Chapitre 6. Logiciels

par G. BOUXIN°

Table des matières

Introduction

Ce chapitre présente des programmes et packages, principalement ceux permettant de travailler dans l'esprit développé dans tous nos textes, c'est-à-dire dans l'esprit d'analyse de données et sortant des méthodes

° rue des Sorbiers, 33 à B.5101 Erpent adresse électronique : guy.bouxin@proximus.be

dites d'ordination. Nous présentons maintenant des logiciels d'analyse multivariées uniquement en langage R.

Nous présentons d'abord brièvement plusieurs quelques logiciels et programmes en langage R [ceux proposés respectivement par HUSSON et al. (2009), par l'Université de Lyon, par JEMBER (2012) et par LOMBARDO, R. & BEH (2019)]. Viennent ensuite nos programmes personnels qui ont été tout spécialement adaptés aux analyses de tableaux phytosociologiques, avec les espèces en lignes et les relevés en colonnes, contrairement aux autres logiciels R qui mettent les variables en colonnes et les relevés en colonnes. Des programmes sont prévus pour transformer facilement des tableaux bruts, avec les données habituelles d'abondances-dominance (i, +, r, 1, 2 , 3 , 4 et 5) en tableaux numériques adaptés aux analyses statistiques.

Le package GINKGO de l'Université de Barcelone est aussi illustré. Quelques autres packages sont cités.

Comme exemples, nous utilisons le fichier Royer tiré de la littérature (avec l'accord de l'auteur) et un second fichier personnel appelé Crupetvegenv113 avec des données floristiques d'abondance et des données environnementales qualitatives et quantitatives. Le plus simple est d'entrer les fichiers avec un tableur, ici Excel et de sauver les tableaux dans un format .txt par exemple.

Le système R

Le langage R est de plus en plus utilisé et se révèle indispensable en analyse de données (R Core Team, 2018). Il offre une multitude de fonctions et de possibilités graphiques. Il est gratuit et est téléchargeable à partir de l'adresse suivante :
http://cran.r-projec.org/

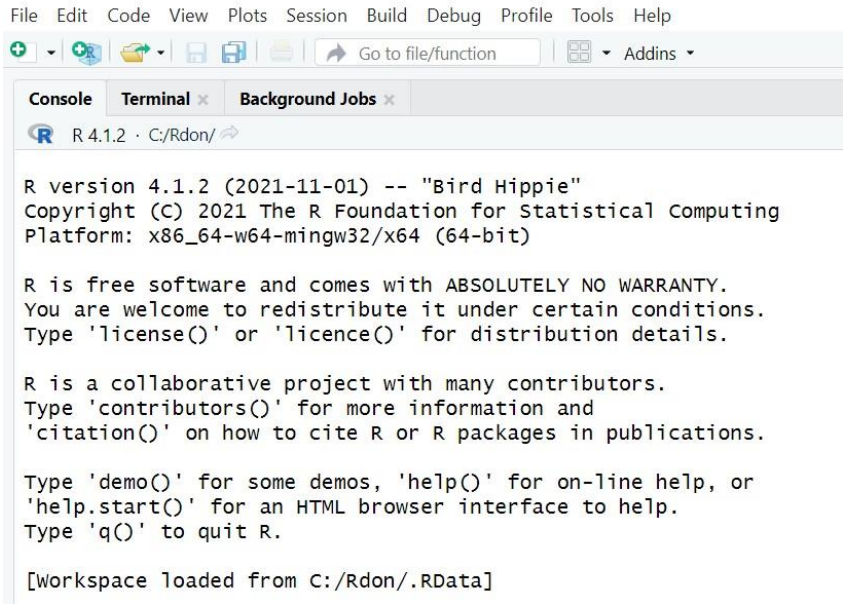

Figure 1. Console du logiciel R.

L'utilisation demande que l'on charge l'un ou l'autre package. Il faut aller dans « packages » et choisir un site miroir de CRAN, ex. France (Montpellier) et puis charger l'un ou l'autre package.

Packages et programmes d'analyse des données en « R »

Plusieurs packages et programmes de la littérature sont présentés parmi d'autres. Plusieurs programmes personnels sont ensuite proposés.

Il faut préciser que la tradition veut que les fichiers soient présentés avec les variables en colonnes et les individus en lignes. **Les tableaux phytosociologiques habituels avec les espèces en lignes et les relevés en colonnes, doivent d'abord être transposés**.

Package Rcmdr (R commander)

L'installation de R commander (interface de R avec des menus) est aisée. Il suffit de suivre les étapes suivantes :

1- Ouvrir R.
2- Aller dans « Packages » et sélectionner « Installer le(s) package(s) ». Choisir un site miroir de CRAN (exemple : France(Montpellier), puis OK.
3- Sélectionner Rcmdr dans la liste de packages disponibles, puis OK.

4- Lancer R commander à l'aide de la commande library(Rcmdr) dans la console de R et le menu apparaît (figure 1).

5- Il faut alors aller dans "Outils" du menu et cliquer sur RcmdrPlugin.FactoMineR.

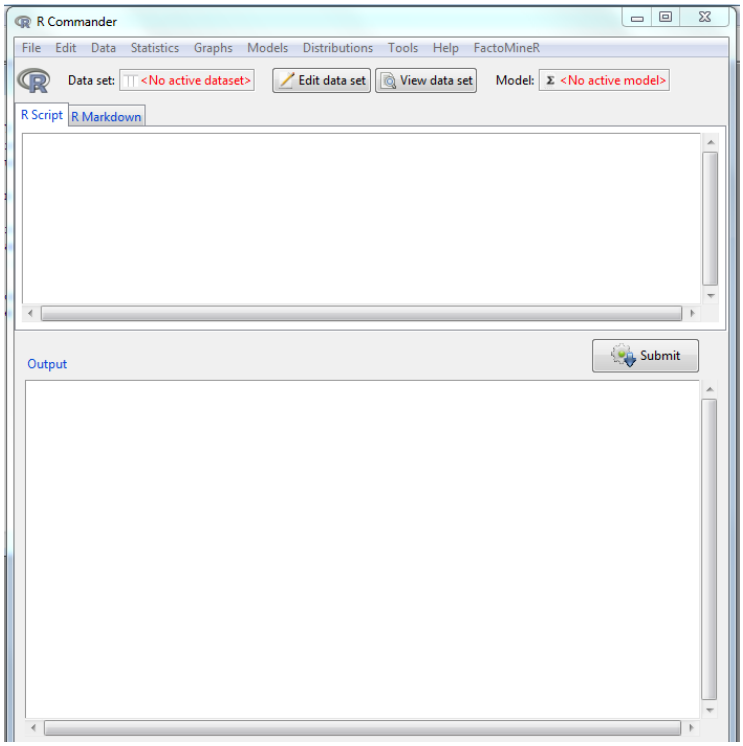

Figure 2. Menu du package Rcmdr.

Il faut maintenant définir l'emplacement des données et des résultats en allant dans File (change working directory).

Il faut alors entrer dans FactoMineR (figure 2) et définir les paramètres du fichier (ici un fichier .txt, figure 3).

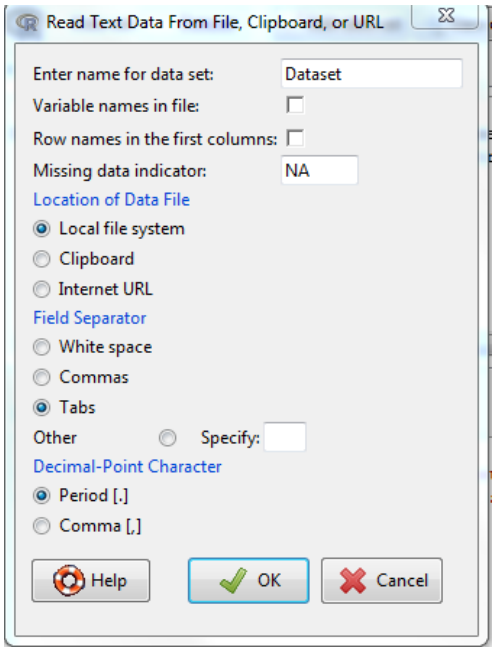

Figure 3. Fenêtre pour charger un fichier avec ses paramètres.

Le fichier est visible comme suit (Figure 4). Dans cette analyse, les intitulés des lignes et colonnes ne sont pas indiqués. On peut les mettre dans des fichiers séparés.

