Package ‘sigmaNet’

April 23, 2018

**Title**  Render Graphs Using 'Sigma.js'

**Version**  1.1.0

**Maintainer**  Ian Kloo <iankloo@gmail.com>

**URL**  https://github.com/iankloo/sigmaNet

**BugReports**  https://github.com/iankloo/sigmaNet/issues

**Description**  Create interactive graph visualizations using 'Sigma.js' (<http://sigmajs.org/>). This package is meant to be used in conjunction with 'igraph', replacing the (somewhat underwhelming) plotting features of the package. The idea is to quickly render graphs, regardless of their size, in a way that allows for easy, iterative modification of aesthetics. Because 'Sigma.js' is a 'javascript' library, the visualizations are inherently interactive and are well suited for integration with 'Shiny' apps. While there are several 'htmlwidgets' focused on network visualization, they tend to underperform on medium to large sized graphs. 'Sigma.js' was designed for larger network visualizations and this package aims to make those strengths available to 'R' users.

**Depends**  R (>= 2.10)

**License**  MIT + file LICENSE

**Encoding**  UTF-8

**LazyData**  true

**Imports**  htmlwidgets, igraph, jsonlite, RColorBrewer

**RoxygenNote**  6.0.1

**Suggests**  magrittr, knitr, rmarkdown, shiny

**VignetteBuilder**  knitr

**NeedsCompilation**  no

**Author**  Ian Kloo [aut, cre]

**Repository**  CRAN

**Date/Publication**  2018-04-23 19:34:26 UTC
addEdgeColors

Modify the edge colors of a 'sigmaNet' object.

Description

Modify the edge colors of a 'sigmaNet' object by providing either: (1) a single color to use for every edge; or (2) an attribute of the initial 'igraph' object that will be used to determine color.

Usage

```r
addEdgeColors(sigmaobj, oneColor = NULL, colorAttr = NULL, colorPal = "Set2")
```

Arguments

- `sigmaobj`: A 'sigmaNet' object - created using the 'sigmaFromIgraph' function
- `oneColor`: A single color to color all of the nodes (hex format)
- `colorAttr`: An attribute from the original 'igraph' nodes to color the nodes by
- `colorPal`: The color palatte to use - only used if colorAttr is specified

Details

If the 2nd option is used, you can also specify a color palette from 'RColorBrewer.'

Value

A 'sigmaNet' object with modified node labels. This object can be called directly to create a visualization, or modified by additional functions.
### Examples

```r
library(igraph)
library(sigmaNet)
library(magrittr)

data(lesMIS)

l <- layout_nicely(lesMis)

# one color for all edges
sig <- sigmaFromIgraph(graph = lesMis, layout = l) %>%
  addEdgeColors(oneColor = '#D95F02')

sig
```

### Description

Modify the edge size of a ‘sigmaNet’ object by providing one of the following: (1) a single size to use for all edges; or (2) an attribute in the initial igraph to be used to size the edges.

### Usage

```r
addEdgeSize(sigmaObj, sizeAttr = NULL, minSize = 1, maxSize = 5,
           oneSize = NULL)
```

### Arguments

- `sigmaObj` A ‘sigmaNet’ object - created using the ‘sigmaFromIgraph’ function
- `sizeAttr` The attribute to use to create edge size (width)
- `minSize` The minimum size of the edges (for scaling)
- `maxSize` The maximum size of the edges (for scaling)
- `oneSize` A single size to use for all edges

### Details

If the 2nd method is used, the `minSize` and `maxSize` attribute will control lower and upper bounds of the scaling function.

### Value

A ‘sigmaNet’ object with modified node labels. This object can be called directly to create a visualization, or modified by additional functions.
addInteraction

Examples

```r
library(igraph)
library(sigmaNet)
library(magrittr)

data(lesMis)

l <- layout_nicely(lesMis)

# specify a single edge size
sig <- sigmaFromIgraph(graph = lesMis, layout = l) %>%
  addEdgeSize(oneSize = 5)

# specify an attribute and min/max
sig <- sigmaFromIgraph(graph = lesMis, layout = l) %>%
  addEdgeSize(sizeAttr = 'value', minSize = .1, maxSize = 2)
```

---

**addInteraction**

*Modify the interactivity of a 'sigmaNet' object.*

**Description**

Modify the interactivity of a 'sigmaNet' object using the below options. By default, visualizations include on-click neighbor events, double-click zoom, and mouse-wheel zoom. These can all be disabled or modified per the below options.

