Comparing Empirical ROC Curves Using a Java Application: CERCUS*

Daniel Moreira[1][0000−0002−8765−1484] and Ana C. Braga[2][0000−0002−1991−9418]

[1] Department of Informatics, School of Engineering, University of Minho
[2] ALGORITMI Centre, University of Minho,
4710-057 Braga, Portugal
a.daniel.t.moreira@gmail.com, acb@dps.uminho.pt

Abstract. Receiver Operating Characteristic (ROC) analysis is a methodology that has gained much popularity in our days, especially in Medicine, since through the empirical ROC curve, it provides a useful tool to evaluate and specify problems in the performance of a diagnostic indicator. The area under empirical ROC curve(AUC) it's an indicator that can be used to compare two or more ROC curves.
This work, arose from the necessity of the existence of software that allows the calculation of the necessary measures to compare systems based on ROC curves.
Several software, commercial and non-commercial, are available that perform the calculation of the measures associated to the ROC analysis. However, they present some flaws, especially when there is a need to compare independent samples with different dimensions, or also to compare two ROC curves that intersect.
In this paper is presented a new application called **CERCUS** (Comparison of Empirical ROC Curves). This was developed using programming language (Java) and stands out for the possibility of comparing two or more ROC curves that cross each other.
The main objective of **CERCUS** is the calculation of several ROC estimates using different methods and make the ROC curves comparison, even if there is intersection, either for independent or paired samples. It also allows the graph representation of the ROC curve in a unitary plan as well the graph of the area between curves in comparison.
This paper presents the program's versatility in data entry, test menus and visualization of graphs and results.

Keywords: ROC curve · CERCUS · Java · R.

1 Introduction

The ROC analysis was developed between 1950 and 1960 that emerged from the decision theory more concretely in the theory of signal detection [3,6]. This

* This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019

emerged as a need to identify and differentiate in a radar operator, a reliable signal (allies, enemies) of a noise (clouds, birds, etc). Since this time the ROC analysis has gained a lot of popularity because although it is a useful tool to evaluate the performance of an indicator, it is able to compare different indicators and to select in a practical way an optimal threshold [4].

This methodology has been applied to several scientific areas and in the field of medicine has been an important factor in medical decisions making, as well as in the areas of epidemiology, diagnostic tests, radiology and bioinformatics [9].

The graph of the ROC curve in the unit plane is a technique that can be used to organize and select classifiers by evaluating their performance. This technique consists of a two-dimensional graphical representation that has as x axis, *"1-specificity"* and in the ordinate y, *"sensitivity"*, that vary from 0 to 1 [6], [**?**]. In terms of data in medicine, the sensitivity corresponds to the probability of an illness being present, when in reality the individual is sick and the specificity corresponds to the probability to exclude the disease, when in reality it is absent [3].

For comparison of two ROC curves, there is a method that obtains from the ROC graph, a scalar value that represents its performance, the area under the curve, AUC. Since AUC is a portion of the unit plane, its values vary from 0 to 1.

If you draw a line diagonally from the origin in this plane, it represents a value of AUC of 0.5 , so no realistic classifier should have a AUC less than 0.5 [6], and in practice the value of AUC ranges from 0.5 to 1.0. Hanley and McNeil [10] can compare two ROC curves, through a Z statistic, which uses this estimate as indicator for the two systems to be compared. Estimations of the AUC can be obtained by the trapezoid rule, to compute the area below the curve, or by the Wilcoxon statistic, where the statistical properties of this can be used to predict the properties of the AUC of the ROC curve.

When comparing ROC curves when they intersect, [1] present a methodology that allows the comparison of ROC curves in different regions of the space, by the determination of partial areas. Using a methodology based on the comparison of Pareto curves in multi-objective optimization, the `Comp2ROC` package, developed in R [2], is the result of this methodology.

The analysis through the ROC curves is important in different fields of applicability, however there are few applications available to systematize this analysis, namely with regard to graphical representation and comparison of two systems. The growing use of this methodology in different areas, such as in the medical field, requires the existence of a unique tool that encompasses the most important methodologies of the study of ROC curves. The development of simple and intuitive software capable of analysing ROC curves using the `Comp2ROC` functionalities is the great motivation of this work.