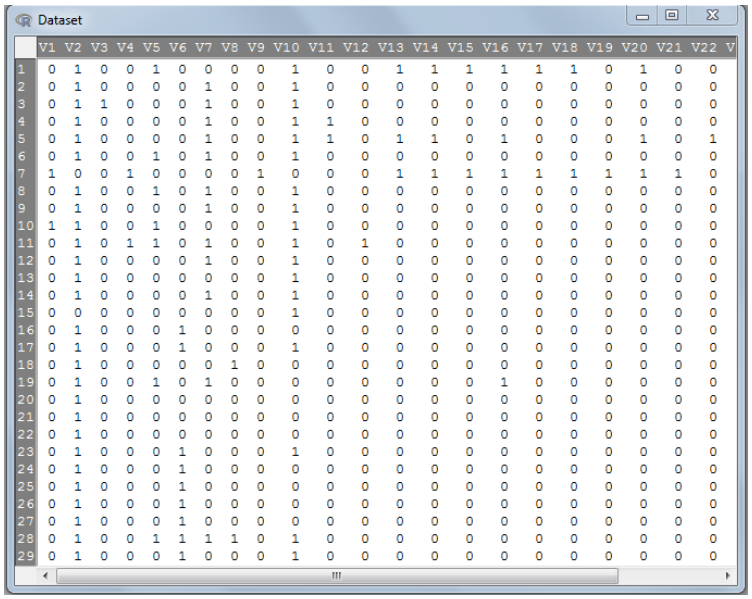

Figure 4. Exemple de fichier .txt chargé.

On choisit alors l'analyse toujours dans FactoMineR et on définit les paramètres des graphiques et des output (Figure 5).

Les analyses proposées sont les suivantes :

Principal component analysis,

Correspondence analysis,

Multiple correspondence analysis,

Multiple factor analysis,

Hierarchical multiple factor analysis,

Dual multiple factor analysis,

Factor analysis of mixed data,

General Procrustes analysis,

Scatter plot with additional variables,

Description of categories,

Hierarchical clustering on principal components.

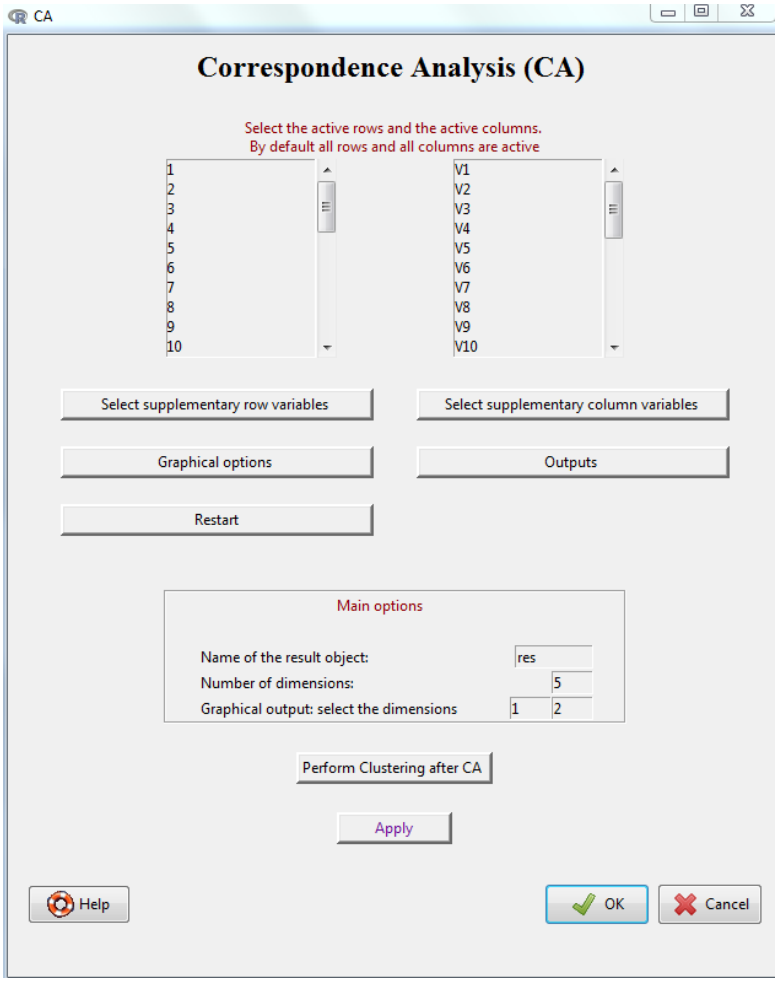

Figure 5. Fenêtre de l'analyse des correspondances.

Les résultats sont sauvés dans un fichier .csv et sont transférables dans d'autres logiciels (Excel par exemple). Les graphiques peuvent aussi être sauvés ou copiés directement et introduits dans un fichier Word.

Cet ensemble est donc assez complet et peut être utilisé avec de très grands tableaux.

Package ADE4

Ce package développé par l'Université de Lyon1 est également très utile et riche en analyses possibles. On y accède par l'adresse URL suivante :

http://pbil.univ-lyon1.fr/ade4TkGUI/

La page d'accueil (Figure 6) donne une idée du contenu

Figure 6. Page d'accueil du package ADE-4.

Ce package est assez complet et est aussi recommandable pour de grands tableaux.

Dans la fenêtre R, il faut charger le package ade4TkGUI, puis entrer library(ade4TkGUI) et enfin la fonction ade4TkGUI(). La fenêtre suivante apparaît :

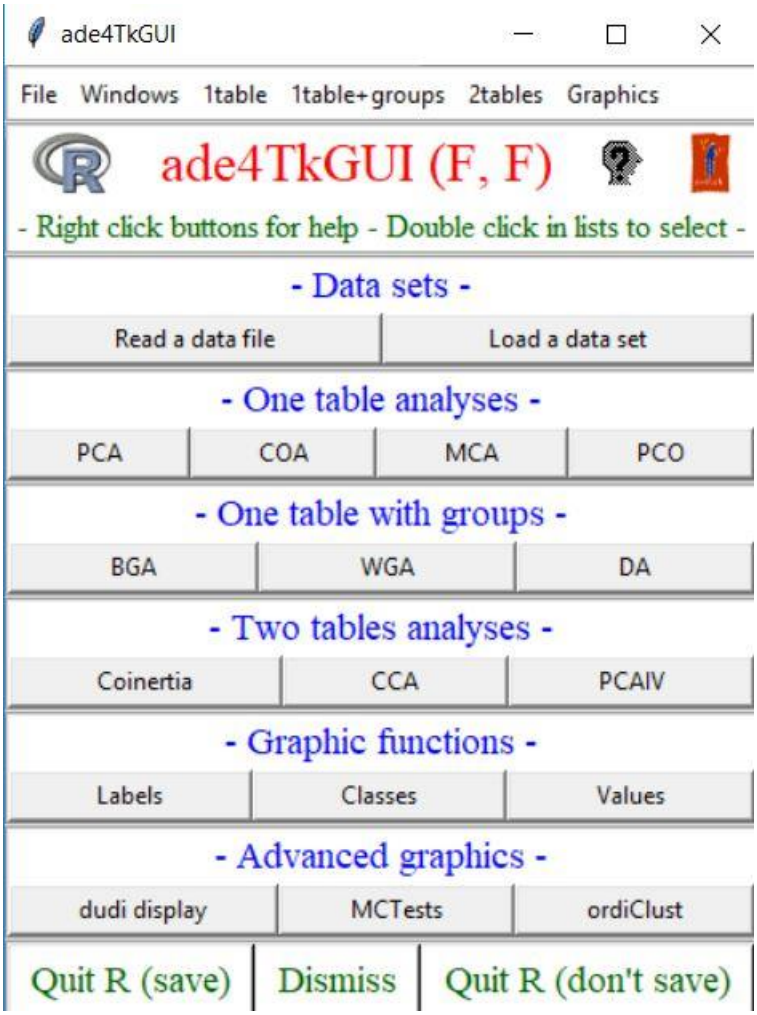

Figure 7. Fenêtre du package Ade4TkGUI

Programmes R proposés par Tessema Genanew Jember

Plusieurs programmes utiles sont fournis par l'auteur (JEMBER, 2012) dans son ouvrage « Multivariate Data Analysis Using R Software ».

