

Package: geneClassifiers (Version 1.0.0)

R.Kuiper

October 27, 2016

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. The workflow is illustrated in Figure 1. The raw gene expression data obtained by microarray experiments is normalized using existing techniques (independent of this package). The choice of normalization method is dictated by the classifier. Some classifiers were developed using MAS5.0. In that case, the data to be classified should be normalized using MAS5.0. Normalization is followed by preprocessing (this package) and generating scores/classifications (this package). This package is suitable only for datasets of at least 20 patients.

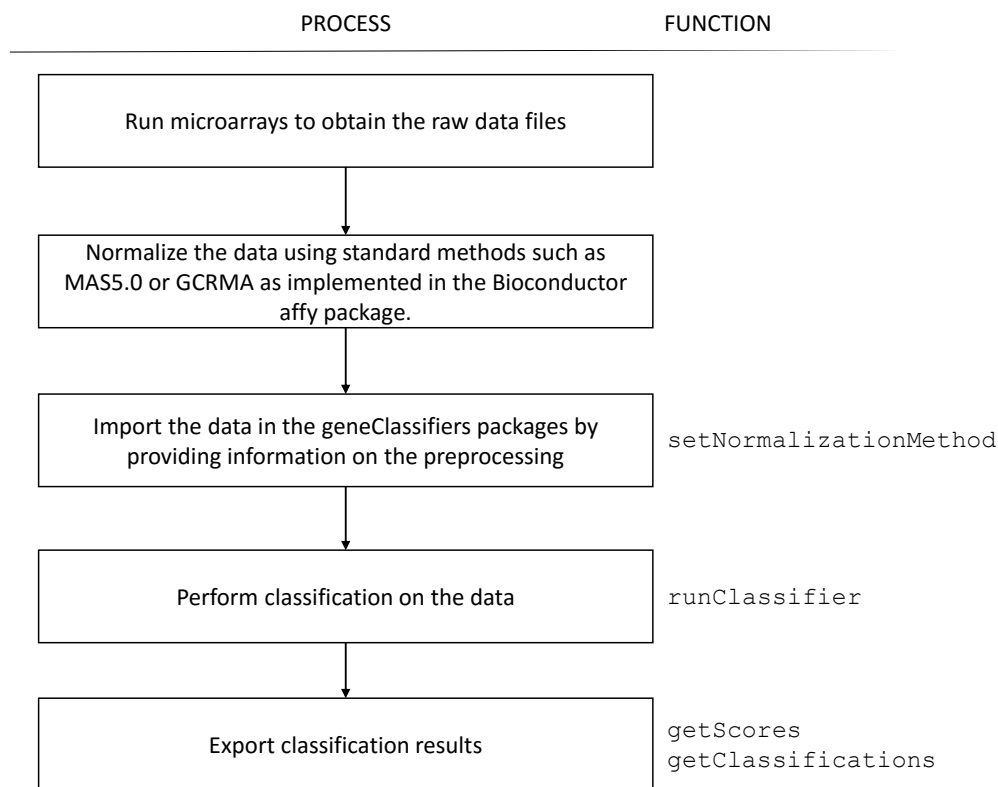


Figure 1: The workflow of the geneClassifiers package. The raw gene expression data is normalized. This normalized data is used as input in the geneClassifiers package. The processes with their relevant functions are shown.

1 Classifiers

The currently implemented list of classifiers can be obtained with the command:

```
> showClassifierList()
```

```

      name          normalizationMethod
[1,] "EMC92"       "MAS5.0"
[2,] "UAMS70"     "MAS5.0"
[3,] "UAMS17"     "MAS5.0"
[4,] "UAMS80"     "MAS5.0"
[5,] "HM19"       "GCRMA"
[6,] "IFM15"     "MAS5.0"
[7,] "MRCIX6"    "MAS5.0"
[8,] "MILLENNIUM100" "MAS5.0"
      description
[1,] "A risk classifier for multiple myeloma"
[2,] "A risk classifier for multiple myeloma"
[3,] "A risk classifier for multiple myeloma"
[4,] "A risk classifier for multiple myeloma"
[5,] "A risk classifier for multiple myeloma"
[6,] "A risk classifier for multiple myeloma"
[7,] "A risk classifier for multiple myeloma"
[8,] "A risk classifier for multiple myeloma based on Affy HG-U133 A/B chip"

