SoftwareX 12 (2020) 100569

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

MiRNA-QC-and-Diagnosis: An R package for diagnosis based on MiRNA expression

Michele Castelluzzo [a], Alessio Perinelli [a], Simone Detassis [b], Michela Alessandra Denti [b], Leonardo Ricci [a,c,*]

[a] Department of Physics, University of Trento, 38123 Trento, Italy
[b] Department of Cellular, Computational and Integrative Biology (CIBIO), University of Trento, 38123 Trento, Italy
[c] CIMeC, Center for Mind/Brain Sciences, University of Trento, 38068 Rovereto, Italy

ARTICLE INFO

Article history:
Received 5 May 2020
Received in revised form 30 June 2020
Accepted 6 July 2020

Keywords:
MiRNA
Biomarkers
Diagnosis
Statistical analysis

ABSTRACT

The possibility of using microRNA (miRNA) levels as diagnostic and prognostic tools to detect different pathologies requires the implementation of reliable classifiers, whose training and use call for quality control of data corresponding to miRNA expression. In this work we present the MiRNA-QC-and-Diagnosis package. The package provides a set of functions for the R environment that implement the required quality control steps and thereupon allow to train, use and optimize a Bayesian classifier for diagnosis based on the measured miRNA expressions. The package thus makes up a complete and dedicated analytical toolbox for miRNA-based diagnosis.

Code metadata

Current code version	v1.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_197
Code Ocean compute capsule	
Legal Code License	GNU GPL v3
Code versioning system used	None
Software code languages, tools, and services used	R
Compilation requirements, operating environments & dependencies	Linux, Windows and MacOS. Requires R ($\geq$ 4.0)
If available Link to developer documentation/manual	User manual
Support email for questions	leonardo.ricci@unitn.it

1. Introduction

MicroRNA (miRNA) expressions are known to make up suitable biomarkers to detect and classify different kinds of pathologies [1–3], most notably cancer [4,5]. Consequently, many studies have been devoted to the investigation of classifiers that rely on the measured expressions of miRNAs [6–9], obtained for example by means of quantitative polymerase chain reaction (PCR) assays [10]. A possible classification approach is to compute a sample "score" out of the measured miRNA expressions and then compare this score against a threshold that has to be suitably set in a "training step" by relying on pre-classified datasets.

In this work we present the MiRNA-QC-and-Diagnosis package that implements a binary classifier for miRNA-based diagnosis. The classifier training, namely the determination of the diagnostic threshold, is carried out following a Bayesian approach [5,11]. Once a trained classifier is available, the package provides a function to classify new data. Quality control (QC) of datasets, which makes up a crucial, preliminary step both in training and diagnosis, is also implemented in the package through functions that allow to identify and remove outliers. In addition, the package provides a tool to carry out a statistical analysis of miRNA expressions, including the evaluation of cross-correlation between miRNA expressions, which can give insight into possible ways to improve the classifier's performance [5].

The algorithm implemented in the package was first devised and tested on a prototypical case of classifying samples either

* Corresponding author at: Department of Physics, University of Trento, 38123 Trento, Italy.
E-mail address: leonardo.ricci@unitn.it (L. Ricci).

https://doi.org/10.1016/j.softx.2020.100569

to adenocarcinomas or to squamous cell carcinomas. The mathematical aspects of the algorithm, as well as the clinical case, are thoroughly described in a work by Ricci et al. [5] entitled "Statistical analysis of a Bayesian classifier based on the expression of miRNAs", which makes up the main reference to the present work.

2. Bayesian classifier relying on miRNA expression multiplets

The method described in this package is graphically summarized in Fig. 1.

The input data required by the algorithm are subject-related measurements of miRNA expressions. Typically, for each subject, a set – i.e. a multiplet – of measured values is available for each miRNA of interest. In the framework of miRNA-based diagnosis, miRNA expressions are usually provided in triplicates [5]. A dataset that is used to train the classifier must contain, for each subject, the *a priori* classification, i.e. a clinical diagnosis that is typically inferred by immunohistochemical analysis and gene profiling.

In the QC step, data are first prepared by condensing the information contained in each multiplet into a pair given by the sample mean $\bar{x}$ and the sample standard deviation s of the multiplet's original values. To this purpose, the size m of the multiplets has to be set by the user. Multiplets with size different from the selected one are discarded.

