Analyses multivariées avec ade4 dans R

K. JACQUET & R. PRODON

EPHE BEV Montpellier

« *L'analyse statistique n'est vraiment utile qu'à des personnes qui n'en n'ont pas la maîtrise, et n'est maîtrisée que par des personnes qui n'en n'ont pas vraiment l'usage* » (D. Chessel).

Le présent "mode d'emploi", empirique et non technique, s'adresse à la première catégorie de personnes – en sollicitant l'indulgence de la seconde ! Il part du principe que la pratique est une des voies menant à la compréhension. Il ne donne aucune justification mathématique (pour cela, se reporter aux fiches de biostatistique mentionnées ci-dessous ; pour les méthodes standard comme l'ACP et l'AFC, il existe aussi une abondante documentation sur internet). Il ne traite qu'un choix de méthodes et ne détaille pas toute la syntaxe des fonctions et options.

– Une biblio succincte est proposée seulement pour les méthodes les moins familières.

– On trouvera une très riche documentation sur le site de biostatistique de l'université Lyon-1 : http://pbil.univ-lyon1.fr/R/enseignement_div.php?contents=html/entree

– Pour une présentation simple des rapports entre analyses multivariées et concepts écologiques :
Prodon R., Lebreton J.D. 1994. Analyses multivariées des tableaux espèces-milieu: structure et interprétation écologique. *Vie et Milieu* 44 : 69-91.

– Le manuel de référence de ade4 est en préparation :
Dray S., Dufour A.-B., Jombart T., Pavoine S., Thioulouse J. *Multivariate analysis of ecological data with ade4*. Springer.

Téléchargement et installation du « package » ade4

Se télécharge directement à partir de la ligne de commande de R: cliquer sur l'onglet « Packages » puis sur « Installer le(s) package(s) ». Choisir un site miroir proche (Lyon, Toulouse, ou Paris), puis le package « ade4 ». Si l'ordinateur n'est pas connecté à internet, il suffit de récupérer le fichier .zip du package ade4, et de faire « installer un package depuis un fichier .zip » dans le menu de R.

Une fois le logiciel R ouvert, taper : **> library(ade4)** pour charger la bibliothèque ade4. Il faut faire cette manœuvre chaque fois que l'on ouvre R et que l'on veut utiliser ce package.

Les fonctions du package ade4 (commandes, syntaxe)

Lorsqu'on cherche dans R une fonction dont on a oublié le nom, aller sur le site CRAN du logiciel R (http://www.r-project.org/) et cliquer sur « **search** » (à gauche sur la page web). Google cherche alors les pages web liées au logiciel R qui contiennent le mot clef.

On obtient les références du package ade4 et la ***liste*** de toutes ses fonctions et jeux de données par:

> library(help=ade4)

On obtient une documentation générale sur une fonction donnée (et notamment sa syntaxe complète et toutes les options) par :

> ?nom_de_la_fonction()

Pour commencer les analyses, il est recommandé de se mettre immédiatement dans le bon répertoire de travail (c'est là que seront stockés les fichiers et figures que l'on désire sauvegarder) :

barre d'outils de la console / Fichier / Changer le répertoire courant

☛ Il faut attribuer un nom d'objet à chaque analyse, car ce nom pourra être rappelé dans des commandes ultérieures. Il est utilisé comme préfixe des noms des fichiers de sortie, que l'on pourra ainsi afficher et utiliser dans des calculs (voir plus loin: § 4). Il suffit de taper ce nom pour obtenir la liste de tous les fichiers produits par l'analyse.

Dans les exemples du présent document, tous les *noms* et *titres* donnés arbitrairement en exemples sont ***en italiques***. Pour les analyses, on utilisera en général comme nom la lettre « ***x*** ».

☛ Voir en ANNEXE quelques informations élémentaires sur la syntaxe, la saisie, la manipulation et la visualisation des tableaux sous R et ade4.

PLAN

Les méthodes d'analyse à **un tableau** (mesures environnementales, morphométrie, tableaux de comptage, etc.) [n° **1** à **9**] – essentiellement descriptives – mettent en évidence les ressemblances/différences entre lignes du tableau (individus statistiques : objets, échantillons, relevés) et les liaisons entre colonnes (variables, descripteurs, caractères, attributs), sans hypothèse *a priori*, donc sans tests statistiques.

Les méthodes d'analyse à **un tableau avec individus en classes** [n° **10** à **12**] font de même, en mettant de plus en évidence les ressemblances/différences intra- et inter-classes, et en permettant de tester la réalité de l'effet-classe sur les ressemblances/différences entre individus.

Les méthodes d'analyse à **deux tableaux** [n° **13** à **17**] révèlent les structures (gradients, blocs) communes à deux tableaux, ou mettent en évidence le pouvoir prédictif de l'un des tableaux (en général des mesures environnementales) sur l'autre (en général des mesures biologiques), le tout en permettant de tester la relation entre les deux tableaux.

Les méthodes d'analyse à **trois tableaux** [n° **18** à **20**] révèlent, et permettent de tester, les relations entre variables environnementales et caractéristiques des espèces (traits biologiques) au travers d'un tableau espèces-relevés.

Les méthodes d'analyse à ***k* tableaux** [n° **21** à **22**], appliquées à des tableaux de comptage répétés dans le temps, permettent de mettre en évidence l'évolution de structures spatiales en fonction du temps, ou les variations de structures temporelles dans l'espace, ou les différences de structures spatio-temporelles selon les espèces, selon le point de vue duquel on se place.

Simon POPY a soigneusement révisé une version antérieure de ce document. Nous avons intégré des informations de *adelist* (notamment de Stéphane DRAY et Jean THIOULOUSE) et fait des emprunts à Bastien MERIGOT (formation CEFE-CNRS) et à Stéphane DRAY (workshop ISEC 2014).

