Package: RedeR (via r-universe)

November 24, 2026

Type Package

Title Interactive visualization and manipulation of nested networks

Version 3.10.0

Depends R (>= 4.4), methods

Imports scales, igraph

Suggests knitr, rmarkdown, BiocStyle, TreeAndLeaf

SystemRequirements Java Runtime Environment (Java>= 11)

Description RedeR combines an R package with a stand-alone Java application for interactive visualization and manipulation of nested networks. Graph, node, and edge attributes can be configured using either graphical or command-line methods, following igraph syntax rules.

License GPL-3

biocViews GUI, GraphAndNetwork, Network, NetworkEnrichment, NetworkInference, Software, SystemsBiology

VignetteBuilder knitr

BugReports https://github.com/sysbiolab/RedeR/issues

URL https://doi.org/10.1186/gb-2012-13-4-r29

Collate AllChecks.R AllValidations.R AllClasses.R AllGenerics.R internalLegendMethods.R internalOtherMethods.R exportedCallMethods.R exportedOtherMethods.R exportedAddGraphMethods.R exportedGetGraphMethods.R exportedRelaxMethods.R exportedNestMethods.R exportedAttributeSeters.R exportedWrappers.R Misc.R zzz.R

Encoding UTF-8

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libglpk-dev default-jdk libxml2-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:31:05 UTC

RemoteUrl https://github.com/bioc/RedeR

RemoteRef RELEASE_3_24

RemoteSha b83b72a3a08a5d464bfea053268538b478d095d9

Contents

addEdges,character-method . . . 3
addGraph,RedPort-method . . . 4
addGraphToRedeR . . . 5
addLegendToRedeR . . . 7
addNodes,character-method . . . 8
calld,RedPort-method . . . 9
deleteEdges,character-method . . . 10
deleteNodes,character-method . . . 11
exitd,RedPort-method . . . 12
exitRedeR . . . 13
getGraph,RedPort-method . . . 14
getGraphFromRedeR . . . 15
mergeOutEdges,numeric_Or_missing-method . . . 16
nestNodes,character-method . . . 17
ping,RedPort-method . . . 20
pingRedeR . . . 21
RedeR-data1 . . . 21
RedeR-data2 . . . 22
rederInfo . . . 23
RedPort . . . 24
RedPort-class . . . 25
relax,RedPort-method . . . 25
relaxRedeR . . . 27
resetd,RedPort-method . . . 29
resetRedeR . . . 30
selectEdges,character-method . . . 30
selectNodes,character-method . . . 32
startRedeR . . . 33
subg . . . 34
transform.attributes . . . 35
updateLayoutFromRedeR . . . 38
version,RedPort-method . . . 39

Index **40**

addEdges,character-method
addEdges

Description

Add edges to an active RedeR application.

Usage

```
## S4 method for signature 'character'
addEdges(edges, ...)

## S4 method for signature 'data.frame'
addEdges(edges, ...)
```

Arguments

edges	A vertex sequence <vector of strings> or data frame of ncol=2.
...	Arguments passed to internal checks (ignore).

Value

Add edges to an active RedeR session.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR, getGraphFromRedeR.

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create some edges as a vertex sequence
edges <- c("n1", "n2", "n1", "n3", "n1", "n4", "n1", "n5")

# ...or as a data.frame
edges <- data.frame(
    A = c("n1", "n1", "n1", "n1"),
    B = c("n2", "n3", "n4", "n5")
)
```

```
# Start the RedeR interface and add edges
startRedeR()
addEdges(edges)
```

---

addGraph,RedPort-method

Adding 'igraph' objects to RedeR

Description

Methods to display igraph objects in the RedeR application.

Usage

```
## S4 method for signature 'RedPort'
addGraph(
  obj,
  g,
  layout = NULL,
  gscale = 75,
  zoom = 100,
  update.coord = TRUE,
  verbose = TRUE,
  isNested = FALSE,
  ...
)
```

Arguments

obj	A RedPort-class object.
g	An igraph object. It must include coordinates and names assigned to x, y, and name vertex attributes.
layout	an optional numeric matrix with two columns for x and y coordinates <numeric>.
gscale	Expansion factor related to the app panel area <numeric>
zoom	A zoom scale for the app panel (range: 0.0 to 100.0) <numeric>.
update.coord	A logical value, whether to update x and y coordinates in the app.
verbose	A logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
isNested	A logical value, whether to nest all nodes into a new container.
...	Additional arguments passed to the nestNodes function (used when isNested = TRUE).

Value

Send igraph objects to RedeR.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR

Examples

```
# Initialize RedeR and igraph
library(RedeR)
library(igraph)

gtoy <- graph.lattice(c(5, 5, 5))

rdp <- RedPort()

## Not run:
calld(rdp)
addGraph(rdp, g = gtoy, layout = layout_nicely(gtoy))

## End(Not run)
```

---

addGraphToRedeR *Adding 'igraph' objects to RedeR*

Description

Methods to display igraph objects in the RedeR application.

Usage

