



Interactive Multivariate Data Analysis in R with the `ade4` and `ade4TkGUI` Packages

Jean Thioulouse
Université de Lyon

Stéphane Dray
Université de Lyon

Abstract

`ade4` is a multivariate data analysis package for the R statistical environment, and `ade4TkGUI` is a Tcl/Tk graphical user interface for the most essential methods of `ade4`. Both packages are available on CRAN. An overview of `ade4TkGUI` is presented, and the pros and cons of this approach are discussed. We conclude that command line interfaces (CLI) and graphical user interfaces (GUI) are complementary. `ade4TkGUI` can be valuable for biologists and particularly for ecologists who are often occasional users of R. It can spare them having to acquire an in-depth knowledge of R, and it can help first time users in a first approach.

Keywords: `ade4`, multivariate data analysis, graphical user interface, Tcl/Tk.

1. Introduction

The `ade4` package is a port of a previous software that was written in C (“Classical **ADE-4**”: <http://pbil.univ-lyon1.fr/ADE-4/>) to R (R Development Core Team 2007). This software was mainly used by ecologists, and it had a feature rich and useful GUI, written in HyperTalk and based successively on HyperCard, WinPlus and MetaCard (Thioulouse, Chessel, Dolédec, and Olivier 1997). Switching to R and to the command line version of `ade4` was a hard task for many users, so we decided to make it easier by providing them with a GUI.

So the first aim of this GUI was to give to the users of “Classical **ADE-4**” an easy access to the main functions of `ade4`. As most users would be also new to R, we wanted it to be easy to install, and using Tcl/Tk was a guarantee of easiness and multi-platform compatibility.

In this paper, we wanted to give a few details about the implementation of `ade4TkGUI`, to give both a general overview and a detailed example of its use, and finally to discuss the interest of a GUI for ecological data analysis in R.

The basic concepts behind **ade4** are given in another issue of this volume (Dray and Dufour 2007), so we shall not detail them here. Briefly, it is based on the duality diagram, a mathematical scheme defined in the early seventies by French statisticians. The data analysis methods available in **ade4** are described in two papers published recently in *R News*: Chessel, Dufour, and Thioulouse (2004) (one-table methods) and Dray, Dufour, and Chessel (2007) (two-tables and K-tables methods). One table methods include principal components analysis (PCA) with several variants, simple and multiple correspondence analysis (CA) with several variants, and principal coordinates analysis. Two-tables methods include coinertia analysis, principal components analysis with respect to instrumental variables (PCAIV), and for example canonical correspondence analysis (CCA), or redundancy analysis (RDA), as particular cases of PCAIV. K-tables methods include partial triadic analysis, the STATIS method, the Foucart analysis (analysis of a series of contingency tables), multiple coinertia analysis, multiple factor analysis, and STATICO (analysis of a series of couples of ecological tables). Only one-table and two-tables methods are currently available in **ade4TkGUI**. K-tables methods are not included in the present version, but they will be in future versions.

2. Implementation

In R, a graphical user interface (GUI) can be implemented at several levels, mainly the R level, the package level and the function level. Many GUIs already exist at the R level, some are platform independent, like R Commander (**Rcmdr**, Fox 2005), and some are platform dependent (like the **R.app** GUI for MacOS, the **RGUI.exe** GUI for Windows, or **SciViews** (Grosjean 2003)). A GUI can also implement a limited subset of functions oriented toward a particular statistical area, like basic statistics for **pmg** (Verzani 2007). The package level is an intermediate level, where the GUI gives access to (some of) the functions of a particular package. This is the case for **ade4TkGUI**, which is a separate package implementing a GUI for the **ade4** package, like **QCAGUI** (Dusa 2007b) is a package implementing a GUI for the **QCA** package (Dusa 2007a). Eventually, a GUI can also be written for only one function. Many of the functions of **ade4TkGUI** can be used this way, providing independent GUIs for some functions of **ade4**.

GUIs can also be implemented in several ways, using different software solutions. Web interfaces are appealing, because they can be used to offer a simplified access to advanced statistical functionalities. The user has no software installation step to go through and can use the proposed methods directly in a web browser. Many packages are available in this area (Mineo and Pontillo 2006), for example **CGIwithR** (Firth and Temple Lang 2005), **Rpad** (Short and Grosjean 2006), and **R-php** (Mineo and Pontillo 2006). We have also used the **Rweb** system (Banfield 1999), together with **ade4** (Chessel *et al.* 2004), **seqinR** (Charif and Lobry 2006) and the **ACNUC** system (Gouy, Gautier, Attimonelli, Lanave, and di Paola 1985) to provide multivariate analysis services in the field of bioinformatics, particularly for sequence and genome structure analysis at the PBIL (Pôle de Bio-Informatique Lyonnais: <http://pbil.univ-lyon1.fr/>). An example of these services is the automated analysis of the codon usage of a set of DNA sequences by correspondence analysis (<http://pbil.univ-lyon1.fr/mva/coa.php>). Moreover, the integration with Bioconductor data sets and classes can be achieved with **made4** (Culhane, Thioulouse, Perrière, and Higgins 2005; Culhane and Thioulouse 2006).

Another solution to build a GUI is to use Tcl/Tk widgets. The **tcltk** package is now included

in the R base distribution, so it is widely available, and it is also platform independent. It has been used to write a GUI in many packages, like **Rcmdr**, **pinktoe** (Nason 2005), or **QCAGUI**. This is also the solution we adopted to write **ade4TkGUI**. Other solutions to write GUIs include using the Gtk toolkit, with the **RGtk2** package, and the Java language, like in the **JGR** package. Many other solutions have been explored, like **R-wxPython** (now abandoned), based on the wxWidgets toolkit, through the Python language with **RSPython** and the wxPython GUI toolkit. The “R GUI projects” page (http://www.sciviews.org/_rgui/) gives informations on this subject.

