

Package: splitTypeR (via r-universe)

October 25, 2026

Type Package

Version 0.99.9

Date 2026-10-24

Title Transcriptomic Classification using Mixture of Normal Distributions

Description The splitTypeR package enables the classification of bulk transcriptomic samples using gene signature lists. For each signature, an enrichment score is first calculated for every sample with Bioconductor GSVA, and the uncertainty around each score is estimated through permutations of the samples. Values drawn from the sample-specific normal distributions are then pooled to fit a mixture of two normal distributions that represents the whole cohort. Each sample is finally tested against the distribution with the lower mean, and samples with a significantly higher score are assigned to the signature. Published gene signatures related to pancreatic ductal adenocarcinoma subtyping are also provided.

Encoding UTF-8

License Artistic-2.0

Depends R (>= 4.5.0)

Imports S4Vectors, mixtools, ggplot2, stringr, rlang, GSVA, methods

Suggests BiocStyle, knitr, rmarkdown, SummarizedExperiment, testthat (>= 3.0.0)

biocViews BiologicalQuestion, Classification, GeneSetEnrichment, Software

VignetteBuilder knitr

URL <https://github.com/adeschen/splitTypeR>

BugReports <https://github.com/adeschen/splitTypeR/issues>

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Config/pak/sysreqs cmake make libmagick++-dev gsfonts libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev zlib1g-dev

Repository <https://biocstaging.r-universe.dev>

Date/Publication 2026-10-25 03:45:21 UTC

RemoteUrl <https://github.com/BiocStaging/splitTypeR>

RemoteRef HEAD

RemoteSha f19ea104fd7c0c822734497a6d723d10007e4d92

Contents

splitTypeR-package	3
classification	4
classification,SplitTypeRdata-method	5
classification<-	5
classification<-,SplitTypeRdata-method	6
expNormalCountsDemo	7
getGeneSignatures	8
getGeneSignaturesNames	9
getGenesInSignature	10
gsvaResults	11
gsvaResults,SplitTypeRdata-method	11
gsvaResults<-	12
gsvaResults<-,SplitTypeRdata-method	13
ListOrNULL-class	14
MatrixOrNULL-class	14
model	14
model,SplitTypeRdata-method	15
model<-	16
model<-,SplitTypeRdata-method	17
permutations	18
permutations,SplitTypeRdata-method	19
permutations<-	19
permutations<-,SplitTypeRdata-method	20
plotDensityWithUpscalingSamplesBimodal	21
plotPermutationSamplesDistribution	23
plotPermutationSamplesNormal	24
runSubtypingBimodal	26
show,SplitTypeRdata-method	27
signatures	28
signatures,SplitTypeRdata-method	29
signatures<-	30
signatures<-,SplitTypeRdata-method	30
signaturesDemo	31
SplitTypeRdata	32
SplitTypeRdata-class	34
standardDeviation	35

standardDeviation,SplitTypeRdata-method	35
standardDeviation<-	36
standardDeviation<-,SplitTypeRdata-method	37
threshold	38
threshold,SplitTypeRdata-method	39
threshold<-	39
threshold<-,SplitTypeRdata-method	40
upscaling	41
upscaling,SplitTypeRdata-method	42
upscaling<-	43
upscaling<-,SplitTypeRdata-method	43
Index	45

splitTypeR-package	<i>Transcriptomic classification using multimodal distributions</i>
--------------------	---

Description

The splitTypeR package provides an automated statistical framework for classifying heterogeneous biological samples based on gene signature lists, effectively isolating signature-positive samples. This classification method is designed for bulk transcriptomic datasets.

Details

The workflow begins by calculating a Gene Set Variation Analysis (GSVA) score for each sample to quantify the relative pathway activity. A subsequent permutation step extracts sample-specific standard deviations to capture the variance. By combining these metrics, the package establishes a unique normal distribution for each sample. Through an up-scaling step, values are randomly drawn from these individual distributions and pooled to construct a comprehensive mixture of normal distributions that represent the entire sample population.

Within this mixture model, the sub-distribution with the lower mean value is designated as the alternative distribution. The framework applies a hypothesis-testing approach in which the null hypothesis posits that a sample belongs to this alternative distribution. Samples that successfully reject the null hypothesis are assigned to the signature classification. Conversely, samples that fail to reject the null hypothesis remain unclassified.

By automating the classification workflow, the splitTypeR package provides an objective method to identify patient or sample subgroups in heterogeneous transcriptomic datasets.

Author(s)

Astrid Deschênes, Pascal Belleau

Maintainer: Astrid Deschênes <adeschen@hotmail.com>

See Also

- [runSubtypingBimodal](#) for the classification of heterogeneous biological samples based on gene signature lists using a mixture of two normal distributions
- [getGeneSignaturesNames](#) for the names of the available and ready-to-use gene signature lists
- [getGeneSignatures](#) for the available and ready-to-use gene signature lists

classification

Generic function for getting the classification slot in a class

Description

A generic function for getting the classification slot in a class. The function is implemented for the `SplitTypeRdata` class.

Usage

```
classification(x)
```

Arguments

`x` an object.

Value

an object.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(classification="character"))

# Create an instance
obj <- new("MyClass", classification="123")

# Call the generic (assuming a method is implemented)
# classification(obj)
```

`classification, SplitTypeRdata-method`*Getter for the classification slot in a SplitTypeRdata class*

Description

A function for getting the classification slot in a SplitTypeRdata class.

Usage

```
## S4 method for signature 'SplitTypeRdata'  
classification(x)
```

Arguments

`x` a SplitTypeRdata object.

Value

NULL or a list NULL or a list of the classification for each signature. The list should have one entry per signature.

Author(s)

Astrid Deschênes

Examples

```
## Create a SplitTypeRdata object  
splitData <- SplitTypeRdata()  
  
## Extract the model slot for the object  
classification(splitData)
```

`classification<-`*Generic function for replacement of the classification slot in a class*

Description

A generic function for replacement of the classification slot in a S4 object.

Usage

```
classification(x) <- value
```

Arguments

`x` a S4 object.
`value` the new value to assign or update.

Value

the modified S4 object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(classification="character"))

# Create an instance
obj <- new("MyClass", classification="123")

# Call the generic (assuming a method is implemented)
# classification(obj) <- "333"
```

`classification<-`, *SplitTypeRdata*-method

Replacement of the classification slot in a SplitTypeRdata object

Description

A function for replacement of the classification slot in a *SplitTypeRdata* class.