The core of the QC step of the algorithm consists in removing outliers from the dataset. This operation is carried out by considering the variance of each multiplet: given a miRNA, and assuming the multiplet variances to be distributed as a χ^2, a multiplet can be identified as an outlier whenever its variance exceeds a threshold $\sigma^2_{\max}$ corresponding to a given level of significance α. The corresponding standard deviation $\sigma_{\max}$, which is referred to in the package as "quality threshold", can be either inferred through a statistical analysis on a given dataset or already known from previous assessments.

The second step of the algorithm consists in training a Bayesian classifier to distinguish between two sets of classes, henceforth referred to as the "Target set" T and the "Versus set" V. Both T and V are sets of classes – possibly a single one – into which subjects have been *a priori* classified by using the alternative diagnostic techniques mentioned above. The key quantity a classifier relies on is taken to be a linear combination of the sample means of suitably selected miRNAs [5]; the numerical value of the linear combination makes up the subject-related score.

The selection of miRNAs and the choice of the related coefficients, i.e. the definition of the linear combination providing the score, is a crucial step of the algorithm. Unless a prior knowledge is given, this step requires a preliminary statistical analysis, which is performed separately on each miRNA, as follows. Subjects are grouped according to their *a priori* classification into the two T and V sets and thereupon the sample mean and sample standard deviation of that miRNA's values are computed for both sets. The analysis provides the p value of the Shapiro–Wilk test for normality as well as the p value of the Student's t-test to check the null hypothesis that the Target and Versus sets have the same population mean. The result is used to determine whether a miRNA is suitably discriminating between the two sets.

In addition, a matrix containing the linear correlation coefficients between the pairs of all available miRNAs is computed, along with the corresponding p value matrix. The reason is that correlation can be used to improve the accuracy of a classifier by combining a discriminating miRNA with a second, not necessarily discriminating one [5]. Besides the matrices of the linear correlation coefficients and of the p values, a matrix of the optimal coefficients for each miRNA pair is produced.

The method also allows to take into account a possible experimental bias in the measurement of miRNA expressions. For example, measurements taken at different times and/or by using different setups can lead to different values of the multiplets' sample means. A possible solution (see for example Ref. [12]) is the use of a specific miRNA as a "normalizer". Consequently, rather than considering a single miRNA value, the analysis is carried out on the difference between that value and the corresponding normalizer value.

When a suitable linear combination is identified by specifying the set of miRNAs $\{F_i\}$ and the coefficients $\{c_i\}$, the training phase consists in assessing a diagnostic threshold χ so that, if a score does (does not) overcome it, the corresponding subject is classified into the T (V) set. The value of χ is assessed in order to obtain the maximum accuracy (rate of correct responses). Two additional thresholds are assessed, namely $\chi_{90:10}$ and $\chi_{10:90}$, corresponding to the 90% and 10% likelihood of the subject to belong to the Target set, respectively.

Once diagnostic thresholds are given, diagnosis can be performed on any new subject according to the very same rule.

3. Software framework

3.1. Software architecture

The package consists of a set of functions for the R environment [13]. Documentation for each function is included in the package and is directly accessed from the R environment by means of the `help()` command. A pdf user manual concerning download, setup, and use of the functions is provided within the package.

3.2. Quality control: preprocessing and outlier removal

Four functions, `miRNA_expressionPreprocessing`, `miRNA_assessQualityThreshold`, `miRNA_loadQualityThreshold`, `miRNA_removeOutliers`, concern the QC stage leading to the removal of outliers from the input dataset. The diagram in Fig. 2 shows the typical workflow that uses these functions.

The input data have to be formatted as a data frame object, with columns labelled as 'Subject', 'miRNA' and 'Value'. A training dataset has to contain an additional column labelled 'Class' corresponding to the *a priori* diagnosis.

The preprocessing occurs through the function `miRNA_expressionPreprocessing`. A first action consists in removing rows containing non-numerical entries in the 'Value' column. If the resulting number of rows for a given subject and miRNA coincides with the multiplet size m, the sample mean $\bar{x}$ and the sample standard deviation s are computed for that multiplet. The preprocessed data frame contains the columns 'Subject', 'miRNA', 'Mean', 'StdDev', 'SampleSize' and, in the case of a training dataset, the 'Class' column too.