3. Overview of the **ade4TkGUI** package

It is not possible to give here a detailed description of all the functions of **ade4TkGUI**, so only the main characteristics will be presented. The core of the package is the **ade4TkGUI()** function, which opens the main GUI window (Figure 1).

This function takes two parameters, **show** and **history**. The first one determines whether the R commands generated by the GUI should be printed in the console. When the user interacts with the GUI, he modifies the status of some widgets and when he clicks on the “Submit” button, an R command is generated from these widgets. This command is executed and can optionally be displayed in the console. If the second parameter is set to **TRUE**, the commands generated by the GUI are also stored in the **.Rhistory** file, where they can easily be retrieved by the user. The state of the two parameters is recalled in the main window heading “**ade4TkGUI(T,T)**”.

In the main GUI window, buttons are grouped according to their function: Data sets, One table analyses, One table analyses with groups of rows, Two tables analyses, Graphic functions, and Advanced graphics. To avoid cluttering this window, only a limited subset of functions is displayed. Less frequently used functions are available through the menus of the menu bar,

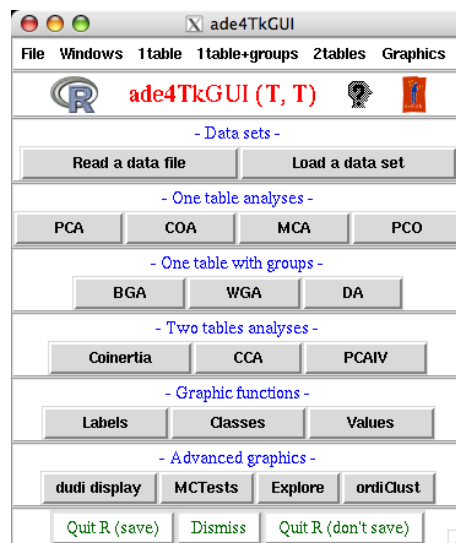


Figure 1: The main **ade4TkGUI** window.

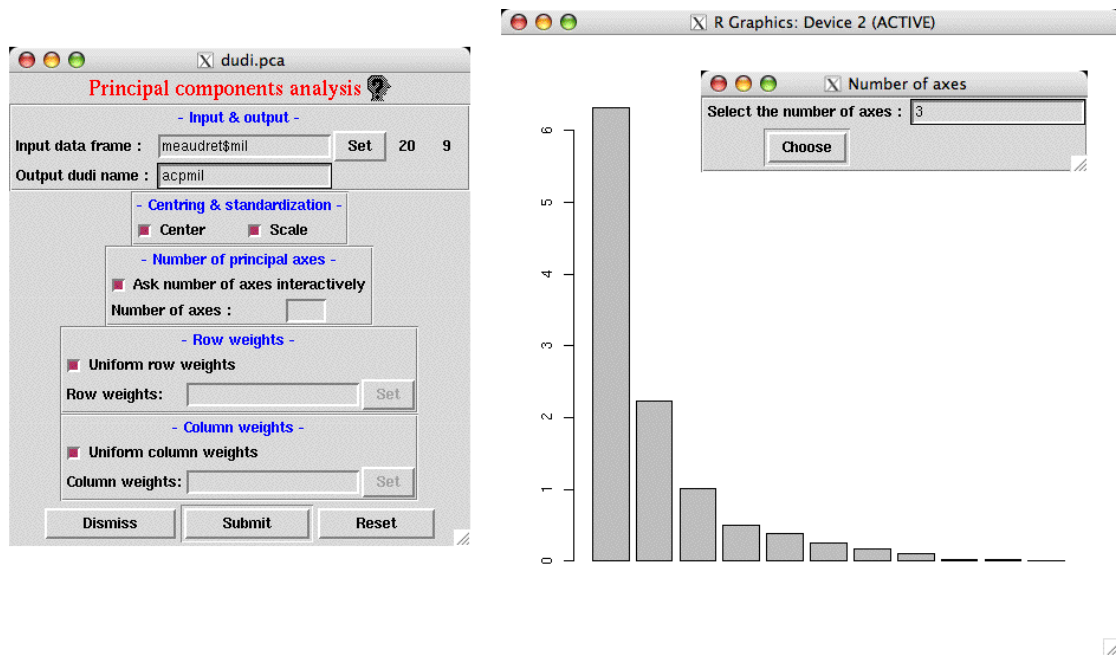


Figure 2: The `dudi.pca()` function GUI window (left), the eigenvalues bar chart (right) and the selection of axis number by the user (top-right).

located at the top of the window. Right-clicking the buttons opens the **ade4** help window for the corresponding function. The questionhead button opens the help window of **ade4TkGUI**. When the “PCA” button is clicked, a new window appears (Figure 2). This is the GUI window of the `dudi.pca()` function (Dray and Dufour 2007). The “Set” button can be used to choose the PCA input dataframe through a listbox showing all the available dataframes in the user global environment. After the “Input data frame” text field has been filled by the user, the number of rows and columns (20, 9) are displayed next to it. The output of the `dudi.pca()` function is an object of class `dudi` (a “duality diagram”, *i.e.*, a list with many components) and the user can type the name of this object in the “Output dudi name” field. If this field is left empty, the name “Untitled1” is used automatically. The remaining widgets can be used to set particular options for the PCA: centring and standardization, number of principal axes, row and column weights.