En plus de programmes de calcul de corrélation, de régression, de comparaison de moyennes et d'analyse de la variance, on trouve un ensemble de programmes classiques de classification, avec ses annexes graphiques (par ex. construction de dendrogrammes), et d'ordination comme l'analyse en composantes principales, l'analyse des principales coordonnées, l'analyse non-métrique multidimensionnelles et des analyses sous contraintes comme l'analyse dite « canonique des correspondances ». Ces programmes font appel à de nombreux packages et fonctions R ; ils sont facilement mis en œuvre.

Package CAvariants

Ce package, paru en 2019, offre des solutions simples pour six analyses multivariées :

Il calcule :

- L'analyse des correspondances simple (catype="CA")
- L'analyse des correspondances doublement ordonnée (catype="SOCA")
- L'analyse des correspondances simplement ordonnée (catype="DOCA")
- L'analyse non symétrique des correspondances (catype="NSCA")
- L'analyse des correspondances non symétrique doublement ordonnée (catype="NSCA")
- L'analyse des correspondances non symétrique simplement ordonnée (catype="DONSCA").

Plusieurs fonctions sont disponibles pour la représentation des résultats.

Dans la fenêtre de Rstudio faire, par exemple, au départ d'un fichier x :
library(CAvariants) et entrer
resfichier<-CAvariants(x, catype= "CA",firstaxis=1, lastaxis=2)
summary(resfichier).

Logiciels personnels

Pourquoi refaire de nouveaux programmes alors qu'il en existe déjà suffisamment ? Les programmes personnels qui suivent sont tout spécialement destinés aux phytosociologues. **Les espèces sont en ligne et les relevés en colonnes** ; il n'est donc pas nécessaire de transposer les tableaux comme dans les autres programmes en R.

La présentation détaillée illustre les étapes du calcul et conduit à une meilleure compréhension des algorithmes. Sans cela, l'utilisation est peu productive et souvent décevante. La programmation n'est probablement pas aussi élaborée que celle d'un informaticien, mais cela fonctionne, même avec de très grands tableaux. Ce n'est pas une boîte noire que je présente mais un outil d'analyse. En plus, tous les résultats utiles sont groupés dans un seul tableau .csv, à l'exception de certains programmes d'analyse factorielle multiple avec un second petit fichier.

Pour nos propres programmes, nous utilisons Rstudio (figure 8), qui est un environnement de développement intégré (IDE) créé spécifiquement pour travailler avec R (GOULET, 2016). On le charge à partir de l'adresse suivante :
https://www.rstudio.com/products/rstudio/download/

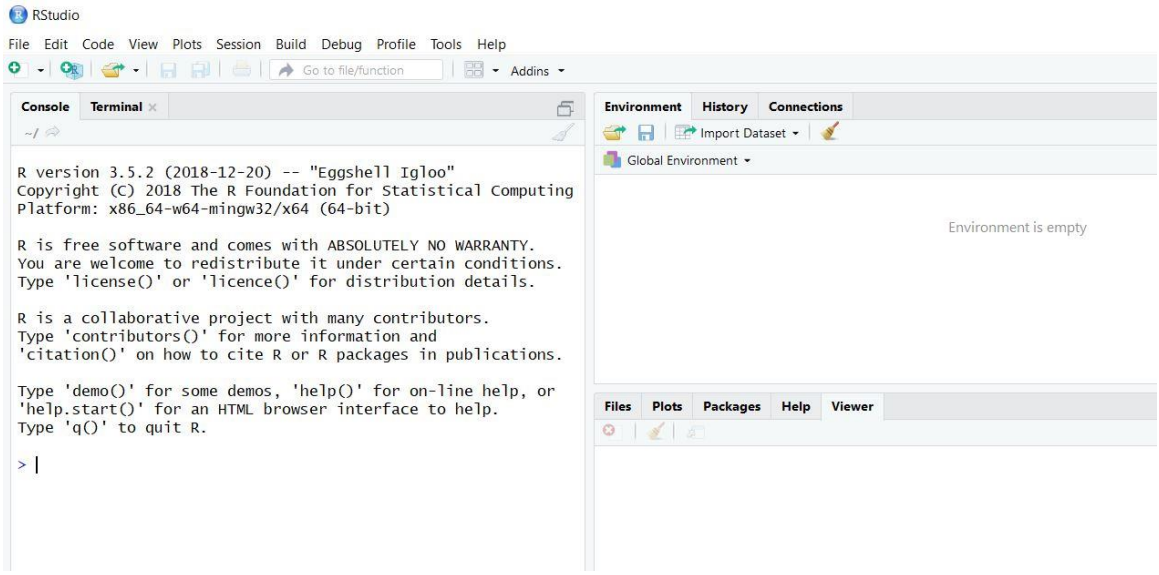

Figure 8. Fenêtre Rstudio.

Il permet de consulter dans une interface conviviale, ses fichiers de script, les lignes de commande R, les rubriques d'aide, les graphiques. Nous recommandons aussi de charger également le logiciel Visual Studio Code, qui permet de visualiser de manière élégante les programmes en format .R.

Les programmes et fonctions présentées ici ont d'abord été rédigés à partir d'une page Word, puis entrés dans l'environnement Rstudio en faisant File → New File → R Script, et ensuite sauvé en format .R. On peut les créer ou les visionner à partir de Visual Studio Code (chargement gratuit) qui donne une présentation élégante du programme.

Dans ce chapitre, nous insistons sur l'aspect calcul, mais nous ajoutons des logiciels de présentation graphique en R, profitant ainsi de la puissance de ce langage dans ce domaine. Tous nos programmes en format.R. sont regroupés dans un seul fichier .docx présent dans l'annexe. Il suffit de les reprendre et les placer dans Visual Studio Code par un simple copier-coller et de les renvoyer dans le répertoire choisi. Ils sont exécutés à partir de petits scripts en Word, de quelques lignes, fournis dans ce chapitre. Il faut d'abord définir un répertoire de travail, dans lequel se trouvent les programmes et fonctions et dans lequel les résultats des calculs sont sauvés. Par exemple, c:/Rdon/, mais chacun définit son propre répertoire. Certains programmes fonctionnent avec des fonctions personnelles. Il faut d'abord les charger dans la console en faisant Code → Source file et choisir la fonction.

Les tableaux phytosociologiques ont cette particularité de montrer beaucoup de cases vides (souvent de l'ordre de 80 à 90 %). Dans l'échelle d'abondance-dominance, on trouve les abondances de 1 à 5 mais aussi les caractères +, i, r. Il importe d'abord de les transformer en chiffres, en remplaçant les cellules vides par 0

et les +, i, r (ou autres symboles) par 1 (ou autres valeurs numériques), ce qui se fait facilement avec un petit programme. Rappelons qu'il s'agit d'une échelle ordinale. Comme on l'a vu dans les chapitres précédents, soumettre directement ces tableaux aux analyses multivariées, comme l'analyse des correspondances ou l'analyse en composantes principales, n'apporte que de pauvres résultats. Plusieurs transformations sont nécessaires. La première consiste à créer un tableau de présences (0-1), la seconde serait de créer un tableau disjonctif complet (une ligne pour les « 0 », une pour les « 1 », ainsi de suite), mais ce type ne convient pas aux tableaux floristiques avec beaucoup trop de cases vides. Une troisième solution, d'abord essayée de manière purement empirique, semble apporter les meilleurs résultats ; elle consiste à créer, pour chaque variable floristique, trois lignes : une pour les données de présence, une pour les abondances supérieures à 1 et une troisième pour les abondances supérieures à 3. Cette troisième proposition semble faire ses preuves et nous appelons ces tableaux, des tableaux disjonctifs simplifiés. Découper plus les lignes apporte une trop grande dispersion des données et produit généralement des résultats moins intéressants. Toutefois, cette dernière transformation peut sembler un peu lourde et multiplie le nombre de lignes du tableau. Dans les programmes d'analyse non symétrique des correspondances, avec des tableaux habituels d'abondance-dominance, il est aussi plus simple d'opérer une transformation logarithmique des coefficients 1, 2, 3, 4, 5 avec la fonction log1p. Cette transformation est intégrée dans le programme mfanscaVP (pondération par les premières valeurs propres des sous-programmes).

Voici les programmes avec leurs petits scripts de mise en œuvre. Les fichiers Royer et Crupetenv113 sont pratiques pour tester les progtammes.