Remarques, critiques et corrections à : roger.prodon@cefe.cnrs.fr

Dernière version : 31 août 2014

ANALYSES A UN TABLEAU

1) L'Analyse en composantes principales (ACP) et ses variantes

En anglais : *Principal Component Analysis*

La fonction correspondant est nommé **dudi.pca()** dans le package ade4. Le terme dudi se réfère au fait que l'ACP (tout comme l'AFC) produit des « ***du****ality* ***di****agrams* » (voir http://pbil.univ-lyon1.fr/R/enseignement.html pour plus de détails).

Cette fonction prend en compte les arguments suivants (tous optionnels, sauf le nom du tableau ***df***) :

```
> dudi.pca(df, row.w = rep(1, nrow(df))/nrow(df),
  col.w = rep(1, ncol(df)), center = TRUE, scale = TRUE,
  scannf = TRUE, nf = 2)
```

avec :

- ***df*** = jeu de données (*data frame*) à *n* lignes (= individus statistiques) et *p* colonnes (=variables).
- **row.w** = le poids des lignes (optionnel ; par défaut = 1 pour tous les individus).
- **col.w** = le poids des colonnes (optionnel ; par défaut = 1 pour toutes les variables).
- **center** = si TRUE, les données sont centrées à la moyenne (par défaut). Si FALSE, les données ne sont pas centrées ; dans ce dernier cas, un échantillon n'ayant que des valeurs nulles est à l'origine. L'ordination obtenue n'est pas la même. Le choix de centrer ou non dépend des données et des buts de l'analyse.
- **scale** = si TRUE, les données sont normées (ACP normée ou *correlation matrix PCA* en anglais : c'est l'option par défaut). Si FALSE, les données ne sont pas normées (ACP non normée = *covariance matrix PCA* ou *centered PCA*).
  - ☛ L'ACP normée est obligatoire lorsque les variables sont en unités hétérogènes. Si les unités sont homogènes, on a le choix : ACP normée pour donner la même importance à toutes les variables, ACP non normée pour différencier les variables ayant de fortes ou faibles valeurs dans le tableau.
- **scannf** = si TRUE, l'histogramme des valeurs propres est représenté (il faut alors choisir le nombre d'axes quand la question est posée par le programme). Si FALSE, l'histogramme n'est pas représenté, et il faut alors préciser le nombre d'axes à garder avec l'option *nf* ci-après
- **nf** = nombre de valeurs propres à conserver dans l'analyse (par défaut = 2).

Comme rappelé en introduction, Il faut attribuer un nom d'objet à l'analyse :

```
> x <- dudi.pca(df)  ou x = dudi.pca(df)        # on dit que x est un objet de classe dudi
```

Nombres d'axes significatifs (ACP normée)

Utiliser le pourcentage d'inertie cumulée comme critère de choix du nombre d'axes n'est pas une bonne idée. La valeur du pourcentage d'inertie d'un plan factoriel n'a pas de sens en elle-même. Elle ne peut être interprétée qu'en la comparant à la courbe de décroissance des valeurs propres. Selon les cas, une même inertie sur un axe peut indiquer une structure forte intéressante, une trivialité à éliminer, ou une variabilité résiduelle à ignorer (Thioulouse, adelist).

On peut, empiriquement, ne retenir que les axes dont les valeurs propres (cf. 3 ci-après) sont supérieures à la moyenne des valeurs propres, ou rechercher une cassure sur l'histogramme des val. propres. Mais combien d'axes sont réellement significatifs (à un seuil donné) ?

La fonction suivante teste la similarité entre la configuration sur la $i^{ème}$ dimension et sur toutes les dimensions pour savoir si elle apporte réellement de l'information (d'après Dray, adelist).

```
> test = testdim(x, nrepet = 999, alpha = 0.05)
```

Le nombre d'axes significatifs est dans ***test*$nb.cor** (la significativité de chaque axe peut être lue dans ***test***, mais seuls les premiers axes donnés comme significatifs sont à retenir).

(Cf. Dray 2008. *Computational Statistics & Data Analysis* 52: 2228 – 2237)

Mais ne pas oublier la fonction descriptive de l'ACP, qui justifie l'examen des axes non retenus par le test, ne serait-ce par exemple que pour repérer des individus "hors norme" (*outliers*).

2) L'Analyse [Factorielle] des Correspondances (AFC)

En anglais : *Correspondence Analysis* (~ *Reciprocal Averaging*)

La fonction correspondant à l'AFC d'un tableau ***df*** est **dudi.coa()**, avec les arguments suivants :

> ***x*** **<- dudi.coa(*df*, scannf = TRUE, nf = 2)**

avec **scannf** et **nf** optionnels comme en ACP (autres options non détaillées), ***x*** étant le nom donné à l'analyse.

☛ L'AFC est basée sur la métrique du chi2. Ceci impose que les deux marges du tableau aient un sens. Ne pas utiliser cette analyse si les variables sont en mesurées en unités hétérogènes.

☛ L'AFC d'un tableau est identique à l'AFC de son transposé. Il est néanmoins recommandé de prendre l'habitude de garder les échantillons (individus statistiques) en lignes, par exemple pour des calculs ultérieurs prenant en compte des variables environnementales.

☛ Les éventuelles lignes ou colonnes vides du tableau sont placées à l'origine des axes. Elles ne modifient pas les résultats de l'analyse.