Usage

```
## S4 replacement method for signature 'SplitTypeRdata'
classification(x) <- value
```

Arguments

`x` a *SplitTypeRdata* object.
`value` NULL or a list of the classification data for each signature. The list should have one entry per signature.

Value

the modified *SplitTypeRdata* object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
## Two demo signatures
sign <- list("SignatureA"=c("EGFR", "CTTN", "ACTB"),
            "SignatureB"=c("ACTR2", "AKT1"))

## Create a SplitTypeRdata object with the signatures
demo <- new("SplitTypeRdata", signatures=sign)

## Create the classification of 3 samples for each signature
## Each entry contains the GSVA score of each sample and its
## classification (the signature name or "unclassified")
demoClass <- list()
demoClass[["SignatureA"]] <- data.frame(GSVA_score=c(0.45, -0.32, 0.12),
    classification=c("SignatureA", "unclassified", "unclassified"),
    row.names=c("Sample_1", "Sample_2", "Sample_3"))
demoClass[["SignatureB"]] <- data.frame(GSVA_score=c(-0.51, 0.38, 0.29),
    classification=c("unclassified", "SignatureB", "SignatureB"),
    row.names=c("Sample_1", "Sample_2", "Sample_3"))

## Assign the classification to the object
classification(demo) <- demoClass

## Get the classification for the first signature
classification(demo)[["SignatureA"]]
```

expNormalCountsDemo	<i>Simulated normalized expected counts for 30 demonstration samples</i>
---------------------	--

Description

A small normalized expected gene counts matrix for 30 patients generated to be used as a demonstration set in this package.

Usage

```
data(expNormalCountsDemo)
```

Format

a matrix containing normalized gene counts for 30 patients (columns). Only genes related to 2018 Tiriatic PDAC PDO classical and basal-like signatures are present (row).

Source

Simulated values generated for demonstration purposes. The genes correspond to the genes of the PDAC patient-derived organoids (PDO) classical and basal-like signatures published in Tiriác et al. (2018).

See Also

- [runSubtypingBimodal](#) for the classification of heterogeneous biological samples based on gene signature lists using a mixture of two normal distributions
- [getGeneSignaturesNames](#) for the names of the available and ready-to-use gene signature lists
- [getGeneSignatures](#) for the available and ready-to-use gene signature lists

Examples

```
## Load the demo normalized expected count matrix
data("expNormalCountsDemo")

## Load the demo signatures
data("signaturesDemo")

## Fix seed for reproducibility
set.seed(221)

## Run classification on the 30 patients using 25 permutations on 70% of
## the cohort, and 10 points per patient for the up-scaling step
results <- runSubtypingBimodal(geneLists=signaturesDemo,
  expectedCountsMatrix=expNormalCountsDemo,
  permRatio=0.70, permNbr=25, upscaleNbr=10)
```

getGeneSignatures	<i>Return the selected gene list signatures</i>
-------------------	---

Description

The function returns the selected gene list signatures.

Usage

```
getGeneSignatures(nameList = NULL)
```

Arguments

nameList	a list of signature names. At least one name is required. The signature names must all be present in the gene list signatures. When NULL, all gene signatures are returned. Default: NULL.
----------	--

Details

The PDAC PDO classical and basal-like signatures are associated with this publication:

Tiriac et al. Organoid Profiling Identifies Common Responders to Chemotherapy in Pancreatic Cancer. Cancer Discov. 2018 Sep;8(9):1112-1129. doi: 10.1158/2159-8290.CD-18-0349. Epub 2018 May 31. PMID: 29853643; PMCID: PMC6125219.

Value

a list of selected signatures. Each signature is composed of a vector of gene names.

Author(s)

Astrid Deschênes

Examples

```
## Extract the basal-like signature from Tiriac et al 2018  
getGeneSignatures(nameList=c("Tiriac_PDAC_PDO_Basal_like"))
```

`getGeneSignaturesNames`

List the names of the available signatures

Description

The function lists the names of all the available signatures.

Usage

```
getGeneSignaturesNames()
```

Details

The PDAC PDO classical and basal-like signatures are associated with this publication:

Tiriac et al. Organoid Profiling Identifies Common Responders to Chemotherapy in Pancreatic Cancer. Cancer Discov. 2018 Sep;8(9):1112-1129. doi: 10.1158/2159-8290.CD-18-0349. Epub 2018 May 31. PMID: 29853643; PMCID: PMC6125219.

Value

a vector of available signature names

Author(s)

Astrid Deschênes

Examples

```
## Print the names of the available signatures  
getGeneSignaturesNames()
```

getGenesInSignature	<i>Return the genes for a specific signature</i>
---------------------	--

Description

The function returns the genes for the selected signature.

Usage

```
getGenesInSignature(name)
```

Arguments

name	a character string representing the name of the signature names. The signature name must all be present in the gene list signatures.
------	--

Details

The PDAC PDO classical and basal-like signatures are associated with this publication:

Tiriác et al. Organoid Profiling Identifies Common Responders to Chemotherapy in Pancreatic Cancer. Cancer Discov. 2018 Sep;8(9):1112-1129. doi: 10.1158/2159-8290.CD-18-0349. Epub 2018 May 31. PMID: 29853643; PMCID: PMC6125219.

Value

a list of genes associated to a specific signature.

Author(s)

Astrid Deschênes

Examples

```
## Extract the genes for the basal-like signature from Tiriác et al 2018  
getGenesInSignature(name="Tiriác_PDAC_PDO_Basal_like")
```

gsvaResults

Generic function for getting the gsvaResults slot in a class

Description

A generic function for getting the gsvaResults slot in a class. The function is implemented for the SplitTypeRdata class.

Usage

```
gsvaResults(x)
```

Arguments

x an object.

Value

an object.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(gsvaResults="character"))

# Create an instance
obj <- new("MyClass", gsvaResults="123")

# Call the generic (assuming a method is implemented)
# gsvaResults(obj)
```

gsvaResults,SplitTypeRdata-method

Getter for the signatures slot in a SplitTypeRdata class

Description

A function for getting the gsvaResults slot in a SplitTypeRdata class.

