ROSY Application for Selecting R Packages that Perform ROC Analysis

José Pedro Quintas[1(✉)], Francisco Machado e Costa[1], and Ana Cristina Braga[2]

[1] School of Engineering, University of Minho, Campus Gualtar, Braga, Portugal
pquintasbcl@gmail.com, frmachadoecosta@gmail.com
[2] ALGORITMI Centre, University of Minho, Campus de Azurem, Guimarães, Portugal
acb@dps.uminho.pt

Abstract. The empirical ROC curve is a powerful statistical tool to evaluate the precision of tests in several fields of study. This is a two-dimensional plot where the horizontal and vertical axis represent false positive and true positive fraction respectively, also referred to as 1-specificity and sensitivity, where precision is evaluated through a summary index, the area under the curve (AUC). Several computer tools are used to perform this analysis one of which is the R environment, this is an open source and free to use environment that allows the creation of different packages designed to perform the same tasks in distinct ways often resulting in different customization and features often providing similar results. There is a need to explore these different packages to provide an experienced user with the simplest and most robust execution of a needed analysis. This work catalogued the different R packages capable of ROC analysis exploring their performance. A shiny web application is presented that serves as a repository allowing for efficient use of all of these packages.

Keywords: Shiny application · ROC curve · R tools · Packages

1 Introduction

ROC curves are considered one of the best tools to evaluate performance in statistical tests in different areas such as machine learning, data mining, medical diagnosis, experimental psychology, economics and sociology among others. These curves are represented in a bi-dimensional plane where the x- and y-axis represent false positive and true positive fraction respectively, also referred to as 1-specificity and sensitivity, where precision is given by the area under the curve [1].

The current volumes of generated and stored data demand an efficient and swift use leading to the development of software tools in the ROC curve area which in turn popularized the method further. The ubiquitous nature of the

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12251, pp. 199–213, 2020.
https://doi.org/10.1007/978-3-030-58808-3_16

ROC curve, being used by such diverse areas, combined with the open source, free to use nature of the R environment allowed the development of several packages, that while performing the same basic analysis present several different and specific features according to the needs of the user, necessitating both quality assessment and cataloguing.

In the field of tool comparison for ROC curves it would be remiss not to mention the work of Stephan *et al.* [16], where eight different computer programs were analysed and compared. While an older article of 2003, making the presented results outdated, its methodology heavily influenced this project. The authors' goal was to select, compare, and evaluate different software using five distinct metrics for the task: data input, data output, program comfort, quality of the user manual and quality of the analysis results. All eight programs were analysed individually by each author receiving a grade for each criterion, conditional on performance.

In a different article by Robin *et al.* [13] the package *pROC* is explored, highlighting some of its features such as, smoothing, pauc (partial area under the curve), confidence intervals (CI), visual output with confidence intervals, statistical tests and CRAN availability, comparing it to *ROCR*, *verification*, *ROC* (Bioconductor) and *pcvsuite*. Considering the author of the paper is one of the package's main creators and current maintainer it is understandable that the focus is on each individual function of the *pROC* rather than the comparison with other packages but it provides useful methodology and information for the current work.

In a more recent study, Cunha and Braga [8] compare the packages *pROC*, *ROCR*, *verification*, *caTools*, *Comp2ROC*, *ROC* (Bioconductor) and *Epi*. In addition to providing a comprehensive guide of each command for all packages, it provides a comprehensive table for each package score regarding different metrics: CRAN availability, ROC processing, ROC smoothing, AUC availability, AUC availability in graphical output and CI intervals on curve comparison. This article's structure and result presentation were a great contribution to the methodology employed in this work.

The Shiny R package allows users the creation of interactive web-based applications. Although a recent addition to the R repository the package and its applications have provided an excellent resource for users to work with complex R packages or perform data analysis through a visual framework requiring only an internet connection. An example of such an application is one created by Wojciechowski, Hopkins and Upton [18] allowing the creation and sharing of pharmaceutical simulations models going as far as granting high school students a study resource for simulated variability of Ibuprofen. Jimmy Doi *et al.* [9] notes the ease of use and of application creation in shiny, emphasising its potential role in education and the ability to create user friendly, interactive and visually stimulating applications, comparing it to Java applets utilizing the R language.

