

Package ‘MiRNAQCD’

July 13, 2020

Version 1.1

Encoding UTF-8

Date 2020-07-09

Title Micro-RNA Quality Control and Diagnosis

Description A complete and dedicated analytical toolbox for quality control and diagnosis based on subject-related measurements of micro-RNA (miRNA) expressions. The package consists of a set of functions that allow to train, optimize and use a Bayesian classifier that relies on multiplets of measured miRNA expressions. The package also implements the quality control tools required to preprocess input datasets. In addition, the package provides a function to carry out a statistical analysis of miRNA expressions, which can give insights to improve the classifier's performance. The method implemented in the package was first introduced in L. Ricci, V. Del Vescovo, C. Cantaloni, M. Grasso, M. Barbareschi and M. A. Denti, "Statistical analysis of a Bayesian classifier based on the expression of miRNAs", *BMC Bioinformatics* 16:287, 2015 <doi:10.1186/s12859-015-0715-9>. The package is thoroughly described in M. Castelluzzo, A. Perinelli, S. Detassis, M. A. Denti and L. Ricci, "MiRNA-QC-and-Diagnosis: An R package for diagnosis based on MiRNA expression", *SoftwareX* 12:100569, 2020 <doi:10.1016/j.softx.2020.100569>. Please cite both these works if you use the package for your analysis. **DISCLAIMER:** The software in this package is for general research purposes only and is thus provided **WITHOUT ANY WARRANTY**. It is **NOT** intended to form the basis of clinical decisions. Please refer to the GNU General Public License 3.0 (GPLv3) for further information.

Maintainer Alessio Perinelli <alessio.perinelli@unitn.it>

License GPL-3

RoxygenNote 7.1.0

Imports stats, utils, ggplot2, pROC

Suggests testthat

NeedsCompilation no

Author Michele Castelluzzo [aut],
Alessio Perinelli [cre],
Simone Detassis [aut],
Michela Alessandra Denti [aut],
Leonardo Ricci [aut]

Repository CRAN

Date/Publication 2020-07-13 07:30:03 UTC

R topics documented:

miRNA_assessQualityThreshold	2
miRNA_classifierSetup	3
miRNA_diagnosis	6
miRNA_expressionPreprocessing	8
miRNA_loadDiagnosticThreshold	9
miRNA_loadQualityThreshold	10
miRNA_plotHistograms	10
miRNA_plotROC	11
miRNA_plotThresholds	12
miRNA_removeOutliers	13

Index **15**

miRNA_assessQualityThreshold

Assessment of quality threshold values.

Description

This function assesses a set of quality threshold values (standard deviations), one for each miRNA, out of a dataset.

Usage

```
miRNA_assessQualityThreshold(
  inputDataset,
  significanceLevel = 0.05,
  saveOutputFile = FALSE,
  outputFileName = "",
  sep = "\t"
)
```

Arguments

inputDataset Dataset (data frame) to be used for the assessment. The data frame must comply with the output format of the preprocessing function (`miRNA_expressionPreprocessing`), thus containing the columns 'Subject', 'miRNA', 'Mean', 'StdDev', 'Sample-Size' and possibly 'Class'. Any other column is ignored, and any missing column forbids execution.

significanceLevel Significance level to be used for the assessment (must be greater than zero and less than one). Default is 0.05 (i.e. 5 percent).

saveOutputFile	Boolean option setting whether results are written to file (TRUE) or not (FALSE). Default is FALSE.
outputFileName	Name of the output file where the quality threshold values are to be stored. If not assigned, a filename is automatically generated.
sep	Field separator character for the output files; the default is tabulation.

