Package ‘geneClassifiers’

April 3, 2024

Type  Package
Title  Application of gene classifiers
Version  1.26.0
Description  This package aims for easy accessible application of classifiers which have been published in literature using an ExpressionSet as input.

URL  https://doi.org/10.18129/B9.bioc.geneClassifiers
BugReports  https://github.com/rkuiper/geneClassifiers/issues
License  GPL-2
biocViews  GeneExpression, BiomedicalInformatics, Classification, Survival, Microarray
LazyData  true
Suggests  testthat
Depends  R (>= 3.6.0)
Imports  utils, methods, stats, Biobase, BiocGenerics
         Class_FixedExpressionData.R Class_ClassifierParameters.R
         Class_ClassifierResults.R zzz.R geneClassifiers.R
RoxygenNote  6.1.1
Encoding  UTF-8

Author  R Kuiper [cre, aut] (<https://orcid.org/0000-0002-3703-1762>)
Maintainer  R Kuiper <r.kuiper.emc@gmail.com>
Description

This package aims for easy accessible application of classifiers which have been published in literature using an ExpressionSet as input.

Details

Combining gene expression profiling data with survival data has led to the development of robust outcome predictors (gene classifiers). This package provides a method for running gene classifiers generating patient specific predictive outcomes. This package is intended to support and enable research. To yield stable results, this package requires a dataset of at least 20 patients.

For detail on how to use this package see the vignette: vignette("geneClassifiers")
Author(s)

Maintainer: R Kuiper <r.kuiper.emc@gmail.com> (0000-0002-3703-1762)

References


See Also

Useful links:
- Report bugs at https://github.com/rkuiper/geneClassifiers/issues

ClassifierParameters

An S4 class to store classifier parameters.

Description

This class stores classifier related information. This is information on probe-sets used and their weightings, means, standard deviations and covariance structure as observed in the classifiers training data, and the description of the procedure on how to preprocess new data prior to application of the classifier.

Slots

name A character string indicating the name of the classifier
description A short description of the classifier
 citations A character vector of citations to literature
 normalizationMethod A character string indicating the normalization method to apply
eventChain A list of preprocessing steps
 probeNames A character vector
 intercept A numeric value
 weights A numeric vector
decisionBoundaries A numeric vector with values that separate the risk-groups
 doRun A function which is called for the actual classification
 means A numeric vector of probe-set means as observed in the trainingset (if available)
sds A numeric vector of probe-set standard deviations as observed in the trainingset (if available)
geneClassifierVersion An object of class package_version
ClassifierResults

An S4 class to store classifier results.

Description

This class stores classifier results as obtained after running the `runClassifier` function.

Slots

- `classifierParameters`: An object of class `ClassifierParameters` in which the applied classifier parameters are stored.
- `score`: A numeric vector of resulting classifier scores.
- `batchCorrection`: A character vector indicating whether batch correction was applied.
- `weightingType`: A character string indicating whether the weighting type was complete (i.e., no missing data), reweighted (i.e., missing data was handled based on correction using the covariance structure in the classifiers training data), or reduced (i.e., missing data but not reweighting the original probe set weighting).

---

dim,FixedExpressionData-method

**Dimensions of an Object**

Description

Retrieve the dimension of an object.

Usage

```r
## S4 method for signature 'FixedExpressionData'
dim(x)
```

Arguments

- `x`: an R object, for example a matrix, array or data frame.

Value

Retrieves the ‘dim attribute of the object. It is ‘NULL’ or a vector of mode ‘integer’.

See Also

Other fixed data information extraction functions: `[,FixedExpressionData,ANY,ANY-method`, `getNormalizationMethod`, `getTargetValue`
exampleMAS5

Examples

data(exampleMAS5)
myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
dim(myData)
dim(myData[1:10,1:3])

---

Example MAS5.0 ExpressionSet

Description

An ExpressionSet. The data contains a sample of gene expression data from patients included in the HOVON65/GMMG-HD4 trial on multiple myeloma. The data was MAS5.0 normalized to a target value of 500.

Usage

eexampleMAS5

Format

An object of class ExpressionSet with 374 rows and 25 columns.

---

FixedExpressionData An S4 class to store classifier parameters.

Description

This class stores gene expression data together with information on the normalization method and additional normalization related parameters. In order to ensure the data is not manipulated in unforeseen ways, manipulation is strictly controlled through adding transformations which are predefined in the TransformationProcess-class. Upon reading the data by the exprs function, the transformations are performed in the order they were added.