The main objective of this work is to present an application developed in Java which is a high-level language, for comparing two systems based on ROC curves that cross or not.

The **CERCUS** (Comparison of Empirical ROC Curves) application is a software developed in Java that facilitates analysis through ROC curves, providing the results of the curves and their graphs. The **CERCUS** name, was obtained using key words ("Comparison", "Empirical", "ROC", "Curves" and "Cross"), in an acronym generator available at: `http://acronymcreator.net/`. The logo is original and was inspired by the representation of ROC curves in the unitary ROC space. The application allows the incorporation and edition of data, being possible to compare two or more ROC curves.

2 Methodology

The methodology of the ROC curves is used for the evaluation of the performance of systems and their comparison, for independent and paired samples. For the development of the application, it is used the technique of algorithm (programming by objects), making use of libraries already developed and available in Java.

2.1 Programming in Java

Java is an object-oriented computer programming language that was originally released in 1995 by Sun Microsystems (which was acquired by Oracle Corporation).

The code is compiled to byte code which can be run on any Java virtual machine, regardless of the operating system [8].

Unlike other programming languages, Java is not just a language that consists only of object programming. This is based on an attractive and appropriate programming and application development environment, especially from the Java Development Kit (JDK) system [8].

The main feature of the Java language is that it includes a simple language that can be programmed without extensive programmer training, where key concepts are learned quickly. The robustness and security of this type of language is to have an extensive compilation-time check, followed by a second level of runtime verification. That is, in the development of Java code the system will find errors quickly, where the main problems will not be suspended until there is an update of the code. On the other hand, Java allows to include cryptographic keys in the code itself, thus enabling the identification of the origin of the code [8].

Basically the development of applications using this type of language originates a software of high security and performance that includes multiple architectures, operating systems and graphical interface.

Additionally, developers have access to existing libraries of tested objects that provide additional functionality to the new program.

2.2 R Serve library

Programming languages, such as Java, are widely used for application development, but are not very efficient when it comes to statistical and/or mathematical modelling. To compensate for this gap, there are languages like R, which has a wide range of statistical libraries. By integrating these two technologies, we can create applications based on high quality statistical modelling. The `Rserve` is a library available in Java, which enables communication between Java and R, making it possible to obtain statistical results using functions and libraries available in R.

The interpolation of the application with the `Rserve` is accomplished by incorporating the program R into the project. With this operation, in the application it is possible to open the R, run the algorithm and then close it. Figure 1 illustrates a scheme, explaining superficially the operation of `Rserve` with Java.

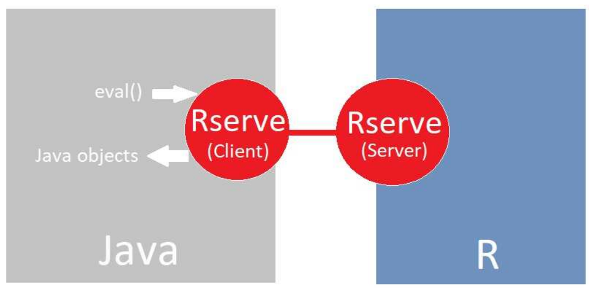


Fig. 1: `Rserve` schema interconnecting Java with R.

2.3 Requirements

Although there are some programs that perform ROC analysis, there are none that can match the graphical presentation with the comparison of two or more ROC systems. To facilitate the information process regarding ROC estimates, the **CERCUS** application contemplates the different ROC methodologies for comparison. The requirements for the construction of the application were:

1. the user must be able to create, open and save data files;
2. the tool should allow the user to edit data files;
3. it should be possible to import / export EXCEL (.xls) files;
4. the tool should have basic commands like copy, cut and paste;
5. it should present the results of ROC estimates in a simple and intuitive way;
6. it should be able to make a graphical representation, which the user can write to image file (.jpeg);
7. it should have a help button to make it easier to use the new program.

2.4 Methodological Approach

The approach chosen was the development of Java software that implements ROC methodologies.

Figure 2 represents a simplified scheme of the new algorithm. The interface class is primarily responsible for the structure of the program and to get results or graphics you will need to use classes like DataFrame and Table. These classes allow you to collect values that the user provides to the program and to calculate the various ROC estimates using specific methods present within them. If there is an intersection of the ROC curves, another class (Comp2Roc) is used, which makes use of the `Rserve` library and the alternative method proposed by Braga et al [1] to calculate the respective metrics for comparison through empirical ROC curves that whether they cross, or not.