Given the preprocessed data frame and a level of significance α set by the user (for example $\alpha = 0.05$), the function `miRNA_assessQualityThreshold` evaluates the quality threshold $\sigma_{\max}$ for each miRNA. The output is a data frame whose columns 'miRNA' and 'QualityThreshold' contain the miRNAs available in the dataset and the corresponding quality thresholds. Alternatively, it is possible to load a pre-determined file by means of the function `miRNA_loadQualityThreshold`.

Outliers are filtered out by the function `miRNA_removeOutliers`. The function, which is fed with the preprocessed data frame and the quality threshold data frame, returns a copy of the preprocessed data frame devoid of outliers.

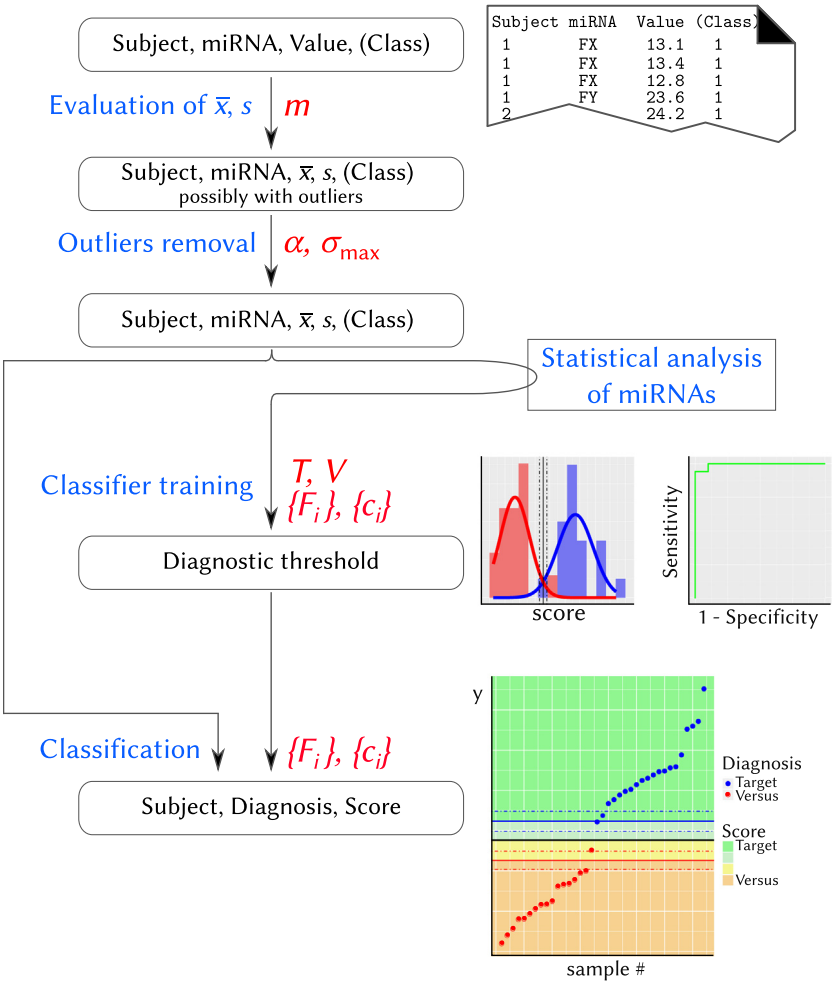

Fig. 1. Graphical summary of the classification pipeline.

