

Package ‘STROMA4’

October 10, 2021

Version 1.16.0

Date 2017-03-23

Title Assign Properties to TNBC Patients

Description This package estimates four stromal properties identified in TNBC patients in each patient of a gene expression datasets. These stromal property assignments can be combined to subtype patients. These four stromal properties were identified in Triple negative breast cancer (TNBC) patients and represent the presence of different cells in the stroma: T-cells (T), B-cells (B), stromal infiltrating epithelial cells (E), and desmoplasia (D). Additionally this package can also be used to estimate generative properties for the Lehmann subtypes, an alternative TNBC subtyping scheme (PMID: 21633166).

Depends R (>= 3.4), Biobase, BiocParallel, cluster, matrixStats, stats, graphics, utils

License GPL-3

LazyData true

RoxygenNote 5.0.1

Suggests breastCancerMAINZ

biocViews ImmunoOncology, GeneExpression, BiomedicalInformatics, Classification, Microarray, RNASeq, Software

git_url <https://git.bioconductor.org/packages/STROMA4>

git_branch RELEASE_3_13

git_last_commit 2107780

git_last_commit_date 2021-05-19

Date/Publication 2021-10-10

Author Sadiq Saleh [aut, cre],
Michael Hallett [aut]

Maintainer Sadiq Saleh <sadiq.mehdiismailsaleh@mail.mcgill.ca>

R topics documented:

assign.properties	2
B.stroma.property	3
BL1.property	4
BL2.property	4
D.stroma.property	5
E.stroma.property	6
IM.property	6
LAR.property	7
M.property	8
MSL.property	8
T.stroma.property	9
Index	10

assign.properties	<i>Function to assign properties to an expression matrix</i>
-------------------	--

Description

assign.properties is a function to assign stromal property and Lehmann generative property levels to a TNBC dataset. Given an expressionset with a gene expression matrix, and a list of HGNC IDs, this function will estimate the status of samples for either the Lehmann generative properties, the stromal properties, or both at the same time. (Saleh et al. , under review at Cancer Research)

Usage

```
assign.properties(ESet, geneID.column = 1, genelists = c("Stroma4",
  "TNBCType"), n = 1000, seed = 123456, mc.cores = snowWorkers(),
  var.method = function(x) rowIQRs(x, na.rm = TRUE))
```

Arguments

ESet	An ExpressionSet object. Rows correspond to genes, columns to samples. If there are genes with multiple probes, they will be collapsed to a single row using the function defined in var.method.
geneID.column	Integer or column name corresponding to column in featureData table corresponding to HGNC ID.
genelists	A vector with either Stroma4, TNBCType, or both to specify which genelists to use.
n	An integer value specifying the number of random samples to generate.
seed	An integer value specifying a seed for the random ranks function. Default value is 123456.
mc.cores	An integer specifying how many cores to use. Defaults to use the function snowWorkers().
var.method	Function for assessing variance to collapse probes. Default is IQR

Details

We defined a method for estimating stromal and Lehmann generative properties in a gene expression dataset (either LCM stroma or whole tumor breast cancer data). The `assign.properties` function extracts the expression data from the expression set using the `exprs()` function. The HGNC ID annotations for the dataset are required as this is what the function uses to identify genes to estimate property levels. The HGNC IDs should be in the `featureData` of the `expressionset`, and the corresponding column ID should be passed to the `geneID.column` argument. The function can be used to estimate the stromal properties, the Lehmann generative properties or both simultaneously depending on whether 'STROMA4' (stromal properties), 'TNBCTYPE' (Lehmann generative properties), or both are passed to the `genelists` argument. This function will then return property assignments.

Value

The function returns a the `ExpressionSet` with the properties added to the `pData` table.

Examples

```
library(breastCancerMAINZ)
data(mainz, package='breastCancerMAINZ')
all.properties <- assign.properties(ESet=mainz, geneID.column='Gene.symbol',
genelists=c('Stroma4', 'TNBCTYPE'), n=10)
```

B.stroma.property	<i>Genelist for B.stroma.property</i>
-------------------	---------------------------------------

Description

This genelist is used to assign the related property using the `assign.property` function

Usage

```
data('B.stroma.property')
```

Format

An object of class `data.frame` with 567 rows and 2 columns.