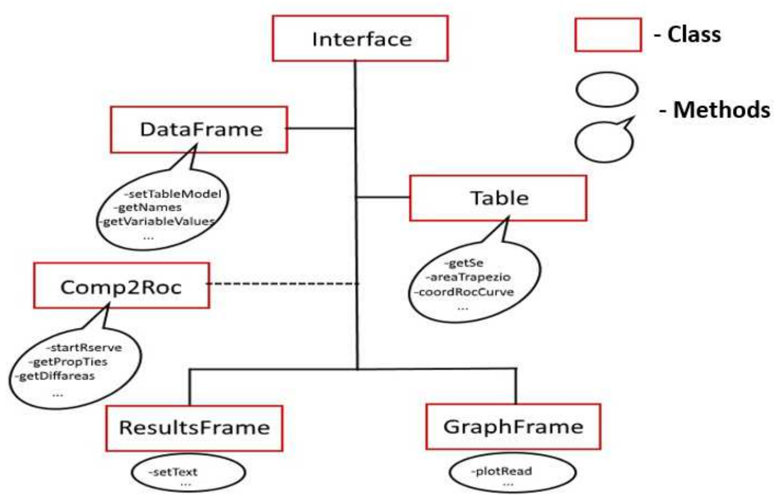


Fig. 2: Schematic of the algorithm.

The **CERCUS** application follows the scheme exemplified in Fig. 2 and all the requirements listed above.

3 CERCUS Application

3.1 Operating Systems

CERCUS is available for Windows and for its use only needs to open the application availabe to download on `https://cercus9.webnode.pt/`. This first version is available to download in a `.zip` file witch includes the `.jar` application, a folder called `fun` that contains additional functions and a folder with examples files.

For the UNIX system, **CERCUS** is also available,to download in a `.zip` file with the same folder but for this system.

To work with this application, in both systems, the user needs to install the R program from CRAN: `https://cran.r-project.org/` choosing the correspondent system, and also install the `Comp2ROC` and `Rserve` packages by using the code instruction `install.packages("Comp2ROC")` and `install.packages("RServe")` in R.

For any doubt in the `zip` file the user has a `readme.txt` with instructions for the requirements of the systems.

3.2 Running CERCUS

The application enables the incorporation and editing of data, making it possible to compare two or more ROC curves. After double click on the application icon, software is initialized by opening a window, as seen in Fig. 3. In this figure it is possible to verify that the application is divided into three sectors listed by 1, 2 and 3.

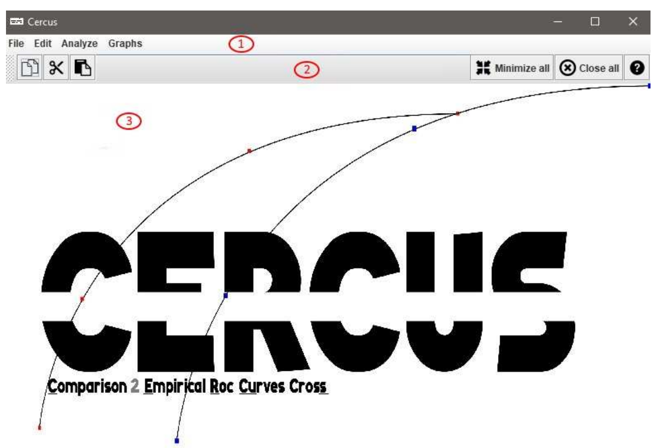


Fig. 3: CERCUS presented in 3 different sectors: 1. Menu bar; 2. Toolbar and 3. Background panel.

Menu Bar The Menu Bar is divided into five groups:

– *File* is a menu for opening, creating and saving data.
– *Edit* is a menu that consists of editing data and windows. This is shown in the toolbar.
– *Analyze* is a menu that is based on calculating ROC estimates.
– *Graphs* is a menu that shows the illustration of the respective graphs.
– *WindowGraph* is a menu that consists of editing the graphics window.