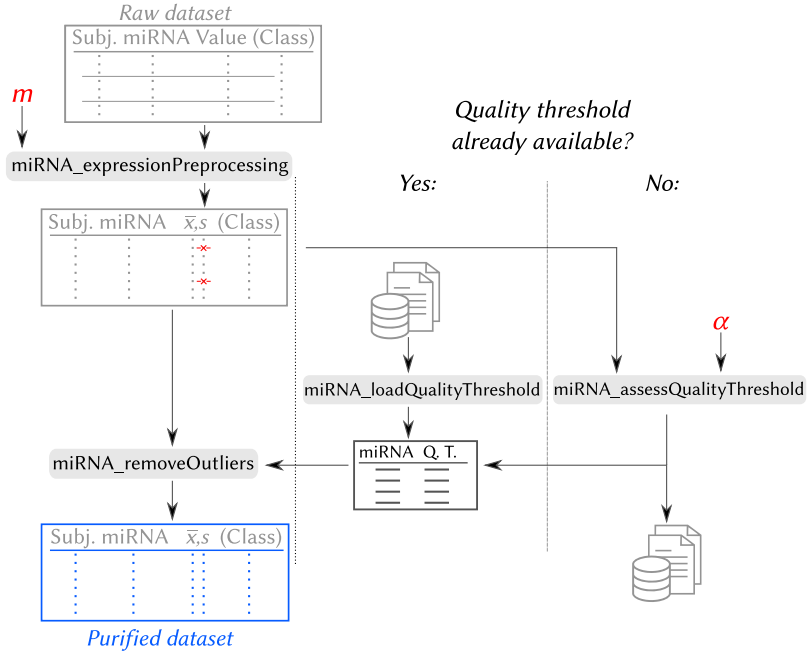

Fig. 2. Pipeline concerning the QC stages, namely preprocessing and outlier removal.

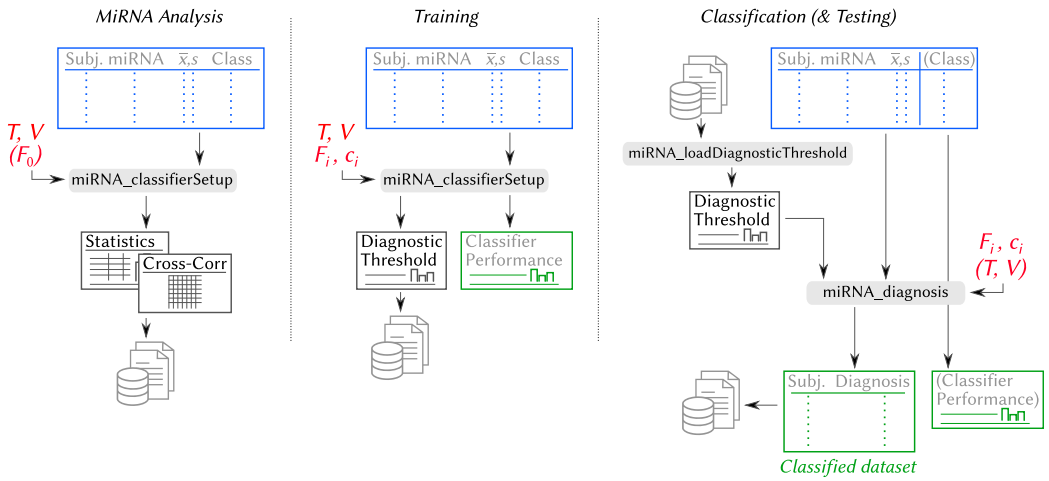

Fig. 3. Pipeline concerning the training of a Bayesian classifier and its use.

3.3. Training of the Bayesian classifier

The functions `miRNA_classifierSetup`, `miRNA_loadDiagnosticThresholds`, `miRNA_diagnosis` concern the miRNA statistical analysis, the training of the Bayesian classifier, and the use of the classifier for diagnosis and testing. Fig. 3 shows the typical workflow concerning these three functions.

The function `miRNA_classifierSetup` performs both miRNA analysis and classifier training. The function takes in a preprocessed data frame, two lists of classes to be used as Target and Versus sets, a list of miRNAs, and a list of coefficients. The last two lists describe the score, namely the linear combination which the classifier relies on. Depending on the inputs to the function, two different function modes can be evoked. The preprocessed data frame and the Target set are mandatory in all cases. If no Versus set is provided, the function automatically assumes all the remaining classes in the dataset to belong to the Versus set. Alternatively, the Versus set can be explicitly provided as an additional input list.

In *Analysis mode*, i.e. when a statistical analysis of the available miRNAs has to be carried out, no further inputs are required. However, if a single miRNA is given in input, it is used as normalizer. In *Analysis mode* the function `miRNA_classifierSetup` produces in output a data frame containing the results of the analysis as well as histogram plots regarding miRNA expression distributions both in the case of the Target set and of the Versus set.

In *Training mode*, a list of miRNAs and the corresponding list of coefficients are fed to the function. The two lists again describe the linear combination which the classifier relies on, i.e. the score. The main output parameter of the classifier is the diagnostic threshold χ, which is produced along with the thresholds $\chi_{90:10}$, $\chi_{10:90}$.