```
addGraphToRedeR(
  g,
  layout = NULL,
  gscale = 75,
  zoom = 100,
  update.coord = TRUE,
  isNested = FALSE,
  unit = c("native", "point", "npc"),
  verbose = TRUE,
  ...
)
```

Arguments

g	An igraph object. It must include coordinates and names assigned to x, y, and name vertex attributes.
layout	an optional numeric matrix with two columns for x and y coordinates <numeric>.
gscale	Expansion factor related to the app panel area <numeric>
zoom	A zoom scale for the app panel (range: 0.0 to 100.0) <numeric>.
update.coord	A logical value, whether to update x and y coordinates in the app.
isNested	A logical value, whether to nest all nodes into a new container.
unit	A string specifying the unit for *lengths*, *widths*, and *sizes* assigned to node and edge attributes. RedeR space coordinate system is native to Java Graphics2D, which uses 'points' by default (a point is 1/72 of an inch). Current options include 'native', 'point', and 'npc'. The 'native' option will used definition from options('RedeR'), which is set to 'point' by default. The 'npc' option will normalize attribute values to RedeR's viewport.
verbose	A logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
...	Additional arguments passed to the nestNodes function (used when isNested = TRUE).

Value

Send igraph objects to RedeR.

Author(s)

Sysbiolab.

See Also

startRedeR, getGraphFromRedeR

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create an igraph
gtoy <- graph.lattice(c(5, 5, 5))


# Start the RedeR interface
startRedeR()

# Send graph to RedeR
addGraphToRedeR(g = gtoy, layout = layout_nicely(gtoy))
```

addLegendToRedeR *Adding graph legends to the RedeR app*

Description

Methods to display legends in the RedeR app.

Usage

```
addLegendToRedeR(
  x,
  type = "nodecolor",
  position = "default",
  orientation = "default",
  title = type,
  font.size = 12,
  stretch = 0,
  ...
)
```

Arguments

x	A vector with legend values (see examples).
type	A legend type. Options: 'nodecolor', 'edgecolor', 'nodesize', 'edgewidth', 'nodeshape', 'edgetype'.
position	Position of the legend in app panel. Options: 'default', 'topleft', 'topright', 'bottomleft', 'bottomright', and 'remove'. Use 'default' to place the legend on a predefined slot, or 'remove' to delete the legend type.
orientation	The orientation of the legend. Options: 'default', 'vertical', 'horizontal'. Use 'default' to automatically set the orientation for the legend type.
title	A string for legend title.
font.size	Font size (unit in points).
stretch	A scaling factor to adjust the legend box (between 0 and 1).
...	Arguments passed to internal checks (ignore).

Value

Send legend objects to RedeR app.

Author(s)

Sysbiolab.

See Also

startRedeR, addGraphToRedeR

Examples

```
# Load RedeR
library(RedeR)


# Start the RedeR interface
startRedeR()

# Adding node and edge color legends
cols <- colorRampPalette(colors = c("red", "blue"))(14)
names(cols) <- 1:length(cols)
addLegendToRedeR(x = cols, type = "nodecolor")
addLegendToRedeR(x = cols, type = "edgecolor", stretch = 0.1)

# Adding node size legend
nsize <- c(10, 20, 30, 40, 50)
addLegendToRedeR(x = nsize, type = "nodesize")

# Adding edge width legend
esize <- c(1:10)
addLegendToRedeR(x = esize, type = "edgewidth")

# Adding node shape legend
shape <- c("ELLIPSE","RECTANGLE","ROUNDED_RECTANGLE","TRIANGLE","DIAMOND")
names(shape) <- shape
addLegendToRedeR(x = shape, type = "nodeshape")

# Adding edge linetype legend
ltype <- c("SOLID", "DOTTED", "DASHED", "LONG_DASH")
names(ltype) <- ltype
addLegendToRedeR(x = ltype, type = "edgetype")
```

---

addNodes,character-method

addNodes

Description

Add nodes to an active RedeR application.

Usage

```
## S4 method for signature 'character'
addNodes(nodes, ...)
```

Arguments

nodes	A vector with node names.
...	Arguments passed to internal checks (ignore).

Value

Add nodes to an active RedeR session.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR, getGraphFromRedeR.

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create a vector with node names
nodes <- c("n1", "n2", "n3", "n4", "n5")


# Start the RedeR interface and add nodes
startRedeR()
addNodes(nodes)
```

---

calld,RedPort-method *Call RedeR app from R*

Description

Method to launch RedeR application from R.

Usage

```
## S4 method for signature 'RedPort'
calld(obj, filepath = "default", maxlag = 20, checkcalls = FALSE)
```

Arguments

obj	A RedPort-class object.
filepath	A path to the 'reder.jar' file available in the RedeR R package <string>
maxlag	Max acceptable lag time for the R-Java callback confirmation (default=20 s) <numeric>.
checkcalls	A logical value, whether to report errors from the R-to-Java calls.

Value

System call to start the RedeR application.

Author(s)

Sysbiolab.

See Also

startRedeR

Examples

```
rdp <- RedPort()


calld(rdp)
```

`deleteEdges,character-method`
deleteEdges

Description

Delete edges from an active RedeR application.

Usage

```
## S4 method for signature 'character'
deleteEdges(edges, ...)

## S4 method for signature 'data.frame'
deleteEdges(edges, ...)
```

Arguments

edges	A vertex sequence <vector of strings> or data frame of ncol=2.
...	Arguments passed to internal checks (ignore).

Value

Remove graph objects from RedeR app.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR, getGraphFromRedeR.

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create some edges as a data.frame
edges <- data.frame(
    A = c("n1", "n1", "n1", "n1"),
    B = c("n2", "n3", "n4", "n5")
)


# Start the RedeR interface
startRedeR()

# Add and delete edges
addEdges(edges)
deleteEdges(c("n1", "n3", "n1", "n6"))
```

---

deleteNodes,character-method *deleteNodes*

Description

Delete nodes from an active RedeR application.