3) Les fichiers en sortie communs aux ACP et aux AFC

Une analyse ***x*** (une ACP ou une AFC) génère sous forme d'objet « liste » un ensemble de fichiers:

☛ On obtient la liste de tous les fichiers créés par une analyse en tapant seulement son nom : ***x***

***x*$eig** : valeurs propres (= « *eigenvalues* »; symbolisées usuellement par la lettre lamda λ)
c'est la part de variance dont rend compte un axe donné
% d'inertie associés à tous les axes : ***x*$eig/sum(*x*$eig)*100**

***x*$li** : coordonnées des lignes (normées à la racine carrée de la valeur propre de l'axe)
la colonne *x*li**Axis**1 contient les coordonnées des lignes sur le 1er axe, etc.
☛ attention au A majuscule de **Axis** (même remarque ci-dessous pour **Comp**) !

***x*$l1** : scores des lignes normés à 1 (leurs coord. sont dans les colonnes **RS1**, **RS2**, etc.)

***x*$co** : coordonnées des colonnes (normées à la racine carrée de la valeur propre de l'axe)
la colonne *x*co**Comp1** contient les coordonnées des colonnes sur le 1er axe, etc.

***x*$c1** : scores des colonnes normés à 1 (*variable loadings*). Les axes s'appellent ici **CS1**, **CS2**…

***x*$cw** : poids des colonnes-variables (par défaut 1 dans l'ACP normée)

***x*$lw** : poids des lignes (par défaut 1/*n* dans l'ACP)

***x*$tab** : jeu de données transformé (les données centrées et normées pour l'ACP standard; seulement centrées pour l'ACP non normée; la matrice transformée et diagonalisée pour l'AFC).

4) Fonctions annexes à l'ACP et l'AFC et aide à l'interprétation

a) Lignes ou colonnes supplémentaires

Il s'agit d'individus statistiques ou de colonnes supplémentaires (de masse nulle) qui n'interviennent pas dans le calcul des axes factoriels de l'ACP ou de l'AFC. On utilise les fonctions **suprow()** pour les individus-lignes et **supcol()** pour les colonnes-variables supplémentaires.

> *z* <- suprow.pca(*x*, *fichier_sup*) ligne(s) supplémentaires(s)
> *z* <- suprow.coa(*x*, *fichier_sup*) ligne(s) supplémentaires(s)
> *z* <- supcol(*x*, *fichier_sup*) colonne(s) supplémentaires(s)

avec : ***x*** = l'objet *dudi* (voir plus haut) correspondant à l'analyse sans individus ni colonnes supplémentaires
fichier_sup = le tableau des données supplémentaires.

Les coordonnées des supplémentaires sont alors dans **z$li** ou **z$co**, que l'on peut concaténer [fonction **rbind()**] avec ***x*$li** ou ***x*$co** pour des représentations ou calculs communs.

Attention : dans **suprow()** et **supcol()**, les variables (colonnes) ou les relevés (lignes), selon le cas respectif, doivent être dans le même nombre et le même ordre que dans le jeu de données de base.

b) Décomposition de l'inertie

Pour mesurer la qualité de l'analyse, c'est-à-dire la détermination des axes et la contribution des variables, utiliser la fonction **inertia.dudi()**, qui est valable pour toutes les analyses de classe *dudi*. Cette fonction prend les arguments suivants :

> inertia.dudi(*x*, row.inertia = TRUE, col.inertia = TRUE)

avec :
- ***x*** = l'objet *dudi* (l'ACP ou l'AFC) (c'est le « *x* » des exemples précédents)
- **row.inertia** = TRUE, donne contributions relatives des individus-lignes du tableau (défaut: F)
- **col.inertia** = TRUE, détaille les contributions relatives des var.-colonnes du tableau (défaut: F)

Cette fonction crée les fichiers suivants :

***x*$TOT** : inertie totale des axes (valeurs propres, par axe ou cumulées; le tableau est affiché)
***x*$col.abs** : contributions absolues (part prise par chaque variable dans la formation des axes)
***x*$col.rel** : contributions relatives (part expliquée par les axes pour chaque variable = cos2)
***x*$col.cum** : contributions relatives cumulées (diviser par 100 pour obtenir les %, comme ↑)
***x*$row.abs**, ***x*$row.rel** et ***x*$row.cum** : idem pour les lignes.

Le cumul (colonne "cum") pour tous les axes correspond à l'inertie totale (= trace)
La contribution de chaque axe à l'inertie (en %, arrondie à 2 décimales) s'obtient par :

> round((*x*$eig/sum(*x*$eig))*100,2)

c) Distances et corrélations

– Corrélations (*r*) entre colonnes d'un tableau "*df*" : **> cor(*df*)**
– Matrice des variances-covariances (pour ACP non normée) : **> cov(*df*)**
– Distance euclidiennes entre ligne d'un tableau "*df*" : **> dist(*df*)**
– Distances du chi-2 entre lignes d'un tableau : faire d'abord l'AFC ***x*** du tableau "***df***" :

***dli* <- dist.dudi(*x*)**

Pour les distances entre colonnes : ***dco* <- dist.dudi(*x*, amongrow = FALSE)**

d) ***Variances conditionnelles des lignes et colonnes dans une AFC***

Dans le cas d'un tableau *df* sites × espèces, les variances conditionnelles des sites-lignes sur un axe de l'AFC *x* (c'est-à-dire les variances sur cet axe des abscisses des espèces de la ligne) représentent une forme de la diversité spécifique du site. On peut les calculer (ici sur l'axe 1) à partir des scores normés des espèces et visualiser la figure correspondante par :

> *mvsites* <- sco.distri(*x*$c1[,1], data.frame(t(*df*))) #les variances sont dans ***mvsites*$var**

Symétriquement, les variances conditionnelles des colonnes-espèces du tableau *df* sur un axe de l'AFC *x* (c'est-à-dire les variances sur cet axe des abscisses des sites où l'espèce est présente) représentent une forme de l'amplitude d'habitat de l'espèce :

> *mvsp* <- sco.distri(*x*$l1[,1], data.frame(*df*)) #les variances sont dans ***mvsp*$var**

Dans ***mvsites*$mean** et ***mvsp*$mean**, on retrouve les ***x*$li** et ***x*$co**, respectivement.