Usage

```
## S4 method for signature 'SplitTypeRdata'
gsvaResults(x)
```

Arguments

`x` a `SplitTypeRdata` object.

Value

NULL or a matrix of the GSVA results for each signature. The matrix should have one entry per signature.

Author(s)

Astrid Deschênes

Examples

```
## Create a SplitTypeRdata object
splitData <- SplitTypeRdata()

## Extract the gsvaResults slot for the object
gsvaResults(splitData)
```

`gsvaResults<-`

Generic function for replacement of gsvaResults slot in a class

Description

A generic function for replacement of `gsvaResults` slot in a S4 object.

Usage

```
gsvaResults(x) <- value
```

Arguments

`x` a S4 object.

`value` the new value to assign or update.

Value

the modified S4 object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(gsvaResults="character"))

# Create an instance
obj <- new("MyClass", gsvaResults="123")

# Call the generic (assuming a method is implemented)
# gsvaResults(obj) <- "333"
```

```
gsvaResults<- , SplitTypeRdata-method
```

Replacement of gsvaResults slot in a SplitTypeRdata object

Description

A function for replacement of the gsvaResults slot in a SplitTypeRdata class.

Usage

```
## S4 replacement method for signature 'SplitTypeRdata'
gsvaResults(x) <- value
```

Arguments

x	a SplitTypeRdata object.
value	NULL or a matrix of the GSVA results for each signature. The matrix should have one row per signature. Default: NULL.

Value

the modified SplitTypeRdata object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
## Two demo signatures
sign <- list("SignatureA"=c("EGFR", "CTTN", "ACTB"),
            "SignatureB"=c("ACTR2", "AKT1"))

## Create a SplitTypeRdata object with the signatures
demo <- new("SplitTypeRdata", signatures=sign)

## Demo study data frame
```

```

signaDemo <- matrix(data=c(0.23772165, -0.71262458, 0.42328775,
  -0.65149268, 0.18324268, 0.007158662), nrow=2, ncol=3, byrow=FALSE)
colnames(signaDemo) <- paste0("Patient_", 1:3)
rownames(signaDemo) <- c("SignatureA", "SignatureB")

gsvaResults(demo) <- signaDemo

```

ListOrNULL-class	<i>Class Union for list or NULL</i>
------------------	-------------------------------------

Description

A virtual class that groups list and NULL together.

Author(s)

Astrid Deschênes

MatrixOrNULL-class	<i>Class Union for matrix or NULL</i>
--------------------	---------------------------------------

Description

A virtual class that groups matrix and NULL together.

Author(s)

Astrid Deschênes

model	<i>Generic function for getting the model slot in a class</i>
-------	---

Description

A generic function for getting the model slot in a class. The function is implemented for the `SplitTypeRdata` class.

Usage

```
model(x)
```

Arguments

`x` an object.

Value

an object.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(model="character"))

# Create an instance
obj <- new("MyClass", model="123")

# Call the generic (assuming a method is implemented)
# model(obj)
```

model, SplitTypeRdata-method

Getter for the model slot in a SplitTypeRdata class

Description

A function for getting the model slot in a SplitTypeRdata class.

Usage

```
## S4 method for signature 'SplitTypeRdata'
model(x)
```

Arguments

x a SplitTypeRdata object.

Value

NULL or a list of the mixture models calculated for each signature. The list should have one entry per signature.

Author(s)

Astrid Deschênes

Examples

```
## Create a SplitTypeRdata object
splitData <- SplitTypeRdata()

## Extract the model slot for the object
model(splitData)
```

`model<-`*Generic function for replacement of model slot in a class*

Description

A generic function for replacement of model slot in a S4 object.

Usage

```
model(x) <- value
```

Arguments

<code>x</code>	a S4 object.
<code>value</code>	the new value to assign or update.

Value

the modified S4 object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(model="character"))

# Create an instance
obj <- new("MyClass", model="123")

# Call the generic (assuming a method is implemented)
# model(obj) <- "333"
```

`model<- ,SplitTypeRdata-method`*Replacement of model slot in a SplitTypeRdata object*

Description

A function for replacement of the model slot in a SplitTypeRdata class.

Usage

```
## S4 replacement method for signature 'SplitTypeRdata'
model(x) <- value
```

Arguments

x	a SplitTypeRdata object.
value	NULL or a list of the mixture models calculated for each signature. The list should have one entry per signature.

Value

the modified SplitTypeRdata object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
## Load required library
library(mixtools)

## Two demo signatures
sign <- list("SignatureA"=c("EGFR", "CTTN", "ACTB"),
            "SignatureB"=c("ACTR2", "AKT1"))

## Create a SplitTypeRdata object with the signatures
demo <- new("SplitTypeRdata", signatures=sign)

## Create demo model
set.seed(121)
modelExp <- normalmixEM(c(sample(1:25, size=12, replace=TRUE),
                          sample(23:45, size=12, replace=TRUE)), k=2, verb=FALSE)

## Create demo models
demoModels <- list()
demoModels[[names(sign)[1]]] <- modelExp
demoModels[[names(sign)[2]]] <- modelExp
```

```
model(demo) <- demoModels
```

permutations

Generic function for getting the permutations slot in a class

Description

A generic function for getting the permutations slot in a class. The function is implemented for the SplitTypeRdata class.

Usage

```
permutations(x)
```

Arguments

x an object.

Value

an object.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(permutations="character"))

# Create an instance
obj <- new("MyClass", permutations="123")

# Call the generic (assuming a method is implemented)
# permutations(obj)
```

`permutations, SplitTypeRdata-method`*Getter for the signatures slot in a SplitTypeRdata class*

Description

A function for getting the permutations slot in a `SplitTypeRdata` class.