Programme phyto1 pour la transformation d'un tableau .+ri12345 en un tableau de présence

On importe le fichier .txt, en allant dans File, puis Import Data set (voir figure avec Heading « Yes » et Row names « Use first column ») et en le chargeant. Il apparaît dans la barre du haut pour chaque utilisation ultérieure. On peut le garder en mémoire en le sauvant. Le programme est prévu uniquement pour des signes +, r, i, 1, 2 , 3 , 4 et 5 (figure 9).

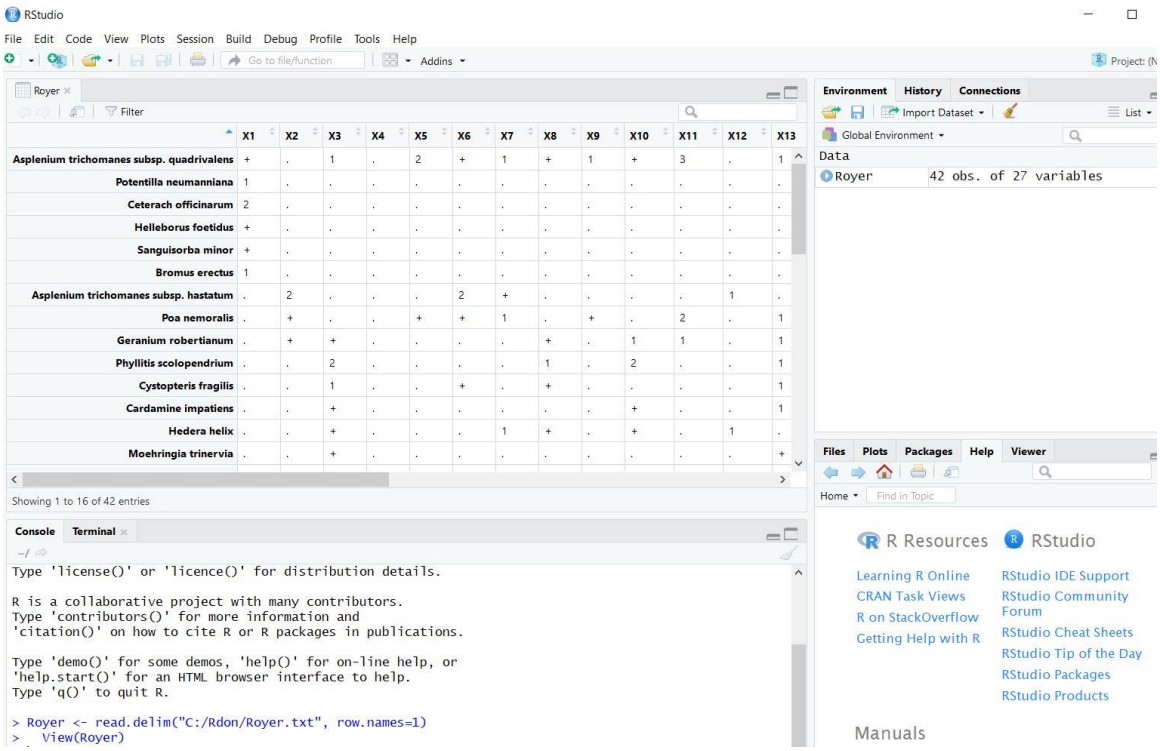

Figure 9. Fenêtre Rstudio avec le tableau à transformer.

Voici le script :

#Transformation of a table ., +, r, i, 1, 2 , 3, 4, 5 into a presence table Guy BOUXIN 2021
filename<-XXX
phyto1<-source("c:/Rdon/phyto1.R")
write.csv2(resul, "c:/Rdon/XXX1.csv")

On remplace, dans le script, XXX par le nom du fichier (deux fois, en deuxième et dernière lignes). On obtient la figure suivante (figure 10), Le tableau1, ainsi produit et sauvé en .txt, est directement utilisable dans les analyses multivariées.

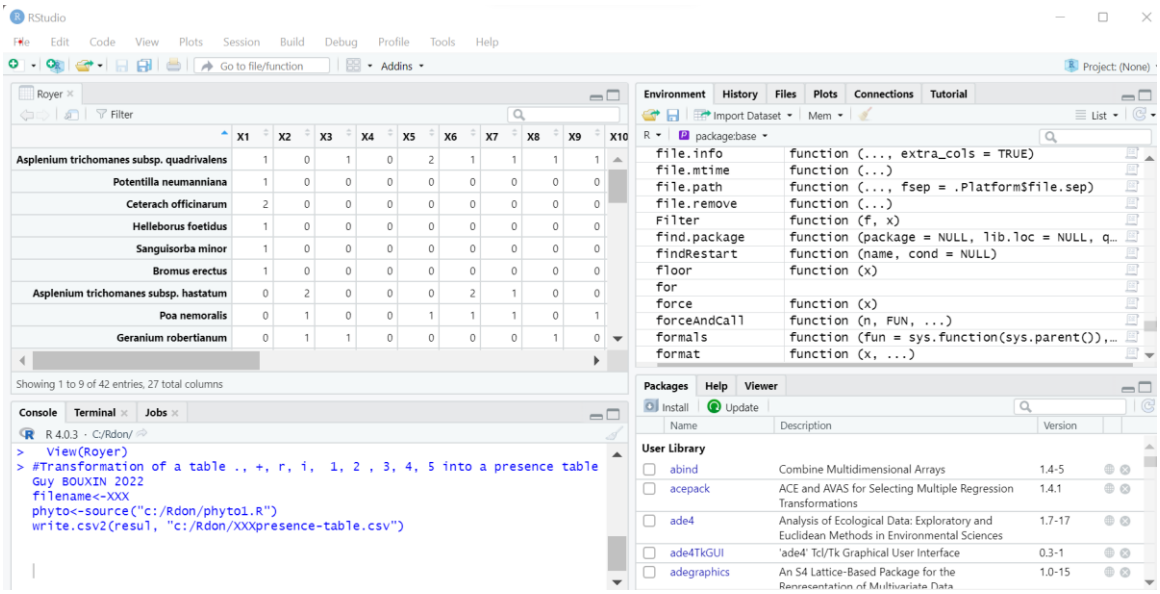

Figure 10. Fenêtre Rstudio avec le programme de transformation.

On obtient le tableau suivant :

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
Asplenium trichomanes subsp. *quad.*	1	0	1	0	1	1	1	1	1	1	1	0	1	1	0	0
Potentilla neumanniana	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Ceterach officinarum	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Helleborus foetidus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sanguisorba minor	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bromus erectus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hast.*	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1
Poa nemoralis	0	1	0	0	1	1	1	0	1	0	1	0	1	0	0	0
Geranium robertianum	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0	1

Tableau 1. Portion du tableau Royer de présence produit par le programme R.

Programme phyto12345 pour la transformation d'un tableau .+ri12345 en un tableau 012345

Le programme est toujours prévu uniquement pour des signes +, r, i, 1, 2, 3, 4 et 5. Dans chaque ligne du tableau, les signes +, r, i sont remplacés par des 1. Il ne fait donc pas de distinction entre ces trois signes. Les chiffres 1, 2, 3, 4 et 5 sont inchangés. Cette transformation des signes +, r et i (ainsi que d'autres éventuellement) est susceptible d'adaptations diverses. On peut attribuer éventuellement un score de 0,1 ou 0,5 à ces trois derniers signes pour leur donner moins d'importance par rapport à l'abondance 1.

Voici le script :

#Transformation of a table ., +, r, i, 1, 2 , 3, 4, 5 into a 12345 table Guy BOUXIN 2021
filename<-XXX

```
phyto12345<-source("c:/Rdon/phyto12345.R")
write.csv2(resul, "c:/Rdon/XXX12345.csv").
```

On remplace, dans le script, XXX par le nom du fichier (deux fois, en deuxième et dernière lignes. Les résultats sont illustrés dans le tableau 2.

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
Asplenium trichomanes subsp. *quad.*	1	0	1	0	2	1	1	1	1	1	3	0	1	1	0	0
Potentilla neumanniana	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Ceterach officinarum	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Helleborus foetidus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sanguisorba minor	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bromus erectus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hast.*	0	2	0	0	0	2	1	0	0	0	0	1	0	0	0	1
Poa nemoralis	0	1	0	0	1	1	1	0	1	0	2	0	1	0	0	0
Geranium robertianum	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0	1

Tableau 2. Portion du tableau Royer transformé en un tableau numérique.