Usage

```
## S4 method for signature 'character'
deleteNodes(nodes, ...)
```

Arguments

nodes	A vector with node names.
...	Arguments passed to internal checks (ignore).

Value

Remove graph objects from RedeR app.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR, getGraphFromRedeR.

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create a vector with node names
nodes <- c("n1", "n2", "n3", "n4", "n5")


# Start the RedeR interface
startRedeR()

# Add and delete nodes
addNodes(nodes)
deleteNodes(c("n1", "n3"))
```

---

exitd,RedPort-method *Exit the RedeR R-to-Java interface*

Description

Close an active RedeR session.

Usage

```
## S4 method for signature 'RedPort'
exitd(obj)
```

Arguments

obj	A RedPort-class object.

Value

Exit/close the RedeR application.

Author(s)

Sysbiolab.

See Also

exitRedeR

Examples

```
rdp <- RedPort()


calld(rdp)
exitd(rdp)
```

---

exitRedeR *Exit the RedeR R-to-Java interface*

Description

Close an active RedeR session.

Usage

```
exitRedeR()
```

Value

Exit/close the RedeR application.

Author(s)

Sysbiolab.

See Also

startRedeR

Examples

```
# Load RedeR
library(RedeR)


# Call 'start' and 'exit' methods
startRedeR()
exitRedeR()
```

getGraph,RedPort-method

Get graphs from RedeR

Description

Methods to wrap up RedeR graphs into igraph's R objects.

Usage

```
## S4 method for signature 'RedPort'
getGraph(
  obj,
  status = c("all", "selected", "notselected"),
  attribs = c("all", "minimal"),
  type = c("node", "container", "all")
)
```

Arguments

obj	A RedPort-class object.
status	A filter (string) indicating the status of the graph elements that should be fetched from the RedeR app (default='all').
attribs	A filter (string) indicating the graph attributes that should be fetched from the RedeR app (default='all').
type	A filter (string) indicating the graph element types that should be fetched from the RedeR app (default='node').

Value

igraph objects from RedeR.

Author(s)

Sysbiolab.

See Also

getGraphFromRedeR

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

g <- graph.lattice(c(5, 5, 5))
```

```
rdp <- RedPort()


calld(rdp)
addGraph(rdp, g, layout_nicely(g))
g <- getGraph(rdp)
```

`getGraphFromRedeR` *Get graphs from RedeR*

Description

Methods to wrap up RedeR graphs into igraph's R objects.

Usage

```
getGraphFromRedeR(
  status = c("all", "selected", "notselected"),
  attribs = c("all", "minimal"),
  type = c("node", "container", "all"),
  unit = c("native", "point", "npc"),
  ...
)
```

Arguments

`status`	A filter (string) indicating the status of the graph elements that should be fetched from the RedeR app (default='all').
`attribs`	A filter (string) indicating the graph attributes that should be fetched from the RedeR app (default='all').
`type`	A filter (string) indicating the graph element types that should be fetched from the RedeR app (default='node').
`unit`	A string specifying the unit for *lengths*, *widths*, and *sizes* assigned to node and edge attributes. RedeR space coordinate system is native to Java Graphics2D, which uses 'points' by default (a point is 1/72 of an inch). Current options include 'native', 'point', and 'npc'. The 'native' option will used definition from `options('RedeR')`, which is set to 'point' by default. The 'npc' option will return attribute values normalized to RedeR's viewport.
`...`	Arguments passed to internal checks (ignore).

Value

igraph objects from RedeR.

Author(s)

Sysbiolab.

See Also

startRedeR, addGraphToRedeR

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create an igraph
gtoy1 <- graph.lattice(c(3, 3, 3))


# Start the RedeR interface
startRedeR()

# Send graph to RedeR
addGraphToRedeR(g = gtoy1)

# Get graph from RedeR
gtoy2 <- getGraphFromRedeR()
```

---

mergeOutEdges,numeric_Or_missing-method
mergeOutEdges

Description

Method to assign out-edges to containers in an active RedeR session. This method transfers edges from nodes to the respective containers.

Usage

```
## S4 method for signature 'numeric_Or_missing'
mergeOutEdges(nlevels = 2, rescale = TRUE, lb = NA, ub = NA, rdp = NA)
```

Arguments

nlevels	Number of levels (>=1) to be merged in the nested network.
rescale	Logical value, whether to rescale out-edge width to not overextend the container size; if 'FALSE', it will run a simple sum when combining the out-edges.
lb	Custom lower bound to rescale edge width between containers.

ub	Custom upper bound to rescale edge width between containers.
rdp	A RedPort-class object used by internal calls (ignore).

Value

Add/change edge assigments.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR, getGraphFromRedeR.

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# create a fully connected graph
g <- igraph::make_full_graph(5)
V(g)$name <- paste0("n", 1:5)


# Start the RedeR interface
startRedeR()

# Add 'g' to the interface
addGraphToRedeR(g, layout.kamada.kawai(g))

# Nest nodes in the interface
nestNodes(c("n1", "n2", "n3"), gcoord = c(30, 30), gscale = 30)
nestNodes(c("n4", "n5"), gcoord = c(70, 70), gscale = 20)

# Merge nodes between containers
mergeOutEdges()
```

---

nestNodes,character-method

nestNodes

Description

Nest nodes into containers.