Slots

normalizationMethod A character string indicating the normalization method that was applied to the data. Possible values are given by getNormalizationMethods.
expressionEnvironment A locked environment in which the expression matrix is stored.
normalizationParameters A list with normalization specific values.
transformationProcess A locked environment to which the transformation processes are added.
geneClassifierVersion An object of class package_version
getBatchCorrection  
*Obtain the batch correction status for a classifier result.*

**Description**

getBatchCorrection returns TRUE or FALSE indicating whether correction was applied.

**Usage**

```r
getBatchCorrection(object)
```

```r
## S4 method for signature 'ClassifierResults'
getBatchCorrection(object)
```

**Arguments**

- `object`  
  An object of class `ClassifierResults` as returned by `runClassifier`.

**Value**

TRUE or FALSE

**See Also**

Other classifier results: `getClassifications`, `getScores`, `getWeightingType`.

**Examples**

```r
myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
results <- runClassifier('EMC92', myData)
getBatchCorrection( results )
```

getcitations  
*Obtain citations to the classifier*

**Description**

getCitations Obtain citations to the classifier.

**Usage**

```r
getcitations(object)
```

```r
## S4 method for signature 'ClassifierParameters'
getcitations(object)
```
getClassifications

Arguments

object

An object of class ClassifierParameters as returned by getClassifier

Value

A character vector

See Also

Other classifier information functions: getClassifier, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getSds, getTrainingData, getWeights

Examples

aClassifier <- getClassifier("EMC92")
getcitations(aClassifier)

R> getClassifications(aClassifier)

得到分类器分类。

Description

getcitations 返回结果的分类。

Usage

getcitations(object)

## S4 method for signature 'ClassifierResults'
getcitations(object)

Arguments

object

An object of class ClassifierResults

Value

A vector of ordered factors with classifications per sample

See Also

Other classifier results: getBatchCorrection, getScores, getWeightingType

Examples

myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
results <- runClassifier('EMC92', myData)
getscores( results )
getcitations( results )
getClassifier

Obtain a classifier definition.

Description

getClassifier returns a requested classifier definition.

Usage

getClassifier(value)

## S4 method for signature 'ClassifierResults'
getClassifier(value)

## S4 method for signature 'character'
getClassifier(value)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Either a text value indicating a classifier name (see <code>showClassifierList</code>), or an object of class <code>ClassifierResults</code> as returned by the <code>runClassifier</code> function.</td>
</tr>
</tbody>
</table>

Value

The return value is a classifier definition which is encoded in an object of class `ClassifierParameters`. This can be used as input argument for the `runClassifier` function.

See Also

`ClassifierParameters` and `runClassifier`

Other classifier information functions: `getCitations, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getSds, getTrainingData, getWeights`

Examples

getClassifier("EMC92")
getDecisionBoundaries

Obtain the decision boundaries defined for the classifier.

Description

getDecisionBoundaries returns a numeric vector of boundary values that separate the risk groups.

Usage

getDecisionBoundaries(object)

## S4 method for signature 'ClassifierParameters'
getDecisionBoundaries(object)

Arguments

object An object of class ClassifierParameters as returned by getClassifier

Value

A numeric vector

See Also

Other classifier information functions: getCitations, getClassifier, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getSds, getTrainingData, getWeights

Examples

aClassifier <- getClassifier("EMC92")
getDecisionBoundaries(aClassifier)

getDescription

Obtain classifiers' description.

Description

getDescription returns the descriptive text associated with the classifier.

Usage

getDescription(object)

## S4 method for signature 'ClassifierParameters'
getDescription(object)
getEventChain

Arguments

object An object of class ClassifierParameters as returned by getClassifier

Value

A character string describing the classifier

See Also

Other classifier information functions: getCitations, getClassifier, getDecisionBoundaries, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getSds, getTrainingData, getWeights

Examples

aClassifier <- getClassifier("EMC92")
getDescription(aClassifier)

getEventChain(object)

### S4 method for signature 'ClassifierParameters'
getEventChain(object)

Arguments

object An object of class ClassifierParameters as returned by getClassifier

Value

Returns the event chain encoded in the in the classifier encoded as a named list.