In addition the function `miRNA_classifierSetup` produces a set of parameters to express the classifier's performance: confusion matrix; accuracy (rate of correct responses); specificity and sensitivity; F1-score; separation d', normalized by the standard deviation, between score distributions of the Target and the Versus set; area-under-curve of the Receiver Operating Characteristic (ROC) curve [14]. The function also produces histogram plots of the score distribution, a scatter plot of the score values against the thresholds, and the ROC curve plot.

3.4. Diagnosis through the Bayesian classifier

If already available, diagnostic threshold values can be loaded by using the function `miRNA_loadDiagnosticThresholds`.

Finally, diagnosis, namely the classification of subjects in a dataset as belonging to the 'Target' or 'Versus' class, is performed through the function `miRNA_diagnosis`. Taking as input a preprocessed dataset and a diagnostic threshold data frame, this function produces a dataset containing the columns 'Subject', 'Diagnosis', 'Score'.

In order to test the classifier's performance, provided that an *a priori* classification is available, the function also produces the set of parameters, and the related plots, described above.

4. Illustrative examples

In this section three examples of analysis pipelines relying on the MiRNA-QC-and-Diagnosis functions are presented. The datasets and the scripts containing the related function calls are stored in the `/examples/` directory within the package:

- Synthetic data
  - `synthetic_dataset_alpha.dat`
  - `synthetic_dataset_beta.dat`
  - `example_synthetic_dataset.R`
- Real data 1 (see Ref. [5])
  - `real_dataset_1_training.dat`
  - `real_dataset_1_testing.dat`
  - `real_dataset_1_additional.dat`
  - `example_real_dataset_1.R`
- Real data 2 (see Ref. [3])
  - `real_dataset_2_training.dat`
  - `real_dataset_2_testing.dat`
  - `example_real_dataset_2.R`

In the first example data are simulated: while they mimic real experimental data, they do not correspond to any real subject. On the other hand, data used in the second and third example are real. The analysis of these datasets are the topics of Refs. [5] and [3]. The data of the second example are also available [15] as supplementary material to Ref. [5].

In the following, only the first example is described. Details of the example pipeline are discussed in the MiRNA-QC-and-Diagnosis user manual (`/docs/manual.pdf`).

With regard to the other two examples, data sources, analysis procedures and outcomes are thoroughly described in the respective references. The two scripts enclosed in the package allow to reproduce the numerical results and the graphical plots presented in the original works.

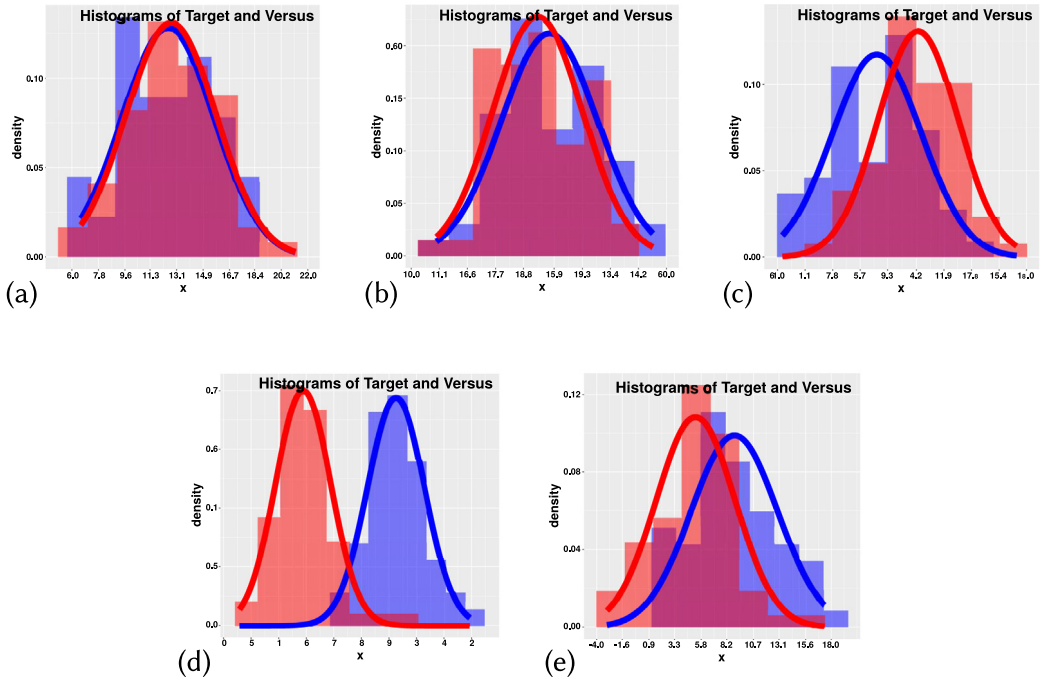

Fig. 4. Histograms, for each analysed miRNA, of the expression values both in the case of the Target (blue) and the Versus (red) set. (a) FX. (b) FY. (c) FZ. (d) ΔFX = FX−FZ. (e) ΔFY = FY−FZ. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.1. Data format, loading and preprocessing

