

Package ‘PathwaySpace’

July 28, 2025

Type Package

Title Spatial Projection of Network Signals along Geodesic Paths

Version 1.0.2

Depends R(>= 4.4), methods, RGraphSpace(>= 1.0.9)

Imports grDevices, stats, scales, RANN, igraph, ggplot2, ggrepel,
lifecycle

Suggests knitr, rmarkdown, testthat, RedeR

Description For a given graph containing vertices, edges, and a signal associated with the vertices, the 'PathwaySpace' package performs a convolution operation, which involves a weighted combination of neighboring vertices and their associated signals. The package then uses a decay function to project these signals, creating geodesic paths on a 2D-image space. 'PathwaySpace' could have various applications, such as visualizing and analyzing network data in a graphical format that highlights the relationships and signal strengths between vertices. It can be particularly useful for understanding the influence of signals through complex networks. By combining graph theory, signal processing, and visualization, the 'PathwaySpace' package provides a novel way of representing and analyzing graph data.

License Artistic-2.0

VignetteBuilder knitr

URL <https://github.com/sysbiolab/PathwaySpace>

BugReports <https://github.com/sysbiolab/PathwaySpace/issues>

Collate pspaceChecks.R pspaceSupplements.R pspaceMisc.R
pspaceWatershed.R pspaceDecay.R pspaceClasses.R
pspaceGenerics.R pspaceMethods.R pspacePlots.R

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Victor Apolonio [ctr],
Vinicius Chagas [ctr],
Mauro Castro [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-4942-8131>>)

Maintainer Mauro Castro <mauro.a.castro@gmail.com>
Repository CRAN
Date/Publication 2025-07-28 02:10:02 UTC

Contents

buildPathwaySpace	2
CGC_20211118	3
circularProjection,PathwaySpace-method	4
expDecay	6
getPathwaySpace,PathwaySpace-method	7
gimage	7
gs_vertex_attr<-,PathwaySpace-method	8
Hallmarks_v2023_1_Hs_symbols	9
linearDecay	10
pathDistances	11
PathwaySpace-class	12
PCv12_pruned_igraph	12
plotPathDistances	13
plotPathwaySpace,PathwaySpace-method	14
polarProjection,PathwaySpace-method	17
pspace.cols	18
signalAggregation	19
signalDecay	20
silhouetteMapping,PathwaySpace-method	21
summitMapping,PathwaySpace-method	22
summitWatershed	23
vertexSignal,PathwaySpace-method	24
weibullDecay	26

Index	27
--------------	-----------

buildPathwaySpace	<i>Constructor of PathwaySpace-class Objects</i>
-------------------	--

Description

buildPathwaySpace is a constructor of PathwaySpace-class objects.

Usage

```
buildPathwaySpace(  
  gs,  
  nrc = 500,  
  verbose = TRUE,  
  g = deprecated(),  
  mar = deprecated()  
)
```

Arguments

gs	A GraphSpace object. Alternatively, an igraph object with node coordinates assigned to x and y vertex attributes, and node labels assigned to name vertex attribute.
nrc	A single positive integer indicating the number of rows and columns (in pixels) for a square image matrix. This argument will affect the resulting image size and resolution.
verbose	A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
g	Deprecated from PathwaySpace 1.0.1; use 'gs' instead.
mar	Deprecated from PathwaySpace 1.0.1; use 'mar' in GraphSpace instead.

Value

A pre-processed [PathwaySpace](#) class object.

Author(s)

Victor Apolonio, Vinicius Chagas, Mauro Castro, and TCGA Network.

See Also

[undirected_graph](#)

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

# Check graph validity
gs <- GraphSpace(gtoy1, mar = 0.1)

# Create a new PathwaySpace object
ps <- buildPathwaySpace(gs, nrc = 100)
# note: adjust 'nrc' to increase image resolution
```

Description

A data frame listing 'GeneSymbol' and 'Entrez' IDs from the COSMIC-CGC database (Sondka et al., 2020). These genes are used to demonstrate the PathwaySpace's summit mapping pipeline, which assigns summits to an image space.

Usage

```
data(CGC_20211118)
```

Format

```
data.frame
```

Value

A data.frame object.

Source

COSMIC-CGC database (release v95, tier 1 collection).

References

Sondka et al. The COSMIC Cancer Gene Census: describing genetic dysfunction across all human cancers. Nat Rev Cancer 18, 696-705, 2018. Doi: 10.1038/s41568-018-0060-1.

Examples