See Also

showClassifierList getClassifier runClassifier

Other classifier information functions: getCitations, getClassifier, getDecisionBoundaries, getDescription, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getSds, getTrainingData, getWeights
getIntercept

Examples

```r
aClassifier <- getClassifier("EMC92")
getEventChain(aClassifier)
```

getIntercept

Obtain classifiers' intercept.

Description

getIntercept returns the numeric value of the classifier's intercept.

Usage

```r
getIntercept(object)
```

## S4 method for signature 'ClassifierParameters'

```r
getIntercept(object)
```

Arguments

- `object`: An object of class `ClassifierParameters` as returned by `getClassifier`

Value

A numeric value

See Also

Other classifier information functions: `getCitations, getClassifier, getDecisionBoundaries, getDescription, getEventChain, getMeans, getNormalizationMethod, getProbeNames, getSds, getTrainingData, getWeights`

Examples

```r
aClassifier <- getClassifier("EMC92")
getIntercept(aClassifier)
```
getMeans

Obtain classifiers’ reference means.

Description
getMeans returns the reference means encoded in the classifier.

Usage
getMeans(object)

## S4 method for signature 'ClassifierParameters'
getMeans(object)

Arguments
object An object of class ClassifierParameters as returned by getClassifier

Value
Returns a numeric vector of probe set means as observed in the reference data

See Also
showClassifierList getClassifier runClassifier

Other classifier information functions: getCitations, getClassifier, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getNormalizationMethod, getProbeNames, getSds, getTrainingData, getWeights

Examples
aClassifier <- getClassifier(“EMC92”)
getMeans(aClassifier)

getName

Obtain object names.

Description
getName returns the name associated with the requested object.
Usage

getName(object)

## S4 method for signature 'TransformationProcess'
getName(object)

## S4 method for signature 'ClassifierParameters'
getName(object)

## S4 method for signature 'ClassifierResults'
getName(object)

Arguments

object The object to get the name of.

Value

The return value is a character string

See Also

ClassifierParameters
ClassifierResults

Examples

aClassifier <- getClassifier("EMC92")
getName(aClassifier)

d<-- getNormalizationMethod

Obtain normalization method

Description

The function `getNormalizationMethod` returns the normalization method associated with the object

`getNormalizationMethods` returns a character vector of currently available normalization methods.
Usage

getNormalizationMethod(object)

getNormalizationMethods()

### S4 method for signature 'FixedExpressionData'
getNormalizationMethod(object)

### S4 method for signature 'ClassifierParameters'
getNormalizationMethod(object)

Arguments

object An object of class `FixedExpressionData` or `ClassifierParameters`

Details

The given normalization methods can be used in the

Value

A character string indicating the normalization method.

See Also

`getNormalizationMethods`

Other classifier information functions: `getCitations, getClassifier, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getMeans, getProbeNames, getSds, getTrainingData, getWeights`

Other fixed data information extraction functions: `[,FixedExpressionData,ANY,ANY-method, dim,FixedExpressionData-method, getTargetValue`

Other workflow functions: `runClassifier, setNormalizationMethod, showClassifierList`

Examples

data(exampleMAS5)
myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
aClassifier <- getClassifier("EMC92")
getNormalizationMethod( myData )
getNormalizationMethod( aClassifier )
data(exampleMAS5)

showClassifierList()
getNormalizationMethods()

myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
results <- runClassifier("UAMS70", myData)

getScores( results )
getProbeNames

getClassifications(results)

getProbeNames
Obtain probe-set names.

Description

getProbeNames returns the probe names associated with the requested classifier.

Usage

getProbeNames(object)

## S4 method for signature 'ClassifierParameters'
getProbeNames(object)

Arguments

object An object of class ClassifierParameters as returned by getClassifier

Value

The return value is a character vector of probe-set names.

See Also

ClassifierParameters

Other classifier information functions: getCitations, getClassifier, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getSds, getTrainingData, getWeights

Examples

aClassifier <- getClassifier("EMC92")
getProbeNames(aClassifier)
**getScores**

*Obtain classifier score.*

**Description**

getScores returns the resulting scores from a classifier run.

**Usage**

```r
getScores(object)

## S4 method for signature 'ClassifierResults'
getScores(object)
```

**Arguments**

- `object` An object of class `ClassifierResults`

**Value**

A numeric vector with scores per sample

**See Also**

Other classifier results: `getBatchCorrection`, `getClassifications`, `getWeightingType`

**Examples**

```r
data(exampleMAS5)
myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
results <- runClassifier("EMC92", myData)
getScores(results)
getClassifications(results)
```

---

**getSds**

*Obtain classifiers’ reference standard deviations.*

**Description**

getSds returns the reference standard deviations encoded in the classifier.