The raw dataset used in this example is stored in the file `synthetic_dataset_alpha.dat` and contains data corresponding to 120 subjects, which are labelled by integer numbers. For each subject, data corresponding to three miRNAs, labelled as FX, FY, FZ, are present. With the exception of the very last subject, for each miRNA three values (a triplicate, i.e. a multiplet of size 3) are provided. The last subject has multiplets of size 2 (doublets) instead. Because a 'Class' column is present, the dataset is eligible for classifier training.

The first step is to load the dataset file. Because the column names are written in the first row, it is sufficient to call the R function `read.table` with the "header" argument set to `TRUE`.

Preprocessing is performed by the `miRNA_expressionPreprocessing` function, with "multipletSize" set to 3. Consequently, subject #120 is excluded from the resulting preprocessed data frame.

4.2. Outlier removal

To remove outliers, a quality threshold has first to be assessed for each miRNA. This assessment is carried out by feeding the preprocessed data frame to the function `miRNA_assessQualityThreshold`. In this example, the significance level α is set to 0.05; the resulting quality threshold values turn out to be 0.51, 0.50, 0.53 for the miRNAs FX, FY, FZ, respectively.

Once the quality thresholds are available, the preprocessed data frame is purified by removing possible outliers through the function `miRNA_removeOutliers`. As a consequence, the purified data frame (345 entries) turns out to be smaller than the preprocessed one (357 entries).

4.3. Features analysis

The purified data frame contains the classes "A", "B", "C". In this example, the first class alone makes up the Target set, while the remaining two classes are assumed to belong to the Versus set. Because the optimal choice of miRNAs and the related coefficients to be used by the classifier is unknown, the function `miRNA_classifierSetup` is first run in *Analysis mode*. The resulting histogram plots are shown in Fig. 4(a)–(c) corresponding to the distributions of the miRNAs FX, FY, FZ. None of the three miRNAs appears to discriminate between the Target and the Versus set. On the other hand, Fig. 4(d), (e) show the histogram plots in the case of FZ taken as a normalizer, i.e. for the linear combinations ΔFX = FX−FZ and ΔFY = FY−FZ. According to the Student's t-test analysis (the Shapiro–Wilk test of normality produced positive outcomes), the difference ΔFX = FX−FZ efficiently discriminates between the Target and the Versus set. It is then possible to use the linear combination $1\cdot\mathrm{FX}+(-1)\cdot\mathrm{FZ}$ as the classifier's score.

4.4. Training of a Bayesian classifier

Once a linear combination is established, the Bayesian classifier is trained by means of the function `miRNA_classifierSetup` set in *Training mode*. For the dataset used in this example, the classifier diagnostic threshold turns out to be $\chi = 4.6 \pm 0.1$. The function `miRNA_classifierSetup` also produces the three graphical files shown in Fig. 5.

4.5. Classification of a dataset

The `/examples/` directory stores a second dataset, namely `synthetic_dataset_beta.dat`, which includes simulated data for 200 subjects. In this case, no 'Class' column is present. Loading and preprocessing are identical to the case of the dataset `synthetic_dataset_alpha.dat` presented above. After QC, the dataset is reduced to 183 subjects.

The dataset is classified by means of the function `miRNA_`diagnosis. The lists of miRNAs and coefficients are the same as the ones used in the training step. In addition, the diagnostic thresholds yielded by the training process are fed to the function. The classification assigns 93 subjects to the Target set and 90 subjects to the Versus set.