Value

A data frame of quality threshold values, containing the columns 'miRNA' and 'QualityThreshold'. Please refer to the user manual installed in "/path-to-library/MiRNAQCD/doc/manual.pdf" for detailed function documentation. The path "/path-to-library" can be shown from R by calling ".libPaths()"

Examples

```
requiredFile = paste(system.file(package="MiRNAQCD"),
"/extdata/test_dataset_alpha_prep.dat", sep='')
myDataFrame <- read.table(file=requiredFile, header=TRUE)
qt <- miRNA_assessQualityThreshold(myDataFrame, significanceLevel=0.05)
```

miRNA_classifierSetup *Analysis of features and training of classifiers.*

Description

This function carries out different tasks depending on the input parameters: → Analysis mode: analyzes the properties of each miRNA (possibly subtracting a normalizer) in terms of Target/Versus separation, normality, etc. A matrix of correlation coefficients between each pair of miRNAs is also assessed. → Training mode: trains a Bayesian classifier by assessing the corresponding diagnostic threshold values and the related uncertainties.

Usage

```
miRNA_classifierSetup(
  inputDataset,
  inputTargetList,
  inputVersusList = character(),
  inputMiRNAList = character(),
  coeffList = double(),
  saveOutputFile = FALSE,
  outputFileBasename = "",
  sep = "\t",
  plotFormat = "pdf",
  scorePlotAscending = TRUE,
  scorePlotParameters = character(),
  histogramParameters = character(),
  colorComplementFlag = FALSE
)
```

Arguments

- inputDataset** Dataset (data frame) to be used for the analysis/training. The data frame must comply with the output format of the quality control functions (`miRNA_expressionPreprocessing` and `miRNA_removeOutliers`), thus containing the columns 'Subject', 'miRNA', 'Mean', 'StdDev', 'SampleSize', 'Class'. Any other column is ignored, and any missing column forbids execution. Please note that in this case the 'Class' column is mandatory.
- inputTargetList** List of classes to use as target for the classification. The chosen target must correspond to at least one of the classes present in the 'Class' column of the `inputDataset`.
- inputVersusList** List of classes to use as versus for the classification. If the argument is left empty, all classes present in the 'Class' column of the `inputDataset`, minus the Target classes, are used as Versus.
- inputMiRNAList** List of miRNAs to be used by the classifier ('Training mode'). The chosen miRNAs must be present in the 'miRNA' column of the `inputDataset`. In 'Analysis mode', this argument has to be omitted (if no normalizer has to be used) or has to contain a single entry (corresponding to the miRNA to be used as normalizer).
- coeffList** List of coefficients for the classifier. In 'Training mode', the number of coefficients must be the same as the number of used miRNAs and listed in the same order. In 'Analysis mode', this argument has to be omitted.
- saveOutputFile** Boolean option setting whether results are written to file (TRUE) or not (FALSE). Default is FALSE.
- outputFileBasename** Name of the output file where the classifier setup results ('Training mode') or the analysis results ('Analysis mode') are to be stored. If not assigned, a filename is automatically generated. File names of other files created by the function are generated by appending suitable labels to the provided "outputFileBasename".
- sep** Field separator character for the output files; the default is tabulation.
- plotFormat** String specifying the format of generated graphic files (plots): can either be "pdf" (default) or "png".
- scorePlotAscending** Boolean option to set the direction in which samples are ordered: TRUE corresponds to samples ordered by ascending score, FALSE corresponds to samples ordered by descending score. Default is TRUE. This argument is meaningful only if `saveOutputFile` is set to TRUE and the function is running in 'Training mode'.
- scorePlotParameters** String specifying the y-axis parameters of the score plot. If empty, the axis is configured by assessing suitable parameters from the data. This argument is meaningful only if `saveOutputFile` is set to TRUE and the function is running in 'Training mode'. The string has to comply with the format "yl_yu_yt", where: yl is the lower y limit; yu is the upper y limit; yt is the interval between tics along the axis.

histogramParameters

String specifying the parameters used to build histograms. If empty, histograms are built by assessing suitable parameters from the data. This parameter is meaningful only if `saveOutputFile` is set to `TRUE`. The string has to comply with the following format: "xl_xu_bw", where `xl` is the lower boundary of the leftmost bin; `xu` is the upper boundary of the rightmost bin; `bw` is the bin width.

colorComplementFlag

Boolean option to switch between the default palette (`FALSE`) and its inverted version (`TRUE`). Default is `FALSE`, corresponding to target samples reported in blue and versus samples in red. This argument is meaningful only if `saveOutputFile` is set to `TRUE`.