**Usage**

```r
getSds(object)

## S4 method for signature 'ClassifierParameters'
getSds(object)
```
**getTargetValue**

Arguments

```
object
```

An object of class `ClassifierParameters` as returned by `getClassifier`

Value

Returns a numeric vector of probe set standard deviations as observed in the reference data

See Also

```
showClassifierList getClassifier runClassifier
```

Other classifier information functions: `getCitations, getClassifier, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getTrainingData, getWeights`

Examples

```
aClassifier <- getClassifier("EMC92")
getSds(aClassifier)
```

---

**Description**

`getTargetValue` returns the current applied targetValue in the MAS5.0 gene expression data.

**Usage**

```
getTargetValue(object)
```

## S4 method for signature 'FixedExpressionData'
```
getTargetValue(object)
```

Arguments

```
object
```

An object of class `FixedExpressionData`

Value

A numeric value

See Also

Other fixed data information extraction functions: `[FixedExpressionData,ANY,ANY-method, dim,FixedExpressionData-method, getNormalizationMethod`
getTrainingData

Examples

data(exampleMAS5)
myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
getTargetValue( myData )

getTrainingData

Obtain classifier training data.

Description

getTrainingData returns the training data that was used for building the classifier.

Usage

getTrainingData(object)

## S4 method for signature 'ClassifierParameters'
getTrainingData(object)

Arguments

object An object of class ClassifierParameters as returned by getClassifier

Value

An object of class ExpressionSet

See Also

Other classifier information functions: getCitations, getClassifier, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getSds, getWeights

Examples

aClassifier <- getClassifier("EMC92")
getTrainingData(aClassifier)
getWeightingType

Obtain the weighting type used to obtain a classifier result.

Description

getWeightingType returns weighting type
getWeightingTypes returns weighting type

Usage

gWeightingType(object)
gWeightingTypes()

## S4 method for signature 'ClassifierResults'
gWeightingType(object)

Arguments

object       An object of class ClassifierResults as returned by runClassifier

Value

one of the values in getWeightingTypes()
either "complete" or "reweighted"

See Also

Other classifier results: getBatchCorrection, getClassifications, getScores

Examples

myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
results <- runClassifier('EMC92', myData)
gWeightingType( results )
gWeightingTypes()
getWeights

**Obtain classifier weights.**

Description

getWeights returns the probe weights associated with the classifier.

Usage

getWeights(object)

## S4 method for signature 'ClassifierParameters'
getWeights(object)

Arguments

object An object of class ClassifierParameters as returned by getClassifier

Value

A numeric vector.

See Also

Other classifier information functions: getCitations, getClassifier, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getSds, getTrainingData

Examples

aClassifier <- getClassifier("EMC92")
getWeights(aClassifier)

runClassifier

**Perform classification.**

Description

runClassifier performs classification by applying a classifier to gene expression data.
runClassifier

Usage

runClassifier(classifierParameters, fixedExpressionData, ...)

## S4 method for signature 'character,FixedExpressionData'
runClassifier(classifierParameters, fixedExpressionData, ...)

## S4 method for signature 'ClassifierParameters,FixedExpressionData'
runClassifier(classifierParameters, fixedExpressionData, ...)

Arguments

classifierParameters
Either a text value indicating a classifier name (see showClassifierList), or an object of class ClassifierParameters as returned by the getClassifier function.

fixedExpressionData
The data to be classified in the form of a FixedExpressionData object as returned by the setNormalizationMethod function.

... see details

Details

A list of possible classifiers is obtained by showClassifierList. The data to be classified is first to be processed by the setNormalizationMethod function. By default the data is assumed to contain many (n>=25) samples with corresponding probe-sets needed for classification. If one of these conditions is not met, a classifier outcome might be seriously affected. By default an error is given. Although strongly discouraged, it is possible to circumvent the security checks. If not all required probe-sets are included in the input set, you can explicitly pass the parameter allow.reweighted = TRUE to the runClassifier function in order to determine the classifier outcome using less probe-sets (e.g. possible if the missing probe-sets are known to have minimal contribution). See vignette("MissingCovariates") for more information. If the input data has a small number of samples, the default batch correction becomes ineffective. If you are aware of the possible negative effects you can force to not use batch correction by passing the parameter do.batchcorrection=FALSE.