Most of the windows created by **ade4TkGUI** are non-blocking (meaning that the user can do other things in the GUI or in the R console before taking the action required by this window.) This was designed to make the interface more flexible and easier to use. Clicking the “Submit” button starts the PCA computations. When they are completed, the barplot of eigenvalues is displayed (Figure 2) and, if this option was chosen in the previous window, the user is asked to select the number of axes on which the row and column scores are to be computed.

After scores are computed, the `dudi` window is displayed (Figure 3, left). This window shows a summary of the analysis, and all the elements of the `dudi` object, under the form of buttons that can be used to draw graphical displays of all these elements. For example, the row and column coordinates buttons draw the classical factor maps. In the lower part of the window, the user can choose which axes are used to draw these graphics. The last row of buttons give

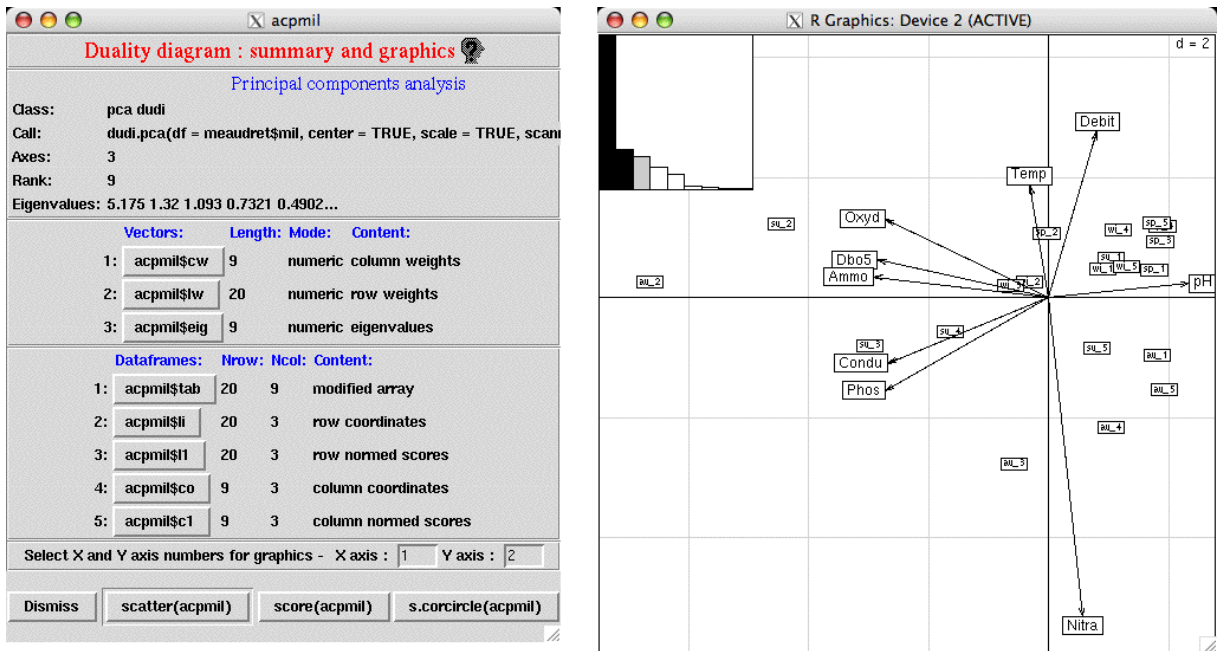


Figure 3: The dudi object display window (left) and the biplot obtained by clicking on the “scatter” button (right).

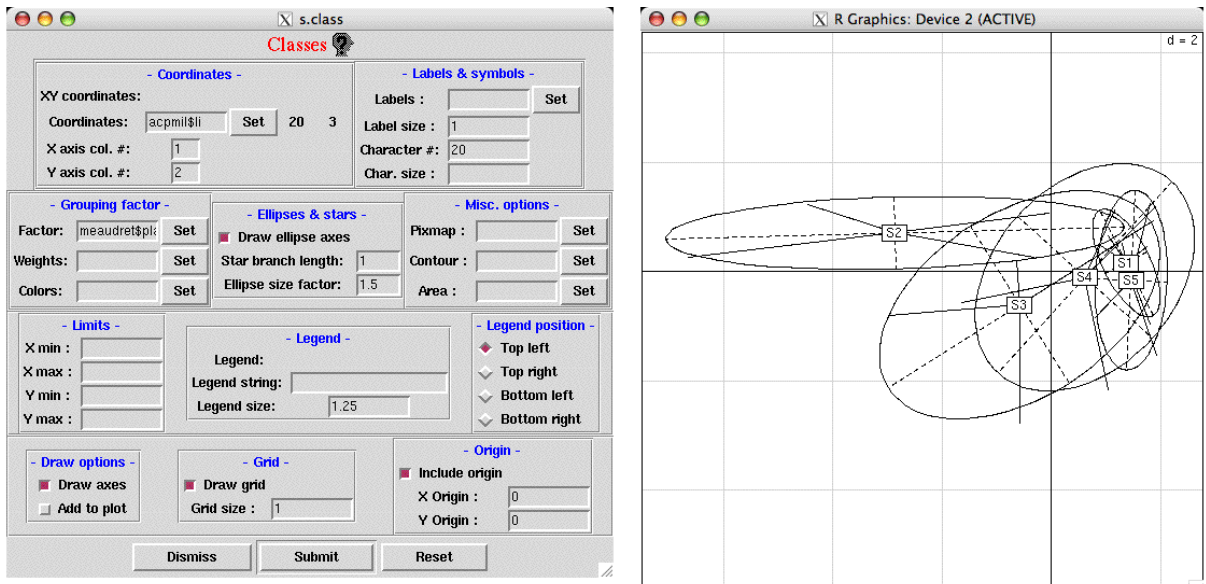


Figure 4: The s.class GUI window (left) and the corresponding graphic (right).