Beware! Cross-correlation coefficients, as well as Shapiro-Wilk tests for normality, require at least three data samples. In case of less than three samples, those tests are skipped and "NA" (not available) is reported in the corresponding output.

Details

In order to select between Analysis and Training mode, the input parameters "inputMiRNAList" and "coeffList" have to comply with the following requirements. → Analysis mode: "coeffList" has to be empty (i.e. omitted in the function call arguments). "inputMiRNAList" can either be empty (i.e. omitted in the function call arguments) or of length 1: in the latter case, the single entry of "inputMiRNAList" is assumed to be the normalizer. → Training mode: "inputMiRNAList" and "coeffList" have to be non-empty and of the same size.

Value

In 'Analysis mode', a data frame containing the columns 'miRNA', 'Diagnosis', 'NumberOfSubjects', 'Mean', 'StdDev', 'NormalityTest', 't-test'. In 'Training mode', a data frame containing the columns 'Threshold', 'DeltaThreshold', 'DPrime', 'Pc', 'ChiUp', 'DChiUp', 'ChiDown', 'DChiDown'.

Examples

```
requiredFile = paste(system.file(package="MiRNAQCD"),
"/extdata/test_dataset_alpha_clean.dat", sep='')
myDataFrame <- read.table(file=requiredFile, header=TRUE)
Target <- c("A")
Versus <- c("B", "C")
## Analysis mode
miRNAstats <- miRNA_classifierSetup(myDataFrame, Target, Versus)
## Analysis mode, with normalizer
miRNAstats <- miRNA_classifierSetup(myDataFrame, Target, Versus, c("FZ"))

## Training mode
mirnaToUse <- c("FX", "FZ")
coefficientsToUse <- c(1.0, -1.0)
threshold <- miRNA_classifierSetup(myDataFrame, Target, Versus,
mirnaToUse, coefficientsToUse)
```

miRNA_diagnosis *Classification of a dataset (diagnosis).*

Description

This function classifies the entries of the input dataset as either target or versus by using the chosen classifier and given the corresponding diagnostic threshold value.

Usage

```
miRNA_diagnosis(
  inputDataset,
  inputMiRNAList,
  coeffList,
  inputThreshold,
  inputTargetList = character(),
  inputVersusList = character(),
  saveOutputFile = FALSE,
  outputFileBasename = "",
  sep = "\t",
  plotFormat = "pdf",
  scorePlotParameters = character(),
  scorePlotAscending = TRUE,
  colorComplementFlag = FALSE,
  histogramParameters = character()
)
```

Arguments

inputDataset	Dataset (data frame) to be classified. The data frame must comply with the output format of the quality control functions (miRNA_expressionPreprocessing and miRNA_removeOutliers), thus containing the columns 'Subject', 'miRNA', 'Mean', 'StdDev', 'SampleSize'. Any other column is ignored, and any missing column forbids execution. If the 'Performance analysis mode' is selected (see inputTargetList), the dataset has to contain the 'Class' column as well.
inputMiRNAList	List of miRNAs to be used by the classifier. The chosen miRNAs must be present in the 'miRNA' column of the inputDataset.
coeffList	List of coefficients for the classifier. The number of coefficients must be the same as the number of used miRNAs and listed in the same order.
inputThreshold	Diagnostic threshold data frame for the classifier. The data frame must comply with the output format of the classifier setup function (miRNA_classifierSetup), thus containing the columns 'Threshold', 'DeltaThreshold', 'ChiUp', 'DChiUp', 'ChiDown', 'DChiDown'. Any other column is ignored.
inputTargetList	List of classes to use as target. Providing this argument corresponds to selecting the 'Performance analysis mode'. Consequently, inputDataset is expected to

contain the 'Class' column as well. The chosen target must correspond to at least one of the classes present in the 'Class' column of the inputDataset.