Value

The classification results as an object of class ClassifierResults.

See Also

Other workflow functions: getNormalizationMethod, setNormalizationMethod, showClassifierList
**Examples**

```r
data(exampleMAS5)
myData<-setNormalizationMethod(exampleMAS5,"MAS5.0",targetValue=500)
runClassifier("EMC92",myData)
```

---

**Description**

`setNormalizationMethod` is to be called prior to running a classifier.

**Usage**

```r
setNormalizationMethod(expressionSet, method, ...)
```

**Arguments**

- `expressionSet` An object of class `ExpressionSet` containing the gene expression data.
- `method` A character string indicating the normalization that was applied to the data. Possible values are given by `getNormalizationMethods()`.
- `...` see details.

**Details**

The `FixedExpressionData` class forms together with the `ClassifierParameters` class the basis for input to the `runClassifier` function. The data inside the `FixedExpressionData`-class has to be stored as it is right after normalization. This function may require some additional arguments:

- `isLog2Transformed = TRUE` Use this argument if the data already underwent a log2transformation, as is common e.g. in case of MAS5.0 normalization.
- `targetValue = value` This is a MAS5.0 specific argument. It is the sample intensity mean when the lowest and highest 2% of intensities are discarded. If only part of the original expression set is given to this function, then this argument is required.

**Value**

An object of class `FixedExpressionData`

**See Also**

Other workflow functions: `getNormalizationMethod`, `runClassifier`, `showClassifierList`
showClassifierList

Examples

data(exampleMAS5)
myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
results <- runClassifier('EMC92', myData)
getScores( results )
getClassifications( results )

showClassifierList

Show classifier names and descriptions.

Description

showClassifierList gives a data.frame of all implemented classifiers.

Usage

showClassifierList(normalizations)

Arguments

normalizations an optional text argument of one or more normalization methods in order to filter the classifiers to be shown.

Details

The names of the classifiers shown can be used as input for the runClassifier function and the getClassifier function.

Value

A data.frame with columns: "name","normalizationMethod" and "description"

See Also

Other workflow functions: getNormalizationMethod, runClassifier, setNormalizationMethod

Examples

showClassifierList()
data(exampleMAS5)
myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
results <- runClassifier('UAMS70', myData)
getScores( results )
getClassifications( results )
Description

Extract Parts of an Object

Usage

```r
## S4 method for signature 'FixedExpressionData,ANY,ANY'
x[i, j, ...], drop = TRUE]

## S4 method for signature 'FixedExpressionData,ANY,missing'
x[i, j, ...], drop = TRUE]

## S4 method for signature 'FixedExpressionData,missing,ANY'
x[i, j, ...], drop = TRUE]

## S4 method for signature 'FixedExpressionData,ANY,missing'
x[[i, j, ...]]

## S4 method for signature 'FixedExpressionData,missing,ANY'
x[[i, j, ...]]

## S4 method for signature 'FixedExpressionData,ANY,ANY'
x[[i, j, ...]]
```

Arguments

- `x` An object of class `FixedExpressionData`
- `i` the rows index
- `j` the column index
- `...` unused
- `drop` unused

Value

An object of class `FixedExpressionData`

See Also

Other fixed data information extraction functions: `dim,FixedExpressionData-method, getNormalizationMethod, getTargetValue`
Examples

data(exampleMAS5)
myData <- setNormalizationMethod(exampleMAS5, "MAS5.0", targetValue=500)
dim(myData)
dim(myData[1:10,1:3])
dim(myData[[1:10,1:3]])
Index

* classifier information functions
  get Citations, 6
  getClassifier, 8
  getDecisionBoundaries, 9
  getDescription, 9
  getEventChain, 10
  getMeans, 12
  getNormalizationMethod, 13
  getProbeNames, 15
  getSds, 16
  getTrainingData, 18
  getWeights, 20

* classifier results
  getBatchCorrection, 6
  getClassifications, 7
  getScores, 16
  getWeightingType, 19

* datasets
  exampleMAS5, 5

* fixed data information extraction functions
  [[,FixedExpressionData,ANY,ANY-method, 24
  dim,FixedExpressionData-method, 4
  getNormalizationMethod, 13
  getTargetValue, 17