```
data(CGC_20211118)
```

circularProjection, PathwaySpace-method

Spatial Projection of Graph-Associated Signals

Description

circularProjection implements a convolution algorithm to project signals onto a 2D-coordinate system.

Usage

```
## S4 method for signature 'PathwaySpace'
circularProjection(
  ps,
  k = 8,
  pdist = 0.15,
  rescale = TRUE,
  verbose = TRUE,
  decay.fun = signalDecay(),
  aggregate.fun = signalAggregation(),
  kns = deprecated()
)
```

Arguments

ps	A PathwaySpace class object.
k	A single positive integer determining the k-top signals in the signal convolution operation.
pdist	A term (in $[0, 1]$) determining a distance unit for the signal decay function. When 'pdist = 1' it will represent the diameter of the inscribed circle within the pathway space. This distance will affect the extent over which the convolution operation projects the signal between source- and destination points.
rescale	A single logical value indicating whether to rescale the signal. If the signal ≥ 0 , then it will be rescaled to $[0, 1]$; if the signal ≤ 0 , then it will be rescaled to $[-1, 0]$; and if the signal in $(-\text{Inf}, +\text{Inf})$, then it will be rescaled to $[-1, 1]$.
verbose	A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
decay.fun	A signal decay function. Available options include 'Weibull', 'exponential', and 'linear' (see signalDecay). Users may also define a custom decay model with at least two arguments, e.g., <code>function(x, signal) { ... }</code> , which should return a vector of projected signals of the same length as x. Additional arguments may include any variable available as graph vertex attribute.
aggregate.fun	A function used to aggregate the projected signals. It must be provided as a unary function, e.g., <code>function(x) { ... }</code> , which should aggregate a vector of signals to a single scalar value. Available options include 'mean', 'wmean', 'log.wmean', and 'exp.wmean' (See signalAggregation).
kns	Deprecated from PathwaySpace 1.0.1; use 'k' instead.

Value

A preprocessed [PathwaySpace](#) class object.

Author(s)

Victor Apolonio, Vinicius Chagas, Mauro Castro, and TCGA Network.

See Also

[buildPathwaySpace](#)

Examples

```
# Load a demo igraph
data('gttoy1', package = 'RGraphSpace')

# Create a new PathwaySpace object
ps <- buildPathwaySpace(gttoy1, nrc = 100)
# note: adjust 'nrc' to increase image resolution

# Set '1s' as vertex signal
vertexSignal(ps) <- 1
```

```
# Create a 2D-landscape image
ps <- circularProjection(ps)
```

expDecay

Exponential decay function

Description

The expDecay function is used by PathwaySpace's methods for signal convolution and projection.

Usage

```
expDecay(x, signal = 1, decay = 0.001)
```

Arguments

x	A numeric vector of distances (in [0,1]).
signal	A single numeric value representing a signal.
decay	A decay factor (in [0,1]). This term indicates how much the signal decreases as a function of distance in pathway space. For example, at a specific distance defined by the pdist parameter (see circularProjection), the signal intensity will be the initial signal multiplied by decay. the weibullDecay function.

Value

A numeric vector; if missing 'x', it will return decay function.

Author(s)

Mauro Castro.

See Also

[weibullDecay](#), [linearDecay](#)

Examples

```
x <- seq(0, 2, 0.01)
y <- expDecay(x, signal = 1, decay = 0.5)
plot(x, y)
```

getPathwaySpace, PathwaySpace-method
<i>Accessors for Fetching Slots from a PathwaySpace Object</i>

Description

getPathwaySpace retrieves information from individual slots available in a PathwaySpace object.

Usage

```
## S4 method for signature 'PathwaySpace'
getPathwaySpace(ps, what = "status")
```

Arguments

ps	A preprocessed PathwaySpace class object
what	A single character value specifying which information should be retrieved from the slots. Options: "nodes", "edges", "graph", "image", "pars", "misc", "signal", "projections", "status", "silhouette", "summits", "summit_mask", "summit_contour"

Value

Content from slots in the [PathwaySpace](#) object.

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

# Create a new PathwaySpace object
ps <- buildPathwaySpace(gtoy1, nrc = 100)
# note: adjust 'nrc' to increase image resolution

# Get the 'status' slot in ps
status <- getPathwaySpace(ps, what = 'status')
```

gimage	<i>An image matrix</i>
--------	------------------------

Description

An image matrix used for workflow demonstrations.