inputVersusList

List of classes to use as versus in 'Performance analysis mode'. If the argument is left empty, all classes present in the 'Class' column of the inputDataset, minus the Target classes, are used as Versus.

saveOutputFile Boolean option setting whether results are written to file (TRUE) or not (FALSE). Default is FALSE.

outputFileBasename

Name of the output file where the diagnosis results are to be stored. If not assigned, a filename is automatically generated.

sep

Field separator character for the output file; the default is tabulation.

plotFormat

String specifying the format of generated graphic files (plots): can either be "pdf" (default) or "png".

scorePlotParameters

String specifying the y-axis parameters of the score plot. If empty, the axis is configured by assessing suitable parameters from the data. This argument is meaningful only if saveOutputFile is set to TRUE. The string has to comply with the format "yl_yu_yt", where: yl is the lower y limit; yu is the upper y limit; yt is the interval between tics along the axis.

scorePlotAscending

Boolean option to set the direction in which samples are ordered: TRUE corresponds to samples ordered by ascending score, FALSE corresponds to samples ordered by descending score. Default is TRUE. This argument is meaningful only if saveOutputFile is set to TRUE.

colorComplementFlag

Boolean option to switch between the default palette (FALSE) and its inverted version (TRUE). Default is FALSE, corresponding to target samples reported in blue and versus samples in red. This argument is meaningful only if saveOutputFile is set to TRUE.

histogramParameters

(Used in 'Performance analysis mode' only). String specifying the parameters used to build the histogram. If empty, the histogram is built by assessing suitable parameters from the data. This parameter is meaningful only if saveOutputFile is set to TRUE. The string has to comply with the format "xl_xu_bw", where: xl is the lower boundary of the leftmost bin; xu is the upper boundary of the rightmost bin; bw is the bin width.

Details

This function can also run in 'Performance analysis mode' to evaluate the performance of a classifier by running it on an already-classified dataset. In order to carry out performance analysis, inputDataset has to contain a 'Class' column. Moreover, a list of Target classes has to be provided to the function via the inputTargetList argument.

Value

A data frame containing the columns 'Subject', 'Diagnosis' and 'Score'.

Please refer to the user manual installed in "/path-to-library/MiRNAQCD/doc/manual.pdf" for detailed function documentation. The path "/path-to-library" can be shown from R by calling ".libPaths()"

Examples

```
requiredDataFile = paste(system.file(package="MiRNAQCD"),
"/extdata/test_dataset_beta_clean.dat", sep='')
myDataFrame <- read.table(file=requiredDataFile, header=TRUE)
requiredThresholdFile = paste(system.file(package="MiRNAQCD"),
"/extdata/test_dataset_alpha_threshold.txt", sep='')
thresholdDataFrame <- read.table(file=requiredThresholdFile, header=TRUE)
mirnaToUse <- c("FX", "FZ")
coefficientsToUse <- c(1.0, -1.0)

## Classification
classifiedDataset <- miRNA_diagnosis(myDataFrame, mirnaToUse, coefficientsToUse,
thresholdDataFrame)
```

miRNA_expressionPreprocessing
Pre-processing of datasets.

Description

This function carries out the pre-processing required by the other functions of the miRNA-QC-and-Diagnosis package.

Usage

```
miRNA_expressionPreprocessing(inputDataset, multipletSize)
```

Arguments

inputDataset	Dataset (data frame) to be pre-processed. The data frame must contain the columns 'Subject', 'miRNA', 'Value' and possibly 'Class'. Any other column is ignored, and any missing column forbids execution.
multipletSize	Size of the multiplets to be considered. Any multiplet of different size is ignored.

Value

A pre-processed data frame, containing the columns 'Subject', 'miRNA', 'Mean', 'StdDev', 'SampleSize', and possibly 'Class'.