* internal
  geneClassifiers-package, 2

* workflow functions
  getNormalizationMethod, 13
  runClassifier, 20
  setNormalizationMethod, 22
  showClassifierList, 23
  [[,FixedExpressionData,ANY,ANY-method, 24
  [[,FixedExpressionData,ANY,missing,ANY-method
    [[,FixedExpressionData,ANY,ANY-method, 24
  [[,FixedExpressionData,ANY,ANY-method
    [[,FixedExpressionData,ANY,ANY-method, 24
  [[,FixedExpressionData,ANY,missing-method
    [[,FixedExpressionData,ANY,ANY-method, 24
  [[,FixedExpressionData,missing,ANY-method
    [[,FixedExpressionData,ANY,ANY-method, 24
    bracket,FixedExpressionData-method
      [[,FixedExpressionData,ANY,ANY-method, 24
  ClassifierParameters, 3, 4, 7–15, 17, 18, 20–22
  ClassifierResults, 4, 6–8, 13, 16, 19, 21
  dim,FixedExpressionData-method, 4
  exampleMAS5, 5
  ExpressionSet, 5, 18, 22
  FixedExpressionData, 5, 14, 17, 21, 22, 24
  geneClassifiers
    (geneClassifiers-package). 2
  geneClassifiers-package, 2
  getBatchCorrection, 6, 7, 16, 19
  getBatchCorrection,ClassifierResults-method
    (getBatchCorrection), 6
  getCitations, 6, 8–12, 14, 15, 17, 18, 20
  getCitations,ClassifierParameters-method
    (getCitations), 6
  getClassifications, 6, 7, 16, 19
getClassifier, 7, 8, 9–12, 14, 15, 17, 18, 20, 21, 23
getClassifier, character-method (getClassifier), 8
getClassifier, ClassifierResults-method (getClassifier), 8
getDecisionBoundaries, 7, 8, 9, 10–12, 14, 15, 17, 18, 20
getDecisionBoundaries, ClassifierParameters-method (getDecisionBoundaries), 9
description, 7–9, 9–10–12, 14, 15, 17, 18, 20
description, ClassifierParameters-method (getDescription), 9
eventChain, 7–10, 10, 11, 12, 14, 15, 17, 18, 20
eventChain, ClassifierParameters-method (getEventChain), 10
intercept, 7–10, 11, 12, 14, 15, 17, 18, 20
intercept, ClassifierParameters-method (getIntercept), 11
means, 7–11, 12, 14, 15, 17, 18, 20
means, ClassifierParameters-method (getMeans), 12
name, 12
name, ClassifierParameters-method (getName), 12
name, ClassifierResults-method (getName), 12
name, TransformationProcess-method (getName), 12
normalizationMethod, 4, 7–12, 13, 15, 17, 18, 20–24
normalizationMethod, ClassifierParameters-method (getNormalizationMethod), 13
normalizationMethod, FixedExpressionData-method (getNormalizationMethod), 13
normalizationMethods, 5, 14
normalizationMethods (getNormalizationMethod), 13
probeNames, 7–12, 14, 15, 17, 18, 20
probeNames, ClassifierParameters-method (getProbeNames), 15
scores, 6, 7, 16, 19
scores, ClassifierResults-method (getScores), 16
ssds, 7–12, 14, 15, 16, 18, 20
ssds, ClassifierParameters-method (getSds), 16
targetValue, 4, 14, 17, 24
targetValue, FixedExpressionData-method (getTargetValue), 17
trainingData, 7–12, 14, 15, 17, 18, 20
trainingData, ClassifierParameters-method (getTrainingData), 18
weightingType, 6, 7, 16, 19
weightingType, ClassifierResults-method (getWeightingType), 19
weightingTypes (getWeightingType), 19
weights, 7–12, 14, 15, 17, 18, 20
weights, ClassifierParameters-method (getWeights), 20
package_version, 3–5
runClassifier, 4, 6, 8, 10, 14, 19, 20, 22, 23
runClassifier, character, FixedExpressionData-method (runClassifier), 20
runClassifier, ClassifierParameters, FixedExpressionData-method (runClassifier), 20
runClassifier, ClassifierResults, FixedExpressionData-method (runClassifier), 20
setNormalizationMethod, 14, 21, 22, 23
showClassifierList, 8, 14, 21, 22, 23