```

To find more information on a specific classifier (e.g. EMC92), the classifier parameters can be obtained by

```

> EMC92Classifier<-getClassifier("EMC92")
> EMC92Classifier

```

```

-----
Classifier: EMC92
Description: A risk classifier for multiple myeloma
Based on n = 92 probe sets
Number of risk groups: n = 2
To be used on MAS5.0 normalized data
R.Kuiper, A.Broyl, Y.de Knegt, et.al.; A gene expression signature for high-risk
multiple myeloma; Leukemia (2012) 26, 2046-2413; doi:10.1038/leu.2012.127

```

```

> HM19Classifier<-getClassifier("HM19")
> HM19Classifier

```

```

-----
Classifier: HM19
Description: A risk classifier for multiple myeloma
Based on n = 19 probe sets
Number of risk groups: n = 2
To be used on GCRMA normalized data
Reme T, Hose D, Theillet C, Klein B. Modeling risk stratification in human cancer.
Bioinformatics. 2013;29(9):1149-1157

```

This is an object of class 'ClassifierParameters' which stores classifier related information, such as probe-sets used and their weights, 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.

Further information can be obtained from this object e.g. obtaining the weights used in a classifier:

```

> getWeights(EMC92Classifier)[1:10]

204379_s_at 210334_x_at 201795_at 38158_at 201307_at 205046_at
0.059427338 0.017489792 0.006685847 0.042315407 0.016492508 0.008662337
204026_s_at 238662_at 220351_at 202542_s_at
0.004558196 0.048955859 0.042045299 0.087027609

```

or the decision boundaries used to decide which class a sample score belongs to

```
> getDecisionBoundaries(HM19Classifier)
```

```
[1] 28.4 54.6
```

or the 'eventChain' which gives information on preprocessing:

```
> getEventChain(EMC92Classifier)
```

```
$targetValue
```

```
[1] 500
```

```
$truncate
```

```
[1] -Inf
```

```
$allow.reweighted
```

```
[1] TRUE
```

```
$to.log
```

```
[1] 2
```

```
$to.meancentering
```

```
[1] TRUE
```

```
$to.unitvariance
```

```
[1] TRUE
```

2 Data to be classified

The input data for the 'geneClassifiers' package is a Bioconductor ExpressionSet which has been prenormalized using existing methods such as MAS5.0 or GCRMA. For more information on these methods see the Bioconductor 'affy' package. The 'geneClassifiers' package contains an example dataset of MAS5.0 normalized (target value = 500) gene expression data of 25 multiple myeloma patients from the HOVON65/GMMG-HD4 trial (Pieter Sonneveld et al., J Clin Oncol, 2012)

```
> library(Biobase)
```

```
> data(exampleMAS5)
```

```
> class(exampleMAS5) #an object of class ExpressionSet
```

```
[1] "ExpressionSet"
```

```
attr(,"package")
```

```
[1] "Biobase"
```

```
> dim(exampleMAS5)
```

```
Features  Samples
```

```
374      25
```

```
> preproc(experimentData(exampleMAS5))
```

```
[[1]]
```

```
[1] "MAS5.0"
```

```
$targetValue
```

```
[1] 500
```

To import this data set into the 'geneClassifiers' package, the setNormalization function is used:

```
> fixedData <- setNormalizationMethod( exampleMAS5, method="MAS5.0", targetValue = 500 )
```

```
> fixedData
```

Fixed expression set

Normalization method: MAS5.0
Number of samples : 25
Number of features : 374

Nb the `targetValue = 500` is only required in the example (see below).

To get reliable results in the classification, the function depends on unmanipulated output from the normalization methods, i.e. read in CEL files into affy functions, obtain `ExpressionSet`, and use this set (without modification) for obtaining classifier scores. The function can detect deviations such as subsets of data sets or log transformed data, but detection of deviations is not guaranteed. When providing an `ExpressionSet` with all probe-sets still included, the `'targetValue=500'` argument is not necessary because the function is able to extract the value from the data. In the example the number of probe-sets was reduced due to space considerations, the MAS5.0 target value cannot be obtained from the data so that the argument has to be provided. See `'?setNormalizationMethod'` for more details.