Ebrahim Jahanshiri and Abdul Shariffna [10], using the package in the field of precision agriculture for mapping and data analysis, go even further and mention both R and shiny as a perfect substitute for python in the statistical analysis field.

The common praise for the shiny package in articles of such different fields with its ease of use and versatility, allowing the construction of complex yet appealing applications, served as a motivating factor to this application framework for this ROC curve and ROC package study.

2 Materials and Methods

Several R packages were chosen for comparison in accordance with the referenced papers. Working with the same dataset and criteria these tools were tested and compared. A generic table of contents was produced to enumerate features either present or missing between packages so as to readily inform the user on notable content some packages might have implemented over others.

These packages were implemented into a shiny application allowing the user to interactively perform ROC analysis with different R packages. Observing their differences and distinctiveness, user can choose the more appropriate tool for a specific task based on the checklist represented on application, desired visual output and his goal.

For replication purposes the dataset was shared to Mendeley Data and is available at https://data.mendeley.com/datasets/zx4r8mgn86/1.

2.1 R Packages and Criteria

Appropriate packages were tested, seven of which were selected for the remainder of this study as can be seen in the list below

- *pROC* [13]
- *Comp2ROC* [3]
- *ROCR* [15]
- *ROSE* [11]
- *Epi* [6]
- *ROC* - Bioconductor [5]
- *caTools* [17]

Eleven objective criteria were selected based on the works of Stephan *et al.* [16] and Cunha and Braga [8] to compare features and performance of these seven packages,

- Version
- Last update
- Empirical Curve
- AUC (Area under the curve)
- More than one curve on a chart
- Compare curves
- CI (Confidence Intervals)
- Smooth curve
- Works with inverted scales

– pAUC (Partial AUC)
– Print only one ROC curve

Version and **Last update** are meant as general information regarding the maintenance and frequency of update of a given package while **Empirical Curve** and **AUC**, referencing the ability to create the ROC curve and calculate its respective AUC, are fundamental characteristics necessary to allow a standard ROC study. In ROC tests it is often important a visual representation of different ROC curves using in this work a criterion called **More than one curve on a chart**. This is closely linked with the **Compare curves**, this criterion means ability to compare two (or more) different curves to infer their statistical differences. These were separated because a package might allow visual but not statistical comparison and vice-versa. The **CI** detail the ability of the package to produce CIs in its analysis, for example in the AUC estimation. Smoothing or curve adjustment is often used whether statistically relevant or just to create a more appealing output, therefore the ability to do so was added to the criterion - **Smooth curve**. The **Works with inverted Scales** parameter details a niche but often needed and overlooked detail of many packages. This parameter refers to scales that work inversely in a dataset. In ROC curves analysis, the values of a scale are generally used where a higher score is associated with a positive event (1), but there are scales where the values work in reverse, i.e., a lower score may correspond to the positive event of the result variable, that is equal 1, for instance, in the case of the weight of the newborn, the higher value of the weight corresponding to the greater probability of the event surviving, this is a result equal to zero. However, this is not always the case and different tools might allow a user to work with these non-standard scales or demand a manual conversion of the dataset to fit standard parameters. Another often overlooked requirement is the **pAUC** estimation, referring to a AUC on a given interval of false positive rates, a partial AUC. Finally, the **Print only one ROC curve** details the ability of a package to exclusively work with an isolated ROC curve or a need to compare groups of ROC curves and is meant as a distinguishing factor between packages focused on ROC comparison and the more generalist ROC packages.

After this selection process and package feature assessment, Table 1 was constructed to better guide users on the best packages to fit their needs on the shiny application.

2.2 Dataset

To perform a ROC analysis a dataset must include an indicator, or variable of interest and a binary outcome (negative/positive, healthy/disease, etc). A valid evaluation of all packages requires the use of the same sample of data. To this effect, two distinct paired and independent sample datasets were used.

