Package 'cyjShiny'

<div align="center">March 28, 2023</div>

Title Cytoscape.js Shiny Widget (cyjShiny)

Version 1.0.42

Date 2023-03-28

Description Wraps cytoscape.js as a shiny widget. cytoscape.js <https://js.cytoscape.org/> is a Javascript-based graph theory (network) library for visualization and analysis. This package supports the visualization of networks with custom visual styles and several available layouts. Demo Shiny applications are provided in the package code.

Depends R (>= 3.5.0), htmlwidgets, shiny, jsonlite, graph, base64enc

Suggests RUnit, knitr, rmarkdown, markdown, BiocStyle

License MIT + file LICENSE

Encoding UTF-8

LazyData false

Collate 'graphsToJSON.R' 'cyjShiny.R'

VignetteBuilder knitr

RoxygenNote 7.2.0

NeedsCompilation no

Author Omar Shah [ctb],
Paul Shannon [aut],
Augustin Luna [aut, cre] (<https://orcid.org/0000-0001-5709-371X>)

Maintainer Augustin Luna <augustin_luna@hms.harvard.edu>

Repository CRAN

Date/Publication 2023-03-28 21:00:02 UTC

R topics documented:

addGraphFromDataFrame   *Add graph from data.frame*

Description

Add graph from data.frame

Usage

```
addGraphFromDataFrame(session, tbl.edges, tbl.nodes = NULL)
```

Arguments

session	a Shiny Server session object.
tbl.edges	a data.frame with source, traget, interaction columns (and option other attributes)
tbl.nodes	(optional; nodes can be deduced from tbl.edges) a data.frame with nodes and their attributes

Value

Nothing

Examples

```
## Not run:
addGraphFromDataFrame(session)

## End(Not run)
```

---

addGraphFromJsonFile   *Add graph from JSON file*

Description

Add graph from JSON file

Usage

```
addGraphFromJsonFile(session, jsonFilename)
```

Arguments

session	a Shiny Server session object.
jsonFilename	of a text file with JSON representation of a cytoscape.js graph

Value

Nothing

Examples

```
## Not run:
addGraphFromJsonFile(session)

## End(Not run)
```

---

clearSelection   *Clear selection all node and edge selections removed*

Description

Clear selection all node and edge selections removed

Usage

```
clearSelection(session)
```

Arguments

session	a Shiny Server session object.

Value

Nothing

---

cyjShiny   *cyjShiny cyjShiny*

Description

This widget wraps cytoscape.js, a full-featured Javsscript network library for visualization and analysis.

Usage

```
cyjShiny(
  graph,
  layoutName,
  styleFile = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

Arguments

graph	a graph in json format; converters from graphNEL and data.frame/s offered ("see also" below)
layoutName	character one of:"preset", "cose", "cola", "circle", "concentric", "breadthfirst", "grid", "random"
styleFile,	default NULL, can name a standard javascript cytoscape.js style file
width	integer initial width of the widget.
height	integer initial height of the widget.
elementId	string the DOM id into which the widget is rendered, default NULL is best.

Value

a reference to an htmlwidget.

See Also

[dataFramesToJSON](#)

[graphNELtoJSON](#)

Examples

```
tbl.nodes <- data.frame(
  id = c("A", "B", "C"),
  type = c("kinase", "TF", "glycoprotein"),
  lfc = c(-3, 1, 1),
  count = c(0, 0, 0),
  stringsAsFactors = FALSE
)

tbl.edges <- data.frame(
  source = c("A", "B", "C"),
  target = c("B", "C", "A"),
  interaction = c("phosphorylates", "synthetic lethal", "unknown"),
  stringsAsFactors = FALSE
)

  #  simple legitimate graph, nodes implied, but no node attributes
graph.json.v1 <- dataFramesToJSON(tbl.edges)
  # nodes and edges both explicit,  attributes specified
graph.json.v2 <- dataFramesToJSON(tbl.edges, tbl.nodes)

g <- graphNEL(nodes = c("A", "B", "C"), edgemode = "directed")
g <- addEdge("A", "B", g)
graph.json.v3 <- graphNELtoJSON(g)

# output$cyjShiny <- renderCyjShiny(cyjShiny(graph.json.v[123]))
```

---

cyjShinyOutput   *Standard shiny ui rendering construct*

Description

Standard shiny ui rendering construct

Usage