Toolbar The toolbar is divided in two sectors, these is the data editing sector (located in the left side of the Fig. 4) and the window editing sector (located in the right side of the Fig. 4), except for the **help** button.

Fig. 4: Toolbar image.

In the data edit group, the three buttons are respectively for copying, cutting and pasting values in the data window. These will only work after the input and subsequent selection of the data window.

In the window editing group, two buttons are available to minimize and close all available windows in the application background.

3.3 Data Entry

The introduction of data in the application can be done in three different ways:

– by creating a new data file that can be saved for editing;
– from a file previously saved in the application;
– from an Excel (.xls) file.

Figure 5 serves to better understand the **File** menu presented in CERCUS. This menu was inspired in ROCNPA program ([**?**,12])

The red boxes in Fig. 5 correspond to the different choices to introduce data.

There is a prerequisite for data entry that is associated with the operating conditions of the `Comp2ROC` package. In the data files, in the result variable must be listed first all values of the distribution of negative cases (0), followed by the positive ones (1).

Creating a New File To create a new data file, it is necessary to press the **New** button of the **File** menu where the program will present three dialogues for complete the definition of variables as shown in Fig. 6.

The first one, is used to characterize the sample, that is, it questions how many variables are being studied and identifies the type of data (paired or independent samples)(Fig. 6a).

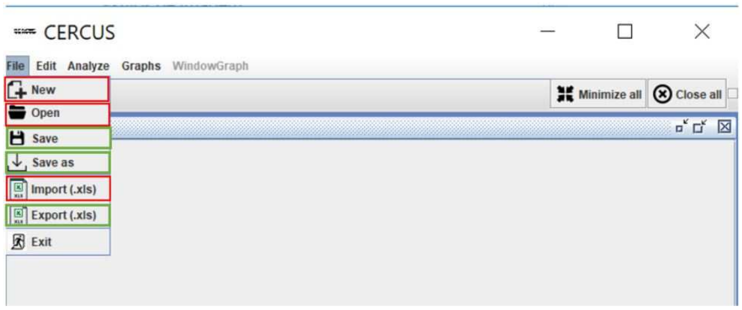
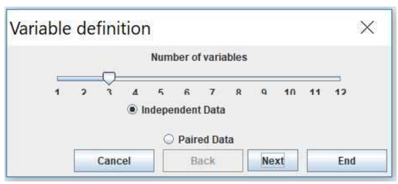
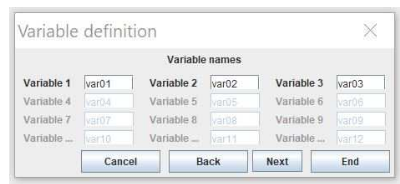


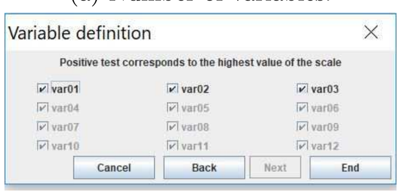
Fig. 5: Window of **File** menu.



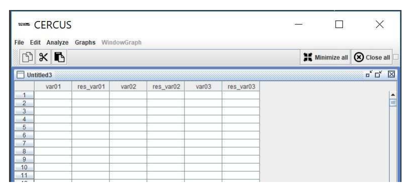
(a) Number of variables.



(b) Name of variables.



(c) Scale of measure.



(d) Example of data sheet.

Fig. 6: Example of new data creation for 3 independent samples.

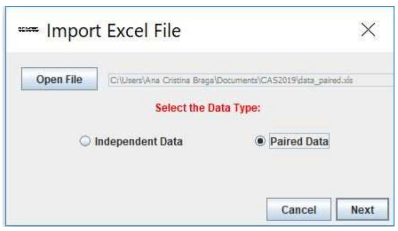
After pressing the **Next** button, a second window will be displayed. The purpose of this will be to define the names of the variables as shown in Fig. 6b.

In the next window (Fig. 6c) the user is asked about the scale of the variables under study. By default it is assumed that the highest value of the scale corresponds to the positive result. If some of the variables change inversely, the visa must be withdrawn.

Pressing the end button, the data sheet appear like in image of Fig. 6d.