Programme disj12345 pour la transformation d'un tableau 012345 en un tableau disjonctif simple

Ce programme est utilisé de la même manière que les précédents. Chaque ligne du tableau est remplacée par autant de lignes qu'il y a de chiffres différents dans la ligne, sauf le zéro qui n'est pas pris en compte. Il ne s'agit pas d'un tableau disjonctif complet qui ne serait pas adapté aux données de végétation, comme expliqué dans le chapitre 2.

Voici le script :

```
#Transformation of a numerical table into a disjunctive file 12345. Guy BOUXIN 2022
filename<-XXX
disj12345<-source("c:/Rdon/disj12345.R")
write.csv2(resul, "c:/Rdon/XXXdisj12345.csv")
```

On remplace, dans le script, XXX par le nom du fichier (deux fois, en deuxième et dernière lignes). Le résultat est illustré dans le tableau 3.

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
Asplenium trichomanes subsp. *quad.*	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0
Asplenium trichomanes subsp. *quad.* 2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *quad.* 3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Potentilla neumanniana	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ceterach officinarum 2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bromus erectus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hastatum*	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Asplenium trichomanes subsp. *hastatum* 2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Poa nemoralis	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
Poa nemoralis 2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Geranium robertianum	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0

Tableau 3. Portion du tableau Royer transformé en tableau disjonctif simple.

Programme disj113 pour la transformation d'un tableau 012345 en un tableau disjonctif simplifié 113

Ce programme est utilisé de la même manière que les précédents. Il est prévu uniquement pour des données de type 0, 1, 2, 3, 4 et 5.

Chaque ligne du tableau est remplacée par une, deux ou trois lignes :

- s'il n' a que des données 0 et 1, la ligne reste inchangée,
- s'il y a des données 0, 1 et des abondances 2 et 3, la ligne originelle est remplacée par deux lignes: une indiquant la presence de la variable, quelle que soit l'abondance et une seconde remplaçant les abondances deux et trois par le chiffre 1; la variable est suivie du signe ">1",
- s'il y a des données 0, 1 et des abondances 2, 3, 4 et 5, il faut ajouter une troisième ligne pour les abondances 4 et 5; la variable est suivie du signe ">3".

Toutefois, si une variable ne présente que des abondances 2, 3, 4 et 5 sans des abondace 1, le programme ne prévoit que deux lignes, une pour les abondances >1 et une pour les abondances >3; Il env a de même s'il n'y a que des abondances 4 et 5; il n'y a plus alors qu'une seule ligne marque ">3". Cette façon de procéder évite de donner trop de poids aux espèces abondantes, qui se retrouveraint avec deux ou trois lignes identiques.

Voici le script :

#**Transformation of a numerical table 12345 table into a disjunctive table 1>1>3.** Guy BOUXIN 2022
filename<-XXX
disj113<-source("c:/Rdon/disj113.R")
write.csv2(resul, "c:/Rdon/XXXdisj113.csv")

On remplace, dans le script, XXX par le nom du fichier (deux fois, en deuxième et dernière lignes). Les résultats sont illustrés dans le tableau 4.

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
Asplenium trichomanes subsp. *quad.* >1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
Asplenium trichomanes subsp. *quad.* >1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
Potentilla neumanniana	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ceterach officinarum >1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ceterach officinarum	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bromus erectus	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hast.* >1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Asplenium trichomanes subsp. *hast.* >1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
Poa nemoralis >1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Poa nemoralis	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0
Geranium robertianum	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0

Tableau 4. Tableau disjonctif simplifié prêt pour les analyses multivariées.

Programmes PCoA et NMDS

Ces programme sont utilisés de la même manière que les précédents.

```
#PCoA
filename2<-as.matrix(XXX)
library(vegan)
library(permute)
k=6
nbrrel= ncol(filename2)
nomrel<-list(1 :nbrrel)
nomcol<-list(1 :k)
nomcol2<-list(1 :2)
nomrel<-colnames(filename2)
nomrel
k=6
filename<-as.matrix(t(filename2))
#Matdis<-dist(filename, method = "euclidean", diag=TRUE, upper=TRUE)
Matdis<-vegdist(filename, method = "chord", diag=TRUE, upper=TRUE)
Matdis
for(j in 1:k)                    #
{
nomcol[1+(j-1)]<-paste("coord",j)
}
nomcol2[1]<-paste("eigenvalues")
nomcol2[2]<-paste("% of trace")
hist(Matdis, breaks=40, include.lowest = TRUE, right = TRUE,
labels=FALSE, border = NULL,
main = paste("Simple disjunctive table – chord distance "), plot=TRUE,
axes = TRUE,
warn.unused = TRUE)
x<-cmdscale(Matdis, eig=TRUE, k=6)
resulpcoacoord<-matrix(1: nbrrel*k,nrow=nbrrel,ncol=k,byrow=TRUE)
colnames(resulpcoacoord)<- nomcol
for(i in 1 :nbrrel)
{
for(j in 1:k)
{
resulpcoacoord[i,j]<-x$points[i,j]
}
}
rownames(resulpcoacoord)<-nomrel
```

```
trace=sum(x$eig)
Lam<-matrix(1:nbrrel*1,nrow=nbrrel,ncol=1)
Lam<-x$eig
Lam
for(i in 1:nbrrel)
{
Lam[i]<-Lam[i]/trace*100
}
resulpcoaeig<-matrix(1: nbrrel*1+1,nrow=nbrrel,ncol=2,byrow=TRUE)
colnames(resulpcoaeig)<-nomcol2
for(i in 1 :nbrrel)
{
resulpcoaeig[i,1]<-x$eig[i]
resulpcoaeig[i,2]<-Lam[i]
}
write.csv2(resulpcoacoord, "c:/Rdon/ XXXPCoAchordcoordrel.csv")
write.csv2(resulpcoaeig, "c:/Rdon/XXXPCoAchordeig.csv")
```

Remarque : pour calculer la distance « chord » il faut installer les packages vegan et permute et utiliser la fonction vegdist()

```
#NMDS
library(MASS)
filename2<-as.matrix(XXX)
library(vegan)
library(permute)
k=3
nbrrel= ncol(filename2)
nomrel<-list(1 :nbrrel)
nomcol<-list(1 :k)
nomcol2<-list(1 :2)
nomrel<-colnames(filename2)
k=6
filename<-as.matrix(t(filename2))
#Matdis<-dist(filename, method = "euclidean")
Matdis<-vegdist(filename, method = "chord", diag=TRUE, upper=TRUE)
for(j in 1:k)
{
nomcol[1+(j-1)]<-paste("coord",j)
}
nomcol2[1]<-paste("eigenvalues")
nomcol2[2]<-paste("% of trace")
hist(Matdis, breaks=40, include.lowest = TRUE, right = TRUE,
labels=FALSE, border = NULL,
main = paste(" Abundance table- distance matrix"), plot=TRUE,
axes = TRUE,
warn.unused = TRUE)
x<-isoMDS(Matdis, k=6, maxit = 50)
resulNMDScoord<-matrix(1: nbrrel*k,nrow=nbrrel,ncol=k,byrow=TRUE)
colnames(resulNMDScoord)<- nomcol
for(i in 1 :nbrrel)
{
for(j in 1:k)
```

```
{
resulNMDScoord[i,j]<-x$points[i,j]
}
}
rownames(resulNMDScoord)<-nomrel
write.csv2(resulNMDScoord, "c:/Rdon/ XXXNMDScoordrel.csv")
```

Programmes pca, caf et nascaf

Les résultats numériques des programmes pca et ca correspondent à ceux produits par RCmdr, ceux de nsca à ceux fournis par le programme nsca de CAvariants().

Ces trois programmes fonctionnent identiquement. Il suffit de remplacer XXX par le nom du fichier, de préciser le nombre de permutations (nper) et le nombre d'axes à calculer (nax).

Ils produisent un fichier de résultats en colonnes avec,

- pour la première ligne, successivement pour chaque axe, la valeur propre correspondante, le pourcentage par rapport à la trace de la matrice de corrélation ou de covariance et la probabilité associée à la valeur propre,
- Pour les lignes suivantes et pour chaque espèce ou relevé, la coordonnée sur l'axe correspondant, la contribution relative (ou $\cos^2$ dans le cas de PCA) et la probabilité associée à la contribution relative (ou $\cos^2$).

