

visHexMulComp

January 18, 2018

visHexMulComp	<i>Function to visualise multiple component planes of a supra-hexagonal grid</i>
---------------	--

Description

visHexMulComp is supposed to visualise multiple component planes of a supra-hexagonal grid

Usage

```
visHexMulComp(sMap, which.components = NULL, rect.grid = NULL,  
margin = rep(0.1, 4), height = 7, title.rotate = 0, title.xy = c(0.45,  
1), colormap = c("bwr", "jet", "gbr", "wyr", "br", "yr", "rainbow",  
"wb"),  
ncolors = 40, zlim = NULL, border.color = "transparent",  
gp = grid::gpar(), newpage = TRUE)
```

Arguments

sMap	an object of class "sMap"
which.components	an integer vector specifying which compopnets will be visualised. By default, it is NULL meaning all components will be visualised
rect.grid	a vector specifying the number of rows and columns for a rectangle grid wherein the component planes are placed. By default, it is NULL (decided on according to the number of component planes that will be visualised)
margin	margins as units of length 4 or 1
height	a numeric value specifying the height of device
title.rotate	the rotation of the title
title.xy	the coordinates of the title
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

<code>ncolors</code>	the number of colors specified
<code>zlim</code>	the minimum and maximum z values for which colors should be plotted, defaulting to the range of the finite values of z. Each of the given colors will be used to color an equispaced interval of this range. The midpoints of the intervals cover the range, so that values just outside the range will be plotted
<code>border.color</code>	the border color for each hexagon
<code>gp</code>	an object of class <code>gpar</code> , typically the output from a call to the function <code>gpar</code> (i.e., a list of graphical parameter settings)
<code>newpage</code>	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

[visVp](#), [visHexComp](#), [visColorbar](#)

Examples

```
# 1) generate data with an iid matrix of 1000 x 3
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 multiple component planes of a supra-hexagonal grid
visHexMulComp(sMap, colormap="jet", ncolors=20, zlim=c(-1,1),
gp=grid::gpar(cex=0.8))
# 3a) visualise only the first 6 component planes
visHexMulComp(sMap, which.components=1:6, colormap="jet", ncolors=20,
zlim=c(-1,1), gp=grid::gpar(cex=0.8))
# 3b) visualise only the first 6 component planes within the rectangle grid of 3 X 2
visHexMulComp(sMap, which.components=1:6, rect.grid=c(3,2),
colormap="jet", ncolors=20, zlim=c(-1,1), gp=grid::gpar(cex=0.8))
```