Package 'spicyR'

May 25, 2024

Type Package

Title Spatial analysis of in situ cytometry data

Version 1.17.1

Description The spicyR package provides a framework for performing inference on changes in spatial relationships between pairs of cell types for cell-resolution spatial omics technologies. spicyR consists of three primary steps: (i) summarizing the degree of spatial localization between pairs of cell types for each image; (ii) modelling the variability in localization summary statistics as a function of cell counts and (iii) testing for changes in spatial localizations associated with a response variable.

License GPL (>=2)

LazyData true

biocViews SingleCell, CellBasedAssays, Spatial

Encoding UTF-8

Depends R (>= 4.1)

VignetteBuilder knitr

BugReports https://github.com/SydneyBioX/spicyR/issues

URL https://ellispatrick.github.io/spicyR/
https://github.com/SydneyBioX/spicyR

Imports ggplot2, concaveman, BiocParallel, spatstat.explore, spatstat.geom, lmerTest, S4Vectors, methods, pheatmap, rlang, grDevices, stats, data.table, dplyr, tidyr, scam, SingleCellExperiment, SpatialExperiment, SummarizedExperiment, ggforce, ClassifyR, tibble, magrittr, cli

Suggests BiocStyle, knitr, rmarkdown, pkgdown, imcRtools, testthat (>= 3.0.0)

RoxygenNote 7.3.1

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/spicyR

git_branch devel

git_last_commit 94657ba

git_last_commit_date 2024-05-19

Repository Bioconductor 3.20

Date/Publication 2024-05-24

Author Nicolas Canete [aut],
Ellis Patrick [aut, cre],
Nicholas Robertson [ctb]

Maintainer Ellis Patrick <ellis.patrick@sydney.edu.au>

Contents

`bind` *Produces a dataframe showing L-function metric for each imageID entry.*

Description

Produces a dataframe showing L-function metric for each imageID entry.

Usage

```
bind(results, pairName = NULL)
```

Arguments

`results`	Spicy test result obtained from spicy.
`pairName`	A string specifying the pairwise interaction of interest. If NULL, all pairwise interactions are shown.

Value

A data.frame containing the colData related to the results.

Examples

```
data(spicyTest)
df <- bind(spicyTest)
```

`colTest` *Perform a simple wilcoxon-rank-sum test or t-test on the columns of a data frame*

Description

Perform a simple wilcoxon-rank-sum test or t-test on the columns of a data frame

Usage

```
colTest(df, condition, type = NULL, feature = NULL, imageID = "imageID")
```

Arguments

`df`	A data.frame or SingleCellExperiment, SpatialExperiment
`condition`	The condition of interest
`type`	The type of test, "wilcox", "ttest" or "survival".
`feature`	Can be used to calculate the proportions of this feature for each image
`imageID`	The imageID's if presenting a SingleCellExperiment

Value

Proportions

Examples

```
# Test for an association with long-duration diabetes
# This is clearly ignoring the repeated measures...
data("diabetesData")
diabetesData <- spicyR::.format_data(
  diabetesData, "imageID", "cellType", c("x", "y"), FALSE
)
props <- getProp(diabetesData)
condition <- spicyR:::getImagePheno(diabetesData)$stage
names(condition) <- spicyR:::getImagePheno(diabetesData)$imageID
condition <- condition[condition %in% c("Long-duration", "Onset")]
test <- colTest(props[names(condition), ], condition)
```

convPairs *Converts colPairs object into an abundance matrix based on number of nearby interactions for every cell type.*

Description

Converts colPairs object into an abundance matrix based on number of nearby interactions for every cell type.

Usage

```
convPairs(cells, colPair, cellType = "cellType", imageID = "imageID")
```

Arguments

cells	A SingleCellExperiment that contains objects in the colPairs slot.
colPair	The name of the object in the colPairs slot for which the dataframe is constructed from.
cellType	The cell type if using SingleCellExperiment.
imageID	The image ID if using SingleCellExperiment.