Table 1. Packages checklist.

	pROC	ROSE	ROCR	Comp2ROC	Epi	ROC (Bioconductor)	caTools
Version	1.16.1	0.0–3	1.0–7	1.1.4	2.40	1.62.0	1.18.0
Last update	2020	2015	2015	2016	2019	2020	2020
Empirical curve	✓	✓	✓	✓	✓	✓	✓
AUC	✓	✓	✓	✓	✓	✓	✓
More than one curve on a chart	✓	✓	✓	✓	✗	✓	✗
Compare curves	✓	✗	✗	✓	✗	✗	✗
CI (Confidence Intervals)	✓	✗	✗	✓	✗	✗	✗
Smooth curve	✓	✗	✗	✗	✗	✗	✗
Works with inverted scales	✓	✓	✗	✓	✗	✗	✓
Partial AUC	✓	✗	✗	✗	✗	✓	✗
Print only one ROC curve	✓	✓	✓	✗	✓	✓	✓

The paired sample dataset, with a total of 169 observations, represents neonatal mortality, where five indicator variables, WEIGHT_AG, CRIB, NTISS, SNAP, SNAPPE access mortality risk and the resulting outcome is represented by the *result* column. Note that, unlike the remaining indicators, WEIGHT_AG is an inverted scale where higher values predict lower mortality. The independent dataset shows different sample sizes of CRIB score measured in four distinct hospitals, *Hospital1–4*, and their resulting mortality, *Res1–4* [4].

Figure 1 and Fig. 2 use the mentioned data format used to test the various packages.

	A	B	C	D	E	F
1	WEIGHT_AG	CRIB	NTISS	SNAP	SNAPPE	result
2	8	6	19	15	15	0
3	9	1	13	22	62	0
4	5	5	22	19	29	0
5	10	0	7	3	3	0

Fig. 1. First rows of the paired sample dataset.

	A	B	C	D	E	F	G	H
1	Hospital1	Res1	Hospital2	Res2	Hospital3	Res3	Hospital4	Res4
2	2	0	2	0	1	0	2	0
3	4	0	12	0	8	0	8	0
4	1	0	2	0	0	0	0	0
5	2	0	3	0	1	0	0	0

Fig. 2. First rows of the independent sample dataset.

2.3 Shiny

Shiny aims to create applications to either be run through a web browser or locally in an R environment. While this tool is mostly aimed at statisticians and data analysts as a way to share their findings, it is mostly available to all areas thanks to its basis on R, making it far more accessible [7].

The package is based around reactive programming, a programming paradigm that facilitates the automatic propagation of change of dataflows. A practical understanding of this concept is, while working with several inputs generating a specific output, be it plots, tables or text, any modification on the input will automatically generate and update the output without requiring a new command or refresh action on the user's side. Another user friendly implementation is the personalization of the shiny interface using widgets and code chunks promoted by RStudio and implemented across several other apps available on their archive [10,14].

How to Get and Use Shiny. As mentioned, this package was created and made available by RStudio and the easiest way of using it is by installing the RStudio software where all dependencies are included and even a basic template is provided.

When it comes to using created shiny applications, an R interpreter is needed, of which RStudio is the standard. However, the second popular option is using the online platform *shinyapps.io* accessible through an internet browser.

How to Build a Shiny Application. To create the application two components are necessary, a web interface and a web server, as well as an update shiny. A web interface allows users to upload files and supply instructions required for R to calculate the desired output.

To design the structure, either two files, *server.R* and *ui.R* are required or a single *app.R* file, the latter option is merely the former's structure components organized in the same file.

For our purposes, the app must allow a ROC study to be conducted in accordance with user necessity after consulting the checklist for the desired features and selecting the appropriate package.

Shinyapp Flowchart. Initial development of the application saw the creation of a flowchart, that can be seen in Fig. 3, to list possible choices and pathways taken by the user so as to more easily structure the app's development.

The user will initially receive two sources of information regarding each package, the aforementioned checklist followed by each individual and in depth guide of each package.

Afterwards the data must be uploaded, this is the first fork in the flowchart given that *Comp2ROC* has a distinct process for data entry. Following the *Comp2ROC* pathway, the user must select whether the data details paired or independent samples and select the relevant result columns.