3 Performing classifications

To perform the classification using a classifier described in section 1 on the data described in section 2, the `'runClassifier'` function is called using both arguments:

```
> resultsEMC92 <- runClassifier( "EMC92" , fixedData )  
> resultsUAMS70 <- runClassifier( "UAMS70", fixedData )  
> resultsEMC92
```

Note: Research use only

Classifier: EMC92

> 0.827 : Risk-II

classifications

Risk-I Risk-II

19 6

Batch corrected : yes

weighting type : complete

> resultsUAMS70

Note: Research use only

Classifier: UAMS70

> 0.66 : Risk-II

classifications

Risk-I Risk-II

23 2

Batch corrected : yes

weighting type : complete

The scores and classifications can be extracted using the `'getScores'` and `'getClassifications'` function

```
> data.frame(  
+ "score_EMCC92" = getScores( resultsEMC92 ),  
+ "class_EMCC92" = getClassifications( resultsEMC92 ),  
+ "score_UAMS70" = getScores( resultsUAMS70 ),  
+ "class_UAMS70" = getClassifications( resultsUAMS70 )  
+ )
```

	score EMC92	class EMC92	score UAMS70	class UAMS70
GSM493958	-0.71624019	Risk-I	-0.52762419	Risk-I
GSM493959	-0.27585287	Risk-I	-0.33862140	Risk-I
GSM493960	-0.78728255	Risk-I	-0.30964906	Risk-I
GSM493961	-1.50733627	Risk-I	-0.71023469	Risk-I
GSM493962	-1.36569438	Risk-I	-0.59723884	Risk-I
GSM493963	-0.99382605	Risk-I	-0.35502398	Risk-I
GSM493964	1.20345559	Risk-II	0.87469597	Risk-II
GSM493965	-0.23111232	Risk-I	0.69126041	Risk-II
GSM493966	0.90148041	Risk-II	0.06231283	Risk-I
GSM493967	0.05209633	Risk-I	-0.12333754	Risk-I
GSM493968	0.33663874	Risk-I	0.07822597	Risk-I
GSM493969	-0.18411181	Risk-I	-0.75836587	Risk-I
GSM493970	-0.77179375	Risk-I	-0.58808060	Risk-I
GSM493971	1.18945257	Risk-II	-0.03323022	Risk-I
GSM493972	1.34454785	Risk-II	0.28384630	Risk-I
GSM493973	-1.02682845	Risk-I	0.10755853	Risk-I
GSM493974	0.67574560	Risk-I	0.07488403	Risk-I
GSM493975	-1.73924311	Risk-I	-0.52809629	Risk-I
GSM493976	1.23116416	Risk-II	0.58564808	Risk-I
GSM493977	0.43139670	Risk-I	-0.97249674	Risk-I
GSM493978	0.60432989	Risk-I	0.58399765	Risk-I
GSM493979	0.63122523	Risk-I	0.10798259	Risk-I
GSM493980	-0.88162933	Risk-I	0.05694285	Risk-I
GSM493981	0.73216640	Risk-I	0.28518870	Risk-I
GSM493982	1.14725160	Risk-II	0.25455012	Risk-I

4 Caution: non standard situations

The geneClassifiers package performs a batch correction by applying a linear transformation of the probe-set means and standard deviations to the values observed in the classifiers' training set. In order to accurately do this the data must contain a sufficient number of samples ($n \geq 20$) to estimate the means and standard deviations. If less samples are available, the 'runClassifier' function will give a warning and suggest to consider setting 'do.batchcorrection = FALSE'. Please note this will most likely result in invalid classifications (or certainly different classifications).