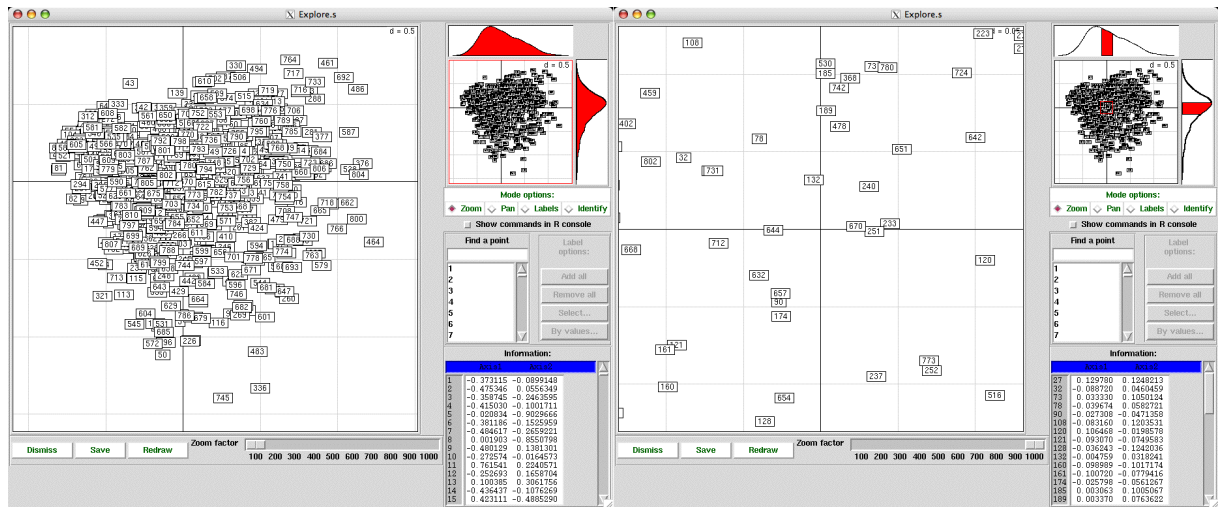


Figure 5: The window of the `explore` function, which implements dynamic graphics, before (left) and after (right) zooming toward the center of the factor map.

access to special graphics, according to the particular properties of the `dudi` that is displayed. For example, in the case of a normed PCA, the “`s.corcircle`” button allows to draw a correlation circle. The scatter button draws a biplot (see for example [Udina 2005](#)), with a small bar chart for eigenvalues (Figure 3, right). These additional buttons are adapted to the type of `dudi` that is displayed, and they allow to draw graphics that illustrate particular properties of this `dudi`.

An example of GUI for one of the graphics functions of `ade4` is given in Figure 4. This is the `s.class()` function, which allows to draw factor maps with groups of individuals. The user can choose the dataframe containing the row scores (here they come from the “`acpmil`” `dudi`), and the factor that should be used to draw the groups on the factor map. Many other options can be set to enrich the graphics.

The window of the `explore` function appears in Figure 5. This function can be used to dynamically explore factor maps, by way of operations like zooming, panning, and searching on labels. This is particularly useful for large datasets, as it is difficult to identify one particular individual on a cluttered factor map.

4. Example of use

A complete example of use of `ade4TkGUI` ([Herrington 2006](#)) can be found online at this address: <http://www.unt.edu/benchmarks/archives/2006/october06/rss.htm>. However, we shall present here the use of `ade4TkGUI` on the same data set as [Dray and Dufour \(2007\)](#) in this JSS special volume, *i.e.*, the famous dune meadow data ([Jongman, ter Braak, and Van Tongeren 1987](#)).

[Dray and Dufour \(2007\)](#) used the `dudi.hillsmith` function to analyse a mixture of numeric and factor variables. This function is not yet available in `ade4TkGUI`, so we shall use a variant, `dudi.mix`. The `dudi.hillsmith` function differs from `dudi.mix` in two things: `dudi.mix`

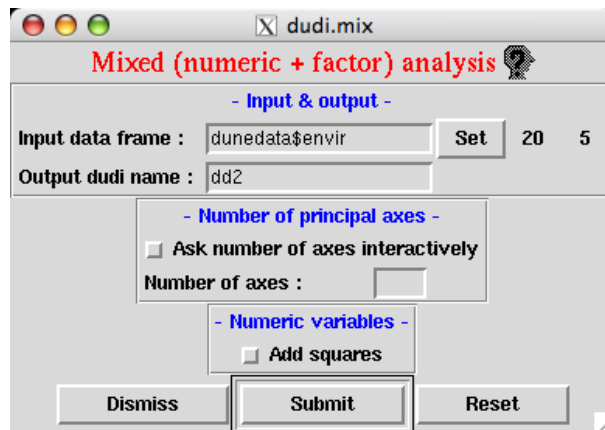


Figure 6: Mixed analysis of the dune data set.

considers quantitative variables, factors, and ordered factors, while `dudi.hillsmith` does not consider ordered factors. On the other hand, `dudi.mix` is restricted to uniform row weights, while non-uniform row weights can be set by the user in `dudi.hillsmith`, using the `row.w` argument. In `dudi.mix`, ordered factors are replaced by `poly(x,deg=2)` and an additional parameter (`add.square`) can be used to add the squares of quantitative variables to the data set. This allows to consider non linear (*i.e.*, quadratic) relationships between variables. Functions `dudi.hillsmith` and `dudi.mix` will probably be gathered into one in future versions of `ade4` and `ade4TkGUI`.

