Package ‘PathwaySpace’

April 8, 2025

Type Package

Title Spatial Projection of Network Signals along Geodesic Paths

Version 1.0.1

Depends R(>= 4.4), methods, RGraphSpace

Imports grDevices, stats, scales, igraph, RANN, 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] (<`https://orcid.org/0000-0003-4942-8131`>)

Maintainer Mauro Castro <`mauro.a.castro@gmail.com`>

Repository CRAN

Date/Publication 2025-04-08 19:20:02 UTC

Contents

`buildPathwaySpace` *Constructor of PathwaySpace-class objects.*

Description

`buildPathwaySpace` is a constructor of PathwaySpace-class objects.

Usage

```
buildPathwaySpace(g, nrc = 500, mar = 0.075, verbose = TRUE)
```

Arguments

`g`	An `igraph` object. It must include graph layout information, with vertex coordinates assigned to `x` and `y` vertex attributes. It must also include vertex labels assigned to the `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.

`mar`	A single numeric value (in `[0,1]`) indicating the size of the outer margins as a fraction of the image matrix.
`verbose`	A single logical value specifying to display detailed messages (when `verbose=TRUE`) or not (when `verbose=FALSE`).

Value

A preprocessed PathwaySpace class object.

Author(s)

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

See Also

`undirected_graph`

Examples

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

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

`CGC_20211118` *COSMIC-CGC genes mapped to PathwaySpace images.*

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`

Creating 2D-landscape images from graph objects.

Description

`circularProjection` implements a convolution algorithm to project signal across a 2D-coordinate system.

Usage

```
## S4 method for signature 'PathwaySpace'
circularProjection(
  pts,
  kns = 8,
  pdist = 0.15,
  rescale = TRUE,
  verbose = TRUE,
  decay_fun = weibullDecay,
  knn = deprecated(),
  ...
)
```

Arguments

`pts`	A PathwaySpace class object.
`kns`	A single positive integer determining the k-nearest signal sources used in the signal convolution operation.
`pdist`	A term (in `[0,1]`) determining a distance unit for the signal convolution related to the image space. This distance will affect the extent over which the convolution operation projects the signal between source- and destination points. The signal projection tends to zero at ‘pdist‘, and when ‘pdist = 1‘ the projection will extend to its maximum distance, defined by the diameter of the inscribed circle within the pathway space.

`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 `(-Inf,+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: 'Weibull', 'exponential', and 'linear' functions (see `weibullDecay`).
`knn`	Deprecated from PathwaySpace 1.0; use ‘kns‘ instead.
`...`	Additional arguments passed to the decay function.

Value

A preprocessed PathwaySpace class object.

Author(s)

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

See Also

`buildPathwaySpace`

Examples

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

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

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

`expDecay` *Exponential decay function.*

Description

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

Usage

```
expDecay(x, signal, decay = 0.999, ...)
```

Arguments

`x`	A numeric vector of distances (in [0,1]).
`signal`	A single numeric value representing a signal.
`decay`	The rate (in [0,1]) at which the signal decays. This term indicates how much the `signal` decreases at a certain distance in `x`. At the distance defined by the `pdist` term (see `circularProjection`), the signal's value will correspond to the initial signal multiplied by `1 - decay`.
`...`	Not used; argument implemented for call compatibility with the `weibullDecay` function.

Value

A numeric vector.

Author(s)

Vinicius Chagas, Victor Apolonio, and Mauro Castro (<`mauro.castro@ufpr.br`>)

See Also

`weibullDecay`, `linearDecay`

Examples

```
x <- c(1:100) / 100
y <- expDecay(x, 1)
plot(x, y)
```

---

getPathwaySpace,PathwaySpace-method

Accessors for fetching slots from a PathwaySpace object.

Description

`getPathwaySpace` retrives information from individual slots available in a PathwaySpace object.

Usage

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

Arguments

`pts`	A preprocessed PathwaySpace class object
`what`	A single character value specifying which information should be retrieved from the slots. Options: 'graph','gxy','gxyz','pars','misc','status','summits', 'summit_mask', and 'summit_contour'.

Value

Content from slots in the PathwaySpace object.

Examples

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

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

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

`gimage` *An image matrix.*

Description

An image matrix used for workflow demonstrations.

Usage

```
data(gimage)
```

Format

matrix

Value

An image matrix.

Source

This package.

Examples

```
data(gimage)
```

`gtoys` *Toy 'igraph' objects.*

Description

Small 'igraph' objects used for workflow demonstrations. All graphs include 'x', 'y', and 'name' vertex attributes.

Usage

```
data(gtoy1)
```

Format

igraph

Value

A pre-processed igraph object.

Source

This package.