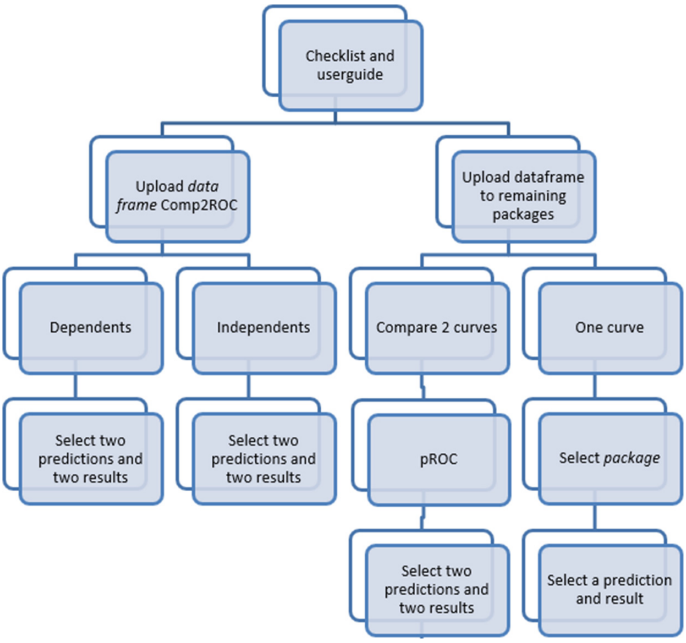


Fig. 3. Flowchart for shiny application creation.

For the remaining packages, data upload and standard ROC analysis are similar with the introduction of a single prediction and result. There is only one exception in the *pROC* package feature when comparing curves that behaves out similarly to *Comp2ROC*, where more than one indicator and result column may be necessary.

3 Results

The developed shiny application is available for use through *shinyapp.io* using the URL https://pquintasbcl.shinyapps.io/ROSY/.

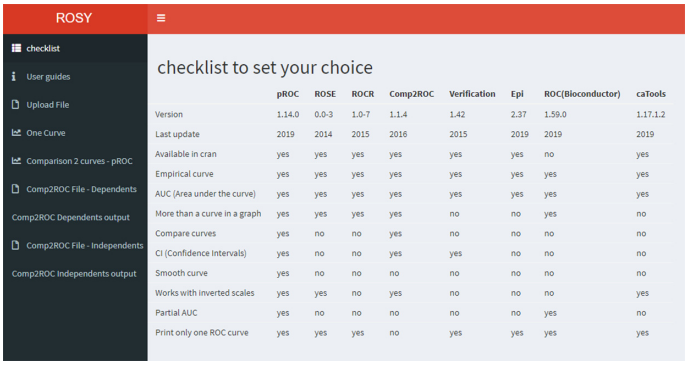
The adopted name, ROSY, came from a mixture of ROC with Shiny through an acronym generator.

This section will detail how each tab of the application works exampling its use with the mentioned datasets. The tool is subject to updates that may result in changes to both visuals and outputs presented at the time of this article's publication.

3.1 Checklist and Userguide

As mentioned, the user is first greeted with the checklist presented in Table 1 implemented into the application as can be seen in Fig. 4, followed by the user

guide tab, in Fig. 5 with official documentation for each of the seven packages provided by either CRAN or Bioconductor. Several notes are also provided with some miscellaneous information about each package or criteria to better aid the user in their choice.



	pROC	ROSE	ROCR	Comp2ROC	Verification	Epi	ROC(Bioconductor)	caTools
Version	1.14.0	0.0-3	1.0-7	1.1.4	1.42	2.37	1.59.0	1.17.1.2
Last update	2019	2014	2015	2016	2015	2019	2019	2019
Available in cran	yes	yes	yes	yes	yes	yes	no	yes
Empirical curve	yes	yes	yes	yes	yes	yes	yes	yes
AUC (Area under the curve)	yes	yes	yes	yes	yes	yes	yes	yes
More than a curve in a graph	yes	yes	yes	yes	no	no	yes	no
Compare curves	yes	no	no	yes	no	no	no	no
CI (Confidence Intervals)	yes	no	no	yes	yes	no	no	no
Smooth curve	yes	no	no	no	no	no	no	no
Works with inverted scales	yes	yes	no	yes	no	no	no	yes
Partial AUC	yes	no	no	no	no	no	yes	no
Print only one ROC curve	yes	yes	yes	no	yes	yes	yes	yes

Fig. 4. Checklist for R packages within the application.