**Usage**

```r
addInteraction(sigmaObj, neighborEvent = "onClick", doubleClickZoom = TRUE,
               mouseWheelZoom = TRUE)
```

**Arguments**

- **sigmaObj** A 'sigmaNet' object - created using the 'sigmaFromIgraph' function
- **neighborEvent** Enable/disable event that highlights a node's neighbors. Can either be onClick, onHover, or None.
- **doubleClickZoom** Enable/disable zoom event on double click
- **mouseWheelZoom** Enable/disable zoom event on mouse wheel
**addListener**

Add a "listener" to report data from a 'sigmaNet' object in 'Shiny' back to the R session.

### Description

Add a "listener" to report data from a 'sigmaNet' object in 'Shiny' back to the R session.

### Usage

```r
addListener(sigmaObj, listener)
```

### Arguments

- **sigmaObj**: A 'sigmaNet' object - created using the 'sigmaFromIgraph' function
- **listener**: Either "clickNode" to listen to node clicks or "hoverNode" to listen to node hover

**addNodeColors**

Modify the node colors of a 'sigmaNet' object.

### Description

Modify the node colors in an existing 'sigmaNet' object by providing one of the following: (1) a single color to use for all nodes or; (2) a vertex attribute from your original 'igraph' object. If you are using a vertex attribute, you can also specify a color palette from the 'RColorBrewer' package.

### Usage

```r
addNodeColors(sigmaObj, oneColor = NULL, colorAttr = NULL, colorPal = "Dark2")
```
addNodeLabels

**Arguments**

- `sigmaObj` A `sigmaNet` object - created using the `sigmaFromIgraph` function
- `oneColor` A single color to color all of the nodes (hex format)
- `colorAttr` An attribute from the original `igraph` nodes to color the nodes by
- `colorPal` The color palette to use - only used if `colorAttr` is specified

**Details**

*It is most useful to use the pipe operator from the `magrittr` package with this function.*

**Value**

A `sigmaNet` object with modified node colors. This object can be called directly to create a visualization, or modified by additional functions.

**Examples**

```r
library(igraph)
library(sigmaNet)
library(magrittr)

data(lesMis)

l <- layout_nicely(lesMis)

# one color for all nodes
sig <- sigmaFromIgraph(graph = lesMis, layout = l) %>%
    addNodeColors(oneColor = '#D9F802')

sig

# color based on attribute (edge betweenness cluster)
clust <- cluster_edge_betweenness(lesMis)$membership
V(lesMis)$group <- clust

sig <- sigmaFromIgraph(graph = lesMis, layout = l) %>%
    addNodeColors(colorAttr = 'group', colorPal = 'Set1')

sig
```

---

**addNodeLabels**

*Modify the node labels of a `sigmaNet` object.*

**Description**

Modify the node labels of an existing `sigmaNet` object by providing an attribute from the initial `igraph` to use as the labels.
Usage

addNodeLabels(sigmaObj, labelAttr = NULL)

Arguments

sigmaObj A 'sigmaNet' object - created using the 'sigmaFromIgraph' function
labelAttr The attribute to use to create node labels

Value

A 'sigmaNet' object with modified node labels. This object can be called directly to create a visualization, or modified by additional functions.

Examples

library(igraph)
library(sigmaNet)
library(magrittr)

data(lesMis)

l <- layout_nicely(lesMis)
sig <- sigmaFromIgraph(graph = lesMis, layout = l) %>%
  addNodeLabels(labelAttr = 'label')
sig

---

addNodeSize

Modify the node size of a 'sigmaNet' object.

Description

Modify the node size of an existing 'sigmaNet' object by providing one of the following: (1) A single size to use for all nodes; (2) a vector of node sizes (this must be the same length as the number of nodes in the graph); or (3) a metric to use to scale the nodes.

Usage

addNodeSize(sigmaObj, minSize = 1, maxSize = 3, sizeMetric = "degree",
            sizeVector = NULL, oneSize = NULL)

Arguments

sigmaObj A 'sigmaNet' object - created using the 'sigmaFromIgraph' function
minSize The minimum node size on the graph (for scaling)
maxSize The maximum node size on the graph (for scaling)
sizeMetric  The metric to use when sizing the nodes. Options are: degree, closeness, betweenness, pageRank, or eigenCentrality.

sizeVector  An optional vector with the sizes for each node (overrides sizeMetric and min/maxSize)

oneSize  A single size to use for all nodes

Details

If using the 2nd or 3rd approach, specifying the minSize and maxSize attributes will scale the nodes according to your specification, between these min- and max sizes.

Value

A 'sigmaNet' object with modified node sizes This object can be called directly to create a visualization, or modified by additional functions.