Usage

```
## S4 method for signature 'character'
nestNodes(
  nodes,
  isAssigned = TRUE,
  isAnchored = TRUE,
  gscale = 40,
  gcoord = c(50, 50),
  status = c("plain", "hide", "transparent"),
  theme = c("th0", "th1", "th2", "th3"),
  gatt = list(),
  parent = NULL,
  verbose = TRUE,
  rdp = NA
)
```

Arguments

nodes	A vector with node names available in the RedeR app.
isAssigned	Logical value, whether to assign the container name to the nested nodes
isAnchored	Logical value, whether to anchor the container in dynamic layout sessions.
gscale	Expansion factor of the nest area related to a parent nest, or related to the app panel.
gcoord	A numeric vector with 'x' and 'y' coordinates for the center of nest related to the app panel or a parent container. Coordinates between 0 and 100 are set to visible areas of the app panel.
status	Status of the container on the screen: 'plain', 'transparent', or 'hide'.
theme	Some pre-defined graph attributes. Options: 'th0', 'th1', 'th2', and 'th3'.
gatt	List of container attributes (see details).
	nestShape A single string.
	nestSize A single number >=0.
	nestColor A single color name or hexadecimal code.
	nestLabel A single string.
	nestLabelSize A single number >=0.
	nestLabelColor A single color name or hexadecimal code.
	nestLabelCoords A numeric vector with two numbers (e.g. c(x=0, y=0)).
	nestLineType A single string.
	nestLineWidth A single number >=0.
	nestLineColor A single color name or hexadecimal code.
parent	Optional argument, a nest ID of a parent nest. It must be used with 'isAssign=TRUE'.
verbose	A logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
rdp	A RedPort-class object used by internal calls (ignore).

Details

The gatt argument can be used to pass detailed attributes to containers, for example, gatt = list(nestLabel="Nest1").

- Options for nestShape: "ELLIPSE", "RECTANGLE", "ROUNDED_RECTANGLE", "TRIANGLE", and "DIAMOND"
- Options for nestLineType: "SOLID", "DOTTED", "DASHED", "LONG_DASH".
- When nestLabelCoords = c(x=0, y=0) then the label will be centered and placed at the top of the container.

Value

Add/change graph objects.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR, getGraphFromRedeR.

Examples

```
# Initialize RedeR and igraph
library(RedeR)
library(igraph)

# create a graph from an edge list
el <- matrix(c("n1", "n2", "n3", "n4"), ncol = 2, byrow = TRUE)
g <- graph.edgelist(el)


# Start the RedeR interface
startRedeR()

# Add 'g' to the interface
addGraphToRedeR(g, layout.kamada.kawai(g))

# Nest nodes in the interface
nestNodes(c("n1", "n2"), gcoord = c(30, 30))
nestNodes(c("n3", "n4"), gcoord = c(70, 70))
```

`ping,RedPort-method` *Ping RedeR app*

Description

Test the R-to-Java interface of an active RedeR session.

Usage

```
## S4 method for signature 'RedPort'
ping(obj)
```

Arguments

`obj`	A `RedPort`-class object.

Value

Ping test for RedeR app, either '1' (accessible) or '0' (not accessible)

Author(s)

Sysbiolab.

See Also

`pingRedeR`

Examples

```
rdp <- RedPort('MyPort')


ping(rdp)
# [1] 0
calld(rdp)
ping(rdp)
# [1] 1
```

pingRedeR *Ping RedeR app*

Description

Test the R-to-Java interface of an active RedeR session.

Usage

```
pingRedeR()
```

Value

Ping test for RedeR app.

Author(s)

Sysbiolab.

See Also

startRedeR

Examples

```
# Load RedeR
library(RedeR)


# Call 'start' and 'ping' methods
startRedeR()
pingRedeR()
```

RedeR-data1 *Pre-processed igraph object for RedeR case studies.*

Description

Preprocessed Human interactome extracted from the Human Protein Reference Database (HPRD) in April 2011 <igraph object> ('name' attribute is mapped to ENTREZ ID).

Usage

```
data(hs.inter)
```

Format

igraph

Value

A pre-processed igraph object.

Source

This package.

Examples

```
data(hs.inter)
```

RedeR-data2 *Pre-processed dataset for RedeR case studies.*

Description

Preprocessed data from a time-course gene expression and ChIP-on-chip analysis of estrogen receptor (ER) binding sites in the MCF7 cell line (Carroll et al, 2006).

Usage

```
data(ER.limma)
```

Format

igraph

Details

The 'ER.limma' dataset contains results from a differential gene expression analysis described elsewhere (Castro et al., 2012). This dataset also includes annotation of ER-binding sites. The original gene expression dataset (Carroll et al.) consists of 12 time-course Affymetrix U133Plus2.0 microarrays: 3 replicates at 0h, 3 replicates at 3h, 3 replicates at 6h and 3 replicates at 12h. The original dataset is available at the GEO database (GSE11324).

ER.limma A data-frame containing pre-processed results from a 'limma' analysis listing the DE genes only. The data-frame columns list the following information: annotation (ENTREZ and Symbol), time-course fold change (logFC.t3, logFC.t6, logFC.t12), p values (p.value.t3, p.value.t6, p.value.t12), DE genes (degenes.t3, degenes.t6, degenes.t12) and kb distance of the nearest ER-binding site to the TSS (ERbdist).