Usage

```
data(gimage)
```

Format

matrix

Value

An image matrix.

Source

This package.

Examples

```
data(gimage)
```

```
gs_vertex_attr<-,PathwaySpace-method
```

Accessor Functions for PathwaySpace Objects

Description

Get or set edge and vertex attributes in [PathwaySpace](#) class object.

Usage

```
## S4 replacement method for signature 'PathwaySpace'  
gs_vertex_attr(x, name, ...) <- value
```

```
## S4 replacement method for signature 'PathwaySpace'  
gs_edge_attr(x, name, ...) <- value
```

Arguments

x	A PathwaySpace class object.
name	Name of the attribute.
...	Additional arguments passed to igraph methods.
value	The new value of the attribute.

Value

Updated [PathwaySpace](#) object.

Examples

```
data('gtoy1', package = 'RGraphSpace')
ps <- buildPathwaySpace(gtoy1, nrc = 100)

# Get vertex count
gs_vcount(ps)

# Get edge count
gs_ecount(ps)

# Access a specific vertex attribute
gs_vertex_attr(ps, "signal")

# Replace an entire vertex attribute
gs_vertex_attr(ps, "signal") <- 1

# Modify a single value within a vertex attribute
gs_vertex_attr(ps, "signal")["n1"] <- 1

# Access a specific edge attribute
gs_edge_attr(ps, "weight")

# Replace an entire edge attribute
gs_edge_attr(ps, "weight") <- 1
```

Hallmarks_v2023_1_Hs_symbols

A list with Hallmark gene sets (v2023.1)

Description

A list with Human gene symbols from the MSigDB's Hallmark gene set collection (Liberzon et al., 2015). These gene sets are used to demonstrate the PathwaySpace's summit mapping pipeline, which assigns summits to an image space.

Usage

```
data(Hallmarks_v2023_1_Hs_symbols)
```

Format

list

Value

A list object.

Source

MSigDB database (v2023.1).

References

Liberzon et al. The Molecular Signatures Database (MSigDB) hallmark gene set collection. Cell Systems 1(5):417-425, 2015 Doi: 10.1016/j.cels.2015.12.004

Examples

```
data(Hallmarks_v2023_1_Hs_symbols)
```

linearDecay	<i>A simple linear decay function</i>
-------------	---------------------------------------

Description

The linearDecay function is used by PathwaySpace's methods for signal convolution and projection.

Usage

```
linearDecay(x, signal = 1)
```

Arguments

x	A numeric vector of distances (in [0,1]).
signal	A single numeric value representing a signal. weibullDecay and expDecay functions.

Value

A numeric vector; if missing 'x', it will return decay function.

Author(s)

Mauro Castro.

See Also

[weibullDecay](#), [expDecay](#)

Examples

```
x <- seq(0, 2, 0.01)
y <- linearDecay(x, signal = 1)
plot(x, y)
```

pathDistances	<i>Calculate a pathway space distance between two vectors</i>
---------------	---

Description

Calculate a pathway space distance between two vectors

Usage

```
pathDistances(gdist, from, to, nperm = 1000, verbose = TRUE)
```

Arguments

gdist	A distance matrix computed by the igraph's distances function. Rows and columns must be named with vertex labels as listed in the 'igraph' object.
from	A vector with valid vertex names.
to	A vector with valid vertex names.
nperm	Number of permutations.
verbose	A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).

Value

A list with pathway space distances and a 'ggplot' object.

See Also

[plotPathwaySpace](#)

Examples

```
# Load a vertex-wise distance matrix (distance between nodes in a graph)
data("gdist.toy", package = "PathwaySpace")

# Get two vertex lists
from <- sample(colnames(gdist.toy), 50)
to <- sample(colnames(gdist.toy), 50)

# Calculate distances between lists, and between random lists
res <- pathDistances(gdist.toy, from, to)
names(res)
# "p_dist" "z_score"
```

PathwaySpace-class	<i>PathwaySpace: An S4 class for signal propagation on image spaces</i>
--------------------	---

Description

PathwaySpace: An S4 class for signal propagation on image spaces

Value

An S4 class object.

Slots

nodes A data frame with xy-vertex coordinates.

edges A data frame with edges.

graph An igraph object.

image A raster background image matrix.

pars A list with parameters.

misc A list with intermediate objects for downstream methods.

projections A list with processed objects for downstream methods.

status A vector containing the processing status of the PathwaySpace object.

Constructor

see [buildPathwaySpace](#) constructor.