Value

Matrix of abundances

Examples

```
data("diabetesData")
images <- c("A09", "A11", "A16", "A17")
diabetesData <- diabetesData[
  , SummarizedExperiment::colData(diabetesData)$imageID %in% images
]

diabetesData_SPE <- SpatialExperiment::SpatialExperiment(diabetesData,
  colData = SummarizedExperiment::colData(diabetesData)
)
SpatialExperiment::spatialCoords(diabetesData_SPE) <- data.frame(
  SummarizedExperiment::colData(diabetesData_SPE)$x,
  SummarizedExperiment::colData(diabetesData_SPE)$y
) |>
  as.matrix()

SpatialExperiment::spatialCoordsNames(diabetesData_SPE) <- c("x", "y")

diabetesData_SPE <- imcRtools::buildSpatialGraph(diabetesData_SPE,
  img_id = "imageID",
  type = "knn",
  k = 20,
```

```
  coords = c("x", "y")
)

pairAbundances <- convPairs(diabetesData_SPE,
  colPair = "knn_interaction_graph"
)
```

`diabetesData` *Diabetes IMC data in SCE format.*

Description

This is a subset of the Damond et al 2019 imaging mass cytometry dataset. The data contains cells in the pancreatic islets of individuals with early onset diabetes and healthy controls. The object contains single-cell data of 160 images from 8 subjects, with 20 images per subject.

Usage

```
data("diabetesData")
```

Format

diabetesData_SCE a SingleCellExperiment object

Details

Converted into a SingleCellExperiment format.

`getPairwise` *Get statistic from pairwise L curve of a single image.*

Description

Get statistic from pairwise L curve of a single image.

Usage

```
getPairwise(
  cells,
  from = NULL,
  to = NULL,
  window = "convex",
  window.length = NULL,
  Rs = c(20, 50, 100),
  sigma = NULL,
```

```
  minLambda = 0.05,
  edgeCorrect = TRUE,
  includeZeroCells = TRUE,
  BPPARAM = BiocParallel::SerialParam(),
  imageIDCol = "imageID",
  cellTypeCol = "cellType",
  spatialCoordCols = c("x", "y")
)
```

Arguments

cells	A SummarizedExperiment that contains at least the variables x and y, giving the location coordinates of each cell, and cellType.
from	The 'from' cellType for generating the L curve.
to	The 'to' cellType for generating the L curve.
window	Should the window around the regions be 'square', 'convex' or 'concave'.
window.length	A tuning parameter for controlling the level of concavity when estimating concave windows.
Rs	A vector of the radii that the measures of association should be calculated.
sigma	A numeric variable used for scaling when fitting inhomogeneous L-curves.
minLambda	Minimum value for density for scaling when fitting inhomogeneous L-curves.
edgeCorrect	A logical indicating whether to perform edge correction.
includeZeroCells	A logical indicating whether to include cells with zero counts in the pairwise association calculation.
BPPARAM	A BiocParallelParam object.
imageIDCol	The name of the imageID column if using a SingleCellExperiment or SpatialExperiment.
cellTypeCol	The name of the cellType column if using a SingleCellExperiment or SpatialExperiment.
spatialCoordCols	The names of the spatialCoords column if using a SingleCellExperiment.

Value

Statistic from pairwise L curve of a single image.

Examples

```
data("diabetesData")
# Subset by imageID for fast example
selected_cells <- diabetesData[
  , SummarizedExperiment::colData(diabetesData)$imageID == "A09"
]
pairAssoc <- getPairwise(selected_cells)
```

getProp *Get proportions from a SummarizedExperiment.*

Description

Get proportions from a SummarizedExperiment.

Usage

```
getProp(cells, feature = "cellType", imageID = "imageID")
```

Arguments

cells	A SingleCellExperiment, SpatialExperiment or data.frame.
feature	The feature of interest
imageID	The imageID's

Value

Proportions

Examples

```
data("diabetesData")
prop <- getProp(diabetesData)
```

signifPlot *Plots result of signifPlot.*

Description

Plots result of signifPlot.

Usage

```
signifPlot(
  results,
  fdr = FALSE,
  type = "bubble",
  breaks = NULL,
  comparisonGroup = NULL,
  colours = c("#4575B4", "white", "#D73027"),
  marksToPlot = NULL,
  cutoff = 0.05
)
```

Arguments

results	Data frame obtained from spicy.
fdr	TRUE if FDR correction is used.
type	Where to make a bubble plot or heatmap.
breaks	Vector of 3 numbers giving breaks used in pheatmap. The first number is the minimum, the second is the maximum, the third is the number of breaks.
comparisonGroup	A string specifying the name of the outcome group to compare with the base group.
colours	Vector of colours to use in pheatmap.
marksToPlot	Vector of marks to include in pheatmap.
cutoff	significance threshold for circles in bubble plot

Value

a pheatmap object

Examples

```
data(spicyTest)

p <- signifPlot(spicyTest, breaks = c(-3, 3, 0.5))
# plot includes unicode characters, do not use default pdf device
ggplot2::ggsave(p, filename = tempfile(), device = cairo_pdf)
```

---

spicyBoxPlot *Plots boxplot for a specified cell-cell relationship*

Description

Plots boxplot for a specified cell-cell relationship

Usage

```
spicyBoxPlot(results, from = NULL, to = NULL, rank = NULL)
```

Arguments

results	Data frame obtained from spicy.
from	Cell type which you would like to compare to the to cell type.
to	Cell type which you would like to compare to the from cell type.
rank	Ranking of cell type in terms of p-value, the smaller the p-value the higher the rank.