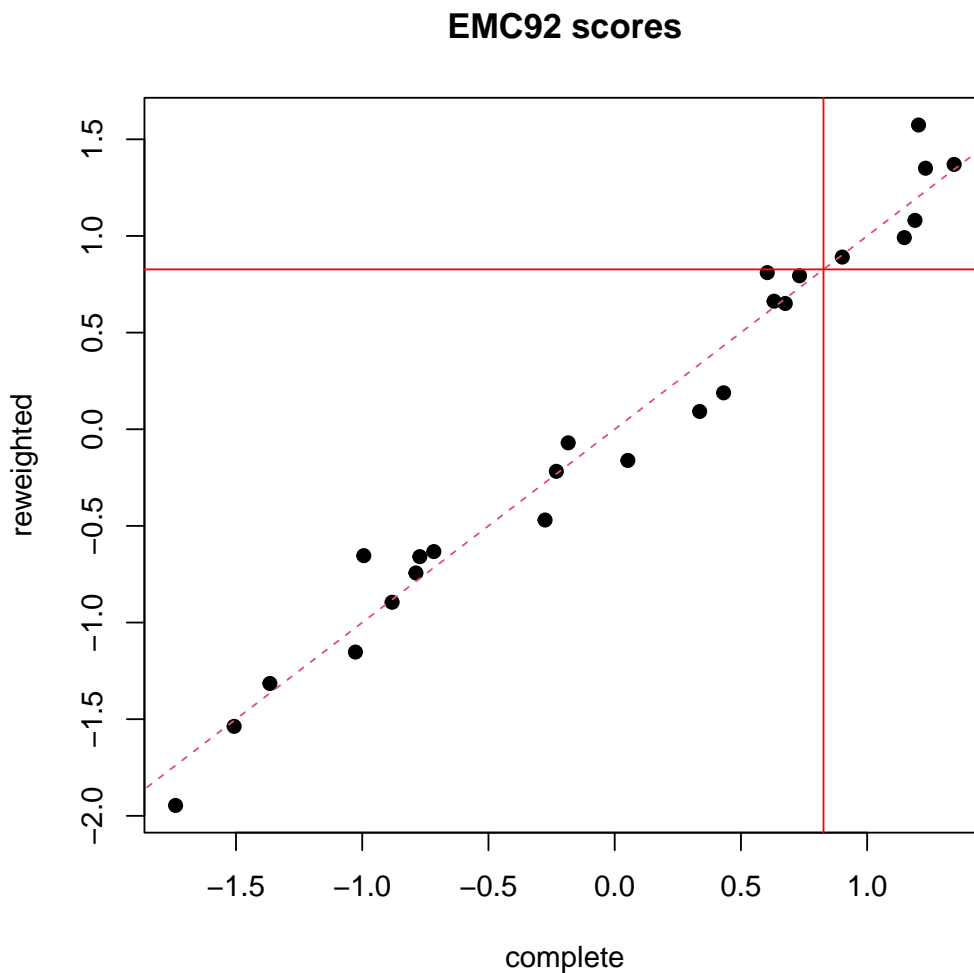
Besides the requirements of a matching normalization method between data and classifier and sufficient samples, the assumption is that the probe-sets needed for classification are present in the data. If this is not true, simply ignoring the missing probe-set may heavily bias the results. Therefore, when detecting missing probe-sets, the 'run-classifier' function will give an error message and suggest to consider using the argument 'allow.reweighted = TRUE'. This will reweight the weightings for the probe-sets which are present, based on the covariance structure of the classifiers' trainings data. See the vignette 'MissingCovariates' for more information. Please note this is not how the classifiers are intended and consequentially will result in different classifications.

```
> resultsEMC92.reWeighted <- runClassifier(
+   "EMC92" ,
+   fixedData[1:70,] ,
+   allow.reweighted=TRUE
+ )
> resultsEMC92.reWeighted
```

```
Note: Research use only
Classifier: EMC92_reweighted
> 0.827 : Risk-II
classifications
Risk-I Risk-II
19      6
```

```
Batch corrected : yes
weighting type  : reweighted
```

```
-----
> plot(
+   x = getScores(resultsEMC92),
+   y = getScores(resultsEMC92.reWeighted),
+   xlab = "complete",
+   ylab = "reweighted",
+   main = "EMC92 scores",
+   pch = 21,
+   bg = 'black'
+ )
> lines(c(-10,10),c(-10,10),col=2,lty=2)
> abline(
+   v = getDecisionBoundaries( getClassifier( resultsEMC92          )),
+   h = getDecisionBoundaries( getClassifier( resultsEMC92.reWeighted)),
+   col='red'
+ )
```



```
> sessionInfo()

R version 4.5.0 RC (2025-04-04 r88126)
Platform: x86_64-pc-linux-gnu
```

Running under: Ubuntu 24.04.2 LTS

Matrix products: default

BLAS: /home/biocbuild/bbs-3.21-bioc/R/lib/libRblas.so

LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0 LAPACK version 3.12.0

locale:

[1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
[3] LC_TIME=en_GB LC_COLLATE=C
[5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8 LC_NAME=C
[9] LC_ADDRESS=C LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: America/New_York

tzcode source: system (glibc)

attached base packages:

[1] stats graphics grDevices utils datasets methods base

other attached packages:

[1] Biobase_2.68.0 BiocGenerics_0.54.0 generics_0.1.3

[4] geneClassifiers_1.32.0

loaded via a namespace (and not attached):

[1] compiler_4.5.0 tools_4.5.0