Usage

```
## S4 method for signature 'SplitTypeRdata'  
permutations(x)
```

Arguments

`x` a `SplitTypeRdata` object.

Value

a list.

Author(s)

Astrid Deschênes

Examples

```
## Create a SplitTypeRdata object  
splitData <- SplitTypeRdata()  
  
## Extract the permutations slot for the object  
permutations(splitData)
```

`permutations<-`*Generic function for replacement of permutations slot in a class*

Description

A generic function for replacement of permutations slot in a S4 object.

Usage

```
permutations(x) <- value
```

Arguments

`x` a S4 object.
`value` the new value to assign or update.

Value

the modified S4 object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(permutations="character"))

# Create an instance
obj <- new("MyClass", permutations="123")

# Call the generic (assuming a method is implemented)
# permutations(obj) <- "333"
```

permutations<-, SplitTypeRdata-method

Replacement of permutations slot in a SplitTypeRdata object

Description

A function for replacement of the permutations slot in a SplitTypeRdata class.

Usage

```
## S4 replacement method for signature 'SplitTypeRdata'
permutations(x) <- value
```

Arguments

`x` a SplitTypeRdata object.
`value` NULL or a list of the permutations results for each signature. The list should have one entry per signature.

Value

the modified SplitTypeRdata object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
## Two demo signatures
sign <- list("SignatureA"=c("EGFR", "CTTN", "ACTB"),
            "SignatureB"=c("ACTR2", "AKT1"))

## Create a SplitTypeRdata object with the signatures
demo <- new("SplitTypeRdata", signatures=sign)

## Demo study data frame
demoPerm1 <- matrix(data=c(0.23772165, -0.71262458, 0.42328775,
                          -0.65149268, 0.18324268, 0.007158662), nrow=2, ncol=3, byrow=FALSE)
rownames(demoPerm1) <- paste0("Patient_", 1:2)

demoPerm2 <- matrix(data=c(0.29772165, 0.21262458, 0.42218775,
                          0.65133268, 0.18324268, -0.017158662), nrow=2, ncol=4, byrow=FALSE)
rownames(demoPerm2) <- paste0("Patient_", 1:2)
demoPerm <- list()
demoPerm[[names(sign)[1]]] <- demoPerm1
demoPerm[[names(sign)[2]]] <- demoPerm2

permutations(demo) <- demoPerm
```

plotDensityWithUpscalingSamplesBimodal

Graph both the alternative and the signature distributions obtained for a specific signature

Description

The function generates a graph showing the signature and alternative distributions obtained from the upscaling step followed by the extraction of the mixture of two normal distributions for each selected signature. The graph also contains the histogram of the enriched scores for the upscaled dataset (not the original dataset).

Usage

```
plotDensityWithUpscalingSamplesBimodal(
  x,
  signature,
  breaks = 22L,
  colorSignature = "black",
  colorAlternative = "#A9A9AD",
```

```

    colorBorderSamples = "azure4",
    colorFillingSamples = "azure3"
  )

```

Arguments

<code>x</code>	a "SplitTypeRdata" object, the output object from <code>runSubtypingfunction</code> , to be graphed.
<code>signature</code>	a character string representing the signature that will be used to create the graph. The signature must be present in the object.
<code>breaks</code>	a single integer value giving the number of cells for the histogram,
<code>colorSignature</code>	a character string representing the color of the signature distribution curve. Default: "black".
<code>colorAlternative</code>	a character string representing the color of the alternative distribution curve. Default: "#A9A9AD".
<code>colorBorderSamples</code>	a character string representing the color of the histogram border for the sample distribution. Default: "azure4".
<code>colorFillingSamples</code>	a character string representing the color of the histogram filling for the sample distribution. Default: "azure3".

Value

NULL, invisibly. The function is called for its side effect: a histogram of the values generated during the upscaling step, with the two normal distributions of the mixture model, is drawn on the current graphics device.

Author(s)

Astrid Deschênes

Examples

```

## Loading signatures
data("signaturesDemo")

## Load demo normalized expected counts for 30 patients
data("expNormalCountsDemo")

## Fix seed for reproducibility
set.seed(1221)

## Run classification on the 30 patients using 20 permutations on 75% of
## the dataset, and 10 points per patient for the up-scaling step
results <- runSubtypingBimodal(geneLists=signaturesDemo,
  expectedCountsMatrix=expNormalCountsDemo,
  permRatio=0.75, permNbr=20, upscaleNbr=5)

```

```
## Graph result
plotDensityWithUpscalingSamplesBimodal(x=results,
  signature="Classical")
```

```
plotPermutationSamplesDistribution
```

Graph showing both the alternative and the signature distributions obtained for a specific signature

Description

The function generates a graph of all the enrichment results obtained during the permutation step for the selected signature and the specified samples. The graph also includes a compact display of the continuous distribution of the values for each sample.

Usage

```
plotPermutationSamplesDistribution(
  x,
  signature,
  samples,
  violinColor = "black",
  pointColor = "black",
  positionJitter = 0.2,
  alpha = 0.25,
  size = 0.95,
  seed = NA
)
```

Arguments

x	a list of class "SplitTypeRdata", the output object from runSubtypingfunction, to be graphed.
signature	a character string representing the signature that will be used to create the graph. The signature must be present in the object.
samples	a list of character string representing the names of the samples that will be used to create the graph. The samples must be present in the object.
violinColor	a character string representing the color of the line for the kernel density distributions in the graph. Default: "black".
pointColor	a character string representing the color of the dots representing the enrichment scores for the selected samples in the graph. Default: "black".
positionJitter	a numeric representing the amount of vertical and horizontal jitter added to the position of the points on the graph. Default: 0.20.
alpha	a numeric representing the amount of the opacity of the points on the graph. If NA, the color is completely transparent. Default: 0.25.

size a numeric that represents the size of the points on the graph. Default: 0.95.

seed a integer that will be used as seed to make the jitter reproducible. If NA, the seed is initialized with a random value. Default: NA.