Examples

library(igraph)
library(sigmaNet)
library(magrittr)

data(lesMis)
l <M layout_nicely(lesMis)

#one size for all nodes
sig <- sigmaFromIgraph(graph = lesMis, layout = l) %>%
  addNodeSize(oneSize = 3)
sig

#using a size attribute
sig <- sigmaFromIgraph(graph = lesMis, layout = l) %>%
  addNodeSize(sizeMetric = 'degree', minSize = 2, maxSize = 8)
sig

#using a vector
customSize <- log10(degree(lesMis))
sig <- sigmaFromIgraph(graph = lesMis, layout = l) %>%
  addNodeSize(sizeVector = customSize)
sig

lesMis  Co-appearances of characters in "Les Miserables"

Description

A graph where the nodes are characters in "Les Miserables" and the edges are times that the characters appeared together in the novel.
renderSigmaNet

Usage

lesMis

Format

An igraph object with 77 nodes and 254 edges

id  numeric id of nodes
label  character label (names) of nodes
value  numeric weight of the edges (number of co-appearances)

Source


renderSigmaNet  Render a 'sigmaNet' visualization in 'Shiny'

Description

Render a 'sigmaNet' visualization in 'Shiny'

Usage

renderSigmaNet(expr, env = parent.frame(), quoted = FALSE)

Arguments

expr  An expression that creates a 'sigmaNet' visualization
env  Defaults to parent.frame() - see 'Shiny' docs for more info
quoted  Defaults to FALSE - see 'Shiny' docs for more info
saveSigma

Save 'sigmaNet' object as html - a wrapper for saveWidget()

Description

Save an 'sigmaNet' object as an html file (without rendering it). This is especially helpful when dealing with very large graphs that could crash your R session if you attempt to render them in the 'Rstudio' viewer pane.

Usage

saveSigma(sigmaObj, fileName = NULL)

Arguments

sigmaObj A 'sigmaNet' object - created using the 'sigmaFromIgraph' function
fileName A name for your html output (with or without .html at the end)

Value

An html file in your working directory (or other specified directory). This file is a standalone representation of your 'Sigma.js' visualization that can be shared and moved freely. This object will maintain its interactivity.

Examples

library(igraph)
library(sigmaNet)
library(magrittr)

data(lesMis)
l <- layout_nicely(lesMis)
sig <- sigmaFromIgraph(graph = lesMis, layout = l)

## Not run:
saveSigma(sig, fileName = file.path(tempdir(), 'myFile.html'))

## End(Not run)
**sigmaFromIgraph**  
*Make a basic 'sigmaNet' graph object from an 'igraph' object*

**Description**

Create a 'sigmaNet' object from an 'igraph' object. The 'sigmaNet' object will be a basic visualization of the 'igraph' object and is meant to be the starting point for the development of a useful 'Sigma.js' visualization. If you are familiar with the 'ggplot' syntax, this is similar to the basic 'ggplot' function.

**Usage**

```r
sigmaFromIgraph(graph, layout = NULL, width = NULL, height = NULL, 
                 elementId = NULL)
```

**Arguments**

- `graph`  
  An 'igraph' object
- `layout`  
  The output of one of the 'igraph' layout functions. If not provided, layout_nicely() will be used (note, this will slow things down).
- `width`  
  Width of the resulting graph - defaults to fit container, probably leave this alone
- `height`  
  Height of the resulting graph - defaults to fit container, probably leave this alone
- `elementId`  
  Do not specify, this is used by the 'htmlwidgets' package

**Value**

A 'sigmaNet' object (which is an 'htmlwidget'). This object is meant to be called directly to render a default 'Sigma.js' visualization, or it can be passed to other arguments to change visualization attributes (colors, sizes, interactivity, etc.).

**Examples**

```r
library(igraph)
library(sigmaNet)

data(lesMis)

l <- layout_nicely(lesMis)
sig <- sigmaFromIgraph(graph = lesMis, layout = l)

# render basic visualization by calling the object
sig
```
Create a UI element for a 'sigmaNet' visualization in 'Shiny'

Usage

`sigmaNetOutput(outputId, width = "100\%", height = "400px")`

Arguments

- `outputId`  The ID of the UI element
- `width`  The width of the UI element
- `height`  The height of the UI element
Index

*Topic datasets
  lesMis, 8

addEdgeColors, 2
addEdgeSize, 3
addInteraction, 4
addListener, 5
addNodeColors, 5
addNodeLabels, 6
addNodeSize, 7

lesMis, 8
renderSigmaNet, 9

saveSigma, 10
sigmaFromIgraph, 11
sigmaNetOutput, 12