Author(s)

Mauro Castro, <mauro.castro@ufpr.br>

PCv12_pruned_igraph	<i>A pruned and laid out igraph object from Pathway Commons V12</i>
---------------------	---

Description

This igraph object was created from a 'sif' file available from the Pathway Commons V12 (Rodchenkov et al., 2020), which was filtered to keep interactions from the following sources: CTD, Recon, HumanCyc, DrugBank, MSigDB, DIP, BioGRID, IntAct, BIND, and PhosphoSite. The igraph was additionally pruned and laid out by a force-directed algorithm aiming signal projection on PathwaySpace's images. Edges with the smallest betweenness centrality were pruned using 'backward elimination' and 'forward selection' strategies. The resulting graph represents the main connected component with the minimum number of edges.

Usage

```
data(PCv12_pruned_igraph)
```

Format

igraph

Value

An igraph object.

Author(s)

Chris Wong, Mauro Castro, and TCGA Network.

Source

Pathway Commons V12.

References

Rodchenkov et al. Pathway Commons 2019 Update: integration, analysis and exploration of pathway data. Nucleic Acids Research 48(D1):D489–D497, 2020. doi:[10.1093/nar/gkz946](https://doi.org/10.1093/nar/gkz946)

Examples

```
data(PCv12_pruned_igraph)
## Suggestion to visualize this igraph in R:
library(RGraphSpace)
plotGraphSpace(PCv12_pruned_igraph)
```

plotPathDistances	<i>Accessory function to plot pathway space distances</i>
-------------------	---

Description

Accessory function to plot pathway space distances

Usage

```
plotPathDistances(pdist, z.transform = FALSE)
```

Arguments

pdist	A list generated by the pathDistances function.
z.transform	A single logical value specifying to convert pathway distances into z-score values.

Value

A 'ggplot' object.

Examples

```
# Load a vertex-wise distance matrix (distance between nodes in a graph)
data("gdist.toy", package = "PathwaySpace")

# Get two gene lists
from <- sample(colnames(gdist.toy), 50)
to <- sample(colnames(gdist.toy), 50)

# Calculate distances between lists, and between random lists
res <- pathDistances(gdist.toy, from, to)

# Plot observed and null distances
plotPathDistances(res)
```

plotPathwaySpace, PathwaySpace-method

Plotting 2D-landscape images for the PathwaySpace package

Description

plotPathwaySpace is a wrapper function to create dedicated ggplot graphics for PathwaySpace-class objects.

Usage

```
## S4 method for signature 'PathwaySpace'
plotPathwaySpace(
  ps,
  colors = pspace.cols(),
  trim.colors = c(3, 2, 1, 2, 3),
  bg.color = "grey95",
  si.color = "grey85",
  si.alpha = 1,
  theme = c("th0", "th1", "th2", "th3"),
  title = "PathwaySpace",
  xlab = "Pathway coordinates 1",
  ylab = "Pathway coordinates 2",
  zlab = "Density",
  font.size = 1,
  font.color = "white",
  zlim = NULL,
  slices = 25,
  add.grid = TRUE,
```

```

    grid.color = "white",
    add.summits = TRUE,
    label.summits = TRUE,
    summit.color = "white",
    add.marks = FALSE,
    marks = NULL,
    mark.size = 3,
    mark.color = "white",
    mark.padding = 0.5,
    mark.line.width = 0.5,
    use.dotmark = FALSE,
    add.image = FALSE
  )

```

Arguments

ps	A PathwaySpace class object.
colors	A vector of colors. Each color is a specific hue used to create a customized color palette, interpolated according to the provided sequence in the vector of colors. The proportion of each color hue can be adjusted by the 'trim.colors' argument. This palette is designed to fine-tune the visibility of summits and valleys within the image space. To bypass this automatic palette generation and use the 'colors' input as-is, simply set 'trim.colors' to NULL.
trim.colors	An vector with 5 positive integer numbers. This argument can be used to adjust the proportion of each color hue in the palette.
bg.color	A single color for background.
si.color	A single color for silhouette. (see silhouetteMapping).
si.alpha	A transparency level in [0, 1], used to adjust the opacity of the silhouette. This parameter is useful for improving the perception of a background image, when one is available.
theme	Name of a custom PathwaySpace theme. These themes (from 'th0' to 'th3') consist mainly of preconfigured ggplot settings, which the user can subsequently refine using ggplot2 .
title	A string for the title.
xlab	The title for the 'x' axis of a 2D-image space.
ylab	The title for the 'y' axis of a 2D-image space.
zlab	The title for the 'z' axis of the image signal.
font.size	A single numeric value passed to ggplot themes.
font.color	A single color passed to ggplot themes.
zlim	The 'z' limits of the plot (a numeric vector with two numbers). If NULL, limits are determined from the range of the input values.
slices	A single positive integer value used to split the image signal into equally-spaced intervals.

