of ColorMap

MapToDiscreteColors *map data to discrete color*

Description

map data to discrete color

Usage

```
MapToDiscreteColors(data, cmp = CMPar(), given.cm = NULL)
```

Arguments

data	numeric vector
cmp	an color map parameter object of class CMPar

Value

an object of ColorMap

plot.WHeatmap *plot WHeatmap*

Description

plot WHeatmap

Usage

```
## S3 method for class 'WHeatmap'
plot(hm, cex = 1, layout.only = FALSE,
  stand.alone = TRUE)
```

Arguments

hm	heatmap to plot

print.WDendrogram *WPlot*

Description

WPlot

Usage

```
## S3 method for class 'WDendrogram'
print(dend, stand.alone = TRUE, layout.only = FALSE,
  cex = 1)
```

print.WGroup *Draw WGroup*

Description

Draw WGroup

Usage

```
## S3 method for class 'WGroup'
print(group, mar = c(0.03, 0.03, 0.03, 0.03),
  stand.alone = TRUE, cex = 1, layout.only = FALSE)
```

Arguments

group	plot to display
cex	for scale fonts

print.WHeatmap *plot WHeatmap*

Description

plot WHeatmap

Usage

```
## S3 method for class 'WHeatmap'
print(hm, cex = 1, layout.only = FALSE,
  stand.alone = TRUE)
```

Arguments

hm	an object of class WHeatmap

Value

NULL

RightOf *RightOf*

Description

Generate dimension to the right of another object

Usage

```
RightOf(x = NULL, width = NULL, pad = 0.01, min.ratio = 0.02,
  v.aln = NULL, h.scale = NULL, h.scale.proportional = FALSE)
```

Arguments

x	an object with dimension
width	the width of the new object (when NULL, set proportional to data)
pad	padding between the target and current
min.ratio	minimum ratio of dimensions when auto-scale
v.aln	object for vertical alignment (when NULL, set to x)
h.scale	object for horizontal scaling (when NULL, set to x)
h.scale.proportional	when h.scale is provided, whether to make proportional to data

Value

a dimension to the right of x

row.cluster *row cluster a matrix*

Description

row cluster a matrix

Usage

```
row.cluster(mat, hc.method = "ward.D2")
```

Arguments

mat	input matrix
hc.method	method to use in hclust

Value

a list of clustered row, column and matrix

ScaleGroup *Scale group*

Description

Scale group to incorporate text on margins

Usage

```
ScaleGroup(group.obj, mar = c(0.03, 0.03, 0.03, 0.03))
```

Arguments

group.obj	group object that needs to be scaled

Value

scaled group obj

text.width *font width and scale to specified font size*

Description

font width and scale to specified font size

Usage

```
## S3 method for class 'width'
text(txt, fontsize = NULL)
```

ToAffine *To affine coordinates*

Description

To affine coordinates

Usage

```
ToAffine(dm, dm.sys)
```

Arguments

dm	absolute coordinate
dm.sys	the affine system

Value

dm.affine on the affine coordiante

TopOf *Top of*

Description

Generate dimension top of another object

Usage

```
TopOf(x = NULL, height = NULL, pad = 0.01, min.ratio = 0.02,
  h.aln = NULL, v.scale = NULL, v.scale.proportional = FALSE)
```

Arguments

x	an object with dimension
height	the height of the new object (when NULL, set to proportional to data)
pad	padding between the target and current
min.ratio	minimum ratio of dimensions when auto-scale
h.aln	object for horizontal alignment (when NULL, set to x)
v.scale	object for vertical scaling (when NULL, set to x)
v.scale.proportional	when v.scale is provided, whether to make proportional to data

Value

a dimension generator on top of x

WColorBarH *WColorBarH*

Description

a horizontal color bar

Usage

```
WColorBarH(data, ...)
```

Arguments

data	numeric vector

Value

an object of class WColorBarH

WColorBarV *WColorBarV*

Description

a vertical color bar

Usage

```
WColorBarV(data, ...)
```

Arguments

data	numeric vector

Value

an object of class WColorBarV

WColumnBind *column bind non-overlapping objects*

Description

column bind non-overlapping objects

Usage

```
WColumnBind(..., nr = NULL, nc = NULL)
```

Arguments

...	plotting objects
nr	number of rows
nc	number of columns

Value

an object of class WDim

WDendrogram *WDendrogram class*

Description

WDendrogram class

Usage

```
WDendrogram(clust = NULL, dm = WDim(0, 0, 1, 1), name = "",
  facing = c("bottom", "top", "left", "right"))
```