```
cyjShinyOutput(outputId, width = "100%", height = "400")
```

Arguments

outputId	the name of the DOM element to create.
width	integer optional initial width of the widget.
height	integer optional initial height of the widget.

Value

a reference to an htmlwidget

Examples

```
## Not run:
mainPanel(cyjShinyOutput("cyjShiny"), width = 10)

## End(Not run)
```

---

dataFramesToJSON	*Create a cytoscape.js JSON graph from one or two data.frames.*

Description

Create a cytoscape.js JSON graph from one or two data.frames.

Usage

```
dataFramesToJSON(tbl.edges, tbl.nodes = NULL)
```

Arguments

tbl.edges	data.frame, with source, target and interaction columns, others option for edge attributes
tbl.nodes	data.frame, options, useful for orphan nodes, and necessary for adding node attributes

Value

a string with a cytoscape.js JSON graph

---

doLayout	*Layout the current graph using the specified strategy.*

Description

Layout the current graph using the specified strategy.

Usage

```
doLayout(session, strategy)
```

Arguments

session	a Shiny Server session object.
strategy	a character string, one of cola, cose, circle, concentric, grid, breadthfirst, random, dagre, cose-bilkent.

Value

Nothing

Examples

```
## Not run:
doLayout(session, "cola")

## End(Not run)
```

---

fit   *Set zoom and center of the graph display so that graph fills the display.*

Description

Set zoom and center of the graph display so that graph fills the display.

Usage

```
fit(session, padding = 50)
```

Arguments

session	a Shiny server session object.
padding	integer, default 50 pixels.

Value

Nothing

See Also

[fitSelected](#)

Examples

```
## Not run:
fit(session, 100)

## End(Not run)
```

---

fitSelected   *Set zoom and center of the graph display so that the currently selected nodes fill the display*

Description

Set zoom and center of the graph display so that the currently selected nodes fill the display

Usage

```
fitSelected(session, padding = 50)
```

Arguments

session   a Shiny server session object.

padding   integer, default 50 pixels.

Value

Nothing

See Also

[fit](#)

Examples

```
## Not run:
fitSelected(session, 100)

## End(Not run)
```

---

getNodePositions   *Get node positions*

Description

Get node positions

Usage

```
getNodePositions(session)
```

Arguments

session   a Shiny Server session object.

Value

Nothing

---

getSelectedNodes *Get Selected Nodes*

Description

Get Selected Nodes

Usage

```
getSelectedNodes(session)
```

Arguments

session   a Shiny server session object.

Value

a data.frame with (at least) an id column

`getSelectedNodes` get the selected nodes

---

graphNELtoJSON *Convert R graphNEL object to cytoscape.js JSON.*

Description

Convert R graphNEL object to cytoscape.js JSON.

Usage

```
graphNELtoJSON(g)
```

Arguments

g   a graphNEL

Value

a string with a cytoscape.js JSON graph

Examples

```
## Not run:
g.json <- graphNELtoJSON(graphNEL())

## End(Not run)
```

---

hideSelection *Hide selection all selected nodes and their edges are hidden*

Description

Hide selection all selected nodes and their edges are hidden

Usage

```
hideSelection(session)
```

Arguments

session   a Shiny Server session object.

Value

Nothing

---

invertSelection *Invert selection all selected nodes and their edges are hidden*

Description

Invert selection all selected nodes and their edges are hidden

Usage

```
invertSelection(session)
```

Arguments

session   a Shiny Server session object.

Value

Nothing

`loadNetworkFromJSONFile`

Load a standard cytoscape.js JSON network file

Description

Load a standard cytoscape.js JSON network file

Usage

```
loadNetworkFromJSONFile(filename)
```

Arguments

`filename`	character string, either relative or absolute path.

Value

Nothing

Examples

```
## Not run:
loadNetworkFromJSONFile(system.file(package = "cyjShiny", "extdata", "galFiltered.cyjs"))

## End(Not run)
```

`loadStyleFile` *Load a standard cytoscape.js style file*

Description

Load a standard cytoscape.js style file

Usage

```
loadStyleFile(styleFile)
```

Arguments

`styleFile`	character string, either relative or absolute path.

Value

Nothing

Examples

```
## Not run:
loadStyleFile(system.file(package = "cyjShiny", "extdata", "yeastGalactoseStyle.js"))

## End(Not run)
```

---

readAndStandardizeJSONNetworkFile

Read in a JSON network file, identify (or add) elements field return JSON

Description

Read in a JSON network file, identify (or add) elements field return JSON

Usage