Examples

```
data(gtoy1)
data(gtoy2)
```

`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)
```

`length,PathwaySpace-method`

Accessor function for PathwaySpace objects.

Description

Get length of a PathwaySpace object.

Usage

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

Arguments

`x`	A PathwaySpace class object.

Value

A non-negative integer of length 1.

Examples

```
data('gtoy1', package = 'PathwaySpace')
pts <- buildPathwaySpace(gtoy1, nrc = 100)
length(pts)
```

`linearDecay` *A simple linear decay function.*

Description

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

Usage

```
linearDecay(x, signal, ...)
```

Arguments

`x`	A numeric vector of distances (in [0,1]).
`signal`	A single numeric value representing a signal.
`...`	Not used; argument implemented for call compatibility with `weibullDecay` and `expDecay` functions.

Value

A numeric vector.

Author(s)

Vinicius Chagas, Victor Apolonio, and Mauro Castro (<`mauro.castro@ufpr.br`>)

See Also

`weibullDecay`, `expDecay`

Examples

```
x <- c(1:100) / 100
y <- linearDecay(x, 1)
plot(x, y)
```

`names,PathwaySpace-method` *Accessor functions for PathwaySpace objects.*

Description

Get and set 'vertex' names of a PathwaySpace class object.

Usage

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

Arguments

`x`	A PathwaySpace class object.

Value

A character vector.

Examples

```
data('gtoy1', package = 'PathwaySpace')
pts <- buildPathwaySpace(gtoy1, nrc = 100)
names(pts)
```

`pathDistances` *Calculate a pathway space distance between two vectors.*

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` *PathwaySpace: An S4 class for signal propagation on image spaces.*

Description

PathwaySpace: An S4 class for signal propagation on image spaces.

Value

An S4 class object.

Slots

`vertex`	A character vector with vertex names.
`vsignal`	A numerical vector with vertex signals.
`vweight`	A numerical vector with vertex weights.
`edges`	A data frame with edges.
`gxy`	A data frame with xy-vertex coordinates (numerical).
`gxyz`	A numerical matrix with x-cols and y-rows coordinates, and a z-signal.
`pars`	A list with parameters.
`misc`	A list with intermediate 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` *A pruned and laid out igraph object from Pathway Commons V12.*

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

Examples

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

`plotPathDistances` *Accessory function to plot pathway space distances.*

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(
  pts,
  colors = pspace.cols(),
  trim.colors = c(3, 2, 1, 2, 3),
  bg.color = "grey85",
  theme.name = c("th0", "th1", "th2", "th3"),
  title = "PathwaySpace",
  font.size = 1,
  font.color = "white",
  xlab = "Pathway coordinates 1",
  ylab = "Pathway coordinates 2",
  zlab = "Density",
  zlim = NULL,
  slices = 25,
  add.grid = TRUE,
  grid.color = "white",
  add.contour = TRUE,
  contour.color = "white",
  label.summits = TRUE,
  marks = FALSE,
  mark.size = 3,
  mark.color = "white",
  mark.padding = 0.5,
  mark.line.width = 0.5,
  use.dotmark = FALSE
)
```

Arguments

`pts`	A PathwaySpace class object.
`colors`	A vector of colors. Each color is a specific tone used to create a customized color palette that matches the signal type projected in the image space, such as binary or numeric values in (-Inf, +Inf). The palette will be created with 5-color tones, interpolated according to the provided sequence in the vector of colors. The proportion of each color tone can be adjusted by the 'trim.colors' argument. The

	primary objective of this color palette construction is to enhance the visibility of summits and valleys in the image space. If you wish to bypass the 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 tone in the palette.
`bg.color`	A single color for background.
`theme.name`	Name of a custom PathwaySpace theme. These themes (from ’th0’ to ’th3’) consist mainly of preconfigured ggplot settings, which the user can subsequently fine-tune within the resulting ggplot object.
`title`	A string for the title.
`font.size`	A single numeric value passed to ggplot themes.
`font.color`	A single color passed to ggplot themes.
`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.
`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.
`add.grid`	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`).
`grid.color`	A color passed to `geom_point`.
`add.contour`	A logical value indicating whether to add contour lines to ’summits’ (see `summitMapping`).
`contour.color`	A color passed to `geom_tile`.
`label.summits`	A logical value indicating whether to label summits, (when summits are available).
`marks`	A logical value indicating whether to add ’marks’ to vertex positions. Alternatively, this could be a vector listing vertex names.
`mark.size`	A font size argument passed to `geom_text`.
`mark.color`	A color passed to `geom_text_repel`.
`mark.padding`	A box padding argument passed to `geom_text_repel`.
`mark.line.width`	A line width argument passed to `geom_text_repel`.
`use.dotmark`	A logical value indicating whether "marks" should be represented as dots.