5) Graphiques des ACP ou AFC

ade4 offre de nombreuses fonctions graphiques pour illustrer les résultats des analyses :

Graphiques communs à l'ACP et à l'AFC

> scatter(*x*) : représente directement trois des graphiques issus de l'analyse x (graphique du nuage de points des lignes et des colonnes superposés (vecteurs des variables dans le cas de l'ACP), plus le graphique des valeurs propres (que l'on peut supprimer avec l'option **posieig="none"**). Voir aussi la fonction **biplot(*x*)**. Voir aussi plus loin p.7.

> s.label(*x*$li) ou **s.label(*x*$co)** : représente les nuages des points-lignes ou des points colonnes (projection des relevés et des espèces/variables dans le plan factoriel), par défaut sur le plan 1-2 ; sinon préciser par exemple : **s.label(*x*$li, xax = 2, yax = 3)**

Pour modifier la taille des caractères des étiquettes : ajouter **clabel=** *un nombre* dans la parenthèse, après une virgule

Pour ajouter des étiquettes contenues dans une colonne "nom" du tableau: ajouter **label=nom**.

> s.traject(*x*$li, *fac*) : trace la trajectoire des points d'un nuage de sous forme de flèches. Un facteur ***fac*** permet de représenter plusieurs trajectoires sur le même graphe. L'ordre des points est par défaut celui du tableau, mais il peut aussi être défini sous forme de vecteur-index. Par exemple si on veut ordonner selon des scores croissants sur l'axe F1, remplacer *x*$li par ***x*$li[ordre,]** avec **ordre=order(*x*$li[,1])** ou tout autre vecteur donnant un ordre d'utilisation de chaque ligne.

> s.hist(*x*$li) : un "scatter" des points-lignes + plus les deux histogrammes marginaux.

> s.kde2d(*x*$li) : un "scatter" des points-lignes + les courbes d'iso-densités (kernels).

> screeplot(*x*) : histogramme des valeurs propres (inerties) associées à chaque axe.

♦ ***Représentater la contribution des points***

On peut visualiser la plus ou moins forte contribution des points à un axe (ci-dessous par ex. l'axe 2) en figurant (par ex. avec des nuances de gris) les contributions de ces points à cet axe, préalablement sauvegardées dans un fichier *i* après un **inertia.dudi** (cf. 4*b*) :

> s.value(x$li, *i*$row.abs[,2], method="greylevel", csize=0.4)

♦ ***Représenter des classes de points***

> s.class(*x*$li, *fac*, add.plot = TRUE) : à la suite d'un **s.label()**, trace des ellipses regroupant des classes de points et des étoiles joignant chaque point au barycentre. Les classes sont définies par le facteur *fac* (=identificateur de classes) indiquant pour chaque ligne, la classe à laquelle elle appartient.

La longueur des axes de l'ellipse est k fois l'écart-type des coordonnées sur les axes. Avec k=1.5, valeur par défaut, ~67% des points à l'intérieur de l'ellipse (95% si k=2.5). Noter toutefois que cette ellipse (*inertia ellipse*) est un simple résumé graphique du nuage de points; ça n'est pas une ellipse de confiance comme en analyse discriminante (voir : http://pbil.univ-lyon1.fr/R_svn/pdf/qr3.pdf)

Si on ne veut que les ellipses, ajouter **cstar = 0** dans la parenthèse; que les étoiles: **cellipse = 0**.

> s.chull(*x*$li) : idem, avec des polygones englobant 100, 75, 50 et 25% des points, tels des courbes de niveau (option optchull=1 si on ne veut que le polygone extérieur à 100%). Exemple :

> s.chull(*x*$li[,c(2,3)], *x*$classe, optchull = 1) # polygones extérieurs des classes dans F2-F3

x$li seul suppose qu'on n'a retenu que les deux 1er axes. Pour les autres, préciser leur numéros : ***x*$li[,c(*num_axe_horiz*, *,num_axe_vert*)]**, ou ***x*$li[,2:3]** avec les axes 2 et 3, etc.

On peut tester l'effet de la variable qualitative classe sur l'ordination des individus-lignes sur un axe:

> *m* <- lm(*x*liAxis$_i$ ~*fac*)

> anova(*m*)

Rem) Ici la variable classe n'intervient pas dans l'ordination; on regarde seulement *a posteriori* son effet. Si on souhaite maximiser son effet, cf. ci-après l'analyse discriminante (**n°10**).

♦ ***« Matcher » deux classes de points***

– **s.match() : p**our relier, sur un graphique, des paires de coordonnées par des flèches (par exemple pour relier des relevés faits à deux dates différentes. Il suffit de donner les deux fichiers des scores des lignes correspondant aux deux dates [voir aussi la fonction **arrows()**]. Par exemple :

> s.match(*date1*$li, *date2*$li, clabels = 0.5)

Attention : les échantillons-lignes des deux périodes doivent avoir leurs lignes et colonnes dans le même nombre et le même ordre.

Graphiques pour l'ACP

> s.corcircle(*x*$co) : [pour l'***ACP normée uniquement***] trace le cercle de corrélation, par défaut selon les 2 premières composantes.

> s.corcircle(*x*$co[, c(2,3)]) : [idem mais selon les composantes 2 et 3]

Les corrélations entre variables n'étant pas forcément bien représentées dans un plan, il est conseillé de toujours consulter la *matrice* des corrélations (voir plus haut)

> s.arrow(*x*$co) : [pour l'***ACP non normée***] la longueur des flèches représente la contribution à l'inertie de chaque variable (cf. le **$col.abs** après un **inertia.dudi()**)..