<code>add.grid</code>	A logical value indicating whether to add gridlines to the image space. However, gridlines will only appear when the image is decorated with graph silhouettes (see silhouetteMapping).
<code>grid.color</code>	A color passed to geom_point .
<code>add.summits</code>	A logical value indicating whether to add contour lines to 'summits' (when summits are available; see summitMapping).
<code>label.summits</code>	A logical value indicating whether to label summits.
<code>summit.color</code>	A color passed to 'summits'.
<code>add.marks</code>	A logical value indicating whether to plot vertex labels.
<code>marks</code>	A vector of vertex names to be highlighted in the image space. This argument overrides 'add.labels'.
<code>mark.size</code>	A size argument passed to geom_text .
<code>mark.color</code>	A color passed to geom_text .
<code>mark.padding</code>	A box padding argument passed to geom_text_repel .
<code>mark.line.width</code>	A line width argument passed to geom_text_repel .
<code>use.dotmark</code>	A logical value indicating whether "marks" should be represented as dots.
<code>add.image</code>	A logical value indicating whether to add a background image, when one is available (see GraphSpace).

Value

A ggplot-class object.

Author(s)

Mauro Castro and TCGA Network.

See Also

[circularProjection](#)

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

# # Check graph validity
gs <- GraphSpace(gtoy1)

# Create a PathwaySpace object
ps <- buildPathwaySpace(gs, nrc = 300)
# note: adjust 'nrc' to increase image resolution

# Set '1s' as vertex signal
vertexSignal(ps) <- 1
```



```
# Create a 2D-landscape image
ps <- circularProjection(ps, k = 2, pdist = 0.4)

# Plot a 2D-landscape image
plotPathwaySpace(ps)
```

polarProjection,PathwaySpace-method

Spatial Projection of Graph-Associated Signals

Description

polarProjection implements a convolution algorithm to project signals across a 2D-coordinate system.

Usage

```
## S4 method for signature 'PathwaySpace'
polarProjection(
  ps,
  k = 8,
  pdist = 0.5,
  theta = 180,
  directional = FALSE,
  rescale = TRUE,
  verbose = TRUE,
  decay.fun = signalDecay(),
  aggregate.fun = signalAggregation(),
  kns = deprecated()
)
```

Arguments

ps	A PathwaySpace class object.
k	A single positive integer determining the k-top signals in the signal convolution operation.
pdist	A term (in $[0, 1]$) determining a distance unit for the signal decay function. When 'pdist = 1' it will represent the diameter of the inscribed circle within the pathway space. This distance will affect the extent over which the convolution operation projects the signal between source- and destination points.
theta	Angle of projection (degrees in $(0, 360]$).
directional	If directional edges are available, this argument can be used to orientate the signal projection on directed graphs.
rescale	A single logical value indicating whether to rescale the signal. If the signal ≥ 0 , then it will be rescaled to $[0, 1]$; if the signal ≤ 0 , then it will be rescaled to $[-1, 0]$; and if the signal in $(-\text{Inf}, +\text{Inf})$, then it will be rescaled to $[-1, 1]$.

verbose	A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
decay.fun	A signal decay function. Available options include 'Weibull', 'exponential', and 'linear' (see signalDecay). Users may also define a custom decay model with at least two arguments, e.g., <code>function(x, signal) { ... }</code> , which should return a vector of projected signals of the same length as x. Additional arguments may include any variable available as graph vertex attribute.
aggregate.fun	A function used to aggregate the projected signals. It must be provided as a unary function, e.g., <code>function(x) { ... }</code> , which should aggregate a vector of signals to a single scalar value. Available options include 'mean', 'wmean', 'log.wmean', and 'exp.wmean' (See signalAggregation).
kns	Deprecated from PathwaySpace 1.0.1; use 'k' instead.

Value

A preprocessed [PathwaySpace](#) class object.

Author(s)

Mauro Castro

See Also

[buildPathwaySpace](#)

Examples

```
# Load a demo igraph
data('gtoy2', package = 'RGraphSpace')