Value

A pre-processed dataset.

Source

Carroll JS et al., Genome-wide analysis of estrogen receptor binding sites. Nat Genet. 38(11):1289-97, 2006.

Castro MA et al. RedeR: R/Bioconductor package for representing modular structures, nested networks and multiple levels of hierarchical associations. Genome Biology, 13(4):R29, 2012.

Examples

```
data(ER.limma)
```

`rederInfo` *RedeR Attribute Information*

Description

Retrieves information on graph-, vertex-, and edge-level attributes recognized by the RedeR application, including usage descriptions and/or value specifications.

Usage

```
rederInfo(what = c("Usage", "Value", "All"))
```

Arguments

`what`	Character string specifying which attribute fields to return.

Value

Either display info summary or a named list with three elements:

Graph Data frame of graph-level attributes.

Vertex Data frame of vertex-level attributes.

Edge Data frame of edge-level attributes.

Author(s)

Sysbiolab.

Examples

```
## List usage information
rederInfo("Usage")

## List value information
rederInfo("Value")

## List both usage and value information
rederInfo("All")
```

RedPort *Constructor of RedPort-class objects*

Description

Constructor of the RedeR interface for remote procedure calls.

Usage

```
RedPort(title = "default", host = "127.0.0.1", port = 9091, checkJava = FALSE)
```

Arguments

title	A string naming the RedeR interface.
host	Domain name of the remote computer that is running the interface.
port	An integer specifying the port on which the interface should listen for incoming requests.
checkJava	A logical value, whether to check the Java Runtime Environment (JRE) installed on the system.

Value

An object of the RedPort class.

Author(s)

Sysbiolab.

See Also

startRedeR

Examples

```
# Initialize RedeR
library(RedeR)

rdp <- RedPort()

# Set global options used in internal methods
options(RedPort = RedPort())
```

RedPort-class *RedPort: An S4 class for RedeR graphics*

Description

RedPort: An S4 class for RedeR graphics

Value

An S4 class object.

Slots

`title` A string naming the XML-RPC port.

`uri` The uri to the XML-RPC server.

`host` Domain name of the machine that is running the XML-RPC server.

`port` An integer specifying the port on which the XML-RPC server should listen.

Constructor

`RedPort`

Author(s)

Sysbiolab.

See Also

`addGraph`, `getGraph`, `relax`, `calld`, @seealso `resetd`, `exitd`, `ping`, `version`.

---

relax,RedPort-method *Relax*

Description

RedeR's hierarchical force-directed interactive layout.

Usage

```
## S4 method for signature 'RedPort'
relax(
  obj,
  p1 = 100,
  p2 = 100,
  p3 = 100,
  p4 = 100,
  p5 = 100,
  p6 = 10,
  p7 = 10,
  p8 = 100,
  p9 = 10
)
```

Arguments

obj	A RedPort-class object.
p1	Edge target length (unit in points; >= 1) <numeric>.
p2	Edge stiffness (arbitrary unit; >= 0) <numeric>.
p3	Node repulsion factor (arbitrary unit; >= 0) <numeric>.
p4	Node perimeter effect (unit in points; >= 0) <numeric>.
p5	Node speed limit (arbitrary unit; >= 0) <numeric>.
p6	Repulsion radius, i.e., this parameter limits the repulsion factor range (unit as in 'p1'; >= 0) <numeric>.
p7	Central pull (arbitrary unit; >= 0) <numeric>.
p8	Nest-nest edge target length, i.e., edge target between linked containers (unit in points; >= 1) <numeric>.
p9	Nest-node repulsion factor, i.e., repulsion among containers and out-nodes (arbitrary unit; >= 0) <numeric>.

Value

Layout a graph in the app panel.

Author(s)

Sysbiolab.

See Also

addGraph

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

g <- graph.lattice(c(5, 5, 5))

rdp <- RedPort()


calld(rdp)
addGraph(rdp, g, layout.random(g))
relax(rdp)
```

`relaxRedeR` *Relax*

Description

RedeR's hierarchical force-directed interactive layout.

Usage