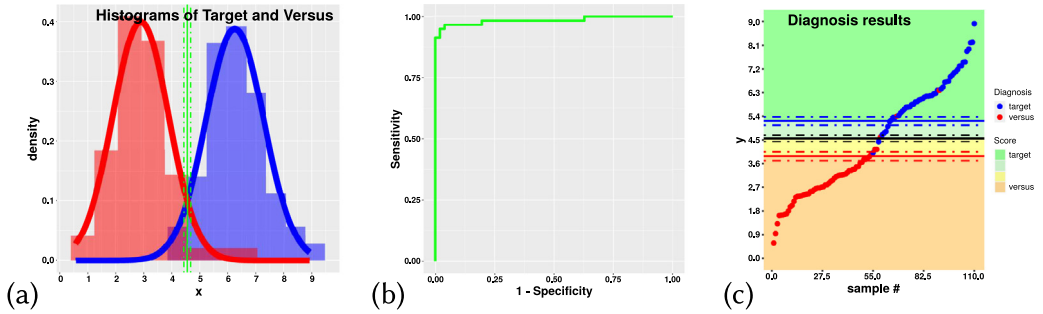

Fig. 5. Classifier training outcomes. (a) Histograms of the score distributions for the Target set (blue) and the Versus set (red). The green lines correspond to the diagnostic threshold (solid line) and the corresponding uncertainty (dash-dotted lines). (b) ROC curve. (c) Scatter plot of the score values against the diagnostic thresholds. The black, blue and red solid lines correspond to the diagnostic threshold χ, $\chi_{90:10}$ and $\chi_{10:90}$, respectively, while the dash-dotted lines correspond to the related uncertainties. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Conclusions

In this work, the MiRNA-QC-and-Diagnosis package is presented. The package implements, in the R environment, a complete set of functions that carry out quality control tasks on input data corresponding to measured values of miRNA expressions, and allow both to train and use a Bayesian classifier for diagnostic purposes.

Many powerful algorithms exist to set up a classifier, like support vector machines [16,17], C4.5 [18,19], linear and quadratic discriminant analysis [20,21], k-nearest neighbours algorithms [22,23]. For each approach, an R implementation is available. Despite these methods are quite versatile to optimize the decisional parameters on a training set, a probabilistic approach is more suited to provide an immediate quantification of the reliability of a classifier's performance. The advantage of using the Bayesian approach implemented within the MiRNA-QC-and-Diagnosis package relies on the verification of the normality of the different distributions of interest. This advantage allows for the estimation of the classifier's reliability even in the case of a limited size of the available data sets.

Beside providing a binary classifier based on a Bayesian approach, the MiRNA-QC-and-Diagnosis package tackles two aspects that are crucial to set up a reliable classifier: a quality control stage for the identification and removal of outliers; a suitable statistical analysis of input data to improve the classifier's performance by exploiting correlations. The MiRNA-QC-and-Diagnosis package thus makes up a dedicated toolbox to develop pipelines for diagnosis based on measured miRNA expressions.

A future development of the method presented in this paper can regard the automatization of the procedure that leads to the optimal choice of the miRNAs to be used within the classifier and to the trimming of the related coefficients.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Grasso M, Piscopo P, Confaloni A, Denti MA. Circulating miRNAs as biomakers for neurodegenerative diseases. Molecules 2014;19:6891–910. http://dx.doi.org/10.3390/molecules19056891.

[2] Grasso M, Piscopo P, Talarico G, Ricci L, Crestini A, Tosto G, Gasparini M, Bruno G, Denti MA, Confaloni A. Plasma microRNA profiling distinguishes patients with frontotemporal dementia from healthy subjects. Neurobiol Aging 2019;84. http://dx.doi.org/10.1016/j.neurobiolaging.2019.01.024, 240.e1–240.e12.

[3] Detassis S, Del Vescovo V, Grasso M, Masella S, Cantaloni C, Cima L, Cavazza A, Graziano P, Rossi G, Barbareschi M, Ricci L, Denti MA. MiR375-3p distinguishes low-grade neuroendocrine from non-neuroendocrine lung tumors in FFPE samples. Front Mol Biosci 2020;7:86. http://dx.doi.org/10.3389/fmolb.2020.00086.

[4] Del Vescovo V, Grasso M, Barbareschi M, Denti MA. MicroRNAs as lung cancer biomarkers. World J Clin Oncol 2014;5:604–20. http://dx.doi.org/10.5306/wjco.v5.i4.604.