After launching `ade4TkGUI` (Figure 1), the user can load the dune data set by clicking on the “Load a data set” button, selecting it in the list of data sets, and clicking on the “Choose” button. As noted by [Dray and Dufour \(2007\)](#), variable “use” is an ordered factor. To obtain the same results as these authors, we need to transform it into a factor, by typing the following command in the R console:

```
R> dunedata$envir$use <- factor(dunedata$envir$use, ordered = FALSE)
```

The `dudi.mix` function is not frequently used, so it is not available as a button in the main `ade4TkGUI` window. It can be found in the “1 table” menu in the main menu bar, and selecting it pops up the `dudi.mix` GUI window (Figure 6).

The user can then click the “Set” button to choose the `dunedata$envir` dataframe in the listbox of available dataframes. He can also set the output dudi name (“dd2” here). Unchecking the “Ask number of axes interactively” checkbox prevents the `dudi.mix` function from prompting the user for the number of axes during computations, and two axes will be kept by default. A click on the “Submit” button starts computations, and the dudi GUI window is displayed when they are finished, which should be nearly instantaneous.

This window (Figure 7, left) displays all the dudi elements as buttons, and in the lower part of the window, the “scatter” button can be used to draw the default scatter plot of this dudi, which is the common biplot (Figure 7, right). This figure is the same as Figure 3 of [Dray and Dufour \(2007\)](#), but inverted vertically. This comes from an inversion of the sign of the second principal axis, and the change is not meaningful. The user can of course also click on the other buttons to get graphical displays of the other dudi elements.

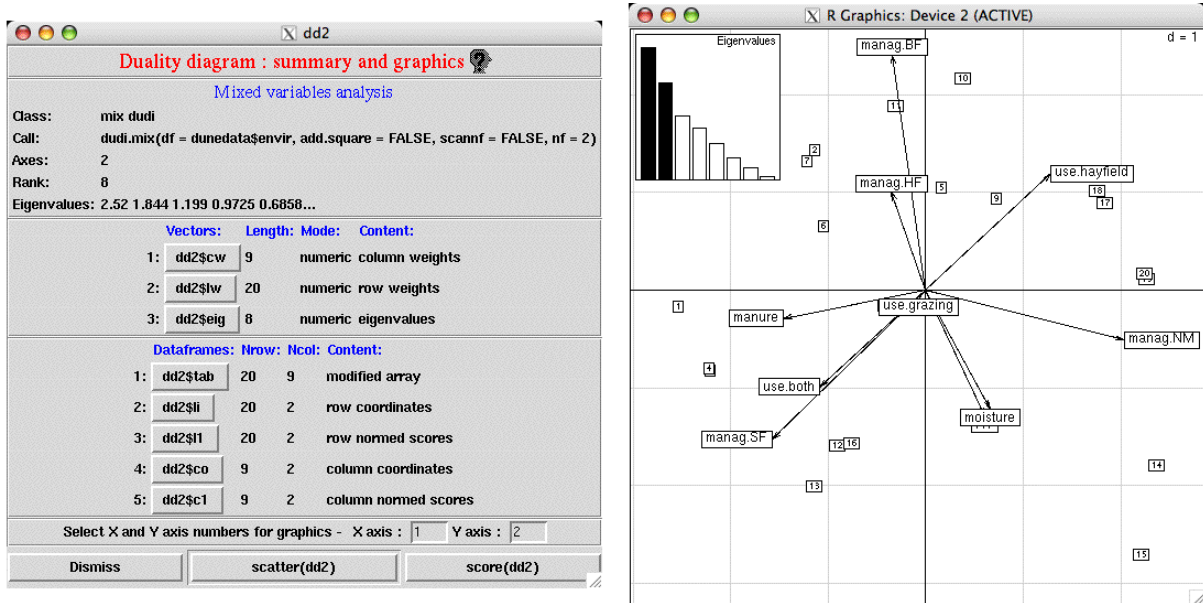


Figure 7: The dudi object (left) and the biplot (right) of the mixed analysis of the dune data set.