☛ Pour visualiser les corrélations (linéaires) entre les variables du tableau et une composante donnée de l'***ACP normée*** (par défaut la première) :

> score.pca(*x*, xax = *num_de_l'axe*) # les coefficients des variables sont dans ***x*$co**

Ce graphique très pédagogique permet de se convaincre qu'une composante de l'ACP est bien une combinaison linéaire (optimale) des variables. Au facteur $1/\sqrt{\lambda}$ près – où λ est la valeur

propre de l'axe (***x*$eig**) – l'abscisse d'un échantillon-ligne est la somme des valeurs de ses variables (centrées-réduites : ***x*$tab**), chacune affectée de son coefficient (***x*$co**).

Graphiques pour l'AFC

☛ ***Diagonalisation du tableau*** : deux possibilités graphiques :

Pour visualiser l'ordination des lignes et des colonnes du tableau sur un axe donné de l'AFC (avec les variances conditionnelles lignes et colonnes : cf n°4*d*) :

> score.coa(*x*, xax = *num_de_l'axe*, dotchart = TRUE) # avec **abline(v=0)** pour visualiser le 0

Ce graphique, très pédagogique lui aussi, visualise le pouvoir prédictif d'une ordination (ligne *vs*. colonne) sur l'autre, autrement dit la correspondance entre lignes et colonnes. Sur un axe, l'abscisse de chaque échantillon-ligne (***x*$li**) est la moyenne pondérée des abscisses des espèces qu'il contient (***x*$co**) – au facteur $1/\sqrt{\lambda}$ près, où λ est la valeur propre de l'axe (***x*$eig**), et inversement. C'est le *reciprocal averaging*.

☛ On peut aussi visualiser le tableau de données *df* réordonné selon un axe (ci-dessous le 1er) :

```
> L = x$li[,1]                    #  définition d'un vecteur-abscisses des lignes sur Axis1
> C = x$co[,1]                    #  définition d'un vecteur-abscisses des colonnes sur Comp1
> table.value(df[order(L),order(C)], grid=T, csize=0.5, col.labels= colnames(df[order(C)]))
```

☛ ***Représentation de la distribution d'une variable-espèce sur le plan factoriel* :**

Pour représenter la j^{ème} espèce (=colonne) du tableau *df* sur le plan 2-3 d'une AFC *x*, avec son nom, une étoile joignant les relevés où elle se trouve, et une ellipse de distribution (nombreuses options)

> s.distri(*x*$li, *df*[, j], xax=2, yax=3, label=names(*df*[i]), clabel=0.5)

On peut aussi représenter la distribution d'une variable-espèce sur le plan factoriel par des carrés proportionnels à ses effectifs dans les individus-lignes du tableau:

> s.value(x$li[, c(1,3)], df[,j]) #ex: l'espèce j dans le plan 1-3 (par défaut le plan1-2)

☛ ***La fonction scatter() en AFC***

Dans le cas de l'AFC, la fonction **scatter()** fournit une ordination simultanée des lignes et colonnes dans laquelle les lignes sont par défaut à la moyenne de leurs distribution dans les colonnes et les colonnes à la moyenne de leurs lignes, dans les deux cas au facteur $1/\sqrt{\lambda}$ près (ce qui pose des problèmes de lecture en cas de faible valeur propre). Cette fonction admet deux variantes:

> scatter(*x*, method=2) *# scaling type 1*

– Les objets-lignes-échantillons (qui ont des scores de variance unité) sont au barycentre de leurs variables. Les variables ont des scores de variance λ.

– Méthode bien adaptée à l'ordination des objets-lignes car les distances du chi-2 entre eux sont préservées : des objets proches ont des variables dans des proportions voisines.

> scatter(*x*, method=3) *# scaling type 2*

– Les variables-colonnes (scores de variance unité) sont aux barycentre des sites-lignes. Les objets-lignes ont des scores de variance λ.

– Bien adaptée à l'ordination des variables car les distances du chi-2 entre elles sont préservées : des variables proches ont des fréquences proches dans les objets-lignes. Dans le cas d'un tableau échantillons × espèces, une espèce très proche d'un point-site est vraisemblablement quasi-exclusive de ce site.

Remarques

La plupart de ces fonctions graphiques peuvent avoir de nombreux paramètres additionnels (voir l'aide). Dans **s.label()** on peut étiqueter chaque point avec un nom contenu dans un vecteur, avec l'argument : **label = *df*$*nom*** (si les noms sont dans la colonne "nom") ou: **label = rownames(*df*)**

☛ Un des inconvénients des fonctions **scatter** de ADE4 est qu'on ne peut pas changer l'échelle des axes (notamment, rapport 1:1 obligatoire). La fonction **plot()** du package de base peut donner des résultats meilleurs que **s.label()**, car elle a plus d'options graphiques.

Par exemple, on obtient le nuage des points-lignes (avec des points) sur les axes 1 et 2 en faisant :

> plot(*x*liAxis1, *x*liAxis2)

ou : **> plot(*x*$li)** # si on n'a sauvegardé que 2 axes

On peut ajouter des labels facilement avec la fonction **text()** qui doit suivre le **plot()**

> text(*x*liAxis1, *x*liAxis2+0.1, label = *df*$nom) # (noms dans colonne « nom » de ***df***)
> text(*x*liAxis1, *x*liAxis2+0.1, label = rownames(*df*) # (noms des lignes de ***df***)

ACP ou AFC ?