Please refer to the user manual installed in "/path-to-library/MiRNAQCD/doc/manual.pdf" for detailed function documentation. The path "/path-to-library" can be shown from R by calling ".libPaths()"

Examples

```
requiredFile = paste(system.file(package="MiRNAQCD"), "/extdata/test_dataset_alpha.dat", sep='')
myDataFrame <- read.table(file=requiredFile, header=TRUE)
myPreprocessedDataFrame <- miRNA_expressionPreprocessing(myDataFrame, 3)
```

miRNA_loadDiagnosticThreshold
Load diagnostic threshold values.

Description

This function loads from file a data frame containing the diagnostic threshold values of a trained classifier.

Usage

```
miRNA_loadDiagnosticThreshold(inputFileName, sep = "")
```

Arguments

inputFileName	Name of the file to be loaded. The file has to contain at least the columns 'Threshold', 'DeltaThreshold', 'ChiUp', 'DChiUp', 'ChiDown', 'DChiDown' (not necessarily in this order).
sep	Field separator character; the default is any white space (one or more spaces or tabulations).

Value

A data frame containing all the columns present in the file.

Please refer to the user manual installed in "/path-to-library/MiRNAQCD/doc/manual.pdf" for detailed function documentation. The path "/path-to-library" can be shown from R by calling ".libPaths()"

Examples

```
requiredFile = paste(system.file(package="MiRNAQCD"),
"/extdata/test_dataset_alpha_threshold.txt", sep='')
threshold <- miRNA_loadDiagnosticThreshold(requiredFile)
```

`miRNA_loadQualityThreshold`*Load quality threshold values.*

Description

This function loads from file a data frame containing the quality threshold values for a set of miRNAs.

Usage

```
miRNA_loadQualityThreshold(inputFileName, sep = "")
```

Arguments

<code>inputFileName</code>	Name of the file to be loaded. The file has to contain at least the columns 'miRNA', 'QualityThreshold' (not necessarily in this order).
<code>sep</code>	Field separator character; the default is any white space (one or more spaces or tabulations).

Value

A data frame containing the columns 'miRNA' and 'QualityThreshold'.

Please refer to the user manual installed in "/path-to-library/MiRNAQCD/doc/manual.pdf" for detailed function documentation. The path "/path-to-library" can be shown from R by calling ".libPaths()"

Examples

```
requiredFile = paste(system.file(package="MiRNAQCD"), "/extdata/test_dataset_alpha_qt.dat", sep='')  
qtDataFrame <- miRNA_loadQualityThreshold(requiredFile)
```

`miRNA_plotHistograms` *Plot of classifier distribution histograms (Target and Versus).*

Description

Generates a plot of the histograms of the score values for the target and versus sets.

Usage

```
miRNA_plotHistograms(
  inputDataset,
  thresholdFrame = character(),
  outputFileLabel,
  plotFormat = "pdf",
  histogramParameters = character(),
  colorComplementFlag = FALSE
)
```

Arguments

inputDataset Dataset (data frame) to be used for the plot.

thresholdFrame Diagnostic threshold values (data frame) to be used for the plot. If omitted, no threshold is drawn on the histogram.

outputFileLabel Label to be used to build the name of the output file.

plotFormat String to set the format of the output file. Can either be 'pdf' (default) or 'png'.

histogramParameters String specifying the parameters used to build the histogram. If empty, the histogram is built by assessing suitable parameters from the data. The string has to comply with the format "xl_xu_bw", where: xl is the lower boundary of the leftmost bin; xu is the upper boundary of the rightmost bin; bw is the bin width.

colorComplementFlag Boolean option to switch between the default palette (FALSE) and its inverted version (TRUE). Default is FALSE, corresponding to target samples reported in blue and versus samples in red.

Value

A ggplot object containing the plot.

This function is not exported to the package NAMESPACE, but it is called by other functions of the same package.

miRNA_plotROC	<i>Plot of classifier ROC curve.</i>
---------------	--------------------------------------

Description

Generates a plot of the ROC curve out of the input dataset.

Usage

```
miRNA_plotROC(inputDataset, outputFileLabel, plotFormat = "pdf")
```

Arguments

`inputDataset` Dataset (data frame) to be used for the plot.

`outputFileLabel` Label to be used to build the name of the output file.