Dans les programmes qui suivent, il faut d'abord charger les fonctions covarca.R et covarnsca.R dans Rstudio, en allant dans Code (barre du haut) puis GoTo File/Function.

Voici les scripts :

1. Pour l'analyse en composantes principales

```
#pca with permutations Guy BOUXIN 2022
filename<-XXX
nper=n1
nax=n2
pca<-source("c:/Rdon/pca.R")
write.csv2(resulpca, "c:/Rdon/XXXpca.csv")
```

2. Pour l'analyse des correspondances

```
#caf with permutations and personal functions Guy BOUXIN 2022
filename<-XXX
nper=n1
nax=n2
caf<-source("c:/Rdon/caf.R")
```

```
write.csv2(resulca, "c:/Rdon/XXXcaf.csv")
```

3. Pour l'analyse non symétrique des correspondances

```
#nscaf with permutations and personal functions Guy BOUXIN 2022
filename<-XXX
nper=n1
nax=n2
nscaf<-source("c:/Rdon/nscaf.R")
write.csv2(resulnsca, "c:/Rdon/XXXnscaf.csv")
```

On remplace, dans le script, XXX par le nom du fichier (deux fois, en deuxième et dernière lignes), n1 par le nombre de permutation souhaité (1000 au minimum, 10000 si possible, en fonction de la taille du nombre de colonnes du tableau), n2 par le nombre de valeurs propres calculées.

Cela produit un tableau comme suit :

	coord 1	Cr% 1	P 1	coord 2	Cr% 2	P 2
Vp,Vp% et P	5,74790056	13,6854775	2,8	4,34982694	10,3567308	8,5
Asplenium trichomanes subsp. *quadrivalens*	0,27685687	0,07664973	47	0,047633	0,0022689	90,4
Potentilla neumanniana	-0,44413484	0,19725576	21	-0,57490655	0,33051755	6,9
Ceterach officinarum	-0,3209514	0,1030098	31	-0,69205829	0,47894467	2,9
Helleborus foetidus	-0,28100306	0,07896272	24	-0,76686229	0,58807777	2,5
Sanguisorba minor	-0,28100306	0,07896272	25	-0,76686229	0,58807777	2,4
Bromus erectus	-0,28100306	0,07896272	24	-0,76686229	0,58807777	2,1
Asplenium trichomanes subsp. *hastatum*	-0,19001115	0,03610424	63	0,02792301	0,00077969	94,2

Tableau 5. Analyse en composantes principales du fichier "Royerpres", résultats pour les deux premiers axes et les sept premières espèces.

Ces programmes affichent aussi, dans la fenêtre viewer de RStudio, l'histogramme de la distribution des coefficients de corrélation ou de covariance, ce qui nous informe sur la qualité de l'analyse. Cet histogramme peut être sauvé.

Les coordonnées des espèces et relevés sont susceptibles d'être reprises pour construire des graphes pour les espèces ou les relevés. Le langage R offre beaucoup de possibilités et nous présentons des logiciels pour dessiner ces graphes.

Programmes mfapca, mfacaf et mfanscaf d'analyse factorielle multiple

Ces programmes fonctionnent toujours comme les précédents, si ce n'est qu'il faut préciser en plus les nombres respectifs de lignes n1 et n2 dans les sous-tableaux.

1. Pour l'analyse factorielle multiple, basée sur l'analyse en composantes principales, la pondération est basée sur les premières valeurs propres des sous-tableaux.

#mfapca Guy BOUXIN 2022
```
filename<-XXX
co<-c(n1,n2,…)
nax=n3
mfapca<-source("c:/Rdon/mfapca.R")
write.csv2(resulmfapca, "c:/Rdon/XXXmfapca.csv")
```

2. Pour l'analyse factorielle multiple, basée sur l'analyse des correspondances. La pondération est basée sur les premières valeurs propres des sous-tableaux.

#mfacaf with personal functions Guy BOUXIN 2022
```
filename<-XXX
co<-c(n1, n2,…)
nper=n3
nax=n4
mfacaf<-source("c:/Rdon/mfacaf.R")
write.csv2(resulmfaca, "c:/Rdon/XXXmfacaf.csv")
```

3. Pour l'analyse factorielle multiple, basée sur l'analyse non symétrique des correspondances, variante 1

La pondération est basée sur la densité des sous-tableaux. Ce programme fonctionne avec des données de présence (par exemple fichier 113).

#mfanscaf with personal functions and permutations, weighting by the densities of the subtables Guy BOUXIN 2022
```
filename<-XXX
co<-c(n1,n2,…)
nper=n3
nax=n4
mfanscafdensity<-source("c:/Rdon/mfanscafdensity.R")
write.csv2(resul1mfansca, "c:/Rdon/ XXX-mfansca -original table & sub-tablesdensity.csv")
write.csv2(resul2mfansca, "c:/Rdon/ XXX-mfansca-density.csv")
```

4. Pour l'analyse factotielle multiple basée sur l'analyse non symétrique des correspondances, variante 2

La pondération est basée sur les premières valeurs propres respectives des sous-tableaux. Ce programme fonctionne avec des données d'abondance. Une transformation logarithmique des données d'abondance est necessaire et incluse dans le logiciel.

#mfanscaf with personal functions and permutations, log1p transformation and weighting by the first eigenvalues of the subtables Guy BOUXIN 2023

```
filename<-XXX
co<-c(n1,n2,…)
nper=n3
nax=n4
mfanscafEV<-source("c:/Rdon/mfanscafEV.R")
write.csv2(resul1mfansca, "c:/Rdon/ XXX-mfansca-EV -original table & sub-tables eigenvalues.csv")
write.csv2(resul2mfansca, "c:/Rdon/ XXX-mfansca-EV.csv")
```

On remplace XXX avec le nom du fichier,bon spécifie les nombres respectifs de lignes des sous-tableaux (n1, n2, …), le nombre de simulations (nper sauf pour mfapca) et le nombre de valeurs propres à calculer (nax).

Dans ces deux derniers programmes, le nombre d'axes à calculer ne peut dépasser le plus petit nombre de lignes parmi les sous-tableaux. Les résultats apparaissent dans deux fichiers, un habituel et un second avec les nombres de lignes, les densités ou les valeurs propres, pour les sous-tableaux et pour le tableau transformé.

Les programmes mfacaf et mfanscaf en R sont quelque peu différents de ceux proposés précédemment. Les programmes anciens étaient rédigés de manière à produire les mêmes résultats numériques que ceux fournis par le logiciel ADE-4 (version utilisée en 2006). Dans ce logiciel, le calcul des valeurs propres des sous-tableaux se fait sans changer les sommes sur les colonnes et de ce fait la somme des traces des sous-tableaux est égale à la trace du tableau complet, comme dans l'analyse factorielle multiple basée sur l'analyse en composantes principales. Dans cette dernière, les lignes des sous-tableaux sont identiques à celles du tableau complet. Dans nos propres programmes en R, les valeurs propres des sous-tableaux sont calculées en tenant compte des sommes respectives sur les colonnes qui sont, bien entendu, différentes d'un sous-tableau à l'autre.

Dans cette nouvelle version d'analyse factorielle, il importe qu'il n'y ait aucune colonne vide dans l'un ou l'autre sous-tableau, du moins dans les versions basées sur l'analyse duale des correspondances ou l'analyse non symétrique.

Les résultats apparaissent dans le tableau 6.

Le fichier Crupetvegenv113 comprend un sous-fichier floristique de 57 lignes et un sous-fichier environnemental de 44 lignes.

	coord 1	CR 1	P 1
Vp, Vp%, P	0.21664321	13.2424065	0
Agrostis stolonifera	-0.2408397	5.80037589	10.05
Alnus glutinosa	-0.049166	0.24172958	41.57
Alnus glutinosa >1	-0.16093688	2.59006786	30.73
Alnus glutinosa >3	0.00905918	0.00820688	72.11
Alnus incana	0.15348452	2.35574976	33.09

Alnus incana >1	0.14713428	2.16484949	16.29
Angelica sylvestris	-0.04813769	0.23172372	72.4
Apium nodiflorum	-0.04966777	0.2466887	27.48
Caltha palustris	0.05544199	0.30738139	38.83
Calystegia sepium	0.00992352	0.00984762	91.2
Cardamine amara	0.24977337	6.23867355	10.06

Tableau 6. Analyse factorielle multiple basée sur l'analyse non symétrique des correspondances du fichier "Crupetvegenv113". Résulats pour le premier axe et les neuf premières espèces, *Alnus glutinosa* étant divisé en trois variables et *Alnus incana* en deux.