# Create a new PathwaySpace object
ps <- buildPathwaySpace(gtoy2, nrc = 100)
# note: adjust 'nrc' to increase image resolution

# Set '1s' as vertex signal
vertexSignal(ps) <- 1

# Create a 2D-landscape image
ps <- polarProjection(ps)
```

pspace.cols

A simple vector of colors for PathwaySpace images

Description

A simple vector of colors for PathwaySpace images

Usage

```
pspace.cols(n = 5)
```

Arguments

n Number of colors.

Value

A vector with hexadecimal color codes.

See Also

[plotPathwaySpace](#)

Examples

```
pspace.cols()
```

signalAggregation	<i>Signal aggregation functions</i>
-------------------	-------------------------------------

Description

Signal aggregation functions for [PathwaySpace](#) internal calls. The aggregation should be symmetric with respect to signal polarity, ensuring that opposite signals produce corresponding outputs.

Usage

```
signalAggregation(method = c("mean", "wmean", "log.wmean", "exp.wmean"))
```

Arguments

method A character string specifying the method for signal aggregation, returning either a customized [mean](#) or [weighted.mean](#) function.

Value

An aggregation function.

Author(s)

Mauro Castro.

See Also

[circularProjection](#), [polarProjection](#), [weighted.mean](#)

Examples

```
signalAggregation()
```

signalDecay

Signal decay functions

Description

Signal decay functions for [PathwaySpace](#) internal calls.

Usage

```
signalDecay(  
  method = c("weibull", "exp", "linear"),  
  decay = 0.001,  
  shape = 1.05  
)
```

Arguments

method	A character string specifying a method for signal decay (any of weibull, exp, or linear), returning weibullDecay , expDecay , or linearDecay functions, respectively.
decay	A decay factor (in (0,1]) passed to the weibullDecay and expDecay functions.
shape	A parameter (≥ 1) passed to the weibullDecay function.

Value

An the function.

Author(s)

Mauro Castro.

See Also

[circularProjection](#) and [polarProjection](#)

Examples

```
signalDecay()
```

`silhouetteMapping,PathwaySpace-method`*Decorating PathwaySpace Images with Graph Silhouettes*

Description

`silhouetteMapping` constructs an image baseline used to outline the graph layout in a `PathwaySpace` image.

Usage

```
## S4 method for signature 'PathwaySpace'
silhouetteMapping(
  ps,
  baseline = 0.01,
  pdist = 0.05,
  fill.cavity = TRUE,
  verbose = TRUE
)
```

Arguments

<code>ps</code>	A PathwaySpace class object.
<code>baseline</code>	A fraction (in $[0, 1]$) of the signal scale of a <code>PathwaySpace</code> image. This term only affects the image baseline projection, which represents a silhouette of the graph's layout outlined in the resulting image. When <code>baseline = 0</code> (i.e. lower level of the signal scale), the baseline will extend over the entire image space, so no silhouette will be visible.
<code>pdist</code>	A term (in $[0, 1]$) determining a distance unit for the signal decay function. This distance will affect the extent over which the convolution operation projects the image baseline.
<code>fill.cavity</code>	A single logical value specifying to fill cavities in the silhouette mask (when <code>verbose=TRUE</code>) or not (when <code>verbose=FALSE</code>).
<code>verbose</code>	A single logical value specifying to display detailed messages (when <code>verbose=TRUE</code>) or not (when <code>verbose=FALSE</code>).

Value

A preprocessed [PathwaySpace](#) class object.

Author(s)

Mauro Castro and TCGA Network.

See Also

[circularProjection](#)

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

# Create a new PathwaySpace object
ps <- buildPathwaySpace(gtoy1, nrc = 100)
# note: adjust 'nrc' to increase image resolution

# Set '1s' as vertex signal
vertexSignal(ps) <- 1

# Map graph silhouette
ps <- silhouetteMapping(ps, pdist = 0.1)
```

summitMapping, PathwaySpace-method

Mapping Summits on PathwaySpace Images

Description

The `summitMapping` method implements a segmentation strategy to identify summits on a 2D-landscape image (see [summitWatershed](#)).

