Package ‘geneClassifiers’

April 25, 2017

Type Package
Title Application of gene classifiers
Version 1.0.0
Date 2017-12-11
Author R Kuiper
Maintainer R Kuiper <r.kuiper.emc@gmail.com>
Description This packages aims for easy accessible application of classifiers which have been published in literature using an ExpressionSet as input.
License GPL-2
biocViews GeneExpression, BiomedicalInformatics, Classification, Survival, Microarray
LazyData true
Suggests testthat
Depends utils
Imports methods, stats, Biobase, BiocGenerics
Class_FixedExpressionData.R Class_ClassifierParameters.R
Class_ClassifierResults.R zzz.R
RoxygenNote 5.0.1
NeedsCompilation no

R topics documented:

ClassifierParameters .............................................................. 2
ClassifierResults ................................................................. 3
dim,FixedExpressionData-method ............................................ 3
exampleMAS5 ................................................................. 4
FixedExpressionData ............................................................. 4
getBatchCorrection ............................................................... 5
getCitations ................................................................. 5
getClassifications .............................................................. 6
getClassifier ................................................................. 7
getDecisionBoundaries ........................................................ 7
getDescription ............................................................... 8
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 training set (if available)
- **sds**: A numeric vector of probe-set standard deviations as observed in the training set (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.

classifierParameters An object of class `ClassifierParameters` in which the applied classifier parameters are stored.

classifierParameters An object of class `ClassifierParameters` in which the applied classifier parameters are stored.

classifierParameters An object of class `ClassifierParameters` in which the applied classifier parameters are stored.

classifierParameters An object of class `ClassifierParameters` in which the applied classifier parameters are stored.

classifierParameters An object of class `ClassifierParameters` in which the applied classifier parameters are stored.

classifierParameters An object of class `ClassifierParameters` in which the applied classifier parameters are stored.

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 probset weighting)

geneClassifierVersion An object of class `package_version`

---

dim,FixedExpressionData-method

Dimensions of an Object

Description

Retrieve the dimension of an object.

Usage

## 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-method, getNormalizationMethod, getTargetValue`

Examples

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

```r
eexampleMAS5
```

### Format

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

### Value

- `FixedExpressionData`

### 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**

getBatchCorrection(object)

### 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**

getCitations(object)

### S4 method for signature 'ClassifierParameters'

getcitations(object)

**Arguments**

object An object of class ClassifierParameters as returned by getClassifier
**Description**

`getClassifications` returns the resulting classifications.

**Usage**

```r
getClassifications(object)
```

## S4 method for signature 'ClassifierResults'

getcollections(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**

```r
myData <- setNormalizationMethod(exampleMAS5, "MASS.0", targetValue=500)
results <- runClassifier('EMC92', myData)
getScores( results )
getClassifications( results )
```
**getClassifier**

*Obtain a classifier definition.*

**Description**

`getClassifier` returns a requested classifier definition.

**Usage**

```r
getClassifier(value)

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

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

**Arguments**

- `value`: Either a text value indicating a classifier name (see `showClassifierList`), or an object of class `ClassifierResults` as returned by the `runClassifier` function.

**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**

```r
getClassifier("EMC92")
```

---

**getDecisionBoundaries**

*Obtain the decision boundaries defined for the classifier.*

**Description**

`getDecisionBoundaries` returns the 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: get Citations, get Classifier, getDescription, get Event Chain, get Intercept, get Means, get Normalization Method, get Probe Names, get Sds, get Training Data, get Weights

Examples

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


description

Obtain classifiers’ description.

description

getDescription returns the descriptive text associated with the classifier.

Usage

getDescription(object)

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

Arguments

object An object of class ClassifierParameters as returned by getClassifier

Value

A character string describing the classifier

See Also

Other classifier information functions: get Citations, get Classifier, get Decision Boundaries, get Event Chain, get Intercept, get Means, get Normalization Method, get Probe Names, get Sds, get Training Data, get Weights
**getEventChain**

**Examples**

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

---

**Description**

`getEventChain` returns the event chain encoded in the classifier. The event chain indicates what preprocessing steps are performed by the `runClassifier` function prior to classification.

**Usage**

```r
getEventChain(object)
```

### S4 method for signature 'ClassifierParameters'

```r
gEventChain(object)
```

**Arguments**

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

**Value**

Returns the event chain encoded 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`

**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)
```

```r
## S4 method for signature 'ClassifierParameters'
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**

```r
getMeans(object)
```

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

**Arguments**

- `object` An object of class `ClassifierParameters` as returned by `getClassifier`
**getName**

**Value**

Retrieves a numeric vector of probe set means as observed in the reference data.

**See Also**

showClassifierList getClassifier runClassifier

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

**Examples**

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

---

**getName**

Obtain object names.

**Description**

getName returns the name associated with the requested object.

**Usage**

```r
getName(object)
```

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

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

```r
## 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**

```r
aClassifier <- getClassifier("EMC92")
getName( aClassifier )
```
**getNormalizationMethod**

Obtain normalization method

Description

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-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 )
getProbeNames

```r
data(exampleMAS5)
showClassifierList()
getNormalizationMethods()

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

getScores(results)
getClassifications(results)
```

---

### Description

`getProbeNames` returns the probe names associated with the requested classifier.