- `Gene.Symbol`: ID of transcript (HGNC where available)
- `Direction`: Direction of expression (up/down)

Value

An object of class `data.frame` with 567 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('B.stroma.property')
```

BL1.property

Genelist for BL1.property

Description

This genelist is used to assign the related property using the assign.property function

Usage

```
data('BL1.property')
```

Format

An object of class data.frame with 473 rows and 2 columns.

- Gene.Symbol: ID of transcript (HGNC where available)
- Direction: Direction of expression (up/down)

Value

An object of class data.frame with 473 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('BL1.property')
```

BL2.property

Genelist for BL2.property

Description

This genelist is used to assign the related property using the assign.property function

Usage

```
data('BL2.property')
```

Format

An object of class `data.frame` with 291 rows and 2 columns.

- Gene.Symbol: ID of transcript (HGNC where available)
- Direction: Direction of expression (up/down)

Value

An object of class `data.frame` with 291 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('BL2.property')
```

<code>D.stroma.property</code>	<i>Genelist for D.stroma.property</i>
--------------------------------	---------------------------------------

Description

This genelist is used to assign the related property using the `assign.property` function

Usage

```
data('D.stroma.property')
```

Format

An object of class `data.frame` with 297 rows and 2 columns.

- Gene.Symbol: ID of transcript (HGNC where available)
- Direction: Direction of expression (up/down)

Value

An object of class `data.frame` with 297 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('D.stroma.property')
```

E.stroma.property	<i>Genelist for E.stroma.property</i>
-------------------	---------------------------------------

Description

This genelist is used to assign the related property using the assign.property function

Usage

```
data('E.stroma.property')
```

Format

An object of class data.frame with 27 rows and 2 columns.

- Gene.Symbol: ID of transcript (HGNC where available)
- Direction: Direction of expression (up/down)

Value

An object of class data.frame with 27 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('E.stroma.property')
```

IM.property	<i>Genelist for IM.property</i>
-------------	---------------------------------

Description

This genelist is used to assign the related property using the assign.property function

Usage

```
data('IM.property')
```

Format

An object of class data.frame with 705 rows and 2 columns.

- Gene.Symbol: ID of transcript (HGNC where available)
- Direction: Direction of expression (up/down)

Value

An object of class `data.frame` with 705 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('IM.property')
```

LAR.property

Genelist for LAR.property

Description

This genelist is used to assign the related property using the `assign.property` function

Usage

```
data('LAR.property')
```

Format

An object of class `data.frame` with 1205 rows and 2 columns.

- Gene.Symbol: ID of transcript (HGNC where available)
- Direction: Direction of expression (up/down)

Value

An object of class `data.frame` with 1205 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('LAR.property')
```

M.property	<i>Genelist for M.property</i>
------------	--------------------------------

Description

This genelist is used to assign the related property using the assign.property function

Usage

```
data('M.property')
```

Format

An object of class data.frame with 710 rows and 2 columns.

- Gene.Symbol: ID of transcript (HGNC where available)
- Direction: Direction of expression (up/down)

Value

An object of class data.frame with 710 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('M.property')
```

MSL.property	<i>Genelist for MSL.property</i>
--------------	----------------------------------

Description

This genelist is used to assign the related property using the assign.property function

Usage

```
data('MSL.property')
```

Format

An object of class data.frame with 1105 rows and 2 columns.

- Gene.Symbol: ID of transcript (HGNC where available)
- Direction: Direction of expression (up/down)

Value

An object of class `data.frame` with 1105 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('MSL.property')
```

T.stroma.property	<i>Genelist for T.stroma.property</i>
-------------------	---------------------------------------

Description

This genelist is used to assign the related property using the `assign.property` function

Usage

```
data('T.stroma.property')
```

Format

An object of class `data.frame` with 458 rows and 2 columns.

- Gene.Symbol: ID of transcript (HGNC where available)
- Direction: Direction of expression (up/down)

Value

An object of class `data.frame` with 458 rows and 2 columns.

Source

Identified in related publication

Examples

```
data('T.stroma.property')
```

Index

* List

- B.stroma.property, [3](#)
- BL1.property, [4](#)
- BL2.property, [4](#)
- D.stroma.property, [5](#)
- E.stroma.property, [6](#)
- IM.property, [6](#)
- LAR.property, [7](#)
- M.property, [8](#)
- MSL.property, [8](#)
- T.stroma.property, [9](#)

assign.properties, [2](#)

B.stroma.property, [3](#)
BL1.property, [4](#)
BL2.property, [4](#)

D.stroma.property, [5](#)

E.stroma.property, [6](#)

IM.property, [6](#)

LAR.property, [7](#)

M.property, [8](#)
MSL.property, [8](#)

T.stroma.property, [9](#)