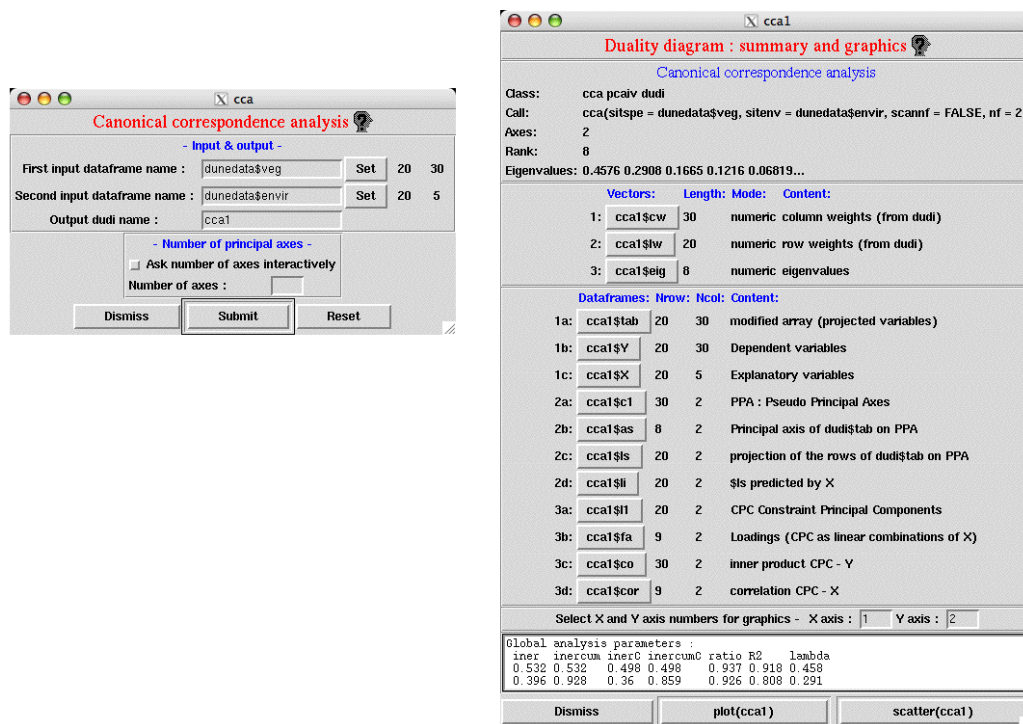


Figure 8: CCA of the dune data vegetation species against the environmental variables (left) and the resulting dudi object (right).

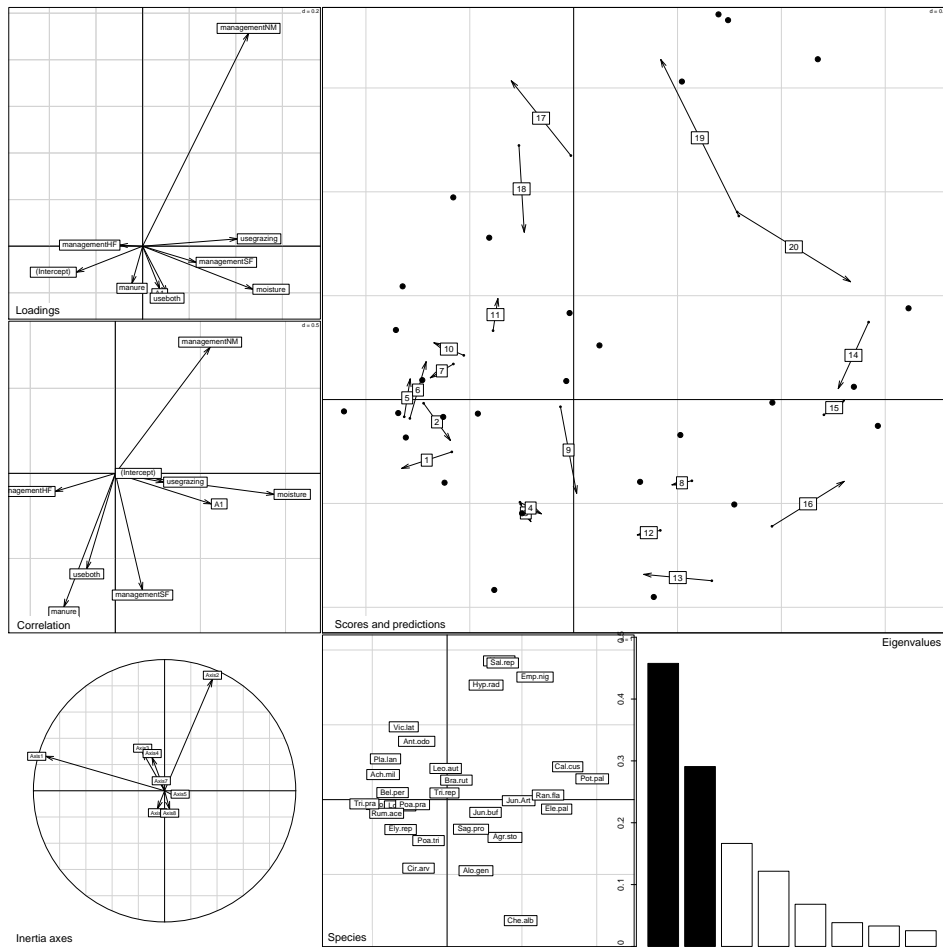


Figure 9: The CCA compound graphic (see text for details).

Doing the CCA of the vegetation species against the environmental variables is just as straightforward. Clicking on the “CCA” button brings up the CCA GUI window (Figure 8, left). The two “Set” buttons are used to set the species dataframe (`dunedata$veg`) and the environmental variables dataframe (`dunedata$envir`). The output `dudi` is named “`cca1`”, and the “submit” button launches the computations. The `dudi` window shows the numerous elements of the CCA `dudi` (Figure 8, right), and clicking on the “`plot(cca1)`” button draws the generic display of PCAIV analyses (Figure 9).

This compound graphic is made of separate graphics for environmental variables loadings and correlations (top and middle left), projection of the axes of the species analysis (*i.e.*, the correspondence analysis) into the CCA (lower left), species scores (bottom middle), eigenvalues bar chart (bottom right), and a biplot of site scores superimposed with their predictions by environmental variables (main graph).

5. Discussion

The advantages of a GUI are particularly important for ecologists who want to use an R package for doing multivariate data analysis. We analyse here the pros and cons of this approach.

5.1. Pros

It is widely recognized that the main advantage of a GUI is the ease of use for beginners, occasional users, or teachers and students. It makes it easier to learn how to use the software (by making the learning curve smoother), and also to get back to work after a long period. This is particularly important in the case of using **ade4** in ecological data analysis, because ecologists are mostly occasional users of R.

