visNetAnimate

July 21, 2015

visNetAnimate	*Function to animate the same graph but with multiple graph node colorings according to input data matrix*

Description

visNetAnimate is supposed to animate the same graph but with multiple colorings according to input data matrix. 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

```
visNetAnimate(g, data, filename = "visNetAnimate", filetype = c("pdf",
"mp4", "gif"), image.type = c("jpg", "png"), num.frame = ncol(data),
sec_per_frame = 1, height.device = 7, margin = rep(0.1, 4),
border.color = "#EEEEEE", colormap = c("bwr", "jet", "gbr", "wyr",
"br",
"yr", "rainbow", "wb"), ncolors = 40, zlim = NULL, colorbar = T,
colorbar.fraction = 0.25, glayout = layout.fruchterman.reingold,
glayout.dynamics = F, mtext.side = 3, mtext.adj = 0, mtext.cex = 1,
mtext.font = 2, mtext.col = "black", ...)
```

Arguments

g	an object of class "igraph" or "graphNEL"
data	an input data matrix used to color-code vertices/nodes. One column corresponds to one graph node coloring. The input matrix must have row names, and these names should include all node names of input graph, i.e. V(g)$name, since there is a mapping operation. After mapping, the length of the patern vector should be the same as the number of nodes of input graph. The way of how to color-code is to map values in the pattern onto the whole colormap (see the next arguments: colormap, ncolors, zlim and colorbar)
filename	the without-extension part of the name of the output file. By default, it is 'visNetAnimate'

`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

`num.frame` a numeric value specifying the number of frames/images. By default, it sets to the number of columns in the input data matrix

`sec_per_frame` 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.

`height.device` a numeric value specifying the height (or width) of device/frame/image.

`margin` margins as units of length 4 or 1

`border.color` the border color of each figure

`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

`ncolors` the number of colors specified over the colormap

`zlim` the minimum and maximum z/patttern 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

`colorbar` logical to indicate whether to append a colorbar. If pattern is null, it always sets to false

`colorbar.fraction` the relative fraction of colorbar block against the figure block

`glayout` either a function or a numeric matrix configuring how the vertices will be placed on the plot. If layout is a function, this function will be called with the graph as the single parameter to determine the actual coordinates. This function can be one of "layout.auto", "layout.random", "layout.circle", "layout.sphere", "layout.fruchterman.reingold", "layout.kamada.kawai", "layout.spring", "layout.reingold.tilford", "layout.fruchterman.reingold.grid", "layout.lgl", "layout.graphopt", "layout.svd" and "layout.norm". A full explanation of these layouts can be found in http://igraph.org/r/doc/layout_nicely.html

`glayout.dynamics` logical to indicate whether graph layout should be dynamic. By default, it always sets to false. If YES, the Fruchterman-Reingold layout algorithm http://igraph.org/r/doc/layout_with_fr.html will be used to stimulate the dynamic layout

`mtext.side` on which side of the mtext plot (1=bottom, 2=left, 3=top, 4=right)

`mtext.adj` the adjustment for mtext alignment (0 for left or bottom alignment, 1 for right or top alignment)

`mtext.cex` the font size of mtext labels

mtext.font	the font weight of mtext labels
mtext.col	the color of mtext labels
...	additional graphic parameters. See http://igraph.org/r/doc/plot.common.html for the complete list.

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 'visNetAnimate.pdf' or 'visNetAnimate.mp4' or 'visNetAnimate.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 xvzf 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

visNetMul

Examples

```
# 1) generate a random graph according to the ER model
g <- erdos.renyi.game(100, 1/100)

# 2) produce the induced subgraph only based on the nodes in query
subg <- dNetInduce(g, V(g), knn=0)

# 3) visualise the module with vertices being color-coded by scores
nnodes <- vcount(subg)
nsamples <- 10
data <- matrix(runif(nnodes*nsamples), nrow=nnodes, ncol=nsamples)
rownames(data) <- V(subg)$name
# output as a <a href="visNetAnimate.pdf">pdf</a> file
visNetAnimate(g=subg, data=data, filetype="pdf")
# output as a <a href="visNetAnimate.mp4">mp4</a> file but with dynamic layout
visNetAnimate(g=subg, data=data, filetype="mp4", glayout.dynamics=TRUE)
# output as a <a href="visNetAnimate.gif">gif</a> file but with dynamic layout
visNetAnimate(g=subg, data=data, filetype="gif", glayout.dynamics=TRUE)
```