L'ACP

– Le nombre d'objets (obligatoirement en lignes) doit être bien supérieur (empiriquement au moins trois fois) au nombre de variables (obligatoirement en colonnes).

– La distance euclidienne mesure la ressemblance entre lignes-objets. En conséquence :

(*1*) l'ACP est très sensible aux totaux-lignes : attention si ceux-ci sont fonction de la pression d'échantillonnage,

(*2*) l'ACP est mal adaptée aux tableaux avec beaucoup de zéros (ce qui est en général le cas des tableaux espèces × échantillons) ; ils peuvent notamment générer de fausses ressemblance entre objets (par exemple rapprocher les échantillons extrêmes d'un gradient qui n'ont aucune espèce en commun : c'est l'effet "fer à cheval").

– Idéalement, les données devraient être multinormales, et les variables en relations linéaires ; mais l'ACP est robuste à ces conditions. Eviter toutefois les « *outliers* », c'est-à-dire les valeurs complètement en dehors de l'intervalle de variation des autres.

– L'ACP est également robuste à la précision des données : les variables peuvent être recodées en classes d'abondance peu nombreuses sans modifier sensiblement l'ordination.

– L'ACP est adaptée aux gradients écologiques "courts", c'est-à-dire sur lesquels les variables ont des variations ***monotones*** (seulement croissantes ou seulement décroissantes). C'est souvent le cas des gradients environnementaux (température, humidité, altitude, latitude, concentrations…)

– Si les unités des variables sont homogènes, et que l'on désire favoriser les variables ayant la plus forte variance dans l'ordination (par ex. dans le cas d'un tableau de % en ligne), l'ACP non normée est appropriée ; le graphe des variables montre alors leurs covariances, non leurs corrélations.

– Si les unités sont hétérogènes, on doit utiliser l'ACP normée; les corrélations entre variables sont visualisées par le cercle des corrélations, mais l'ordination ignore alors les différences de variances.

… / …

L'AFC

– L'AFC est symétrique (la transposition du tableau ne change pas les résultats) ; les lignes peuvent donc ne pas être plus nombreuses que les colonnes.

– Il ne doit pas y avoir de valeurs négatives dans le tableau, mais les données peuvent être en 0/1, et les zéros peuvent être nombreux. L'AFC est également robuste à la précision des données. Elle est bien adaptée aux tableaux de comptage (pour lesquels elle a été originellement conçue) mais admet aussi des variables continues avec décimales.

– La distance du chi-2 mesure la ressemblance entre lignes-objets (et entre colonnes-variables) sans tenir compte des doubles zéros.

– La pondération des valeurs du tableau par ces marges rend l'AFC insensible aux différences dans les totaux-lignes, en particulier lorsque celles-ci résultent de différences de pression d'échantillonnage (toutefois voir ci-après n°9). Mais les unités des variables doivent être homogènes (= les totaux-lignes doivent avoir un sens).

– L'AFC est bien adaptée aux gradients "longs", sur lesquels les variables présentent des variations unimodales (c'est-à-dire avec augmentation puis diminution après un optimum). Cet ***unimodal response model*** est adapté au cas où les espèces se succèdent en fonction de leurs niches le long d'un gradient écologique.

La plupart de ces considérations s'appliquent aux méthodes dérivées (ANSC, ACPVI, AFCVI, etc.)

6) L'Analyse des Correspondances multiples (AFCM)

L'AFCM (en anglais : *Multiple Correspondence Analysis* ; ne pas confondre avec l'AFM : cf. n°**22**) peut s'appliquer directement à un tableau ***df*** où chaque ligne est un individu statistique (=échantillon), et où les variables, qualitatives, sont codées en classes ou modalités dans le data frame avec des lettres ou des noms et non des chiffres. C'est plus commode car ça évite de devoir les déclarer en facteurs avec la fonction **as.factor()**. Il est inutile de coder préalablement les variables en modalités disjonctives, l'analyse le fait elle-même.

Fonction correspondante sous ade4:

> x <- dudi.acm(*df*, scannf=TRUE, nf=2) si ***x*** est le nom donné à l'analyse

avec **scannf** et **nf**, optionnels, similaires à ceux de l'ACP et de l'AFC.

- Corrélations des variables avec les axes (*correlation ratio*) dans le fichier ***x*$cr**
- Décomposition de l'inertie par **inertia.dudi(*x*)** comme pour l'ACP et l'AFC

L'analyse positionne au même endroit les individus dont le codage est identique pour toutes les variables, et à proximité ceux dont le codage est proche, ce que représentent différentes figures:

– **scatter(*x*)** : cartographie des modalités des variables dans les individus (un graphe par variable). Les ellipses correspondent aux variances-covariances de la positions des individus qui possèdent la modalité en question; elles sont centrées sur leurs moyennes.

– **scatter.dudi(*x*) :** représentation simultanée des individus et des modalités des variables.

– **s.label(x$co)** et **s.label(x$li) :** représentation des modalités des variables / des individus.

– **score.acm(*x*)** Visualisation de l'ordination des modalités sur l'axe (par défaul le 1er, sinon ajouter **xax=**2 ou 3...). Sur l'axe des abscisses, les modalités sont à la moyenne des individus où elles sont présentes, lesdits individus sont positionnés (***x*$li**) par de petits traits sur l'axe. L'axe des ordonnées correspond au scores des modalités (***x*$co**).

Il est recommandé que les variables et les classes ne soient pas trop nombreuses – ne serait-ce que pour que les figures soient lisibles – et que les effectifs ne soient pas trop déséquilibrés, car plus une variable a de modalités, plus elle contribue à l'ordination. Le vérifier sur les histogrammes obtenus par : **plot(*df*$nom_de_la_variable)**

☛ Même un tableau de variables quantitatives peut être recodé en classes et soumis à AFCM si on soupçonne que les relations entre variables s'éloignent trop du modèle linéaire sous-jacent à l'ACP.