Usage

```
## S4 method for signature 'PathwaySpace'
summitMapping(
  ps,
  maxset = 30,
  minsize = 30,
  threshold = 0.5,
  verbose = TRUE,
  segm_fun = summitWatershed,
  ...
)
```

Arguments

<code>ps</code>	A PathwaySpace class object.
<code>maxset</code>	A single positive integer indicating the maximum number of summits to be returned by the segmentation function.
<code>minsize</code>	A single positive integer indicating the minimum size of the summits.
<code>threshold</code>	A threshold provided as a fraction (in $[0, 1]$) of the max signal intensity.
<code>verbose</code>	A single logical value specifying to display detailed messages (when <code>verbose=TRUE</code>) or not (when <code>verbose=FALSE</code>).
<code>segm_fun</code>	A segmentation function used to detect summits (see summitWatershed).
<code>...</code>	Additional arguments passed to the segmentation function.

Value

A preprocessed [PathwaySpace](#) class object.

Author(s)

Mauro Castro and TCGA Network.

See Also

[circularProjection](#)

Examples

```
# Load a large igraph
data("PCv12_pruned_igraph", package = "PathwaySpace")

# Continue this example from the PathwaySpace vignette,
# in the 'PathwaySpace decoration' section
```

summitWatershed

Variation of the watershed algorithm for summit detection

Description

The `summitWatershed` function implements a segmentation strategy to identify summits within a landscape image generated by the `PathwaySpace` package. This function is entirely coded in R, which helps alleviating users from the task of loading an excessive number of dependencies. Nonetheless, while this novel implementation prevents the burden a 'dependency heaviness', it still requires optimization as it currently exhibits slower performance compared to well-established implementations such as the `watershed` function from the `EBImage` package. The `summitWatershed` maintain a certain level of compatibility with the `EBImage`'s `watershed` function, and both can be used in the `PathwaySpace` package.

Usage

```
summitWatershed(x, tolerance = 0.1, ext = 1)
```

Arguments

<code>x</code>	A 2D-numeric array in which each point represents the coordinates of a signal in a landscape image.
<code>tolerance</code>	The minimum signal intensity of a summit (in [0,1]), representing a fraction of the maximum signal intensity.
<code>ext</code>	Radius (in pixels) for detecting neighboring objects.

Value

A matrix with labeled summits.

Author(s)

Mauro Castro.

See Also

[summitMapping](#)

Examples

```
# Load a demo landscape image
data('gimage', package = 'PathwaySpace')

# Scale down the image for a quicker demonstration
gimage <- gimage[200:300, 200:300]

# Check signal range
range(gimage, na.rm = TRUE)
# [1] 0 1

# Check image

image(gimage)

# Threshold the signal intensity, for example:
gimage[gimage < 0.5] <- 0

# Run summit segmentation
gmask <- summitWatershed(x = gimage)

# Check resulting image mask

image(gimage)
```

vertexSignal,PathwaySpace-method

Accessor Functions for PathwaySpace Objects

Description

Get or set 'signal' and 'decay' functions in a [PathwaySpace](#) class object.

Usage

```
## S4 method for signature 'PathwaySpace'
vertexSignal(x)

## S4 replacement method for signature 'PathwaySpace'
vertexSignal(x) <- value

## S4 method for signature 'PathwaySpace'
vertexDecay(x)

## S4 replacement method for signature 'PathwaySpace'
vertexDecay(x) <- value
```

Arguments

x	A PathwaySpace class object.
value	The new value of the attribute.

Value

Updated [PathwaySpace](#) object.

Examples

```
data('gtoy1', package = 'RGraphSpace')
ps <- buildPathwaySpace(gtoy1, nrc = 100)

# Check vertex names
names(ps)

# Access signal values from all vertices
vertexSignal(ps)

# Modify signal value of a specific vertex
vertexSignal(ps)[1] <- 1

# Modify signal value of specific vertices
vertexSignal(ps)[c("n2", "n3")] <- 1

# Set '1s' to all vertices
vertexSignal(ps) <- 1

#----

# Access decay function of a specific vertex
vertexDecay(ps)[["n3"]]

# Modify decay function of a specific vertex
vertexDecay(ps)[["n3"]] <- signalDecay(method = "linear")
```

```
# Modify decay functions of two vertices
vertexDecay(ps)[c("n1","n3")] <- list( signalDecay() )

# Modify decay functions of all vertices
vertexDecay(ps) <- signalDecay(method = "weibull", shape = 2)
```

weibullDecay

*Weibull decay function***Description**

The weibullDecay function is used by PathwaySpace's methods for signal convolution and projection.

