Package ‘geneClassifiers’

May 29, 2024

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Application of gene classifiers</td>
</tr>
<tr>
<td>Version</td>
<td>1.28.0</td>
</tr>
<tr>
<td>Description</td>
<td>This package aims for easy accessible application of classifiers which have been published in literature using an ExpressionSet as input.</td>
</tr>
<tr>
<td>BugReports</td>
<td><a href="https://github.com/rkuiper/geneClassifiers/issues">https://github.com/rkuiper/geneClassifiers/issues</a></td>
</tr>
<tr>
<td>License</td>
<td>GPL-2</td>
</tr>
<tr>
<td>biocViews</td>
<td>GeneExpression, BiomedicalInformatics, Classification, Survival, Microarray</td>
</tr>
<tr>
<td>LazyData</td>
<td>true</td>
</tr>
<tr>
<td>Suggests</td>
<td>testthat</td>
</tr>
<tr>
<td>Depends</td>
<td>R (&gt;= 3.6.0)</td>
</tr>
<tr>
<td>Imports</td>
<td>utils, methods, stats, Biobase, BiocGenerics</td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>6.1.1</td>
</tr>
<tr>
<td>Encoding</td>
<td>UTF-8</td>
</tr>
<tr>
<td>git_url</td>
<td><a href="https://git.bioconductor.org/packages/geneClassifiers">https://git.bioconductor.org/packages/geneClassifiers</a></td>
</tr>
<tr>
<td>git_branch</td>
<td>RELEASE_3_19</td>
</tr>
<tr>
<td>git_last_commit</td>
<td>b5d8c4d</td>
</tr>
<tr>
<td>git_last_commit_date</td>
<td>2024-04-30</td>
</tr>
<tr>
<td>Repository</td>
<td>Bioconductor 3.19</td>
</tr>
<tr>
<td>Date/Publication</td>
<td>2024-05-29</td>
</tr>
<tr>
<td>Author</td>
<td>R Kuiper [cre, aut] (<a href="https://orcid.org/0000-0002-3703-1762">https://orcid.org/0000-0002-3703-1762</a>)</td>
</tr>
<tr>
<td>Maintainer</td>
<td>R Kuiper <a href="mailto:r.kuiper.emc@gmail.com">r.kuiper.emc@gmail.com</a></td>
</tr>
</tbody>
</table>
geneClassifiers-package

geneClassifiers: Application of gene classifiers

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:


---

**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 classifier's 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.

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 classifier's training data), or reduced (i.e., missing data but not reweighting the original probeset 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,ANY,ANY-method, getNormalizationMethod, getTargetValue
Examples

```r
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**

`exampleMAS5`

**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

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

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

ggetCitations(object)

## 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**


**Examples**

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

getClassifications(aClassifier)
```

---

**Description**

`getClassifications` returns the resulting classifications.

**Usage**

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

classifications(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, "MAS5.0", targetValue=500)
results <- runClassifier("EMC92", myData)
getScores(results)

classifications(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

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

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: getCitations, getClassifier, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getSds, getTrainingData, getWeights

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)
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)

gEventChain <- getEventChain(aClassifier)

getEventChain

Obtain classifiers' event chain.

Description

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

Usage

gEventChain(object)

## S4 method for signature 'ClassifierParameters'
gEventChain(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 in the classifier.

**Usage**

```r
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**

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

---

**getName**  
*Obtain object names.*

**Description**

getName returns the name associated with the requested object.
getNormalizationMethod

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 )

---

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.
getNormalizationMethod

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")
geNormalizationMethod( myData )
geNormalizationMethod( aClassifier )
data(exampleMAS5)

showClassifierList()
geNormalizationMethods()

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

gScores( 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

gScores(object)

## 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

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

gSds

Obtain classifiers’ reference standard deviations.

Description

getSds returns the reference standard deviations encoded in the classifier.

Usage

gSds(object)

## S4 method for signature 'ClassifierParameters'
gSds(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, getDecisionBoundaries, getDescription, getEventChain, getIntercept, getMeans, getNormalizationMethod, getProbeNames, getTrainingData, getWeights

Examples

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

g << getTargetValue

Description

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

Usage

gTargetValue(object)

## S4 method for signature 'FixedExpressionData'
gTargetValue(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

Obtain classifier training data.

Examples

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

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

getWeightingType(object)
getWeightingTypes()

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

Arguments

object An object of class ClassifierResults as returned by runClassifier

Value

one of the values in getWeightingTypes()

either "complete" or "rewighted"

See Also

Other classifier results: getBatchCorrection, getClassifications, getScores

Examples

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

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
showClassifierList

Examples

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

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

```r
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
  getCitations, 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
  (getClassifications), 7
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
getDescription, 7–9, 10–12, 14, 15, 17, 18, 20
getDescription, ClassifierParameters-method (getDescription), 9
gETEventChain, 7–10, 10, 11, 12, 14, 15, 17, 18, 20
gETEventChain, ClassifierParameters-method (getEventChain), 10
gETIntercept, 7–10, 11, 12, 14, 15, 17, 18, 20
gETIntercept, ClassifierParameters-method (getIntercept), 11
gETMeans, 7–11, 12, 14, 15, 17, 18, 20
gETMeans, ClassifierParameters-method (getMeans), 12
gETName, 12
gETName, ClassifierParameters-method (getName), 12
gETName, ClassifierResults-method (getName), 12
gETName, TransformationProcess-method (getName), 12
gETNormalizationMethod, 4, 7–12, 13, 15, 17, 18, 20–24
gETNormalizationMethod, ClassifierParameters-method (getNormalizationMethod), 13
gETNormalizationMethod, FixedExpressionData-method (getNormalizationMethod), 13
gETNormalizationMethods, 5, 14
gETNormalizationMethods (getNormalizationMethod), 13
gETProbeNames, 7–12, 14, 15, 17, 18, 20
gETProbeNames, ClassifierParameters-method (getProbeNames), 15
gETScores, 6, 7, 16, 19
gETScores, ClassifierResults-method (getScores), 16
gETSds, 7–12, 14, 15, 16, 18, 20
gETSds, ClassifierParameters-method (getSds), 16
gETTargetValue, 4, 14, 17, 24
gETTargetValue, FixedExpressionData-method (getTargetValue), 17
gETTrainingData, 7–12, 14, 15, 17, 18, 20
gETTrainingData, ClassifierParameters-method (getTrainingData), 18
gETWeightingType, 6, 7, 16, 19
gETWeightingType, ClassifierResults-method (getWeightingType), 19
gETWeightingTypes (getWeightingType), 19
gETWeights, 7–12, 14, 15, 17, 18, 20
gETWeights, 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