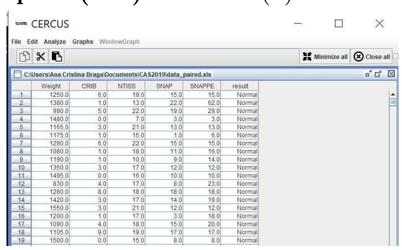
Open or Import a File To open an existing file, the user must click on the **Open** button shown in Fig. 5. A new window will be displayed, providing choice and search options. **CERCUS** allows access to a file with an own extension `.cer`.

To import an EXCEL file it is necessary to click the **Import (.xls)** button located in the **File** menu, shown in Fig. 5. The window in Fig. 7a will be displayed so that clicking on the **Open File** button will allow selection of the file, in this case limited by the `.xls` extension. Still in this window is the selection of data type. If the user presses the **Next** button without characterizing the sample, an error message will appear.



(a) Path way for **Import (.xls)**.

(b) Choice for scale measurement.



(c) Final data sheet.

Fig. 7: Example of import EXCEL file for paired samples.

In Fig. 7b, in the case of the Weight variable, it is verified that the smallest value of the scale corresponds to the positive test (result that corresponds to the death of the newborn).

For the EXCEL file to be imported correctly, it must be in `.xls` (Excell 97-2003) and also be filled in from the first row and column. That is, there can be no white-space between columns in the first row. If this happens the reading of the file will not be correctly processed. Also note that variable names must be in the first row only. If the file encounters non-numeric characters after the first line, importing the file will be impossible. On the other hand, in files for paired samples, the last column should be (0 = negative result or 1 = positive result), and in the files for independent samples the variables and the respective response (0 or 1) should be merged side by side. The Fig. 8 illustrates an example of how data should be distributed in Excel.

It is important to note that in the data structure the response variable (0 and 1) must be ordered according to the data procedure to be processed in Comp2ROC in the R, that is, the negative cases (0) must first be placed and then the positive ones (1), either whether they are paired or independent samples.

H1

	A	B	C	D	E	F
1	Weight	CRIB	NTISS	SNAP	SNAPPE	Result
2	1250	6	19	15	15	0
3	1380	1	13	22	62	0
4	990	5	22	19	29	0
5	1480	0	7	3	3	0
6	1165	3	21	13	13	0
7	1175	1	15	1	6	0
8	1290	6	22	15	15	0
9	1080	1	18	11	16	0
10	1190	1	10	9	14	0
11	1350	3	17	12	12	0
12	1495	0	16	10	10	0
13	830	4	17	8	23	0

(a) Paired samples.

I1

	A	B	C	D	E	F
1	Hospital1	Res1	Hospital2	Res2	Hospital3	Res3
2	2	0	2	0	1	0
3	4	0	12	0	8	0
4	1	0	2	0	0	0
5	2	0	3	0	1	0
6	2	0	0	0	1	0
7	14	0	4	0	8	0
8	1	0	0	0	3	0
9	3	0	1	0	2	0
10	2	0	8	0	0	0
11	3	0	7	0	1	0
12	0	0	14	0	7	0
13	4	0	3	0	2	0

(b) Independent samples.

Fig. 8: Example of data in EXCEL file.

To save the data files we can press the **Save** or **Save As** buttons, where the `.cer` file is automatically associated. In case we want to export to the EXCEL format, we can proceed to **Export As**, where the `.xls` extension is associated.

3.4 Analyze Menu

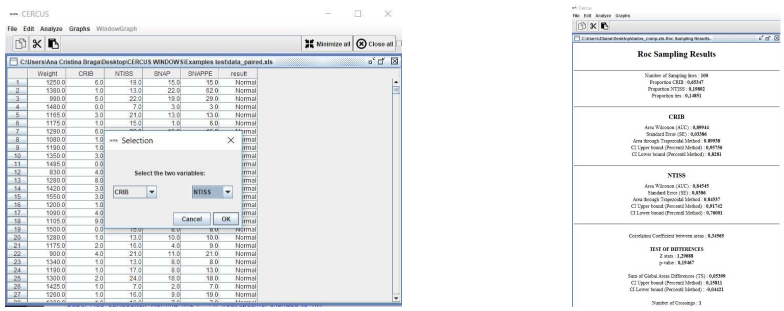
CERCUS allows comparison of two or more ROC curves using two different approaches. Several results will be presented concerning the information present in the data window, whether it is data from independent or paired samples.

