

Package ‘spathial’

March 25, 2020

Type Package

Title Evolutionary Analysis

Version 0.1.1

Author Erika Gardini [aut, cre],
Federico M. Giorgi [aut],
Sergio Decherchi [aut]

Maintainer Erika Gardini <erika.gardini4@unibo.it>

Description A generic tool for manifold analysis. It allows to infer a relevant transition or evolutionary path which can highlights the features involved in a specific process. 'spathial' can be useful in all the scenarios where the temporal (or pseudo-temporal) evolution is the main problem (e.g. tumor progression). The algorithm for finding the principal path is described in: Ferrarotti et al., (2019) <doi:10.1109/TNNLS.2018.2884792>."

License GPL-3

Encoding UTF-8

LazyData TRUE

RoxygenNote 7.0.2

Depends R (>= 3.5)

NeedsCompilation no

Imports pracma, igraph, matrixStats, MASS, Rtsne, class, utils, knitr,
rmarkdown

VignetteBuilder utils, knitr, rmarkdown

Repository CRAN

Date/Publication 2020-03-25 15:30:02 UTC

R topics documented:

spathialBoundaryIds	2
spathialLabels	3
spathialPlot	4
spathialPrefiltering	5
spathialStatistics	6
spathialWay	7

spathialBoundaryIds *Select starting and ending points*

Description

Get the coordinates of the starting and ending points

Usage

```
spathialBoundaryIds(X, X_labels = NULL, mode = 1, from = NULL, to = NULL)
```

Arguments

X	data points
X_labels	labels of the data points
mode	strategy for boundary selection <ul style="list-style-type: none">• 1 - selected by the user• 2 - centroids• 3 - insert the row name of the starting and ending points
from	starting class or row name of the starting point
to	ending class or row name of the ending point

Value

A list of objects

- boundary ids: the indexes of the boundaries
- X: the new data matrix with the boundary
- X_labels: the new labels of the data matrix with the boundary labels

Examples

```
# Load data matrix X
load(system.file('extdata', 'X.rda', package='spathial', mustWork=TRUE))
# Load description vector X_labels
load(system.file('extdata', 'X_labels.rda', package='spathial', mustWork=TRUE))
# Run spathialBoundary
boundaryRes <- spathialBoundaryIds(X, X_labels, mode=2, from=3, to=6)
```

spatialLabels	<i>Find labels</i>
---------------	--------------------

Description

Get the label of each waypoint according to the neighbourhood

Usage

```
spatialLabels(X, X_labels, spathial_res)
```

Arguments

X	data points
X_labels	labels of the data points
spathial_res	principal path from the starting point to the ending point

Value

ppath_labels: labels of the waypoints

Examples

```
#EXAMPLE 1
# Load data matrix X
load(system.file('extdata', 'X.rda', package='spatial', mustWork=TRUE))
# Load description vector X_labels
load(system.file('extdata', 'X_labels.rda', package='spatial', mustWork=TRUE))
# Run spathialBoundary
boundaryRes <- spathialBoundaryIds(X, X_labels, mode=2, from=3, to=6)
X <- boundaryRes$X
X_labels <- boundaryRes$X_labels
boundary_ids <- boundaryRes$boundary_ids
#Set the number of waypoints
NC <- 20
# Run spathialWay
spathial_res <- spathialWay(X, boundary_ids, NC)
#Run spathialLabels with spathial_res
labels <- spathialLabels(X, X_labels, spathial_res)
labels
```

 spathialPlot

2D spathial

Description

Get the 2D coordinates of each waypoint (using t-SNE algorithm for the dimensionality reduction)

Usage

```
spathialPlot(
  X,
  X_labels,
  boundary_ids,
  spathial_res,
  perplexity_value = NULL,
  mask = NULL,
  title = NULL,
  ...
)
```

Arguments

<code>X</code>	data points
<code>X_labels</code>	labels of the data points
<code>boundary_ids</code>	waypoints
<code>spathial_res</code>	principal path from the starting point to the ending point
<code>perplexity_value</code>	the value for TSNE perplexity (default is $nrsamples*3/50$)
<code>mask</code>	the mask of the sample to preserve (when prefiltering is computed)
<code>title</code>	the title of the plot
<code>...</code>	Parameters which will be inherited by <code>plot()</code>