The package *pROC* is clearly the most feature heavy of the packages however it is still important to provide access to different implementations of these function to ensure a diversity of options and a satisfactory outcome, either with a different visual output or with a swifter computation of results.

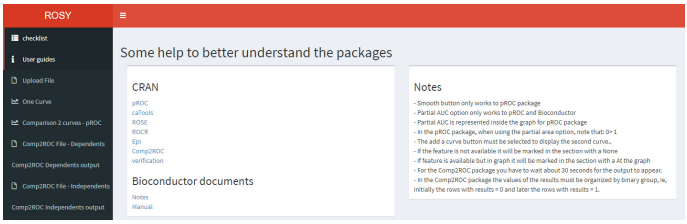


Fig. 5. Packages user guide.

3.2 Upload Dataset

Section 2.3 mentions the need to fork the file upload system due to specific requirements of the *Comp2ROC* package. In this section we will explore the upload systems the app uses for the majority of packages as well as the two systems used for *Comp2ROC*.

Firstly, we examine the generic upload system used for the remaining packages, this functionality uses on the standard shiny upload functions and is compatible with .csv and .txt files. The uploaded dataset is displayed on the side. An example of this display and tab functionality can be seen in Fig. 6.

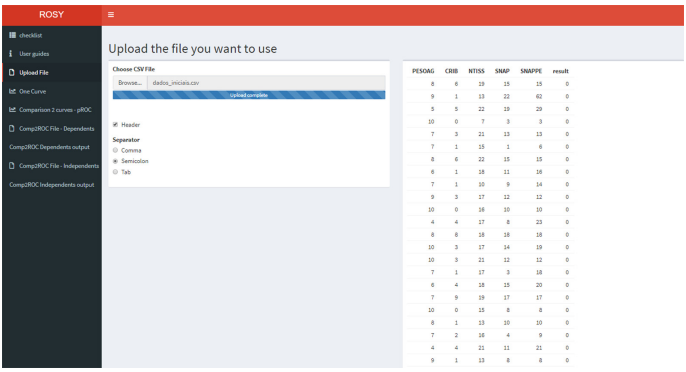


Fig. 6. File upload for all packages except *Comp2ROC*.

This upload function is fairly accessible, needing only the user to press the "Browse" button and search for the relevant files in their directory. After upload the user must specify whether the dataset has headers and how it separates values (comma, semicolon or tab) similarly to the *read.csv* function in R to ensure proper file readability. This tab is used for the *pROC*, *pROC*, *Epi*, *ROC*(Bioconductor) and *caTools* packages.

Dataset upload for *Comp2ROC* was separated between paired and independent samples, creating two distinct tabs for the process to allow simultaneous work on both sections if necessary. As mentioned the specific requirements of data input for this package made it necessary to ease and optimize this process from a user perspective. The code used for this process can be seen below.

```
data2 = reactive({
    req(input$file2)
    inFile2 = input$file2
    df2 = Comp2ROC::read.file(inFile2$datapath, header =
    input$header2 , sep = input$sep2 , ''.'', input$pred1 ,
    input$Direction1, input$pred2,input$Direction2, input$res
    ,T)
    return(df2)})

data3 = reactive({
    input$goPlot
    req(input$file3)
    inFile2 = input$file3
```

```
    df3 = Comp2ROC::read.file(inFile2$datapath, header =
    input$header3 , sep = input$sep3 , ''.'', input$pred3 ,
    input$Direction3, input$pred4,input$Direction4, input$
    res3,FALSE,input$res4)
    return(df3)})
```