An important feature introduced in **ade4** is the *dudi*, an R object (a list with many elements) containing all the informations relating to a *duality diagram* (Holmes 2006). The *dudi* GUI window (Figure 3) was designed to display all the components of a *dudi*, and to allow to draw automatically default graphics for each of these components. Therefore, it offers a centralized and synthetic view of an analysis, and it allows to see rapidly and interactively many graphics. In command line mode, the user must know all the components of a *dudi*, and remember which one is needed to draw a particular graphic: this is also difficult for occasional users.

In “the French way” (Holmes 2006), exploratory multivariate data analysis is mainly a way to look at the data. So graphical functions in **ade4** are extremely important. Using them in command line mode is fast and easy when the user knows the arguments by heart. But if he has to look for the arguments in the help window, most of the interactive aspects are lost. A GUI can bring a useful help here, by showing directly all the possible arguments, and allowing to change them easily. For example, the `s.class()` function has 28 arguments, and it is not easy to remember the meaning of all but the first few ones. The GUI window (Figure 4) displays all these arguments in an ordered way, making it easy to change one, click the “Submit” button to see the change, and change it again if the result is not satisfactory. Doing the same thing in command line mode requires to look at the help window, to use the arrow keys to navigate in past commands (up - down and left - right), and to type in the new value.

The **ade4TkGUI** package also facilitates the use of **ade4** by pre-selecting the types of objects that are proposed to the user when he must do a selection. For example, in the `dudi.pca()` dialog window (Figure 1), when the user clicks on the “Set” button to select a dataframe, the proposed listbox contains only the dataframes in the global environment (or in lists present in the global environment). In the same window, if the user wants to set non uniform row weights, the “Set” button for row weights displays only vectors of length equal to the number of rows of the dataframe. More generally, lists are filtered to propose only objects with consistent properties. In the same way, in the *dudi* window, the displayed buttons and their functions are coherent with the type of *dudi* and with the mathematical properties of its components.

Another area where a GUI is extremely useful, is dynamic graphics. When the number of items in a multivariate analysis is high, factor maps get completely cluttered, and it is impossible to find a particular item. **ade4TkGUI** proposes a factor map exploration function (Figure 5) that can be used to zoom in, pan, and search for particular items. All these possibilities are more easily presented through a GUI, and should be useful for ecologists.

5.2. Cons

The most frequently cited drawback of GUIs is that “it makes it easier to make mistakes”, allowing ignorant users to “click everywhere” and get erroneous results. But it is also easy to make mistakes in command line mode, not speaking about typing mistakes. For example, copying commands in a book or in a scientific article, without understanding what they mean can be the source of many mistakes. This can be even more misleading, as typing a complicated command and obtaining some result can make the user believe that this result is good, particularly if he has received many “syntax error” messages before obtaining this result. Obtaining a result in just a few mouse clicks can remind the user that a few other mouse clicks could give different results. Moreover, setting wrong input in a GUI will raise the same error message as in the CLI.

Another problem with GUIs is that they can be slow and cumbersome when used on old or entry-level computers. **ade4TkGUI** is based on the **tcltk** package, and the requirements in terms of computer power are quite modest.

A more serious drawback of GUIs is the fact that it is difficult to keep track of all the actions that lead to a particular result. Conversely, command line mode provides a history of commands, and it is easy to find out which command gave which result. This drawback was present in the first version of **ade4TkGUI**, but since version 0.2-1, command lines can be printed in the R console, and added to the history file each time a “Submit” button is clicked. This means that actions performed by the user in the GUI are recorded in the session history in exactly the same way as commands typed in the R console. In fact, they can even be freely mixed: it is possible to type R commands in the console and use the GUI simultaneously.

In R, a GUI can also hinder the joint use of several packages. For example, advanced users of **ade4** can benefit from functions and objects defined in other packages, such as **spdep** for spatial multivariate analysis (with the **multispati()** function). In this case, using **ade4TkGUI** could be an obstacle to take full advantage of the joint use of the two packages. The user should thus be careful not get locked up in a particular subset of functions by the GUI.

Obviously, a GUI is not well adapted to scripting, and even to simple repetitive tasks. It is also not good for batch and online or remote use, and it is not easy to integrate into Sweave documents and vignettes. This is probably the main drawback of GUIs: they are made for *personal* and *instant* use, while CLIs allow many operations like scripting, re-doing the same analysis several months later, sharing pieces of code among colleagues, and programmed time consuming computations.

6. Conclusion

GUIs and CLIs should not be opposed, but considered as complementary. GUIs make the learning phase smoother for beginners, and can be used in education to introduce students to CLI mode. CLI mode is more powerful, it allows to build more complex analyses, particularly when using several packages jointly.

When possible, the joint use of both CLI and GUI is attractive, as the user gets the benefits of the two approaches. Joint use can be very intimate: for example, it is possible to use R expressions in the GUI text fields, and the GUI can return R expressions that can be copied and pasted in the console. In the case of **ade4TkGUI**, the strings typed by the user in the text fields of the GUI are parsed, and it is therefore possible to use R expressions

like `doubs$mil[1:20,1:5]` to specify a subset of a dataframe in a PCA. For example, in the `dunedata` analysis (Figure 6), instead of typing a command in the R console to change the type of the `dunedata$envir$use` variable from ordered to factor, the user can type (or paste) the following expression in the GUI window text field for the input dataframe:

```
R> cbind(dunedata$envir[,-4],factor(dunedata$envir[,4], ordered=F))
```

Conversely, when `ade4TkGUI()` is called with argument `"show = TRUE"`, R commands built by the GUI are echoed to the console. It is then possible to copy/paste them and execute them as needed in the console. This is also an effective way for beginners to learn how to use elaborate R function calls. In the CCA example on the dune meadow data, the commands generated by the GUI were:

```
R> data("dunedata")
R> cca1 <- cca(sitspe = dunedata$veg, sitenv = dunedata$envir,
+   scannf = FALSE, nf = 2)
R> plot(cca1,1,2)
```

Ecologists can thus analyse these command lines and possibly adapt them to their needs, with the additional benefit of gradually learning the R language.