Value

a ggplot object that contains all the enrichment scores obtained through the permutation step with the kernel density distributions for the selected samples and the selected signature.

Author(s)

Astrid Deschênes

Examples

```
## Loading signatures
data("signaturesDemo")

## Load demo normalized expected counts for 30 patients
data("expNormalCountsDemo")

## Fix seed for reproducibility
set.seed(1221)

## Run classification on the 30 patients using 20 permutations on 75% of
## the dataset, and 10 points per patient for the up-scaling step
results <- runSubtypingBimodal(geneLists=signaturesDemo,
  expectedCountsMatrix=expNormalCountsDemo,
  permRatio=0.75, permNbr=30, upscaleNbr=5)

## Graph the enrichment results from the permutation for 3 samples
plotPermutationSamplesDistribution(x=results,
  signature="Basal-like",
  samples=c("Patient_9", "Patient_25", "Patient_29"),
  violinColor="darkred", pointColor="darkviolet", positionJitter=0.20,
  size=1.2, seed=121)
```

```
plotPermutationSamplesNormal
```

Graph the enrichment results as violin plots from the permutation step

Description

This function generates a graph showing the enrichment results, as violin plots, as obtained from the permutation step, for the selected samples.

Usage

```
plotPermutationSamplesNormal(x, signature, samples, pointColor = "darkred")
```

Arguments

x	a list of class "splitTypeResults", the output object from runSubtypingfunction, to be graphed.
signature	a character string representing the signature that will be used to create the graph. The signature must be present in the object.
samples	a list of character string representing the names of the samples that will be used to create the graph. The samples must be present in the object.
pointColor	a character string representing the color of the dots representing the enrichment scores for the sampled values, obtained during the upscaling step, in the graph. Default: "darkred".

Value

a ggplot object that contains the density distribution of the enrichment score (shown as dashed lines) from the normal distribution calculated for each selected sample. The mean of the distribution is the enrichment score for the selected sample (shown as a vertical dotted line in the graph) while the standard deviation is obtained from the permutation step. The enrichment scores values sampled at the upscaling step are shown as dots in the graph.

Author(s)

Astrid Deschênes

Examples

```
## Loading signatures
data("signaturesDemo")

## Load demo normalized expected counts for 30 patients
data("expNormalCountsDemo")

## Fix seed for reproducibility
set.seed(1221)

## Run classification on the 30 patients using 20 permutations on 75% of
## the dataset, and 10 points per patient for the up-scaling step
results <- runSubtypingBimodal(geneLists=signaturesDemo,
  expectedCountsMatrix=log2(expNormalCountsDemo + 1),
  permRatio=0.75, permNbr=30, upscaleNbr=5)

## Graph the enrichment results from the permutation for 3 samples
plotPermutationSamplesNormal(x=results,
  signature="Basal-like",
  samples=c("Patient_9", "Patient_25", "Patient_29"))
```

runSubtypingBimodal	<i>Classification of heterogeneous biological samples based on gene signature lists using a mixture of two normal distributions</i>
---------------------	---

Description

This function classifies heterogeneous biological samples based on gene signature lists, effectively isolating signature-positive samples through an automated statistical framework.

The workflow begins by calculating a Gene Set Variation Analysis (GSVA) score for each sample to quantify the relative pathway activity. A subsequent permutation step extracts sample-specific standard deviations to capture the variance. By combining these metrics, a unique normal distribution is established for each sample. Through an up-scaling step, values are randomly drawn from these individual distributions and pooled to construct a comprehensive mixture of normal distributions that represent the entire sample population. A hypothesis-testing approach in which the null hypothesis posits that a sample belongs to this alternative distribution with the lower mean is applied. Samples that successfully reject the null hypothesis are assigned to the signature classification.

Usage

```
runSubtypingBimodal(
  geneLists,
  expectedCountsMatrixLog,
  permRatio = 0.75,
  permNbr = 20,
  upscaleNbr = 10
)
```

Arguments

geneLists	a list containing signature gene lists. At least two signature gene lists must be present.
expectedCountsMatrixLog	a log2-normalized gene expression data matrix with rows corresponding to genes and columns to samples. Other data that also be used as long as it conforms with the Gaussian behavior. As examples, log2(TPM + 1), log2(CPM + 1), or vst(counts) would be acceptable.
permRatio	a numeric between 0 and 1 representing the number of samples that are retained for the permutation step. The rounded value of the number of patients multiplied by this numeric should be inferior to the total number of patients so that the permutation step is not done on the entire cohort. Default: 0.75.
permNbr	a integer, 5 or higher, representing the number of permutation sampling done. In addition, the number of permutations must be equal or inferior to the total number of unique permutations possible with the dataset considering the parameters selected by the user. Default: 20.
upscaleNbr	a integer, 2 or higher, representing the number of values taken for the normal distribution for each sample to run the up-scaling step. Default: 10.

Value

a list of class `splitTypeResults` containing the signature-specific classification for each sample as well as all the results generated at each step of the statistical workflow.

Author(s)

Astrid Deschênes

Examples

```
## Loading signatures
data("signaturesDemo")

## Load demo normalized expected counts for 30 patients
data("expNormalCountsDemo")

## Fix seed for reproducibility
set.seed(121)

## Run classification on the 30 patients using 20 permutations on 75% of
## the dataset, and 10 points per patient for the up-scaling step
results <- runSubtypingBimodal(geneLists=signaturesDemo,
  expectedCountsMatrixLog=expNormalCountsDemo,
  permRatio=0.75, permNbr=20, upscaleNbr=10)
```

`show, SplitTypeRdata-method`

Show a summary of a SplitTypeRdata object

Description

A function that displays a summary of a `SplitTypeRdata` object: the signatures with their number of genes, the number of samples, the steps of the method that have been run and, for each signature, the number of samples assigned to the signature. The function is called when the object is printed at the console.

Usage

```
## S4 method for signature 'SplitTypeRdata'
show(object)
```

Arguments

`object` a `SplitTypeRdata` object.

