

visHexAnimate

January 18, 2018

visHexAnimate	<i>Function to animate multiple component planes of a supra-hexagonal grid</i>
---------------	--

Description

visHexAnimate is supposed to animate multiple component planes of a supra-hexagonal grid. The output can be a pdf file containing a list of frames/images, a mp4 video file or a gif file. To support video output file, the software 'ffmpeg' must be first installed (also put its path into the system PATH variable; see Note). To support gif output file, the software 'ImageMagick' must be first installed (also put its path into the system PATH variable; see Note).

Usage

```
visHexAnimate(sMap, which.components = NULL, filename =  
"visHexAnimate",  
filetype = c("pdf", "mp4", "gif"), image.type = c("jpg", "png"),  
sec_per_frame = 1, 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())
```

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
filename	the without-extension part of the name of the output file. By default, it is 'visHexAnimate'
filetype	the type of the output file, i.e. the extension of the output file name. It can be one of either 'pdf' for the pdf file, 'mp4' for the mp4 video file, 'gif' for the gif file
image.type	the type of the image files temporarily generated. It can be one of either 'jpg' or 'png'. These temporary image files are used for producing mp4/gif output file. The reason doing so is to accommodate that sometimes only one of image types is supported so that you can choose the right one

<code>sec_per_frame</code>	a numeric value specifying how long (seconds) it takes to stream a frame/image. This argument only works when producing mp4 video or gif file.
<code>margin</code>	margins as units of length 4 or 1
<code>height</code>	a numeric value specifying the height of device
<code>title.rotate</code>	the rotation of the title
<code>title.xy</code>	the coordinates of the title
<code>colormap</code>	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 gpar, typically the output from a call to the function gpar (i.e., a list of graphical parameter settings)

Value

If specifying the output file name (see argument 'filename' above), the output file is either 'filename.pdf' or 'filename.mp4' or 'filename.gif' in the current working directory. If no output file name specified, by default the output file is either 'visHexAnimate.pdf' or 'visHexAnimate.mp4' or 'visHexAnimate.gif'

Note

When producing mp4 video, this function requires the installation of the software 'ffmpeg' at <https://www.ffmpeg.org>. Shell command lines for ffmpeg installation in Terminal (for both Linux and Mac) are:

- 1) `wget -O ffmpeg.tar.gz http://www.ffmpeg.org/releases/ffmpeg-2.7.1.tar.gz`
- 2) `mkdir ~/ffmpeg | tar xvfz ffmpeg.tar.gz -C ~/ffmpeg --strip-components=1`
- 3) `cd ffmpeg`
- 4a) # Assuming you want installation with a ROOT (sudo) privilege:
`./configure --disable-yasm`
- 4b) # Assuming you want local installation without ROOT (sudo) privilege:
`./configure --disable-yasm --prefix=$HOME/ffmpeg`
- 5) `make`
- 6) `make install`
- 7) # add the system PATH variable to your ~/.bash_profile file if you follow 4b) route:
`export PATH=$HOME/ffmpeg:$PATH`

- 8) # make sure ffmpeg has been installed successfully:
ffmpeg -h

When producing gif file, this function requires the installation of the software 'ImageMagick' at <http://www.imagemagick.org>. Shell command lines for ImageMagick installation in Terminal are:

- 1) wget <http://www.imagemagick.org/download/ImageMagick.tar.gz>
- 2) mkdir ~/ImageMagick | tar xvfz ImageMagick.tar.gz -C ~/ImageMagick --strip-components=1
- 3) cd ImageMagick
- 4) ./configure --prefix=\$HOME/ImageMagick
- 5) make
- 6) make install
- 7) # add the system PATH variable to your ~/.bash_profile file.
For Linux:
export MAGICK_HOME=\$HOME/ImageMagick
export PATH=\$MAGICK_HOME/bin:\$PATH
export LD_LIBRARY_PATH=\${LD_LIBRARY_PATH:+\$LD_LIBRARY_PATH:}\$MAGICK_HOME/lib
For Mac:
export MAGICK_HOME=\$HOME/ImageMagick
export PATH=\$MAGICK_HOME/bin:\$PATH
export DYLD_LIBRARY_PATH=\$MAGICK_HOME/lib/
- 8a) # check configuration:
convert -list configure
- 8b) # check image format supported:
identify -list format
- Tips:
Prior to 4), please make sure libjpeg and libpng are installed. If NOT, for Mac try this:
brew install libjpeg libpng
To check whether ImageMagick does work, please get additional information from:
identify -list format
convert -list configure
On details, please refer to <http://www.imagemagick.org/script/advanced-unix-installation.php>

See Also

[visHexMulComp](#)

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) animate sMap
# output as a <a href="visHexAnimate.pdf">pdf</a> file
```

```
visHexAnimate(sMap, filename="visHexAnimate", filetype="pdf")
# output as a <a href="visHexAnimate.mp4">mp4</a> file
visHexAnimate(sMap, filename="visHexAnimate", filetype="mp4")
# output as a <a href="visHexAnimate.gif">gif</a> file
visHexAnimate(sMap, filename="visHexAnimate", filetype="gif")
```