7) L'Analyse de Hill & Smith

En anglais : *Hill and Smith's Analysis.* S'applique à un tableau ***df*** où coexistent des mesures quantitatives et des mesures qualitatives en classes ou modalités (par exemple des dimensions corporelles et des colorations, ou bien un mélange de variables environnementales quanti- et qualitatives). Il est commode de coder directement ces classes dans le tableau *df* avec des lettres ou des noms (cf. ci-dessus).

Fonction correspondante sous ade4 :

> dudi.hillsmith(*df*)

L'analyse pondère les variables catégorielles pour éviter tout effet "nombre de classes". Si le tableau ne contient que des variables quantitatives, cette analyse équivaut à une ACP. Si toutes les variables sont catégorielles, elle équivaut à une AFCM.

Comme pour les analyses précédentes, on obtient la représentation simultanée des individus, des

vecteurs-variables (quantitatives) et des vecteurs-modalités (qualitatives) avec **scatter()**, les relations des axes avec les variables continues ou catégorielles avec **score()**, et la décomposition de l'inertie avec **inertia.dudi()**.

Attention, si on fait un **s.corcircle(*x*$co)** après une Hill & Smith, on obtient des corrélations uniquement pour les variables quantitatives. Pour les qualitatives, il n'y a pas de raison que les valeurs soient comprises entre -1 et 1 (on maximise un rapport de corrélation) et les flèches peuvent dépasser le cercle.

Cf. aussi la fonction **dudi.mix()** , qui admet tout type de variables (quantitatives, facteurs, ordre)

8) L'Analyse non symétrique des Correspondances (ANSC)

En anglais : *Non symmetrical Correspondence Analysis.* Fonction correspondante sous ade4 :

> dudi.nsc()

Dans cette variante de l'AFC, toutes les colonnes-espèces ont une pondération uniforme et les échantillons-lignes sont à la moyenne (pondérée par les abondances) de leurs espèces (à un facteur de centrage près).

Les abscisses des espèces maximisent la discrimination entre échantillons, mais l'inverse n'est pas vrai. Contrairement à l'AFC standard, l'ANSC d'un tableau ne donne donc pas les mêmes résultats que l'ANSC de son transposé.

☛ L'ANSC fournit une ordination des échantillons basée essentiellement sur les espèces abondantes, les espèces rares intervenant peu dans l'ordination (contrairement à l'AFC classique). Mais attention, si les effectifs varient beaucoup d'une variable-colonne à l'autre, les résultats sont très différents de ceux de l'AFC simple et les axes respectifs non homologues.

9) L'Analyse des Correspondances décentrée (AFCD)

En anglais: *Decentred Correspondence Analysis*
Fonction correspondante sous ade4:

> dudi.dec(*df*, *effort*)

Cette variante de l'AFC permet de tenir compte d'un effort d'échantillonnage inégal selon les lignes d'un tableau ***df*** espèces(colonne)-relevés(lignes), lorsque celui-ci déséquilibre les fréquences relatives des différentes espèces et fausse la représentation de la structure du peuplement. La méthode corrige les fréquences spécifiques de chaque ligne de ***df*** par l'effort d'échantillonnage correspondant (contenu dans un fichier ***effort*** à une seule colonne et *n* lignes).

Référence
Dolédec, Chessel, Olivier 1995. L'analyse des correspondances décentrée. Application aux peuplements ichtyologiques du Haut-Rhône. *Bull. Fr. Pêche Piscic*.336: 29-40.

ANALYSES A UN TABLEAU
AVEC INDIVIDUS EN CLASSES

Nous avons vu plus haut (n°5) qu'on peut représenter des classes de points dans une ACP ou une AFC. Dans ce cas, le facteur classe n'intervient pas dans le calcul, il n'intervient qu'*a posteriori* pour l'interprétation ou pour vérifier visuellement la pertinence du découpage par rapport à l'ordination.

Lorsque les individus-échantillons-lignes du tableau sont séparés en classes *a priori* par un facteur, les analyses suivantes permettent soit de maximiser la discrimination entre ces classes (n°10 et 11), soit de faire la part des variabilités intra- et inter-classes (n°12).

10) L' ANALYSE DISCRIMINANTE (AD)

En anglais: *Discriminant Analysis*
S'applique aux tableaux de variables quantitatives justifiables d'une ACP, AFC ou ACM *x*.
L'analyse discriminante est réalisée par la fonction **discrimin()**, qui prend comme arguments :

> ***ad*** **<- discrimin(*x*, *fac*, scannf = TRUE, nf = 2)**

avec :
- ***ad*** = le nom donné à l'analyse
- ***x*** = un objet de classe dudi (une ACP effectuée préalablement, ou une AFC)
- ***fac*** = un facteur (au sens anova) définissant les classes d'individus à discriminer. Si les classes sont dénommées par un numéro, le transformer préalablement en facteur avec =**as.factor()**.
- **scannf** et **nf**, optionnels, ont même signification que pour l'ACP et l'AFC

Tables créées lors d'une analyse discriminante ***ad***:

***ad*$nf** : contient le nombre d'axes factoriels conservés dans l'analyse
***ad*$eig** : contient les valeurs propres (= 1 si la discrimination est parfaite)
***ad*$fa** : contient les poids canoniques (= coefficients des combinaisons linéaires discriminantes; cf. les *beta* d'une régression) ; mesure les contributions des variables à la discrimination
***ad*$li** : contient les scores canoniques des individus sur les axes discriminants
***ad*$va** : contient les corrélations entre les scores canoniques et les variables de départ (il est souhaitable que la représentation soit proche de *x*$fa, sous peine de faible fiabilité-robustesse)
***ad*$cp** : contient les corrélations entre les scores canoniques des indiv. et leurs scores dans l'ACP
***ad*$gc** : contient les scores des classes.