The **Analyze** menu, is designed to make this comparison giving the user two options:

- **Traditional Multiple Comparison Test**, provide the user with a series of ROC estimates, that are used when the ROC curves do not cross each other. Statistical computations according the methodology proposed by Hanley and McNeil [11] and Delong et al [5].
- **Roc Sampling Results**, gives the user the analytical results of comparing two ROC curves. This option is used when ROC curves cross each other resulting in more detailed ROC estimates according the methodology developed by Braga et al [1] and implemented in `Comp2ROC` in R ([7]).

Intrinsic to the **CERCUS** program, the `Rserve` library is used in the option **Roc Sampling Results** to calculate the respective ROC comparison results using the **Comp2ROC** library ([2]).

Basically, after selecting **Roc Sampling Results** the dialogue box, shown in Fig. 9a, allows the selection of the variables that the user wants to compare. Internally after selecting the data window, it removes the information about the two variables selected in the dialogue window (see Fig. 9a), calculates the results using the connection with R (`Rserve`) and consequently presents the results in a new window as presented in Fig. 9b that could export to `.html` format .



(a) **Roc Sampling Results** dialogue box. (b) Resulting window.

Fig. 9: Example of **Roc Sampling Results** menu.

3.5 Graphs Menu

The **Graphs** menu is intended for presentation of the graphs associated with the analysis through empirical ROC curves. This representation is divided into three parts and the user can subsequently save the produced graphics to a `.jpeg` file:

- Empirical ROC curve(s);
- Empirical ROC curves (2 by 2);
- Area between curves ROC.

Fig. 10 illustrate the **Graphs** menu presented in **CERCUS**.

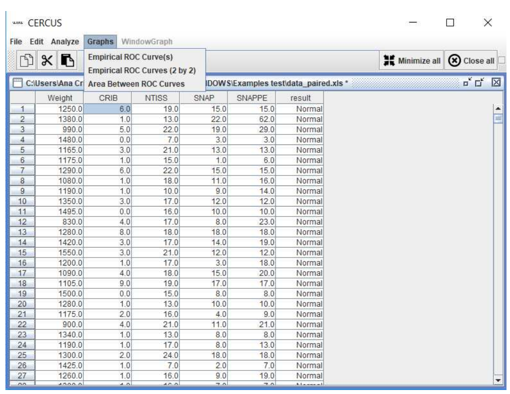


Fig. 10: Menu Graphs

The empirical ROC curves are produced by joining the coordinate points, which correspond to the pairs (1 - specificity, sensitivity), calculated for each case. For the option **Empirical ROC curves (2 by 2)** will first be asked to select the variables and then the union of the coordinate points.

For the areas between ROC curves, after the selection of the variables to be compared, the application once again uses the `Rserve` to obtain the values of *Lower Bound*, *Upper Bound* and *Degrees* using the method proposed by Braga et al. [1] implemented in `Comp2ROC`.

Fig. 11 shows the set of result windows produced by the introduction of five paired samples (Fig. 10).