Pour les premières lignes (deux plus le nombre de sous-fichiers), les trois programmes fournissent les valeurs propres de chaque axe, les contributions relatives correspondantes et les probabilités (uniquement pour mfansca) . Pour les lignes suivantes, la présentation est la même que dans les analyses simples, avec ou sans les probabilités.

Programme Cs pour la définition d'espèces caractéristiques

Ce programme permet d'établir, à partir d'un fichier de présences, la liste des espèces caractéristiques d'un groupement ou association, au sein d'un tableau comprenant plusieurs syntaxons. Cette definition ne vaut qu'à l'échelle du grand tableau retenu. Il faut que les relevés d'un même groupement soient juxtaposés. Il suffit d'entrer le nom du fichier complet avec les colonnes réarrangées, les nombres respectifs de relevés dans chaque groupement ainsi que le nombre de permutations (nper, 10.000 recommandé). Voici le script :

```
# Definition of character species of a presence table. Guy BOUXIN 2022
filename<-XXX
co<-c(n1,n2,…….)
nper=10000
cs<-source("c:/Rdon/cs.R")
write.csv2(resul, "c:/Rdon/XXXcs.csv")
```

Programme Graph pour la construction des graphes à partir des coordonnées des analyses multivariées

Il y a deux programmes : un lorsqu'il n'y a qu'un type de variable, comme des variables floristiques et un second lorsqu'on mélange, par exemple, des variables floristiques et des variables environnementales.

Pour le premier programme, il faut sortir un fichier comprenant une première colonne avec la liste des noms de variables et autant de colonnes qu'il y a de coordonnées considérées, indiquées coord1, coord2, … dans la première ligne. Les décimales doivent être marquée par un point. Un exemple est donné dans le tableau 7.

	coord1	coord2	coord3
1	-0.134601	-0.460529	1.277298
2	1.050046	0.398986	0.937062
3	-0.118671	-0.327056	1.805153

4	-0.270912	-0.216169	3.389558
5	2.972145	0.288958	1.202091
6	2.623666	0.032145	-0.418913
7	-0.656541	-0.272471	1.00179
8	1.581773	0.184523	-0.196533
9	2.203882	0.262407	0.420033
10	2.989747	0.539129	0.600225

Tableau 7. Coordonnées de 10 relevés sur les trois premiers axes d'une analyse multivariée.

Avec le second programme, il faut préciser le type de caractère correspondant aux variables, sur des colonnes complémentaires. Voici un exemple de fichier avec 6 colonnes, une avec les variables, trois avec les coordonnées, la quatrième précisant la couleur et la cinquième, le type de caractères.

coord1	coord2	coord3	colo	cara	
Syzygium rowlandiii G	0,044090554	-0,061651151	-0,122712784	a	3
Syzygium rowlandiii G 20-39	-0,087414582	-0,096412416	-0,011908992	a	3
Syzygium rowlandiii G >40	0,023628466	0,047747795	-0,142272082	a	3
Syzygium rowlandii p	0,025185604	-0,028021546	-0,060402434	a	3
Syzygium rowlandii p 20-39	0,095857426	-0,06464239	0,011357952	a	3
Syzygium rowlandii +	0,153844472	-0,043228559	-0,069207629	a	3
Syzygium rowlandii + 20-39	0,102597138	-0,089898555	0,022400078	a	3
Carapa grandiflora G	-0,002141596	-0,008193828	-0,067967865	a	3
Carapa grandiflora G 20-39	-0,01233258	0,0084035	-0,016967498	a	3
Carapa grandiflora p	-0,04500537	0,133986642	-0,072509291	a	3
pente0	0,08301115	-0,034781038	-0,127867489	b	2
pente 1-9	-0,03090512	-0,044024429	-0,023049512	b	2
pente 10-19	-0,012415985	0,016079666	0,151575951	b	2
pente>19	-0,015671775	0,029473705	0,054554763	b	2
profondeur du sol 1	-0,068887046	0,037281395	0,189510058	b	2
profondeur du sol 2	-0,011419242	0,115827687	0,010251562	b	2
profondeur du sol 3	0,104324559	-0,186361179	-0,144547907	b	2
couvert 1	0,102261718	-0,099365189	0,161534433	b	2
couvert 2	-0,015350527	0,084104893	0,070700006	b	2
couvert 3	-0,094069771	-0,06330587	-0,145327554	b	2
couvert 4	0,03117685	0,04531407	-0,031693173	b	2
hydrographie 1	-0,017978603	-0,000285474	-0,03427732	b	2
hydrographie 2	-0,000100038	-0,020634889	-1,19E-05	b	2

Tableau 8. Fichier .txt utilisé pour construire un graphe avec deux couleurs et deux caractères.

La construction du graphe se base sur la function ggplot(). La position des étiquettes de variables est ra ngée de manière optimale grâce à l'instruction "gridExtra::grid.arrange(p1, ncol = 1)". On peut jouer sur les couleurs (a, b ou autres ou même une seule lettre) et sur le type de caractère : 1 = texte normal, 2 = texte en gras, 3 =texte en italique, 4 = texte en italique gras et 5, caractères grecs.

Voici le script du premier programme :

```
library(ggplot2)
```

```
library(ggrepel)
dat<-XXX
nom = rownames(dat)
nom[dat$coord1 <= 0.05 & dat$coord1> -0.05 & dat$coord2<= 0.05 & dat$coord2> -0.05] = ""
p<-ggplot(dat, aes(x = coord1, y = coord2, label = nom, fontface = 1)) + geom_point() +
geom_hline(yintercept=0) +
geom_vline(xintercept=0) +
ggtitle("Analyse **********") +
xlab("coord1 - % inertie : ****") +
ylab("coord2 - % inertie : ****") +
theme_bw()+
theme( axis.text.x = element_text(size = 24), axis.text.y = element_text(size = 24), plot.title =
element_text(size=24, face=1), axis.title.x = element_text(size=24, face=1), axis.title.y =
element_text(size=24, face=1)) +
geom_text_repel(box.padding = 0.25, min.segment.length= 0.25, segment.curvature = -0.1, segment.ncp =
3, segment.angle = 20, max.overlaps = getOption("ggrepel.max.overlaps", 10),
size = 8)
gridExtra::grid.arrange(p, ncol = 1)
ggsave("c:/Rdon/ XXX.png", plot = p, width = 12, height = 12)
```

Il faut préciser le type d'analyse dans ggtitle et les pourcentages d'inertie des axes. Suivant la taille du fichier, et notamment le nombre de variables à placer, il faut adapter, dans geom_text_repel, le nombre maximum de superposition de variables (getOption("ggrepel.max.overlaps", 10 ou 20), la taille des lettres (8 ou 6).

Pour le second programme, on change la cinquième ligne comme suit :

```
p<-ggplot(dat, aes(x = coord1, y = coord2, label = nom, fontface = cara)) +
```

On peut aussi jouer sur les couleurs des variables avec la cinquième ligne.

```
p<-ggplot(dat, aes(x = coord1, y = coord2, label = nom, colour=colo, fontface = cara)) + geom_point()+ sca
le_color_manual(values = c("black","black")) + labs(title = "mfansca")
```

Il faut préciser les axes considérés (coord1, coord2, …), le type d'analyse dans la function ggtitle et les pourcentages d'inertie des axes.

Programmes de classification

Les programmes de classification sont fortement inspirés de ceux présentés par JEMBER (2012) et bénéficient des nombreuses informations fournies par le blog de Claire Della Vedova (https://delladata.fr/). Le programme "Cluster analysis using base R and euclidean distance as ressemblance measure" est utilisé pour classer les relevés de végétation. Il fonctionne à partir de deux types de fichiers : les coordonnées des relevés ou encore les abondances des espèces. Si on utilise les coordonnées des relevés, les données sont présentées comme dans le tableau 7. Si on utilise les abondances d'un tableau phytosociologique, ce dernier

doit au préalable être transposé. Il suffit de remplacer, comme d'habitude, XXX par le nom du fichier chargé dans RStudio. Un complément permet de dessiner un nombre fixé de clusters dans le dendrogramme.