Value

NULL, invisibly. The function is called for its side effect: the summary of the object is displayed at the console.

Author(s)

Astrid Deschênes

Examples

```
## Two demo signatures
sign <- list("SignatureA"=c("EGFR", "CTTN", "ACTB"),
            "SignatureB"=c("ACTR2", "AKT1"))

## Create a SplitTypeRdata object with the signatures
demo <- SplitTypeRdata(signatures=sign)

## Show the summary of the object
demo
```

signatures

Generic function for getting the signatures slot in a class

Description

A generic function for getting the signatures slot in a class. The function is implemented for the SplitTypeRdata class.

Usage

```
signatures(x)
```

Arguments

x an object.

Value

an object.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(signatures="character"))

# Create an instance
obj <- new("MyClass", signatures="123")

# Call the generic (assuming a method is implemented)
# signatures(obj)
```

signatures, SplitTypeRdata-method

Getter for the signatures slot in a SplitTypeRdata class

Description

A function for getting the signatures slot in a SplitTypeRdata class.

Usage

```
## S4 method for signature 'SplitTypeRdata'
signatures(x)
```

Arguments

x a SplitTypeRdata object.

Value

a list.

Author(s)

Astrid Deschênes

Examples

```
## Create a SplitTypeRdata object
splitData <- SplitTypeRdata()

## Extract the signature slot for the object
signatures(splitData)
```

signatures<-	<i>Generic function for replacement of signatures slot in a class</i>
--------------	---

Description

A generic function for replacement of signatures slot in a S4 object.

Usage

```
signatures(x) <- value
```

Arguments

x	a S4 object.
value	the new value to assign or update.

Value

the modified S4 object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(signatures="character"))

# Create an instance
obj <- new("MyClass", signatures="123")

# Call the generic (assuming a method is implemented)
# signatures(obj) <- "333"
```

signatures<-,SplitTypeRdata-method	<i>Replacement of signatures slot in a SplitTypeRdata object</i>
------------------------------------	--

Description

A function for replacement of the signatures slot in a SplitTypeRdata class.

Usage

```
## S4 replacement method for signature 'SplitTypeRdata'
signatures(x) <- value
```

Arguments

x	a SplitTypeRdata object.
value	NULL or a list of the genes for each signature. The list should have one entry per signature.

Value

the modified SplitTypeRdata object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
## Create a SplitTypeRdata object
paramDemo <- new("SplitTypeRdata")

## Demo study data frame
signaDemo <- list("SignatureA"=c("KRAS", "AKT1", "MTOR"),
  "SignatureB"=c("PIK3CB", "CTTN"))

## Assign the new list to the signatures slot in the object
signatures(paramDemo) <- signaDemo
```

signaturesDemo

Demonstration gene signatures for PDAC subtyping

Description

A short collection of published gene signatures that primarily focuses on PDAC classification for patient-derived organoids (PDO). While the signatures are accurate, this list is mainly compiled for demonstration purposes.

Usage

```
data(signaturesDemo)
```

Format

a list containing one entry per signature:

- "Classical": a vector of gene names associated to the PDAC Patient-derived Organoid (PDO) Classical signature as published in Tiriac et al 2018
- "Basal-like": a vector of gene names associated to the PDAC Patient-derived Organoid (PDO) Basal-like signature as published in Tiriac et al 2018

Details

The PDAC patient-derived organoids (PDO) classical and basal-like signatures are associated with this publication:

Tiriac et al. Organoid Profiling Identifies Common Responders to Chemotherapy in Pancreatic Cancer. Cancer Discov. 2018 Sep;8(9):1112-1129. doi: 10.1158/2159-8290.CD-18-0349. Epub 2018 May 31. PMID: 29853643; PMCID: PMC6125219.

Source

Tiriac H, Belleau P, Engle DD, et al. Organoid Profiling Identifies Common Responders to Chemotherapy in Pancreatic Cancer. Cancer Discov. 2018;8(9):1112-1129. doi:10.1158/2159-8290.CD-18-0349

See Also

- [runSubtypingBimodal](#) for the classification of heterogeneous biological samples based on gene signature lists using a mixture of two normal distributions
- [getGeneSignaturesNames](#) for the names of the available and ready-to-use gene signature lists
- [getGeneSignatures](#) for the available and ready-to-use gene signature lists

Examples

```
## Load the published signature gene list
data(signaturesDemo)
```

SplitTypeRdata

Create a SplitTypeRdata object

Description

This function creates a SplitTypeRdata object using the values present in the parameters.

Usage

```
SplitTypeRdata(
  signatures = NULL,
  gsvaResults = NULL,
  permutations = NULL,
  standardDeviation = NULL,
  upscaling = NULL,
  model = NULL,
  threshold = NULL,
  classification = NULL
)
```

Arguments

signatures	NULL or a list of the genes for each signature. The list should have one entry per signature. Default: NULL.
gsvaResults	NULL or a matrix of the GSEA results for each signature. The matrix should have one row per signature. Default: NULL.
permutations	NULL or a list of the permutations results for each signature. The list should have one entry per signature. Default: NULL.
standardDeviation	NULL or a list of the standard deviation results for each signature. The list should have one entry per signature. Default: NULL.
upscaling	NULL or a list of the upscaling data for each signature. The list should have one entry per signature. Default: NULL.
model	NULL or a list of the mixture models calculated for each signature. The list should have one entry per signature. Default: NULL.
threshold	NULL or a list of the classification thresholds for each signature. The list should have one entry per signature. Default: NULL.
classification	NULL or a list of the classification for each signature. The list should have one entry per signature. Default: NULL.

Value

an object of class `SplitTypeRdata` that contains all the required parameters needed by the RAIDS workflow.

Author(s)

Astrid Deschênes

Examples

```
## New object of class "SplitTypeRdata" with default parameters
newSplitData1 <- SplitTypeRdata()