Usage

```
weibullDecay(x, signal = 1, decay = 0.001, shape = 1.05)
```

Arguments

x	A numeric vector of distances (in [0,1]).
signal	A single numeric value representing a signal.
decay	A decay factor (in [0,1]). This term indicates how much the signal decreases as a function of distance in pathway space. For example, at a specific distance defined by the pdist parameter (see circularProjection), the signal intensity will be the initial signal multiplied by decay.
shape	A parameter (≥ 1) of a Weibull function. When shape=1 the Weibull decay follows an exponential decay. When shape>1 the function is first convex, then concave with an inflection point.

Value

A numeric vector; if missing 'x', it will return decay function.

Author(s)

Mauro Castro.

See Also

[expDecay](#), [linearDecay](#)

Examples

```
x <- seq(0, 2, 0.01)
y <- weibullDecay(x, signal = 1, decay = 0.5, shape = 4)
plot(x, y)
```

Index

* **CGC_20211118**
CGC_20211118, [3](#)

* **Hallmarks_v2023_1_Hs_symbols**
Hallmarks_v2023_1_Hs_symbols, [9](#)

* **PCv12_pruned_igraph**
PCv12_pruned_igraph, [12](#)

* **gimage**
gimage, [7](#)

buildPathwaySpace, [2](#), [5](#), [12](#), [18](#)

CGC_20211118, [3](#)

circularProjection, [6](#), [16](#), [19–21](#), [23](#), [26](#)

circularProjection
(circularProjection, PathwaySpace-method),
[4](#)

circularProjection, PathwaySpace-method,
[4](#)

expDecay, [6](#), [10](#), [20](#), [26](#)

gdist.toy (PCv12_pruned_igraph), [12](#)

geom_point, [16](#)

geom_text, [16](#)

geom_text_repel, [16](#)

getPathwaySpace
(getPathwaySpace, PathwaySpace-method), [19](#), [20](#)
[7](#)

getPathwaySpace, PathwaySpace-method, [7](#)

ggplot2, [15](#)

gimage, [7](#)

GraphSpace, [3](#), [16](#)

gs_edge_attr<-
(gs_vertex_attr<- , PathwaySpace-method),
[8](#)

gs_edge_attr<- , PathwaySpace-method
(gs_vertex_attr<- , PathwaySpace-method),
[8](#)

gs_vertex_attr<- , PathwaySpace-method,
[8](#)

gs_vertex_attr<-
(gs_vertex_attr<- , PathwaySpace-method),
[8](#)

hallmarks
(Hallmarks_v2023_1_Hs_symbols),
[9](#)

Hallmarks_v2023_1_Hs_symbols, [9](#)

igraph, [3](#)

linearDecay, [6](#), [10](#), [20](#), [26](#)

mean, [19](#)

pathDistances, [11](#), [13](#)

PathwaySpace, [3](#), [5](#), [7](#), [8](#), [15](#), [17–25](#)

PathwaySpace-class, [12](#)

PCv12_pruned_igraph, [12](#)

plotPathDistances, [13](#)

plotPathwaySpace, [11](#), [19](#)

plotPathwaySpace
(plotPathwaySpace, PathwaySpace-method),
[14](#)

plotPathwaySpace, PathwaySpace-method,
[14](#)

polarProjection, [19](#), [20](#)

polarProjection
(polarProjection, PathwaySpace-method),
[17](#)

polarProjection, PathwaySpace-method,
[17](#)

pspace.cols, [18](#)

signalAggregation, [5](#), [18](#), [19](#)

signalDecay, [5](#), [18](#), [20](#)

silhouetteMapping, [15](#), [16](#)

silhouetteMapping
(silhouetteMapping, PathwaySpace-method),
[21](#)

silhouetteMapping, PathwaySpace-method,
21

summitMapping, 16, 24

summitMapping
(summitMapping, PathwaySpace-method),
22

summitMapping, PathwaySpace-method, 22

summitWatershed, 22, 23

undirected_graph, 3

vertexDecay
(vertexSignal, PathwaySpace-method),
24

vertexDecay, PathwaySpace-method
(vertexSignal, PathwaySpace-method),
24

vertexDecay<-
(vertexSignal, PathwaySpace-method),
24

vertexDecay<-, PathwaySpace-method
(vertexSignal, PathwaySpace-method),
24

vertexSignal
(vertexSignal, PathwaySpace-method),
24

vertexSignal, PathwaySpace-method, 24

vertexSignal<-
(vertexSignal, PathwaySpace-method),
24

vertexSignal<-, PathwaySpace-method
(vertexSignal, PathwaySpace-method),
24

weibullDecay, 6, 10, 20, 26

weighted.mean, 19