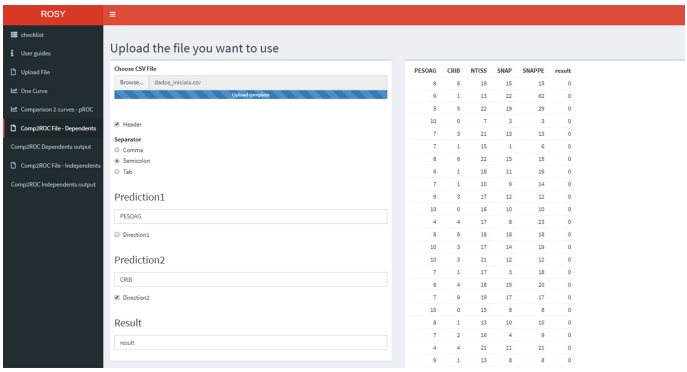


Fig. 7. Data loading for paired samples in *Comp2ROC*.

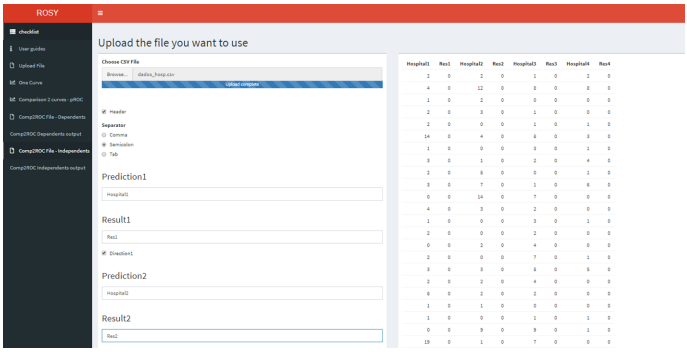


Fig. 8. Data loading for independent samples in *Comp2ROC*.

Much like the previous tab, data uploaded must be supplied with header and separator information. However this data entry must also directly be supplied with information on predictors and result columns. For paired samples these are two predictors for the same result while in independent samples each indicator must have a corresponding result. A "direction" checkbox is also supplied to allow inverted scales without manual inversion. A user must uncheck this box if working with an indicator that does not follow the standard ROC assumption

that higher values on indicator corresponding to a negative result, such as the WEIGHT_AG variable in the paired samples dataset.

Figure 7 and Fig. 8 illustrate the use of paired and independent tabs with their respective datasets.

All uploaded data are inserted into reactive objects to allow manipulation in the next tabs of the ROSY application.

3.3 One Curve Section

The data uploaded with the generic system generates its own output in this tab. Outputs from the *Comp2ROC* package are separated to avoid confusion and will be discussed later.

The One Curve tab supplies several options for the user. First, a section of selection boxes asks the user which package should be used, these packages available of this list are, *pROC*, *ROSE*, *pROC*, *caTools*, *Epi* and *ROC* (Biocondutor). Afterwards the users is asked to identify the prediction and response columns (indicator and outcome, respectively), there are default selections already available that can be changed. With this information the app is already able to generate a ROC curve and calculate its AUC in the left-hand side of the screen. Additional options can be selected however, such as smoothing, available if using the *pROC* package, and pAUC selection and calculation, if using *pROC* or Bioconductor.

Finally, users have the possibility to add a second ROC curve to the generated plot. While not a statistical comparison such as in *Comp2ROC*, a visual comparison can be useful in different ROC analysis. This option is available to *ROSE*,*pROC*, Bioconductor and *pROC* packages. In a different tab *pROC*'s in depth curve comparison will be used.

In Fig. 9 the One Curve tab input options can be seen with their respective output present in Fig. 10.

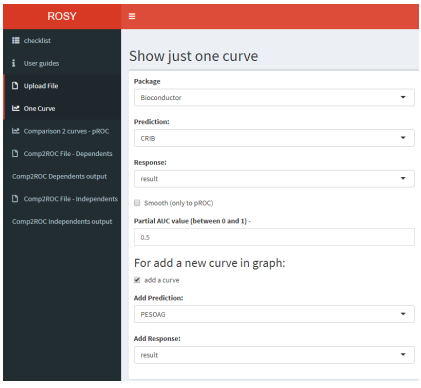


Fig. 9. Visual appearance of the input values of the "one curve" tab.