## New object of class "SplitTypeRdata" with non-default parameters
```

```
newSplitData2 <- SplitTypeRdata(signatures=list("Signature1"=c("ABL1",
  "BLM", "BRCA1"), "Signature2"=c("COP1", "RAD50", "FANCD2", "TERF2")),
  gsvaResults=NULL, permutations=NULL,
  standardDeviation=NULL, upscaling=NULL, model=NULL, threshold=NULL,
  classification=NULL)
```

SplitTypeRdata-class *An S4 class to represent the splitTypeR data*

Description

An S4 class to represent the splitTypeR data

Slots

signatures NULL or a list of the genes for each signature. The list should have one entry per signature. Default: NULL.

gsvaResults NULL or a numeric matrix with the sample GSVA results for each signature. The matrix should have one row per signature. Default: NULL.

permutations NULL or a list of the permutations results for each signature. The list should have one entry per signature. Default: NULL.

standardDeviation NULL or a list of the sample standard deviation for each signature. The list should have one entry per signature. Default: NULL.

upscaling NULL or a list of the upscaling data for each signature. The list should have one entry per signature. Default: NULL.

model NULL or a list of the mixture models calculated for each signature. The list should have one entry per signature. Default: NULL.

threshold NULL or a list of the classification thresholds for each signature. The threshold corresponds to the 95th percentile of the "low-mean" distribution of the mixture model. The list should have one entry per signature. Default: NULL.

classification NULL or a list of the classification for each signature. The list should have one entry per signature. Default: NULL.

Author(s)

Astrid Deschênes

standardDeviation	<i>Generic function for getting the standardDeviation slot in a class</i>
-------------------	---

Description

A generic function for getting the standardDeviation slot in a class. The function is implemented for the SplitTypeRdata class.

Usage

```
standardDeviation(x)
```

Arguments

x an object.

Value

an object.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(standardDeviation="character"))

# Create an instance
obj <- new("MyClass", standardDeviation="123")
```

standardDeviation, SplitTypeRdata-method	<i>Getter for the standardDeviation slot in a SplitTypeRdata class</i>
--	--

Description

A function for getting the standardDeviation slot in a SplitTypeRdata class.

Usage

```
## S4 method for signature 'SplitTypeRdata'
standardDeviation(x)
```

Arguments

`x` a `SplitTypeRdata` object.

Value

NULL or a list of the sample standard deviation for each signature. The list should have one entry per signature.

Author(s)

Astrid Deschênes

Examples

```
## Create a SplitTypeRdata object
splitData <- SplitTypeRdata()

## Extract the sd slot for the object
standardDeviation(splitData)
```

`standardDeviation<-` *Generic function for replacement of standardDeviation slot in a class*

Description

A generic function for replacement of `standardDeviation` slot in a S4 object.

Usage

```
standardDeviation(x) <- value
```

Arguments

`x` a S4 object.

`value` the new value to assign or update.

Value

the modified S4 object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(standardDeviation="character"))

# Create an instance
obj <- new("MyClass", standardDeviation="123")

# Call the generic (assuming a method is implemented)
# standardDeviation(obj) <- "333"
```

```
standardDeviation<-,SplitTypeRdata-method
```

Replacement of the standardDeviation slot in a SplitTypeRdata object

Description

A function for replacement of the standardDeviation slot in a SplitTypeRdata class.

Usage

```
## S4 replacement method for signature 'SplitTypeRdata'
standardDeviation(x) <- value
```

Arguments

x	a SplitTypeRdata object.
value	NULL or a list of the sample standard deviations for each signature. The list should have one entry per signature. Each entry is a named numeric vector containing the standard deviation of the GSVA scores obtained for each sample during the permutation step.

Value

the modified SplitTypeRdata object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
## Two demo signatures
sign <- list("SignatureA"=c("EGFR", "CTTN", "ACTB"),
            "SignatureB"=c("ACTR2", "AKT1"))

## Create a SplitTypeRdata object with the signatures
```

```
demo <- new("SplitTypeRdata", signatures=sign)

## Create standard deviations
demoSD <- list()
demoSD[[names(sign)[1]]] <- c(0.01748988, 0.08992500, 0.11781061, 0.0747440)
demoSD[[names(sign)[2]]] <- c(0.02058364, 0.08476766, 0.14783058, 0.1009961)

standardDeviation(demo) <- demoSD
```

threshold

Generic function for getting the threshold slot in a class

Description

A generic function for getting the threshold slot in a class. The function is implemented for the SplitTypeRdata class.

Usage

```
threshold(x)
```

Arguments

x an object.

Value

an object.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(threshold="character"))

# Create an instance
obj <- new("MyClass", threshold="123")

# Call the generic (assuming a method is implemented)
# threshold(obj)
```

threshold, SplitTypeRdata-method

Getter for the threshold slot in a SplitTypeRdata class

Description

A function for getting the threshold slot in a SplitTypeRdata class.

Usage

```
## S4 method for signature 'SplitTypeRdata'
threshold(x)
```

Arguments

x a SplitTypeRdata object.

Value

NULL or a list of the classification thresholds for each signature. The threshold corresponds to the 95th percentile of the "low-mean" distribution of the mixture model. The list should have one entry per signature.

Author(s)

Astrid Deschênes

Examples

```
## Create a SplitTypeRdata object
splitData <- SplitTypeRdata()

## Extract the threshold slot for the object
threshold(splitData)
```

threshold<-

Generic function for replacement of threshold slot in a class

Description

A generic function for replacement of threshold slot in a S4 object.

Usage

```
threshold(x) <- value
```

Arguments

x	a S4 object.
value	the new value to assign or update.

Value

the modified S4 object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(threshold="character"))

# Create an instance
obj <- new("MyClass", threshold="123")

# Call the generic (assuming a method is implemented)
# threshold(obj) <- "333"
```

```
threshold<- , SplitTypeRdata-method
```

Replacement of threshold slot in a SplitTypeRdata object

Description

A function for replacement of the threshold slot in a *SplitTypeRdata* class.

Usage

```
## S4 replacement method for signature 'SplitTypeRdata'
threshold(x) <- value
```

Arguments

x	a <i>SplitTypeRdata</i> object.
value	NULL or a list of the classification thresholds for each signature. The list should have one entry per signature.