```
readAndStandardizeJSONNetworkFile(filename)
```

Arguments

filename   a JSON file

Value

a string with a cytoscape.js JSON graph

---

readAndStandardizeJSONStyleFile

Read in a JSON file, extract the selector elements, return JSON

Description

this utility function examines the incoming JSON, returns exactly and only an array of selector objects

Usage

```
readAndStandardizeJSONStyleFile(filename)
```

Arguments

filename   a json file

Details

there are at least two JSON object structures used to specify style (see function comments in code for more details):

- simple: an array of selector objects

- more complex, exported from the Cytoscape desktop application this is also an array of objects, one named "style" which (like the simple format described above) contains an array of selectors.

Value

a string with a cytoscape.js JSON graph

---

removeGraph   *Remove the current graph*

Description

Remove the current graph

Usage

```
removeGraph(session)
```

Arguments

session   a Shiny Server session object.

Value

Nothing

Examples

```
## Not run:
removeGraph(session)

## End(Not run)
```

---

renderCyjShiny | *More shiny plumbing - a cyjShiny wrapper for htmlwidget standard rendering operation*

Description

More shiny plumbing - a cyjShiny wrapper for htmlwidget standard rendering operation

Usage

```
renderCyjShiny(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

expr	an expression that generates an HTML widget.
env	environment in which to evaluate expr.
quoted	logical specifies whether expr is quoted ("useuful if you want to save an expression in a variable").

Value

output from htmlwidgets rendering operation

---

savePNGtoFile | *Save a png rendering of the current network view to the specified filename*

Description

Save a png rendering of the current network view to the specified filename

Usage

```
savePNGtoFile(session, filename)
```

Arguments

session	a Shiny Server session object.
filename	a character string

Value

Nothing

---

selectFirstNeighbors   *Select first neighbors of the currently selected nodes*

Description

Select first neighbors of the currently selected nodes

Usage

```
selectFirstNeighbors(session)
```

Arguments

session   a Shiny Server session object.

Value

Nothing

---

selectNodes   *Select Nodes*

Description

Select Nodes

Usage

```
selectNodes(session, nodeNames)
```

Arguments

session   a Shiny Server session object.

nodeNames   character, a list of node IDs

Value

Nothing

---

setEdgeAttributes   *Assign the supplied edge attribute values to the graph structure con-*
                                 tained in the browser.

Description

Assign the supplied edge attribute values to the graph structure contained in the browser.

Usage

```
setEdgeAttributes(
  session,
  attributeName,
  sourceNodes,
  targetNodes,
  interactions,
  values
)
```

Arguments

session	a Shiny Server session object.
attributeName	character string, the attribute to update.
sourceNodes	a character vector, the names of the source nodes of the edges
targetNodes	a character vector, the names of the target nodes of the edgees
interactions	a character vector, further identifying the specific edge whose attributes are updated.
values	a character, logical or numeric vector, the new values.

Value

Nothing

Examples

```
## Not run:
setEdgeAttributes(session,
  attributeName = "score",
  sourceNodes = c("A", "B", "C"),
  targetNodes = c("D", "E", "A"),
  interactions = c("promotes", "promotes", "inhibits"),
  values = new.scores
)

## End(Not run)
```

---

setNodeAttributes *Assign the supplied node attribute values to the graph structure con-*
tained in the browser.

Description

Assign the supplied node attribute values to the graph structure contained in the browser.

Usage

```
setNodeAttributes(session, attributeName, nodes, values)
```

Arguments

session	a Shiny Server session object.
attributeName	character string, the attribute to update.
nodes	a character vector the names of the nodes whose attributes are updated.
values	a character, logical or numeric vector, the new values.

Value

Nothing

Examples

```
## Not run:
setNodeAttributes(session,
  attributeName = attribute,
  nodes = yeastGalactodeNodeIDs,
  values = expression.vector
)

## End(Not run)
```

---

setNodePositions *Set node positions from the supplied data.frame*

Description

Set node positions from the supplied data.frame

Usage

```
setNodePositions(session, tbl.positions)
```

Arguments

session   a Shiny Server session object.

tbl.positions   a data.frame with three columns: id, x, y

Value

Nothing

---

showAll   *Show all all selected nodes and their edges are hidden*

Description

Show all all selected nodes and their edges are hidden

Usage

```
showAll(session)
```

Arguments

session   a Shiny Server session object.

Value

Nothing

Index