Value

A ggplot-class object.

Author(s)

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

See Also

`circularProjection`

Examples

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

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

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

# Map graph silhouette
pts <- silhouetteMapping(pts)

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

`polarProjection,PathwaySpace-method`
Creating 2D-landscape images from graph objects.

Description

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

Usage

```
## S4 method for signature 'PathwaySpace'
polarProjection(
  pts,
  kns = 8,
  pdist = 0.5,
  rescale = TRUE,
  theta = 180,
  directional = FALSE,
  verbose = TRUE,
  decay_fun = weibullDecay,
  knn = deprecated(),
```

```
    ...
)
```

Arguments

`pts`	A PathwaySpace class object.
`kns`	A single positive integer determining the k-nearest signal sources used in the signal convolution operation.
`pdist`	A term (in `[0,1]`) determining a distance unit for the signal convolution related to length between any two connected vertices. This distance will affect the extent over which the convolution operation projects the signal between source- and destination points along the polar coordinates of the edges. The signal projection tends to zero at ‘pdist‘, and when ‘pdist = 1‘ the projection will extend to its maximum distance, defined by the length between the two connected vertices.
`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 `(-Inf,+Inf)`, then it will be rescaled to `[-1,1]`.
`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.
`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: 'Weibull', 'exponential', and 'linear' functions (see `weibullDecay`).
`knn`	Deprecated from PathwaySpace 1.0; use ‘kns‘ instead.
`...`	Additional arguments passed to the decay function.

Value

A preprocessed PathwaySpace class object.

Author(s)

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

See Also

`buildPathwaySpace`

Examples

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

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

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

`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()
```

`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(pts, baseline = 0.01, pdist = 0.05, verbose = TRUE)
```

Arguments

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

Value

A preprocessed PathwaySpace class object.

Author(s)

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

See Also

`circularProjection`

Examples

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

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

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

# Map graph silhouette
pts <- silhouetteMapping(pts)
```

---

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(
  pts,
  maxset = 30,
  minsize = 30,
  threshold = 0.5,
  verbose = TRUE,
  segm_fun = summitWatershed,
  ...
)
```

Arguments

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

Value

A preprocessed PathwaySpace class object.

Author(s)

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

See Also

`circularProjection`

Examples

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

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

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

# Map summits in a 2D-landscape image
pts <- summitMapping(pts)
```

`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

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

Value

A matrix with labeled summits.

Author(s)

Vinicius Chagas, Victor Apolonio, and Mauro Castro (<`mauro.castro@ufpr.br`>)

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 fetching slots from a PathwaySpace object.*

Description

Get or set 'signal' for a PathwaySpace class object.

Usage

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

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

Arguments

`pts`	A PathwaySpace class object.
`value`	A numeric vector with values representing signal intensities. This vector should be aligned to the "vertex" slot.

Value

A numeric vector.

Examples

```
data('gtoy1', package = 'PathwaySpace')
pts <- buildPathwaySpace(gtoy1, nrc = 100)
vertexSignal(pts)
```

---

vertexWeight,PathwaySpace-method

Accessor functions for fetching slots from a PathwaySpace object.

Description

Get or set 'weights' for a PathwaySpace class object.

Usage

```
## S4 method for signature 'PathwaySpace'
vertexWeight(pts)

## S4 replacement method for signature 'PathwaySpace'
vertexWeight(pts) <- value
```

Arguments

`pts`	A PathwaySpace class object.
`value`	A numeric vector with values representing vertex weights. This vector should be aligned to the "vertex" slot.

Value

A numeric vector.

Examples

```
data('gtoy1', package = 'PathwaySpace')
pts <- buildPathwaySpace(gtoy1, nrc = 100)
vertexWeight(pts)
```

`weibullDecay` *Weibull decay function.*

Description

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

Usage

```
weibullDecay(x, signal, decay = 0.999, shape = 1.05)
```

Arguments

`x`	A numeric vector of distances (in [0,1]).
`signal`	A single numeric value representing a signal.
`decay`	The rate (in [0,1]) at which the signal decays. This term indicates how much the `signal` decreases at a certain distance in `x`. At the distance defined by the `pdist` term (see `circularProjection`), the signal's value will correspond to the initial signal multiplied by `1 - 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.

Author(s)

Vinicius Chagas, Victor Apolonio, and Mauro Castro (<`mauro.castro@ufpr.br`>)

See Also

`expDecay`, `linearDecay`

Examples

```
x <- c(1:100) / 100
y <- weibullDecay(x, 1)
plot(x, y)
```

Index