Value

the modified *SplitTypeRdata* object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
## Two demo signatures
sign <- list("SignatureA"=c("EGFR", "CTTN", "ACTB"),
            "SignatureB"=c("ACTR2", "AKT1"))

## Create a SplitTypeRdata object with the signatures
demo <- new("SplitTypeRdata", signatures=sign)

## Assign a classification threshold to each signature
threshold(demo) <- list("SignatureA"=0.12, "SignatureB"=-0.05)
```

upscaling

*Generic function for getting the upscaling slot in a class***Description**

A generic function for getting the upscaling slot in a class. The function is implemented for the SplitTypeRdata class.

Usage

```
upscaling(x)
```

Arguments

x an object.

Value

an object.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(upscaling="character"))

# Create an instance
obj <- new("MyClass", upscaling="123")

# Call the generic (assuming a method is implemented)
```

```
# upscaling(obj)
```

```
upscaling, SplitTypeRdata-method
```

Getter for the upscaling slot in a SplitTypeRdata class

Description

A function for getting the upscaling slot in a SplitTypeRdata class.

Usage

```
## S4 method for signature 'SplitTypeRdata'
upscaling(x)
```

Arguments

x a SplitTypeRdata object.

Value

NULL or a list of the upscaling data for each signature. The list should have one entry per signature.

Author(s)

Astrid Deschênes

Examples

```
## Create a SplitTypeRdata object
splitData <- SplitTypeRdata()

## Extract the upscaling slot for the object
upscaling(splitData)
```

upscaling<-	<i>Generic function for replacement of the upscaling slot in a class</i>
-------------	--

Description

A generic function for replacement of the upscaling slot in a S4 object.

Usage

```
upscaling(x) <- value
```

Arguments

x	a S4 object.
value	the new value to assign or update.

Value

the modified S4 object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
# Define a dummy class to show usage
setClass("MyClass", slots = list(upscaling="character"))

# Create an instance
obj <- new("MyClass", upscaling="123")

# Call the generic (assuming a method is implemented)
# upscaling(obj) <- "333"
```

upscaling<- , SplitTypeRdata-method	<i>Replacement of the upscaling slot in a SplitTypeRdata object</i>
-------------------------------------	---

Description

A function for replacement of the upscaling slot in a SplitTypeRdata class.

Usage

```
## S4 replacement method for signature 'SplitTypeRdata'  
upscaling(x) <- value
```

Arguments

x	a SplitTypeRdata object.
value	NULL or a list of the upscaling data for each signature. The list should have one entry per signature.

Value

the modified SplitTypeRdata object when the new value is valid.

Author(s)

Astrid Deschênes

Examples

```
## Two demo signatures  
sign <- list("SignatureA"=c("EGFR", "CTTN", "ACTB"),  
            "SignatureB"=c("ACTR2", "AKT1"))  
  
## Create a SplitTypeRdata object with the signatures  
demo <- new("SplitTypeRdata", signatures=sign)  
  
## Create demo upscaling data  
demoUp <- list()  
demoUp[[names(sign)[1]]] <- matrix(data=c(0.0174988, 0.0892500, 0.1178101,  
0.0744401, 0.0654692, 0.2546492), nrow=2, ncol=3, byrow=FALSE)  
demoUp[[names(sign)[2]]] <- matrix(data=c(0.3749488, 0.4894500, 0.2118101,  
0.4444401, 0.4154692, 0.8543392), nrow=2, ncol=3, byrow=FALSE)  
  
upscaling(demo) <- demoUp
```

Index

- * **classes**
 - SplitTypeRdata-class, [34](#)
- * **datasets**
 - expNormalCountsDemo, [7](#)
 - signaturesDemo, [31](#)
- * **package**
 - splitTypeR-package, [3](#)
- classification, [4](#)
- classification, SplitTypeRdata-method, [5](#)
- classification<-, [5](#)
- classification<-, SplitTypeRdata-method, [6](#)
- expNormalCountsDemo, [7](#)
- getGeneSignatures, [4](#), [8](#), [8](#), [32](#)
- getGeneSignaturesNames, [4](#), [8](#), [9](#), [32](#)
- getGenesInSignature, [10](#)
- gsvaResults, [11](#)
- gsvaResults, SplitTypeRdata-method, [11](#)
- gsvaResults<-, [12](#)
- gsvaResults<-, SplitTypeRdata-method, [13](#)
- ListOrNULL-class, [14](#)
- MatrixOrNULL-class, [14](#)
- model, [14](#)
- model, SplitTypeRdata-method, [15](#)
- model<-, [16](#)
- model<-, SplitTypeRdata-method, [17](#)
- permutations, [18](#)
- permutations, SplitTypeRdata-method, [19](#)
- permutations<-, [19](#)
- permutations<-, SplitTypeRdata-method, [20](#)
- plotDensityWithUpscalingSamplesBimodal, [21](#)
- plotPermutationSamplesDistribution, [23](#)
- plotPermutationSamplesNormal, [24](#)
- runSubtypingBimodal, [4](#), [8](#), [26](#), [32](#)
- show, SplitTypeRdata-method, [27](#)
- signatures, [28](#)
- signatures, SplitTypeRdata-method, [29](#)
- signatures<-, [30](#)
- signatures<-, SplitTypeRdata-method, [30](#)
- signaturesDemo, [31](#)
- splitTypeR (splitTypeR-package), [3](#)
- splitTypeR-package, [3](#)
- SplitTypeRdata, [32](#)
- SplitTypeRdata-class, [34](#)
- standardDeviation, [35](#)
- standardDeviation, SplitTypeRdata-method, [35](#)
- standardDeviation<-, [36](#)
- standardDeviation<-, SplitTypeRdata-method, [37](#)
- threshold, [38](#)
- threshold, SplitTypeRdata-method, [39](#)
- threshold<-, [39](#)
- threshold<-, SplitTypeRdata-method, [40](#)
- upsampling, [41](#)
- upsampling, SplitTypeRdata-method, [42](#)
- upsampling<-, [43](#)
- upsampling<-, SplitTypeRdata-method, [43](#)