Arguments

clust	hclust object
dm	plotting dimension
name	name of the dendrogram plot
facing	direction of the dendrogram plot

Value

an object of class WDendrogram

WDim *class WDim*

Description

class WDim

Usage

```
WDim(left = 0, bottom = 0, width = 1, height = 1, nr = 1, nc = 1,
  column.split = NULL, row.split = NULL)
```

Arguments

left	left coordinate
bottom	bottom coordinate
width	width
height	height
column.split	a list of WDim objects for column split
row.split	a list of WDim objects for row split

WGroup *Create a WGroup*

Description

Children must be registered already

Usage

```
WGroup(..., name = "", group.dm = WDim(), affine = FALSE, nr = NULL,
  nc = NULL)
```

Arguments

affine	member is on affine coordinate
nr	number of rows
nc	number of columns
dm	dimension

Value

an object of class WGroup

WHeatmap *WHeatmap object*

Description

Create a heatmap

Usage

```
WHeatmap(data = NULL, dm = NULL, name = "", continuous = NULL,
  cmp = CMPar(), cm = NULL, parent = NULL, title = NULL,
  title.fontsize = 12, title.pad = 0.005, title.side = "l",
  xticklabels = NULL, xticklabels.n = NULL, xticklabel.side = "b",
  xticklabel.fontsize = 12, xticklabel.rotat = 90, xticklabel.pad = 0.005,
  yticklabels = NULL, yticklabels.n = NULL, yticklabel.side = "l",
  yticklabel.fontsize = 12, yticklabel.pad = 0.005, alpha = 1,
  sub.name = NULL, gp = NULL)
```

Arguments

data	data matrix
dm	plotting dimension c(left, bottom, width, height)
name	name of the plot
continuous	whether the data is on continuous scale
cmp	an object of CMPar class

Value

one or a list of heatmaps (depends on whether dimension is split)

WLegendH *WLegendH*

Description

a horizontal legend

Usage

```
WLegendH(x = NULL, dm = NULL, name = "", n.stops = 20, n.text = 5,
  label.fontsize = 12, width = 0.1, height = 0.1, ...)
```

Arguments

x	a name or a plotting object, if NULL use the last plotting object
dm	position
name	name of the plotted legend
n.stops	number of stops in computing continuous legend
n.text	number of text labels in continuous legend
label.fontsize	label font size
width	width of each unit in plotted legend
height	height of each unit in plotted legend

Value

an object of class WLegendH

Examples

```
WHeatmap(matrix(1:4,nrow=2))+WLegendH(NULL, Beneath())
```

WLegendV *WLegendV*

Description

a vertical legend

Usage

```
WLegendV(x = NULL, dm = NULL, name = "", n.stops = 20, n.text = 5,
  label.fontsize = 12, width = 0.1, height = 0.1, ...)
```

Arguments

x	a name or a plotting object, if NULL use the last plotting object
dm	position
name	name of the plotted legend
n.stops	number of stops in computing continuous legend
n.text	number of text labels in continuous legend
label.fontsize	label font size
width	width of each unit in plotted legend
height	height of each unit in plotted legend

Value

an object of class WLegendV

Examples

```
WHeatmap(matrix(1:4,nrow=2))+WLegendV(NULL, RightOf())
```

WRowBind *row bind non-overlapping objects*

Description

row bind non-overlapping objects

Usage

```
WRowBind(..., nr = NULL, nc = NULL)
```

Arguments

...	plotting objects
nr	number of rows
nc	number of columns

Value

an object of class WDim

[.WGroup *subset WGroup*

Description

subset WGroup

Usage

```
## S3 method for class 'WGroup'
x[i]
```

Arguments

i	integer indexing element

Index