When possible, package writers should therefore consider adding a GUI to their package, as users can largely benefit from it. The cost of writing a Tcl/Tk GUI for a simple package is not very high, and many good examples can be found on Internet to learn the bases of the R - Tcl/Tk interface (see for example: http://www.sciviews.org/_rgui/tcltk/index.html).

References

- Banfield J (1999). “**Rweb**: Web-based Statistical Analysis.” *Journal of Statistical Software*, **4**(1), 1–15.
- Charif D, Lobry JR (2006). “**SeqinR** 1.0-2: A Contributed Package to the R Project for Statistical Computing Devoted to Biological Sequences Retrieval and Analysis.” In HER U Bastolla M Porto, M Vendruscolo (eds.), “Structural Approaches to Sequence Evolution: Molecules, Networks, Populations,” Biological and Medical Physics, Biomedical Engineering Series. Springer-Verlag, New York.
- Chessel D, Dufour AB, Thioulouse J (2004). “The **ade4** Package – I: One-table Methods.” *R News*, **4**(1), 5–10.
- Culhane AC, Thioulouse J (2006). “A Multivariate Approach to Integrating Datasets Using **made4** and **ade4**.” *R News*, **6**(5), 54–58.
- Culhane AC, Thioulouse J, Perrière G, Higgins DG (2005). “**made4**: An R Package for Multivariate Analysis of Gene Expression Data.” *Bioinformatics*, **21**(11), 2789–2790.
- Dray S, Dufour AB (2007). “The **ade4** Package: Implementing the Duality Diagram for Ecologists.” *Journal of Statistical Software*, **22**(4).

- Dray S, Dufour AB, Chessel D (2007). “The **ade4** Package – II: Two-Table and K -Table Methods.” *R News*. Forthcoming.
- Dusa A (2007a). **QCA**: *Qualitative Comparative Analysis*. R package version 0.4-2, URL <http://CRAN.R-project.org/>.
- Dusa A (2007b). **QCAGUI**: *QCA Graphical User Interface*. R package version 1.2-7, URL <http://CRAN.R-project.org/>.
- Firth D, Temple Lang D (2005). **CGIwithR**: *CGI Programming in R*. R package version 0.72, URL <http://www.omegahat.org/CGIwithR/>.
- Fox J (2005). “The R Commander: A Basic-Statistics Graphical User Interface to R.” *Journal of Statistical Software*, **14**(9). URL <http://www.jstatsoft.org/v14/i09/>.
- Gouy M, Gautier C, Attimonelli M, Lanave C, di Paola G (1985). “**ACNUC** – A Portable Retrieval System for Nucleic Acid Sequence Databases: Logical and Physical Designs and Usage.” *Computer Applications in the Biosciences*, **1**(3), 167–172.
- Grosjean P (2003). **SciViews**: *An Object-oriented Abstraction Layer to Design GUIs on Top of Various Calculation Kernels*. R package version 0.9-5, URL <http://www.sciviews.org/SciViews-R/>.
- Herrington R (2006). “**ade4TkGUI** - A GUI for Multivariate Analysis and Graphical Display in R.” *Benchmarks Online*, **9**(12). URL <http://www.unt.edu/benchmarks/archives/2006/october06/rss.htm>.
- Holmes S (2006). “Multivariate Analysis: The French Way.” In D Nolan, T Speed (eds.), “Festschrift for David Freedman,” IMS, Beachwood, OH.
- Jongman RH, ter Braak CJF, Van Tongeren OFR (1987). *Data Analysis in Community and Landscape Ecology*. Pudoc, Wageningen.
- Mineo MA, Pontillo A (2006). “Using R via PHP for Teaching Purposes: **R-php**.” *Journal of Statistical Software*, **17**(4). URL <http://www.jstatsoft.org/v17/i04/>.
- Nason G (2005). “**pinktoe**: Semi-automatic Traversal of Trees.” *Journal of Statistical Software*, **14**(1). URL <http://www.jstatsoft.org/v14/i01/>.
- R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.
- Short T, Grosjean P (2006). **Rpad**: *Workbook-style, Web-based Interface to R*. R package version 1.3.0, URL <http://www.Rpad.org/Rpad/>.
- Thioulouse J, Chessel D, Dolédec S, Olivier JM (1997). “**ADE-4**: A Multivariate Analysis and Graphical Display Software.” *Statistics and Computing*, **7**(1), 75–83.
- Udina F (2005). “Interactive Biplot Construction.” *Journal of Statistical Software*, **13**(5). URL <http://www.jstatsoft.org/v13/i05/>.

Verzani J (2007). **pmg**: *Poor Man's GUI*. R package version 0.9-33, URL <http://www.math.csi.cuny.edu/pmg/>.

Affiliation:

Jean Thioulouse, Stéphane Dray
Laboratoire de Biométrie et Biologie Evolutive (UMR 5558); CNRS
Université de Lyon
université Lyon 1
43, Boulevard du 11 Novembre 1918
F-69622 Villeurbanne Cedex, France
E-mail: jthioulouse@biomserv.univ-lyon1.fr, dray@biomserv.univ-lyon1.fr
URL: <http://pbil.univ-lyon1.fr/JHome>, <http://biomserv.univ-lyon1.fr/~dray/>