### Usage

```r
getProbeNames(object)
```

```r
## 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

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

*Obtain classifier score.*

**Description**

gScores returns the resulting scores from a classifier run.

**Usage**

gScores(object)

```r
## S4 method for signature 'ClassifierResults'
gScores(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)
gScores( results )
gClassifications( results )
```

---

**getSds**

*Obtain classifiers’ reference standard deviations.*

**Description**

gSds returns the reference standard deviations encoded in the classifier.

**Usage**

gSds(object)

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

**Arguments**

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

**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: getCleanData, getClassifier, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getTrainingData, getWeights

**Examples**

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

---

**getDescription**

Obtain the `getDescription` of the classifier.

**Usage**

`getDescription(object)`

```r
## S4 method for signature 'Classifier'

description(object)
```

**Arguments**

- `object` An object of class `Classifier`

**Value**

A string describing the classifier.

**See Also**

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

**Examples**

```r
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

Usage
getWeightingType(object)

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

Arguments
object       An object of class ClassifierResults as returned by runClassifier
getWeights

Value

complete or reweighted

See Also

Other classifier results: getBatchCorrection, getClassifications, getScores

Examples

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

geweights

Obtain classifier weights.

Description

geweights returns the probe weights associated with the classifier.

Usage

geweights(object)

## S4 method for signature 'ClassifierParameters'

geweights(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")
geweights(aClassifier)
runClassifier  Perform classification.

Description

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

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 re-
turned 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

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

Description

setNormalizationMethod is to be called prior to running a classifier.

Usage

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

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)

[.,FixedExpressionData-method

Extract

Description
Extract Parts of an Object
Usage

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

## S4 method for signature 'FixedExpressionData'
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

**Topic datasets**
- exampleMAS5, 4
- [,FixedExpressionData-method, 20
- [[,FixedExpressionData-method
  ([,FixedExpressionData-method), 20
- bracket,FixedExpressionData-method
  ([,FixedExpressionData-method), 20
- ClassifierParameters, 2, 3, 5–14, 16–19
- ClassifierResults, 3, 5–7, 11, 14, 16, 18
- dim,FixedExpressionData-method, 3
- exampleMAS5, 4
- ExpressionSet, 4, 16, 19
- FixedExpressionData, 4, 12, 15, 18, 19, 21
- getBatchCorrection, 5, 6, 14, 17
- getBatchCorrection,ClassifierResults-method
  (getBatchCorrection), 5
- getCitations, 5, 7–13, 15–17
- getCitations,ClassifierParameters-method
  (getCitations), 5
- getClassifications, 5, 6, 14, 17
- getClassifications,ClassifierResults-method
  (getClassifications), 6
- getClassifier, 5, 6, 7, 8–18, 20
- getClassifier,character-method
  (getClassifier), 7
- getClassifier,ClassifierResults-method
  (getClassifier), 7
- getDecisionBoundaries, 6, 7, 7, 8–13, 15–17
- getDecisionBoundaries,ClassifierParameters-method
  (getDecisionBoundaries), 7
- getDescription, 6–8, 8, 9–13, 15–17
- getDescription,ClassifierParameters-method
  (getDescription), 8
- getEventChain, 6–8, 9, 10–13, 15–17
- getEventChain,ClassifierParameters-method
  (getEventChain), 9
- getIntercept, 6–9, 10, 11–13, 15–17
- getIntercept,ClassifierParameters-method
  (getIntercept), 10
- getMeans, 6–10, 10, 12, 13, 15–17
- getMeans,ClassifierParameters-method
  (getMeans), 10
- getName, 11
- getName,ClassifierParameters-method
  (getName), 11
- getName,ClassifierResults-method
  (getName), 11
- getName,TransformationProcess-method
  (getName), 11
- getNormalizationMethod, 3, 6–11, 12, 13, 15–21
- getNormalizationMethod,ClassifierParameters-method
  (getNormalizationMethod), 12
- getNormalizationMethod,FixedExpressionData-method
  (getNormalizationMethod), 12
- getNormalizationMethods, 4, 12
- getNormalizationMethods
  (getNormalizationMethod), 12
- getProbeNames, 6–12, 13, 15–17
- getProbeNames,ClassifierParameters-method
  (getProbeNames), 13
- getScores, 5, 6, 14, 17
- getScores,ClassifierResults-method
  (getScores), 14
- getSds, 6–13, 14, 16, 17
- getSds,ClassifierParameters-method
  (getSds), 14
- getTargetValue, 3, 12, 15, 21
- getTargetValue,FixedExpressionData-method
  (getTargetValue), 15
- getTrainingData, 6–13, 15, 16, 17
- getTrainingData,ClassifierParameters-method
  (getTrainingData), 16
- getWeightingType, 5, 6, 14, 16
- getWeightingType,ClassifierResults-method
  (getWeightingType), 16
- getWeights, 6–13, 15, 16, 17
- getWeights,ClassifierParameters-method
  (getWeights), 17
package_version, 2–4

runClassifier, 3, 5, 7, 9, 12, 16, 18, 19, 20
runClassifier, character, FixedExpressionData-method
  (runClassifier), 18
runClassifier, ClassifierParameters, FixedExpressionData-method
  (runClassifier), 18
runClassifier, ClassifierResults, FixedExpressionData-method
  (runClassifier), 18

setNormalizationMethod, 12, 18, 19, 20
showClassifierList, 7, 12, 18, 19, 20