```
relaxRedeR(
  p1 = 100,
  p2 = 100,
  p3 = 100,
  p4 = 100,
  p5 = 100,
  p6 = 10,
  p7 = 10,
  p8 = 100,
  p9 = 10
)
```

Arguments

p1	Edge target length (unit in points; >= 1) <numeric>.
p2	Edge stiffness (arbitrary unit; >= 0) <numeric>.
p3	Node repulsion factor (arbitrary unit; >= 0) <numeric>.
p4	Node perimeter effect (unit in points; >= 0) <numeric>.
p5	Node speed limit (arbitrary unit; >= 0) <numeric>.
p6	Repulsion radius, i.e., this parameter limits the repulsion factor range (unit as in 'p1'; >= 0) <numeric>.

p7	Central pull (arbitrary unit; >= 0) <numeric>.
p8	Nest-nest edge target length, i.e., edge target between linked containers (unit in points; >= 1) <numeric>.
p9	Nest-node repulsion factor, i.e., repulsion among containers and out-nodes (arbitrary unit; >= 0) <numeric>.

Details

RedeR's interactive layout uses a force-directed algorithm described elsewhere (Brandes 2001; Fruchterman and Reingold 1991). Here we adapted the method to deal with nested networks. In force-directed graphs, each edge can be regarded as a spring - with a given target length - and can either exert a repulsive or attractive force on the connected nodes, while nodes are analogous to mutually repulsive charged particles that move according to the applied forces. In RedeR, the simulation is additionally constrained by the hierarchical structure of the network. For example, a nested node is constrained to its parent-node by opposing forces applied by the nest, which is regarded as a special node whose nested objects can reach a local equilibrium independently from other network levels. The simulation is adjusted by global options and evolves until the system reaches the equilibrium state. The default values are set to layout sparse networks with few nodes (e.g. 10-100 nodes). For large and dense networks better results can be achieved interactively by tuning one or more parameters.

Value

Layout a graph in the app panel.

Author(s)

Sysbiolab.

References

Brandes U. Drawing graphs: methods and models. In: Lecture notes in computer science. Kaufmann M. and Wagner D. (Ed), vol. 2025. Heidelberg: Springer; 2001: 71-86.

Fruchterman TMJ, Reingold EM. Graph drawing by force-directed placement. Software: Practice and Experience 1991, 21(11):1129-1164.

See Also

startRedeR, addGraphToRedeR

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create an igraph
gtoy <- graph.lattice(c(5, 5, 5))
```

```
# Start the RedeR interface
startRedeR()

# Send the igraph to RedeR
addGraphToRedeR(g = gtoy)

# Start interactive layout
relaxRedeR()
```

`resetd,RedPort-method` *Reset RedeR app*

Description

Reset an active RedeR session.

Usage

```
## S4 method for signature 'RedPort'
resetd(obj)
```

Arguments

`obj`	A `RedPort`-class object.

Value

Reset plotting panel.

Author(s)

Sysbiolab.

See Also

`resetRedeR`

Examples

```
rdp <- RedPort()


calld(rdp)
resetd(rdp)
```

resetRedeR *Reset RedeR app*

Description

Reset an active RedeR session.

Usage

```
resetRedeR()
```

Value

Reset plotting panel.

Author(s)

Sysbiolab.

See Also

startRedeR, addGraphToRedeR

Examples

```
# Load RedeR
library(RedeR)


# Call 'start' and 'reset' methods
startRedeR()
resetRedeR()
```

selectEdges,character-method *selectEdges*

Description

Select edges in an active RedeR application.

Usage

```
## S4 method for signature 'character'
selectEdges(edges, ...)

## S4 method for signature 'data.frame'
selectEdges(edges, ...)
```

Arguments

edges	A vertex sequence <vector of strings> or data frame of ncol=2.
...	Arguments passed to internal checks (ignore).

Value

Mark edges – which can be handled by other methods.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR, getGraphFromRedeR.

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create some edges as a data.frame
edges <- data.frame(
    A = c("n1", "n1", "n1", "n1"),
    B = c("n2", "n3", "n4", "n5")
)


# Start the RedeR interface
startRedeR()

# Add and select edges
addEdges(edges)
selectEdges(c("n1", "n3"))
```

selectNodes,character-method *selectNodes*

Description

Select nodes in an active RedeR application.

Usage

```
## S4 method for signature 'character'
selectNodes(nodes, anchor = FALSE, nid = NULL, ...)
```

Arguments

nodes	A string or array of strings with node names.
anchor	A logical value, whether to anchor nodes, which will prevent the relax function from applying the relaxing algorithm on the selected nodes.
nid	A nest ID. This will restrict searching to a specific container.
...	Arguments passed to internal checks (ignore).

Value

Mark nodes – which can be handled by other methods.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR, getGraphFromRedeR.

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create some edges as a data.frame
edges <- data.frame(
    A = c("n1", "n1", "n1", "n1"),
    B = c("n2", "n3", "n4", "n5")
)


# Start the RedeR interface
startRedeR()
```

```
# Add edges and select nodes
addEdges(edges)
selectNodes(c("n1", "n3"))
```

`startRedeR` *Start RedeR app from R*

Description

Method to launch RedeR application from R.

Usage

```
startRedeR(...)
```

Arguments

...	Arguments passed to the `RedPort` function.

Details

The `startRedeR()` is a wrapper function that launches the RedeR app by calling `RedPort()` and `calld()` methods. Therefore, these methods no longer needed to be called by the user from RedeR version >=3. List of functions that uses `startRedeR()`:

addGraphToRedeR Methods to display igraph objects.

getGraphFromRedeR Methods to wrap up RedeR graphs.

addLegendToRedeR Methods to display legends.

relaxRedeR Start RedeR's interactive layout.

resetRedeR Reset an active RedeR session.

exitRedeR Close an active RedeR session.

pingRedeR Test the R-to-Java interface.

addNodes Add nodes to an active RedeR application.

addEdges Add edges to an active RedeR application.

selectNodes Select nodes in an active RedeR application.

selectEdges Select edges in an active RedeR application.

deleteNodes Delete nodes from an active RedeR application.

deleteEdges Delete edges from an active RedeR application.

nestNodes Nest nodes into containers.

mergeOutEdges Assign 'out-edges' to containers.

Value

System call to start the RedeR application.

Author(s)

Sysbiolab.