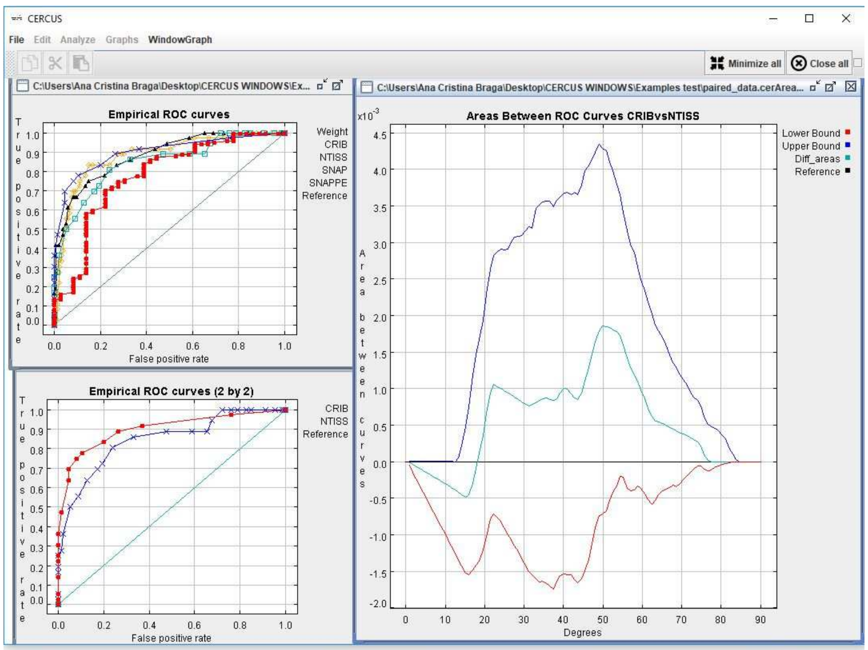


Fig. 11: Menu Graphs

4 Final Remarks

The main focus of this work was the development of an application for personal computers that can integrate the different methodologies of comparison based on empirical ROC curves, allowing the comparison of two systems based on empirical ROC curves that intersect or not.

In the case of the methodology in which the ROC curves that intersect, the **CERCUS** application, in spite of visually identifying the regions of the space where there is better performance of one system in relation to the other, it was not possible to implement a metric conversion algorithm to identify in the unit ROC space which pairs (1-specificity and sensitivity) correspond to this region.

This work allows to answer to the lack of software capable of systematizing the analysis through the ROC curves, particularly with regard to graphical representation and comparison of two or more systems, either for independent data or for paired data.

The elaboration of the algorithm was based on the structure of the ROCNPA program, trying to simplify its functionality to the maximum. Although there are still many improvements to be made to the **CERCUS** interface, it is versatile and robust for analysing samples of any type.

The application was tested on both operating systems using the R version 3.4.4.

We are aware that the work presented here can be improved and supplemented. For example, implementing a button that can translate the results into a text file will help the user make a more detailed comparison of the ROC curves. On the other hand, since windows are not intuitively available, creating a menu that makes windows open will assist the user in making the selection.

Finally, we believe that the implementation of new ROC analysis methodologies, such as curve fitting and presentation of confidence intervals, will improve the future development of **CERCUS**.

References

1. Braga, A.C., Costa, L., Oliveira, P.: An alternative method for global and partial comparison of two diagnostic systems based on roc curves. Journal of Statistical Computation and Simulation **83**(2), 307–325 (2013)
2. Braga, A.C., Frade, H., Carvalho, S., Santiago, A.M.: Package 'Comp2ROC' (2014), `https://cran.r-project.org/web/packages/Comp2ROC/Comp2ROC.pdf`
3. Braga, A.C., Oliveira, P.: Diagnostic analysis based on ROC curves: theory and applications in medicine. International Journal of Health Care Quality Assurance **16**(4), 191–198 (2003)
4. Cheam, A., McNicholas, P.D.: Modelling Receiver Operating Characteristic Curves Using Gaussian Mixtures pp. 1–15 (2014)
5. Delong, E.R., Delong, D.M., Clarke-pearson, D.L., Carolina, N.: Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. Biometrics **44**(3), 837–845 (1988)
6. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters **27**(8), 861–874 (2006)
7. Frade, H., Braga, A.C.: Comp2roc. In: Mohamad, M.S., Nanni, L., Rocha, M.P., Fdez-Riverola, F. (eds.) 7th International Conference on Practical Applications of Computational Biology & Bioinformatics. pp. 127–135. Springer International Publishing, Heidelberg (2013)
8. Greenberg, I., Xu, D., Kumar, D.: Processing Creative Coding and Generative Art in Processing 2. Apress, Berkeley, CA (2013). https://doi.org/10.1007/978-1-4302-4465-3
9. Hajian-Tilaki, K.: Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation (2013)
10. Hanley, A., McNeil, J.: The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. Radiology **143**, 29–36 (1982)

11. Hanley, J.A., McNeil, B.J.: A method of comparing the areas under receiver operating characteristic curves derived from the same cases. Radiology **148**(3), 839–43 (1983)
12. Mourão, M.F., Braga, A.C.: Evaluation of the crib as an indicator of the performance of neonatal intensive care units using the software rocnpa. In: 2012 12th International Conference on Computational Science and Its Applications. pp. 151–154 (June 2012). https://doi.org/10.1109/ICCSA.2012.37