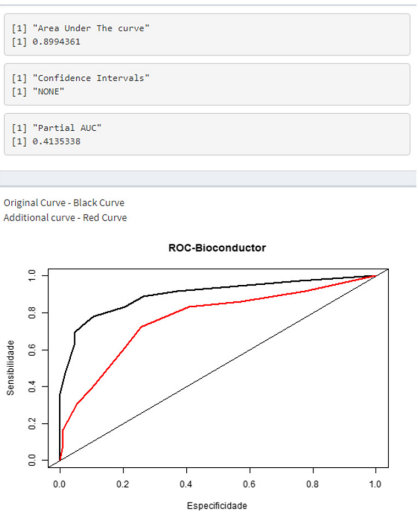


Fig. 10. Output example provided by application.

3.4 Comparison of Two Curves - *pROC*

Since *pROC* allows curve comparison with some statistical parameters a seperate tab was created to highlight this option - Comparison 2 curves - *pROC*.

This tab presents four separate select boxes for input where a user must define predictions and results for both curves, generating an output that represents both curves in the same plot, Z value, p value and AUC.

Figure 11 presents a possible use of this section of the app.

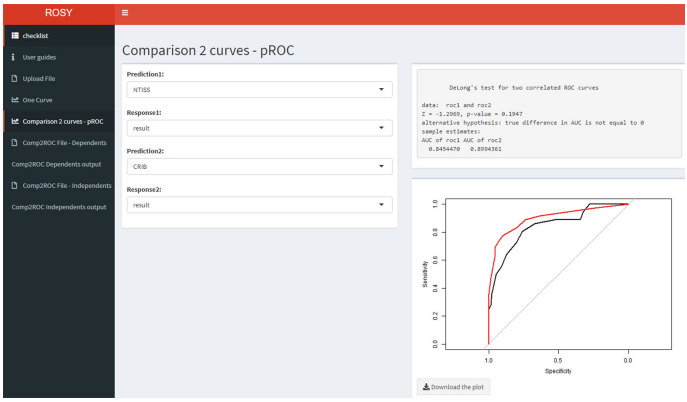


Fig. 11. Example of the output provided on the "Comparison 2 curves - *pROC*" tab.

3.5 Output Comp2ROC - Dependent and Independent Samples

Much like the data input, output for *Comp2ROC*'s analysis is separated in different tabs. After correct data input, "Comp2ROC Dependent output" needs only a press of a button, to display this section's output. Unlike the two previous sections, this tab, due to data entry constraints requires that any change to the inputs on data input must be recalculated with a manual button press, this process takes roughly 30 s to compile all calculations and display them. An example of this output can bee seen in Fig. 12

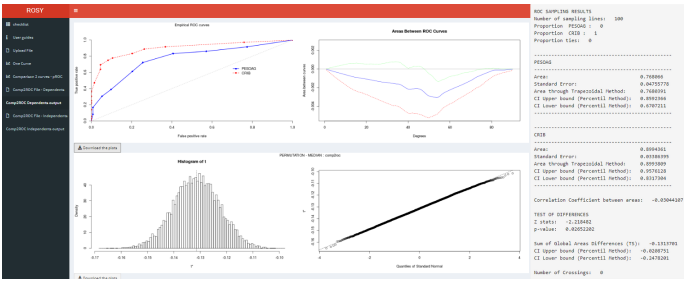


Fig. 12. Example output of roc analysis returned by *Comp2ROC* package on dependent samples tab.

According to the methodology developed by Braga et al. [2] this package is entirely focused on ROC curve comparison therefore, besides the typical plot, provides AUC value comparison, histogram of t and quantiles of standard normal as well as analysis statistics to better compare the two indicators and their performance. These plots can be downloaded using the "Download the plots" button after analysis.

For independent samples, output follows the same format, the only difference being the already discussed input parameters, as can be seen in Fig. 13.



Fig. 13. Example output of roc analysis returned by *Comp2ROC* package on independent samples tab.

4 Conclusions

This work aimed to explore several available packages in R for ROC curve construction. The cataloguing features led to the creation of a checklist to better emphasise the differences between packages in a quick fashion with selected criteria.