See Also

addGraphToRedeR

Examples

```
# Load RedeR
library(RedeR)


# Start the RedeR interface
startRedeR()
```

---

subg *Subgraph of a graph*

Description

Creates a subgraph containing nodes specified from a data frame.

Usage

```
subg(g, dat, refcol = 1, maincomp = TRUE, connected = TRUE, transdat = TRUE)
```

Arguments

g	An 'igraph' object.
dat	A data frame with node names and attributes to be mapped to 'g'.
refcol	The reference column (node names) in the 'dat' object.
maincomp	Logical value, whether to return only the main component of the subgraph.
connected	Logical value, whether to return only connected nodes.
transdat	Logical value, whether to transfer node attributes from the 'dat' object to the subgraph.

Value

An igraph object.

See Also

subgraph

Examples

```
# see 'nested subgraphs' section in RedeR's vignette:
# vignette("RedeR")
```

`transform.attributes` *Transforming edge and vertex attributes*

Description

Given an 'igraph' object, 'att.addv' adds a new attribute with a fixed 'value' to all nodes or selected nodes, while 'att.adde' adds a new attribute with a fixed 'value' to all edges.

The 'att.mapv' and 'att.mape' functions map data frames to an 'igraph' object.

The 'att.setv' and 'att.sete' functions rename attributes available in the 'igraph' object, transforming them into new attribute classes (for example, numeric values into colors or sizes).

Usage

```
att.addv(g, to, value, index = V(g), filter = NULL)

att.adde(g, to, value, index = E(g))

att.setv(
  g,
  from = "name",
  to = "nodeColor",
  pal = 1,
  cols = NULL,
  na.col = "grey70",
  xlim = c(20, 100, 1),
  breaks = NULL,
  nquant = NULL,
  digits = 1,
  title = from,
  isrev = FALSE
)

att.sete(
  g,
  from = "name",
  to = "edgeColor",
  pal = 1,
```

```
  cols = NULL,
  na.col = "grey70",
  xlim = c(20, 100, 1),
  breaks = NULL,
  nquant = NULL,
  title = from,
  digits = 1,
  isrev = FALSE
)

att.mapv(g, dat, refcol = 1)
```

Arguments

g	An 'igraph' object.
to	A valid RedeR attribute name (see addGraph or type 'att.setv()' and 'att.sete()' for a quick list).
value	A single value for an edge or vertex attribute.
index	An optional index to set an attribute to a subset of vertices or edges.
filter	A named list of length = 1, used to filter which nodes will receive the attribute. The attribute 'to' will be added to nodes which have the attribute.
from	An attribute name available in 'g'.
pal	Color palette option (1 or 2); 'pal=1' will use a single color palette, while 'pal=2' will split 'breaks' at the center, generating two color palettes. The 'pal=2' option may be useful to build separated color palettes, for example, negative and positive values.
cols	Vector of colors (either hexadecimals or valid color names).
na.col	A single color for NAs.
xlim	A numeric vector with three boundaries: c(<lower>, <upper>, <NA>). It corresponds to boundary values to be apply to numeric attributes (e.g. nodeSize). Default: c(20, 100, 1).
breaks	A numeric vector of two or more breakpoints to be applied to the attribute values.
nquant	Number of breakpoints to split attribute values by quantiles.
digits	Integer indicating the number of decimal places in the legend of numerical attributes.
title	A legend title.
isrev	A logical value, whether to verse attribute values.
dat	A data frame with the attributes to be mapped to 'g'.
refcol	A reference column in the 'dat' object used to map 'dat' to 'g'. For 'att.mapv', 'refcol' is a single integer value indicating a column with node ids. For 'att.mape', 'refcol' is a vector with two integers indicating columns with edge ids. Also, for 'att.mapv', when 'refcol = 0' rownames will be used to map 'dat' to 'g'.

Value

Add, map, and set igraph attributes to the RedeR application.

See Also

addGraphToRedeR, getGraphFromRedeR

Examples

```
library(igraph)

# Generate a 'toy' graph with vertex names
gtoy <- sample_pa(10, directed=FALSE)
V(gtoy)$name <- paste0("V",1:vcount(gtoy))

# Create data frame with IDs compatible to vertex names
df <- data.frame(ID=sample(V(gtoy)$name))

# Add two variables to 'df' for demonstration
df$var_mumbers <- rnorm(nrow(df))
df$var_letters <- letters[1:nrow(df)]

### Using the 'att.set' functions to transform edge and vertex attributes

# Map 'df' to vertex attributes
gtoy <- att.mapv(g = gtoy, dat = df, refcol = 1)

# Set a new vertex attribute, creating 'nodeLabel' from 'var_letters'
gtoy <- att.setv(gtoy, from = "var_letters", to = "nodeLabel")

# Set a new vertex attribute, creating 'nodeColor' from 'var_mumbers'
gtoy <- att.setv(gtoy,
    from = "var_mumbers", to = "nodeColor",
    breaks = seq(-1, 1, 0.2), pal = 2
)

# Set a new vertex attribute, creating 'nodeSize' from 'var_mumbers'
gtoy <- att.setv(gtoy,
    from = "var_mumbers", to = "nodeSize", nquant = 10,
    isrev = TRUE, xlim = c(5, 40, 1)
)

### Using the 'att.add' functions to add fixed values

# Add a new vertex attribute, creating 'nodeFontSize' from a fixed value
gtoy <- att.addv(gtoy, to = "nodeFontSize", value = 10)

# ...as above, but applied only to three nodes
gtoy <- att.addv(gtoy,
    to = "nodeFontSize", value = 100,
    filter = list("name" = V(gtoy)$name[1:3])
)
```

`updateLayoutFromRedeR` *Update graph layout*

Description

This function updates node coordinates of an igraph object with the node coordinates from the RedeR interface.