Value

a ggplot2 boxplot

Examples

```
data(spicyTest)

spicyBoxPlot(spicyTest,
             rank = 1)
```

`SpicyResults-class` *Performs spatial tests on spatial cytometry data.*

Description

Performs spatial tests on spatial cytometry data.

Usage

```
spicy(
  cells,
  condition,
  subject = NULL,
  covariates = NULL,
  from = NULL,
  to = NULL,
  imageIDCol = "imageID",
  cellTypeCol = "cellType",
  spatialCoordCols = c("x", "y"),
  alternateResult = NULL,
  verbose = FALSE,
  weights = TRUE,
  weightsByPair = FALSE,
  weightFactor = 1,
  window = "convex",
  window.length = NULL,
  BPPARAM = BiocParallel::SerialParam(),
  sigma = NULL,
  Rs = NULL,
  minLambda = 0.05,
  edgeCorrect = TRUE,
  includeZeroCells = FALSE,
  ...
)
```

Arguments

cells	A SummarizedExperiment or data frame that contains at least the variables x and y, giving the location coordinates of each cell, and cellType.
condition	Vector of conditions to be tested corresponding to each image.
subject	Vector of subject IDs corresponding to each image if cells is a data frame.
covariates	Vector of covariate names that should be included in the mixed effects model as fixed effects.
from	vector of cell types which you would like to compare to the to vector
to	vector of cell types which you would like to compare to the from vector
imageIDCol	The image ID if using SingleCellExperiment.
cellTypeCol	The cell type if using SingleCellExperiment.
spatialCoordCols	The spatial coordinates if using a SingleCellExperiment.
alternateResult	An pairwise association statistic between each combination of celltypes in each image.
verbose	logical indicating whether to output messages.
weights	logical indicating whether to include weights based on cell counts.
weightsByPair	logical indicating whether weights should be calculated for each cell type pair.
weightFactor	numeric that controls the convexity of the weight function.
window	Should the window around the regions be 'square', 'convex' or 'concave'.
window.length	A tuning parameter for controlling the level of concavity when estimating concave windows.
BPPARAM	A BiocParallelParam object.
sigma	A numeric variable used for scaling when fitting inhomogeneous L-curves.
Rs	A vector of radii that the measures of association should be calculated.
minLambda	Minimum value for density for scaling when fitting inhomogeneous L-curves.
edgeCorrect	A logical indicating whether to perform edge correction.
includeZeroCells	A logical indicating whether to include cells with zero counts in the pairwise association calculation.
...	Other options.

Value

Data frame of p-values.

Examples

```
data("diabetesData")

# Test with random effect for patient on a pairwise combination of cell
# types.
spicy(diabetesData,
  condition = "stage", subject = "case",
  from = "Tc", to = "Th"
)

# Test all pairwise combinations of cell types without random effect of
# patient.
## Not run:
spicyTest <- spicy(diabetesData, condition = "stage", subject = "case")

## End(Not run)

# Test all pairwise combination of cell types with random effect of patient.
## Not run:
spicy(diabetesData, condition = "condition", subject = "subject")

## End(Not run)
```

`spicyTest` *Results from spicy for diabetesData*

Description

Results from the call: spicyTest <- spicy(diabetesData, condition = "condition", subject = "subject")

Usage

```
data("spicyTest")
```

Format

spicyTest a spicy object

`topPairs` *A table of the significant results from spicy tests*

Description

A table of the significant results from spicy tests

Usage

```
topPairs(x, coef = NULL, n = 10, adj = "fdr", cutoff = NULL, figures = NULL)
```

Arguments

x	The output from spicy.
coef	Which coefficient to list.
n	Extract the top n most significant pairs.
adj	Which p-value adjustment method to use, argument for p.adjust().
cutoff	A p-value threshold to extract significant pairs.
figures	Round to 'figures' significant figures.

Value

A data.frame

Examples

```
data(spicyTest)
topPairs(spicyTest)
```

Index