visDmatCluster

July 21, 2015

`visDmatCluster` *Function to visualise clusters/bases partitioned from a supra-hexagonal grid*

Description

`visDmatCluster` is supposed to visualise clusters/bases partitioned from a supra-hexagonal grid

Usage

```
visDmatCluster(sMap, sBase, height = 7, margin = rep(0.1, 4),
area.size = 1, gp = grid::gpar(cex = 0.8, font = 2, col = "black"),
border.color = "transparent", fill.color = NULL, lty = 1, lwd = 1,
lineend = "round", linejoin = "round", colormap = c("rainbow", "jet",
"bwr", "gbr", "wyr", "br", "yr", "wb"), clip = c("on", "inherit",
"off"),
newpage = T)
```

Arguments

`sMap` an object of class "sMap"

`sBase` an object of class "sBase"

`height` a numeric value specifying the height of device

`margin` margins as units of length 4 or 1

`area.size` an inteter or a vector specifying the area size of each hexagon

`gp` an object of class "gpar". It is the output from a call to the function "gpar" (i.e., a list of graphical parameter settings)

`border.color` the border color for each hexagon

`fill.color` the filled color for each hexagon

`lty` the line type for each hexagon. 0 for 'blank', 1 for 'solid', 2 for 'dashed', 3 for 'dotted', 4 for 'dotdash', 5 for 'longdash', 6 for 'twodash'

`lwd` the line width for each hexagon

`lineend` the line end style for each hexagon. It can be one of 'round', 'butt' and 'square'

`linejoin` the line join style for each hexagon. It can be one of 'round', 'mitre' and 'bevel' `colormap` short name for the colormap. It can be one of "jet" (jet colormap), "bwr" (blue-white-red colormap), "gbr" (green-black-red colormap), "wyr" (white-yellow-red colormap), "br" (black-red colormap), "yr" (yellow-red colormap), "wb" (white-black colormap), and "rainbow" (rainbow colormap, that is, red-yellow-green-cyan-blue-magenta). Alternatively, any hyphen-separated HTML color names, e.g. "blue-black-yellow", "royalblue-white-sandybrown", "darkgreen-white-darkviolet". A list of standard color names can be found in `http://html-color-codes.info/color-names`

`clip` either "on" for clipping to the extent of this viewport, "inherit" for inheriting the clipping region from the parent viewport, or "off" to turn clipping off altogether

`newpage` logical to indicate whether to open a new page. By default, it sets to true for opening a new page

Value

invisible

Note

none

See Also

`sDmatCluster`, `sDmat`, `visColormap`, `visHexGrid`

Examples

```
# 1) generate an iid normal random matrix of 100x10
data <- matrix( rnorm(100*10,mean=0,sd=1), nrow=100, ncol=10)

# 2) get trained using by default setup
sMap <- sPipeline(data=data)

# 3) partition the grid map into clusters using region-growing algorithm
sBase <- sDmatCluster(sMap=sMap, which_neigh=1,
distMeasure="median", clusterLinkage="average")

# 4) visualise clusters/bases partitioned from the sMap
visDmatCluster(sMap,sBase)
# 4a) also, the area size is proportional to the hits
visDmatCluster(sMap,sBase, area.size=log2(sMap$hits+1))
# 4b) also, the area size is inversely proportional to the map distance
dMat <- sDmat(sMap)
visDmatCluster(sMap,sBase, area.size=-1*log2(dMat))

# 5) customise the fill color and line type
my_color <-
visColormap(colormap="PapayaWhip-pink-Tomato")(length(sBase$seeds))[sBase$bases]
my_lty <- (sBase$bases %% 2)
visDmatCluster(sMap,sBase, fill.color=my_color, lty=my_lty,
border.color="black", lwd=2, area.size=0.9)
# also, the area size is inversely proportional to the map distance
visDmatCluster(sMap,sBase, fill.color=my_color, lty=my_lty,
border.color="black", lwd=2, area.size=-1*log2(dMat))
```