`plotFormat` String to set the format of the output file. Can either be 'pdf' (default) or 'png'.

Value

A ggplot object containing the plot.

This function is not exported to the package NAMESPACE, but it is called by other functions of the same package.

`miRNA_plotThresholds` *Plot of scores and thresholds of a Bayes classifier.*

Description

Generates a plot of the classifier scores of a dataset, as well as the corresponding classifier thresholds.

Usage

```
miRNA_plotThresholds(
  inputDataset,
  thresholdsFrame,
  outputFileLabel,
  plotFormat = "pdf",
  scorePlotParameters = character(),
  colorComplementFlag = FALSE
)
```

Arguments

`inputDataset` Dataset (data frame) to be used for the plot.

`thresholdsFrame` Diagnostic threshold values (data frame) to be used for the plot.

`outputFileLabel` Label to be used to build the name of the output file.

`plotFormat` String to set the format of the output file. Can either be 'pdf' (default) or 'png'.

`scorePlotParameters` String specifying the parameters of the score plot y-axis. If empty, the axis is configured by assessing suitable parameters from the data. The string has to comply with the format "yl_yu_yt", where: yl is the lower y limit; yu is the upper y limit; yt is the interval between tics along the axis.

colorComplementFlag

Boolean option to switch between the default palette (FALSE) and its inverted version (TRUE). Default is FALSE, corresponding to target samples reported in blue and versus samples in red.

Value

A ggplot object containing the plot.

This function is not exported to the package NAMESPACE, but it is called by other functions of the same package.

miRNA_removeOutliers *Removal of dataset outliers.*

Description

This function removes outliers from a given dataset according to a set of quality threshold values.

Usage

```
miRNA_removeOutliers(inputDataset, qualityThresholdFrame)
```

Arguments

inputDataset Dataset (data frame) to be cleaned of outliers. The data frame must comply with the output format of the preprocessing function (`miRNA_expressionPreprocessing`), thus containing the columns 'Subject', 'miRNA', 'Mean', 'StdDev', 'SampleSize' and possibly 'Class'. Any other column is ignored, and any missing column forbids execution.

qualityThresholdFrame Critical sigma values (data frame) to be used. The data frame must comply with the output format of the `ebbc` function for critical sigma assessment (`miRNA_assessQualityThreshold`), thus containing the columns 'miRNA' and 'QualityThreshold'. Any other column is ignored, and any missing column forbids execution.
Beware! Entries of the dataset for which 'miRNA' is not present in the data frame of critical sigma values are copied in output without any filtering.

Value

A data frame corresponding to a copy of the input dataset devoid of outliers. The output data frame thus contains the columns 'Subject', 'miRNA', 'Mean', 'StdDev', 'Variance', 'SampleSize' and possibly 'Class'.

Please refer to the user manual installed in `"/path-to-library/MiRNAQCD/doc/manual.pdf"` for detailed function documentation. The path `"/path-to-library"` can be shown from R by calling `".libPaths()"`

Examples

```
requiredDataFile = paste(system.file(package="MiRNAQCD"),
"/extdata/test_dataset_alpha_prep.dat", sep='')
myDataFrame <- read.table(file=requiredDataFile, header=TRUE)
requiredQtFile = paste(system.file(package="MiRNAQCD"),
"/extdata/test_dataset_alpha_qt.dat", sep='')
qtDataFrame <- read.table(file=requiredQtFile, header=TRUE)
myDataFrameCleaned <- miRNA_removeOutliers(myDataFrame, qtDataFrame)
```

Index

miRNA_assessQualityThreshold, [2](#)
miRNA_classifierSetup, [3](#)
miRNA_diagnosis, [6](#)
miRNA_expressionPreprocessing, [8](#)
miRNA_loadDiagnosticThreshold, [9](#)
miRNA_loadQualityThreshold, [10](#)
miRNA_plotHistograms, [10](#)
miRNA_plotROC, [11](#)
miRNA_plotThresholds, [12](#)
miRNA_removeOutliers, [13](#)