Examples

```
load(system.file('extdata', 'X.rda', package='spathial', mustWork=TRUE))
# Load description vector X_labels
load(system.file('extdata', 'X_labels.rda', package='spathial', mustWork=TRUE))
# Run spathialBoundary
boundaryRes <- spathialBoundaryIds(X, X_labels, mode=2, from=3, to=6)
X <- boundaryRes$X
X_labels <- boundaryRes$X_labels
boundary_ids <- boundaryRes$boundary_ids
#Set the number of waypoints
NC <- 20
# Run spathialWay
spathial_res <- spathialWay(X, boundary_ids, NC)
```

```
#Run spatialPlot with spatial_res  
spatialPlot(X, X_labels, boundary_ids, spatial_res, perplexity_value=30)
```

spatialPrefiltering *Prefilter data*

Description

Regularized K-means for principal path: prefiltering

Usage

```
spatialPrefiltering(X, boundary_ids)
```

Arguments

X	data points
boundary_ids	names of the start and ending points, to be treated separately

Value

A list of objects

- mask: indexes of the data points to preserv
- boundary_ids: the filtered boundary ids

Examples

```
# Load data matrix X  
load(system.file('extdata', 'X.rda', package='spatial', mustWork=TRUE))  
# Load description vector X_labels  
load(system.file('extdata', 'X_labels.rda', package='spatial', mustWork=TRUE))  
# Run spatialBoundary  
boundaryRes <- spatialBoundaryIds(X, X_labels, mode=2, from=3, to=6)  
X <- boundaryRes$X  
X_labels <- boundaryRes$X_labels  
boundary_ids <- boundaryRes$boundary_ids  
# Run spatial spatialPrefiltering with the output of the function spatialBoundaryIds  
filterRes <- spatialPrefiltering(X, boundary_ids)  
# Extract prefiltering results  
mask <- filterRes$mask  
boundary_ids <- filterRes$boundary_ids  
X <- X[mask,]
```

spatialStatistics *Correlation*

Description

Get how much the features correlate with the path

Usage

```
spatialStatistics(spatial_res)
```

Arguments

`spatial_res` principal path from the starting point to the ending point

Value

A list of objects

- `correlations`: Pearson's correlation coefficients between each feature and the path (when `p_path_perturbed` is not NULL, a Fisher-integrated correlation coefficient is provided)
- `ranks`: ranks of associations between the `n` features and the path (when `p_path_perturbed` is not NULL, the mean of the ranks is provided)
- `p_values`: p values from the Pearson's correlation scores
- `p_adj`: p values adjusted according to the Benjamini & Hochberg (BH) method

Examples

```
# Load data matrix X
load(system.file('extdata', 'X.rda', package='spatial', mustWork=TRUE))
# Load description vector X_labels
load(system.file('extdata', 'X_labels.rda', package='spatial', mustWork=TRUE))
# Run spatialBoundary
boundaryRes <- spatialBoundaryIds(X, X_labels, mode=2, from=3, to=6)
X <- boundaryRes$X
X_labels <- boundaryRes$X_labels
boundary_ids <- boundaryRes$boundary_ids
#Set the number of waypoints
NC <- 20
# Run spatialWay
spatial_res <- spatialWay(X, boundary_ids, NC)
#Run spatialStatistics with spatial_res
statistics <- spatialStatistics(spatial_res)
```

spatialWay *Compute Principal Path*

Description

Get the coordinates of the waypoints of the principal path

Usage

```
spatialWay(X, boundary_ids, NC = 50)
```

Arguments

X	data points
boundary_ids	starting and ending points
NC	number of waypoints

Value

spatial_res: spatial waypoints

Examples

```
#EXAMPLE 1
# Load data matrix X
load(system.file('extdata', 'X.rda', package='spatial', mustWork=TRUE))
# Load description vector X_labels
load(system.file('extdata', 'X_labels.rda', package='spatial', mustWork=TRUE))
# Run spatialBoundary
boundaryRes <- spatialBoundaryIds(X, X_labels, mode=2, from=3, to=6)
X <- boundaryRes$X
X_labels <- boundaryRes$X_labels
boundary_ids <- boundaryRes$boundary_ids
#Set the number of waypoints
NC <- 20
# Run spatialWay
spatial_res <- spatialWay(X, boundary_ids, NC)
spatial_res
```

Index

`spatialBoundaryIds`, [2](#)
`spatialLabels`, [3](#)
`spatialPlot`, [4](#)
`spatialPrefiltering`, [5](#)
`spatialStatistics`, [6](#)
`spatialWay`, [7](#)