Usage

```
updateLayoutFromRedeR(g, delNodes = FALSE, delEdges = FALSE)
```

Arguments

`g`	An igraph object, which will be updated with the graph layout displayed in the RedeR interface. Note: 'g' must be the same igraph object sent to the RedeR interface by the`addGraphToRedeR` function.
`delNodes`	Option to delete nodes from 'g' when these nodes are not displayed in the RedeR interface.
`delEdges`	Option to delete edges from 'g' when these edges are not displayed in the RedeR interface.

Value

An updated igraph object.

Author(s)

Sysbiolab.

See Also

`startRedeR`, `addGraphToRedeR`

Examples

```
# Load RedeR and igraph
library(RedeR)
library(igraph)

# Create an igraph
gtoy1 <- graph.lattice(c(3, 3, 3))


# Start the RedeR interface
startRedeR()
```

```
# Send graph to RedeR
addGraphToRedeR(g = gtoy1)

# Update 'gtoy1' with changes introduced in the RedeR interface
gtoy2 <- updateLayoutFromRedeR(g = gtoy1)
```

`version,RedPort-method` *Version*

Description

Returns the RedeR application version.

Usage

```
## S4 method for signature 'RedPort'
version(obj)
```

Arguments

`obj`	A `RedPort`-class object.

Value

Version of the running app.

Author(s)

Sysbiolab.

See Also

`pingRedeR`

Examples

```
rdp <- RedPort()


calld(rdp)
version(rdp)
```

Index

∗ **ER**
RedeR-data2, 22
∗ **hs.inter**
RedeR-data1, 21

addEdges, *33*
addEdges *(*addEdges,character-method*)*, 3
addEdges,character-method, 3
addEdges,data.frame-method *(*addEdges,character-method*)*, 3
addGraph, *25, 26*, *36*
addGraph *(*addGraph,RedPort-method*)*, 4
addGraph,RedPort-method, 4
addGraphToRedeR, *3*, 5, *5*, *7*, *9*, *11, 12*, *16, 17*, *19*, *28*, *30–34*, *37, 38*
addLegendToRedeR, 7, *33*
addNodes, *33*
addNodes *(*addNodes,character-method*)*, 8
addNodes,character-method, 8
att.adde *(*transform.attributes*)*, 35
att.addv *(*transform.attributes*)*, 35
att.mapv *(*transform.attributes*)*, 35
att.sete *(*transform.attributes*)*, 35
att.setv *(*transform.attributes*)*, 35

calld, *25*
calld *(*calld,RedPort-method*)*, 9
calld,RedPort-method, 9

deleteEdges, *33*
deleteEdges *(*deleteEdges,character-method*)*, 10
deleteEdges,character-method, 10
deleteEdges,data.frame-method *(*deleteEdges,character-method*)*, 10
deleteNodes, *33*
deleteNodes *(*deleteNodes,character-method*)*, 11
deleteNodes,character-method, 11

ER.limma *(*RedeR-data2*)*, 22
exitd, *25*
exitd *(*exitd,RedPort-method*)*, 12
exitd,RedPort-method, 12
exitRedeR, 13, *13*, *33*

getGraph, *25*
getGraph *(*getGraph,RedPort-method*)*, 14
getGraph,RedPort-method, 14
getGraphFromRedeR, *3*, *6*, *9*, *11, 12*, *14*, 15, *17*, *19*, *31–33*, *37*

hs.inter *(*RedeR-data1*)*, 21

mergeOutEdges, *33*
mergeOutEdges *(*mergeOutEdges,numeric_Or_missing-method*)*, 16
mergeOutEdges,numeric_Or_missing-method, 16

nestNodes, *4*, *6*, *33*
nestNodes *(*nestNodes,character-method*)*, 17
nestNodes,character-method, 17

ping, *25*
ping *(*ping,RedPort-method*)*, 20
ping,RedPort-method, 20
pingRedeR, *20*, 21, *33*, *39*

RedeR-data1, 21
RedeR-data2, 22
rederInfo, 23
RedPort, 24, *25*, *33*
RedPort-class, 25
relax, *25*, *32*
relax *(*relax,RedPort-method*)*, 25
relax,RedPort-method, 25

relaxRedeR, 27, *33*
resetd, *25*
resetd *(*resetd,RedPort-method*)*, 29
resetd,RedPort-method, 29
resetRedeR, *29*, 30, *33*

selectEdges, *33*
selectEdges
*(*selectEdges,character-method*)*, 30
selectEdges,character-method, 30
selectEdges,data.frame-method
*(*selectEdges,character-method*)*, 30
selectNodes, *33*
selectNodes
*(*selectNodes,character-method*)*, 32
selectNodes,character-method, 32
startRedeR, *6, 7*, *10*, *13*, *16*, *21*, *24*, *28*, *30*, 33, *38*
subg, 34
subgraph, *35*

transform.attributes, 35

updateLayoutFromRedeR, 38

version, *25*
version *(*version,RedPort-method*)*, 39
version,RedPort-method, 39