visHexMapping

March 11, 2015

visHexMapping	*Function to visualise various mapping items within a supra-hexagonal grid*

Description

visHexMapping is supposed to visualise various mapping items within a supra-hexagonal grid

Usage

```
visHexMapping(sObj, mappingType = c("indexes", "hits", "dist",
"antidist",
"bases", "customized"), labels = NULL, height = 7, margin = rep(0.1,
4),
area.size = 1, gp = grid::gpar(cex = 0.7, font = 1, col = "black"),
border.color = "black", fill.color = "transparent", lty = 1, lwd = 1,
lineend = "round", linejoin = "round", clip = c("on", "inherit",
"off"),
newpage = T)
```

Arguments

sObj	an object of class "sMap" or "sInit" or "sTopol"
mappingType	the mapping type, can be "indexes", "hits", "dist", "antidist", "bases", and "customized"
labels	NULL or a vector with the length of nHex
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'
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

The mappingType includes:

- "indexes": the index of hexagons in a supra-hexagonal grid
- "hits": the number of input data vectors hitting the hexagons
- "dist": distance (in high-dimensional input space) to neighbors (defined in 2D output space)
- "antidist": the oppose version of "dist"
- "bases": clusters partitioned from the sMap
- "customized": displaying input "labels"

See Also

sDmat, sDmatCluster, visHexGrid

Examples

```
# 1) generate data with an iid matrix of 1000 x 9
data <- cbind(matrix(rnorm(1000*3,mean=0,sd=1), nrow=1000, ncol=3),
matrix(rnorm(1000*3,mean=0.5,sd=1), nrow=1000, ncol=3),
matrix(rnorm(1000*3,mean=-0.5,sd=1), nrow=1000, ncol=3))
colnames(data) <- c("S1","S1","S1","S2","S2","S2","S3","S3","S3")

# 2) sMap resulted from using by default setup
sMap <- sPipeline(data=data)

# 3) visualise supported mapping items within a supra-hexagonal grid
# 3a) for indexes of hexagons
visHexMapping(sMap,mappingType="indexes")
# 3b) for the number of input data vectors hitting the hexagons
visHexMapping(sMap,mappingType="hits")
# 3c) for distance (in high-dimensional input space) to neighbors (defined in 2D output space)
visHexMapping(sMap,mappingType="dist")
# 3d) for clusters/bases partitioned from the sMap
visHexMapping(sMap,mappingType="bases")
```