[5] Ricci L, del Vescovo V, Cantaloni C, Grasso M, Barbareschi M, Denti MA. Statistical analysis of a Bayesian classifier based on the expression of miRNAs. BMC Bioinformatics 2015;16:287. http://dx.doi.org/10.1186/s12859-015-0715-9.

[6] Lebanony D, Benjamin H, Gilad S, Ezagouri M, Dov A, Ashkenazi K, Gefen N, Izraeli S, Rechavi G, Pass H, Nonaka D, Li J, Spector Y, Rosenfeld N, Chajut A, Cohen D, Aharonov R, Mansukhani M. Diagnostic assay based on hsa-miR-205 expression distinguishes squamous from nonsquamous non-small-cell lung carcinoma. J Clin Oncol 2009;27:2030–7. http://dx.doi.org/10.1200/JCO.2008.19.4134.

[7] Tan X, Qin W, Zhang L, Hang J, Li B, Zhang C, Wan J, Zhou F, Shao K, Sun Y, Wu J, Zhang X, Qiu B, Li N, Shi S, Feng X, Zhao S, Wang Z, Zhao X, Chen Z, Mitchelson K, Cheng J, Guo Y, He J. A 5-microRNA signature for lung squamous cell carcinoma diagnosis and hsa-miR-31 for prognosis. Clin Cancer Res 2011;17:6802–11. http://dx.doi.org/10.1158/1078-0432.CCR-11-0419.

[8] Lee HW, Lee EH, Ha SY, Lee CH, Chang HK, Chang S, Kwon KY, Hwang IS, Roh MS, Seo JW. Altered expression of microRNA miR-21, miR-155, and let-7a and their roles in pulmonary neuroendocrine tumors. Pathol Int 2012;62:583–91. http://dx.doi.org/10.1111/j.1440-1827.2012.02845.x.

[9] Huang W, Hu J, Yang DW, Fan XT, Jin Y, Hou YY, Wang JP, Yuan YF, Tan YS, Zhu XZ, Bai CX, Wu Y, Zhu HG, Lu SH. Two microRNA panels to discriminate three subtypes of lung carcinoma in bronchial brushing specimens. Am J Respir Crit Care Med 2012;186:1160–7. http://dx.doi.org/10.1164/rccm.201203-0534OC.

[10] Benes V, Castoldi M. Expression profiling of microRNA using real-time quantitative PCR, how to use it and what is available. Methods 2010;50:244–9. http://dx.doi.org/10.1016/j.ymeth.2010.01.026.

[11] Gorunescu F, Belciug S. Ch. intelligent decision support systems in automated medical diagnosis. In: Advances in biomedical informatics. Springer International Publishing; 2018, p. 161–86. http://dx.doi.org/10.1007/978-3-319-67513-8_8.

[12] Peltier HJ, Latham GJ. Normalization of microRNA expression levels in quantitative RT-PCR assays: identification of suitable reference RNA targets in normal and cancerous human solid tissues. RNA 2008;14:844–52. http://dx.doi.org/10.1261/rna.939908.

[13] R project webpage: https://www.r-project.org/ (accessed Dec. 2019).

[14] Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez J-C, Müller M. PROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinformatics 2011;12:77. http://dx.doi.org/10.1186/1471-2105-12-77.

[15] Data are available at: https://static-content.springer.com/esm/art%3A10.1186%2Fs12859-015-0715-9/MediaObjects/12859_2015_715_MOESM1_ESM.txt.

[16] Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods. Cambridge, UK: Cambridge university press; 2000.

[17] Support vector machines are implemented in R by the svm function of the e1071 package: https://www.rdocumentation.org/packages/e1071.

[18] Quinlan JR. C4.5: Programs for machine learning. California, USA: Morgan Kaufmann Publishers; 1993.

[19] C4.5 algorithm is implemented in R by the J48 function of the RWeka package: https://www.rdocumentation.org/packages/RWeka/.
[20] Ripley BD. Pattern recognition and neural networks. Cambridge, UK: Cambridge University Press; 1996, http://dx.doi.org/10.1017/CBO9780511812651.
[21] Linear and Quadratic Discriminant Analysis are implemented in R by the lda and qda functions of the MASS package: https://www.rdocumentation.org/packages/MASS/.
[22] Duda RO, Hart PE, Stork DG. Pattern classification. New York, USA: John Wiley & Sons; 2012.
[23] k-nearest neighbours classification is implemented in R by the knn function of the class package: https://www.rdocumentation.org/packages/class/.