Le programme "Kmeans avec R" est un autre classique, mais il ne s'agit plus d'une classification au sens propre mais plutôt d'un rangement ou classement des relevés dans un nombre de clusters préalablement fixé.

Les programmes Elbow, Silhouette et Gap permettent en principe de fixer le nombre idéal de clusters, en se basant sur la function Kmeans, mais il faut éviter d'y voir une panacée. Le programme Elbow ne produit pas de résultats très nets, contrairement au programme Silhouette. Le programme Gap fournit souvent des résultats différents des deux autres. Dans nos analyses, j'ai surtout privilégié les résultats du programme silhouette.

Enfin, un dernier programme dessine de ellipses autour de clusters représentés dans un espace à deux dimensions. Cela permet de dessiner de belles figures.

Dans tous ces programmes, il est recommandé d'utiliser des variables transformées comme dans le tableau 8.

Comme précédemment, il faut remplacer XXX par le nom du fichier de données6

1. **Classification basée sur la distance euclidienne comme mesure de ressemblance**

```
newdat<-as.matrix(XXX)
distance =dist(newdat, method= "euclidean") #distance matrix
structure =hclust(distance, method="ward.D")
dendrogram=as.dendrogram(as.hclust(structure))
plot(dendrogram, nodePar = list(lab.cex = 0.8, lab.col= "black", pch = NA, axes = T), cex.axis=1)
mtext("", side = 1, line=1, cex=0.8,font = 2)
mtext("Dissimilarity", side=2,line=2.5,cex=1,font=2)
title(main="HAC",cex.main=1)
#option facultative dessinant les clusters
#k doit être défini, par ex. après une première classification.
k= *
range(distance)
cor(distance,cophenetic(structure))
#division of the dendrogram into specified number of clusters can be done after inspection
rect.hclust(structure, k=k,border=(1:k))
ClusterID=cutree(structure,k=k)
write.csv2(ClusterID, "c:/Rdon/XXXclusters.csv")
```

Il faut remplacer * pour le chiffre choisi.

2. **Kmeans avec R**

Il faut d'abord installer les packages tidyverse, factoextra et NbClust

```
library(tidyverse)
 str(XXX)
km.out = kmeans(XXX,centers=*,nstart =20)
km.out$cluster
resul<-as.matrix(km.out$cluster)
write.csv2(resul, "c:/Rdon/ XXX.csv")
```

Il faut remplacer * par le nombre retenu de centres.

3. Méthode Nbclust et kmeans Elbow

```
library(factoextra)
library(NbClust)# Elbow method
fviz_nbclust(XXX, kmeans, method = "wss") + geom_vline(xintercept = 5, linetype = 2)+
labs(subtitle = "Elbow method")
```

4. Nbclust et kmeans Silhouette method

```
library(factoextra)
library(NbClust)# Silhouette method

fviz_nbclust(XXX, kmeans, method = "silhouette")+
labs(subtitle = "Silhouette method")
```

5. Programme avec ellipses

```
# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
set.seed(123)
fviz_nbclust(XXX, kmeans, nstart = 25, method = "gap_stat", nboot = 50)+
labs(subtitle = "Gap statistic method")
km.out=kmeans(XXX,centers=5,nstart =20)
str(km.out)
pairs(XXX, col=c(1:2)[km.out$cluster])
library(factoextra)
fviz_cluster(km.out, XXX, ellipse.type = "norm")
```

Le logiciel GINKGO de VegAna

Le logiciel Ginkgo, produit par l'Université de Barcelone, permet d'exécuter facilement plusieurs analyses multivariées, qui ont été développées dans un contexte d'écologie numérique. Il est facile à utiliser pour des personnes non spécialisées en statistique. Ce logiciel fait partie du « package » VegAna, un environnement de travail qui fournit plusieurs outils pour éditer et analyser la flore et la végétation.
Le programme est écrit en langage Java.

Ginkgo est accessible à partir du lien :
biodiver.bio.ub.es/**ginkgo**/**Ginkgo**.htm

La fenêtre suivante (Figure 11) montre les possibilités du programme.

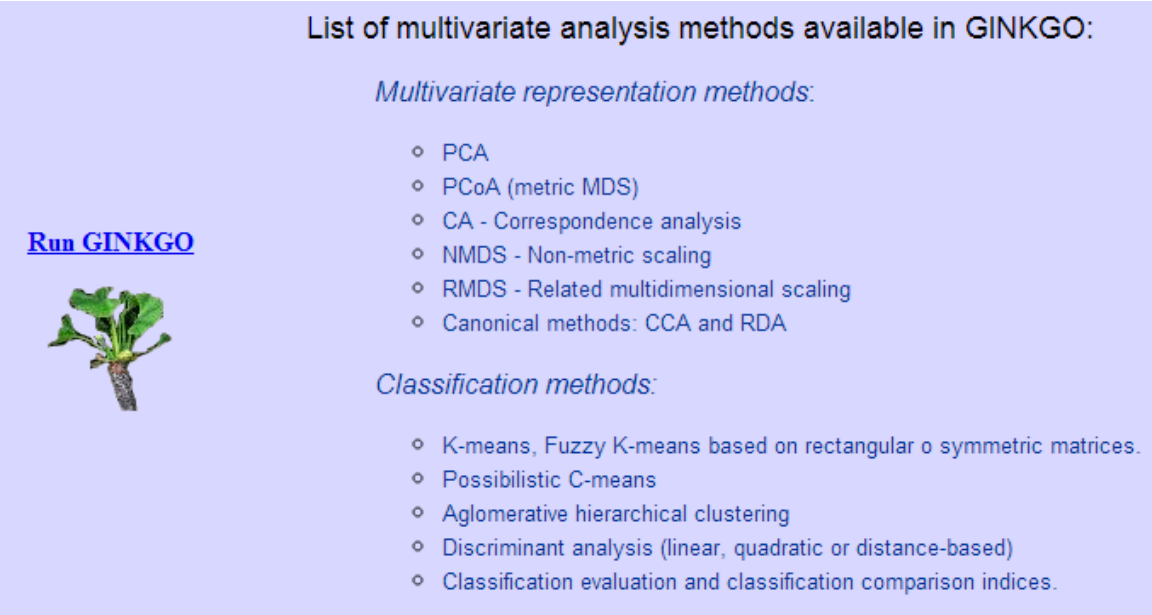

Figure 11. Fenêtre d'entrée du logiciel Ginkgo.

Deux autres ouvrages

Signalons également les ouvrages très détaillés de BOCCARD, GILLET & LEGENDRE (2011) et de WILDI (2013) qui proposent beaucoup de logiciels en R en écologie numérique.

janvier 2024

Références

BERTHET, P., FEYTMANS, E., STEVENS, D. & GENETTE, A. (1976). A new divisive method of classification illustrated by its applications to ecological problems. . *Proc. Ninth Int. Biom. Conf., invited papers*. **Vol. II** : 366-382.

BORCARD, D., GILLET, F. & LEGENDRE, P. (2011). *Numerical Ecology with R*. Springer. 306 pp.

BOUXIN, G. (2005). Ginkgo, a multivariate analysis package. *Journal of vegetation Science* **16**: 355-359.

BOUXIN, G. (2016). *Analyse statistique des tableaux de relevés de végétation. Recherche d'adéquation entre les données de végétation et les techniques statistiques, au moyen d'exemples*. Éditions Universitaires Européennes. 440 pp.

DIXON, P. (2003). VEGAN, a package of R functions for community ecology. *Journal of vegetation Science* **14** : 927-930.

GOULET, V, avec la collaboration de L. CARON (2016). Introduction à la programmern en R, cinquième édition. Université Laval, Québec. 202 pp.

HUSSON, R., LÊ, S. & PAGÈS, J. (2009). *Analyse des données avec R*. Presses Universitaires de France. 224 pp.

JEMBER, T. G. (2012). Multivariate Data Analysis using R software. Practical exercices for multivariate software. LAP Lambert Academic Publishing. 104 pp.

LOMBARDO, R. & BEH, E.J. (2019). Package 'CAvariants'. The R Foundation. https://www.R-project.org

R Core Team (2018). R: *A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

ROUX, M. (1985). *Algorithmes de classification*. Masson, Paris. 151 p.

WILDI, O (2013). Data analysis in vegetation ecology. Second edition. WILEY-BLACKWELL. 301 pp.