To optimize tool selection for ROC analysis a shiny application was created, ROSY, available at https://pquintasbcl.shinyapps.io/ROSY/, giving users the freedom to choose the best package for their desired goal after consulting their features in the checklist and user guide menus.

The application allows any user to use R packages without prior knowledge of programming or scripting replacing command lines and, perhaps, intimidating environments with an easily consumable web based application with select menus and direct outputs.

5 Future Work

This article detailed the first release of the ROSY shiny application and is, as such, subject to change, be it visual improvements or potential bug fixes. Future developments may include an expanding checklist of criteria as well as new packages being implemented to widen the scope of the app and its user appeal.

Acknowledgments. This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

References

1. Braga, A.C.: Curvas ROC: Aspectos funcionais e aplicações, Ph.D. thesis, (Portuguese language) (2000)
2. Braga, A.C., Costa, L., Oliveira, P.: An alternative method for global and partial comparison of two diagnostic systems based on ROC curves. J. Stat. Comput. Simulat. **83**(2), 307–325 (2013)
3. Braga, A.C., Frade, H., Carvalho, S., Santiago, A.: Comp2ROC: Compare Two ROC Curves that Intersect. R package version 1.1.4. (2016). https://CRAN.R-project.org/package=Comp2ROC
4. Braga, A.C., Oliveira, P.: Diagnostic analysis based on ROC curves: theory and applications in medicine. Int. J. Health Care Quality Assurance, **16**(4), 191–198 (2003). https://doi.org/10.1108/09526860310479677
5. Carey, V., Enhancements, H.: ROC: utilities for ROC, with uarray focus. R package version 1.60.0. (2019). http://www.bioconductor.org
6. Carstensen, B., Plummer, M., Laara, E., Hills, M.: EPI: A Package for Statistical Analysis in Epidemiology. R package version 2.37 (2019). https://CRAN.R-project.org/package=Epi
7. Cho, W., Lim, Y., Lee, H., Varma, M., Lee, M., Choi, E.: Big data analysis with interactive visualization using r packages. BigDataScience **14**, 103 (2014)

8. Cunha, D., Braga, A.: Receiver Operating Characteristic (ROC) Packages Comparison in R, pp. 545–559 (2017). https://doi.org/10.1007/978-3-319-62395-5_37
9. Doi, J., Gailwong, J., Irvinchi, P.: Web application teaching tools for statistics using R and shiny. Technol. Innov. Stat. Educ. **9**, 1–32 (2016)
10. Jahanshiri, E., Shariff, R.: Developing web-based data analysis tools for precision farming using R and shiny. IOP Conf. Ser.: Earth Environ. Sci. **20**, 012014 (2014)
11. Lunardon, N., Menardi, G., Torelli, N.: ROSE: a package for binary imbalanced learning. R J. **6**(1), 82–92 (2014)
12. Quintas, J.P., Machado e Costa, F.; Braga, A.C.: "Neonatal mortality indicators in hospitals", Mendeley Data, V1, (2020). https://doi.org/10.17632/zx4r8mgn86.1
13. Robin, X., et al.: pROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinform. **12**(1), 77 (2011). https://doi.org/10.1186/1471-2105-12-77
14. RStudio Team.: RStudio: Integrated Development for R. RStudio Inc, Boston, MA (2019). http://www.rstudio.com/
15. Sing, T., Sander, O., Beerenwinkel, N., Lengauer, T.: ROCR: visualizing classifier performance in R. Bioinformatics, **21**(20), 7881 (2005). http://rocr.bioinf.mpi-sb.mpg.de
16. Stephan, C., Wesseling, S., Schink, T., Jung, K.: Comparison of eight computer programs for receiver-operating characteristic analysis. Clinical Chemistry **49**, 433–9 (2003)
17. Tuszynski, J.: caTools: tools: moving window statistics, GIF, Base64, ROC AUC, etc. R package version 1.17.1.2. (2019). https://CRAN.R-project.org/package=caTools
18. Wojciechowski, J., Hopkins, A., Upton, R.: Interactive pharmacometric applications using R and the shiny package. CPT: Pharmacometrics Syst. Pharmacol. **4**, 146–159 (2015)