La significativité de la discrimination peut être éprouvée par un test de permutation :

> **randtest(*ad*)**

dont on obtient la figure correspondante avec :

> **plot(randtest(*ad*))**

Si la valeur repérée par une flèche est en dehors de l'histogramme, la discrimination est significative

Graphiques de l'analyse discriminante

L'ensemble des graphiques issus d'une analyse discriminante est obtenu par **plot(*ad*)**, avec « *ad* » correspondant au nom donné à l'analyse discriminante [si on veut d'autres axes que les deux premiers, le préciser; par exemple **plot(*ad*, xax = 2 , xay = 3)]**

- [*haut, gauche*] poids canoniques, c'est-à-dire contribution des variables à la discrimination
- [*milieu, gauche*] corrélations entre variables canoniques et variables de départ

- [*bas, gauche*] valeurs propres
- [*haut, droite*] variables canoniques (= coordonnées des individus) et groupes (ellipses)
- [*bas, milieu*] corrélations entre variables canoniques et axes de l'ACP (=*factor loading*)
- [*bas, droite*] moyennes des variables canoniques par classes

☛ Dans le cas où il y seulement deux groupes – donc un seul axe discriminant – la fonction **plot(*ad*)** donne autant de graphes que de variables, avec pour chacun d'eux le score de l'analyse en abscisse et la valeur de la variable en ordonnée.

Pour l'histogramme des individus d'une classe (ici la 1) sur un axe unique de l'analyse (ici le 1er) :

> **hist(*x*liDS1[fact = 1], breaks = 10, xlim=c(-3, 5))** # valeurs -3, 5, 10 par ex.

11) L'AFC DISCRIMINANTE

En anglais *Discriminant Correspondence Analysis.*
Son utilisation est la même que l'analyse discriminante classique, mais elle s'applique aux tableaux de variables qualitatives justifiables d'une AFC préalable et dont les lignes sont réparties en classes (préférer un code en lettre ou alphanumérique) grâce à un facteur contenu dans un fichier à part.
Fonction correspondante sous ade4 :

> **discrimin(*x*, *fac*, scannf = TRUE, nf = 2)**

Ou ***x*** est le nom de l'AFC simple du tableau
fac est un facteur au sens de l'anova.

On obtient une figure synthétique avec : **plot(*ad*)**, où *ad* est le nom donné à l'analyse.

☛ Cette analyse n'est pas recommandée pour les tableaux floro-faunistiques comportant de nombreuses colonnes. Dans ce cas, il vaut mieux utiliser l'analyse inter-classes ci-après.

12) LES ANALYSES INTER- et INTRA-CLASSES (ACP ET AFC)

Après avoir soumis l'ensemble du tableau à une analyse ***x*** de classe *dudi* (ACP, AFC, AFCNS, AFCM, Hill & Smith…), on peut toujours représenter les centres de gravité de chaque groupe sur un axe et le lien entre un échantillon et son groupe d'appartenance avec la fonction :

> **s.class(*x*$li, *fac*, xax = 1, yax = 2, cellipse = 0)**

Mais les analyses inter- et intra-classes permettent de séparer dans l'analyse ***x*** les parts respectives des variabilités intra- et inter-classes :

– **intra-classes** :

> ***awithin* <- wca(*x*, *fac*)** # ***fac*** doit être un facteur, non un data frame
> **plot(*awithin*)** # figure synthétique avec notamment :

- [*haut, droite*] coordonnées des individus-lignes et groupes (ellipses) (***awithin* $li**) ; on peut en lire les étiquettes avec : **s.label(*awithin*$li)**
- [*milieu, gauche*] coordonnées des variables-colonnes (**$co**)
- [*bas, milieu*] inertie projetée sur chaque axe (**$as)**

– **inter-classes** :

> ***abetween* <- bca(*x*, *fac*)**

> plot(*abetween*)

- [*haut, droite*] coordonnées des individus-lignes et groupes (ellipses) (***abetween* $ls**)
- [*milieu, gauche*] coordonnées des variables-colonnes (**$co**)
- [*bas, droite*] coordonnées des classes (**$li**)
- [*bas, milieu*] inertie projetée sur chaque axe (**$as)**

– Dans l'analyse ***intra-classe***, tous les centres des classes sont placés à l'origine des axes (ce qui élimine l'effet classe) et les individus sont représentés avec une variance maximale autour de l'origine. On obtient une typologie générale des lignes, c'est-à-dire qui ignore l'effet classe.

– L'analyse ***inter-classe*** fait ressortir l'effet-classe, c'est-à-dire met l'accent sur la différence entre les groupes, et permet d'identifier les variables responsables de cette différence. C'est un cas particulier d'ACPVI (au sens large; cf. n°14 et 15) avec ici une seule variable instrumentale, qui est le facteur ***fac***. Cela équivaut à faire une ACP sur les moyennes par groupe et par variable.

Un test de permutation permet d'évaluer la significativité de l'effet-classe :

> randtest (abetween, nrepet = 999) # avec 999 répétitions

La trace (somme des valeurs propres) des analyses *within* et *between* est égale à la trace de l'analyse normale préalable (AFC, ACP ou autre). Les parts de variance du tableau général qui sont due aux effets intra-groupe et inter-groupe sont dans les fichiers ***within*$ratio** et ***between*$ratio**, respectivement (leur somme est 1).

☛ Contrairement à une discriminante, l'analyse inter-classe n'optimise pas la discrimination des classes. La différence avec la discriminante se situe au niveau du critère optimisé (la variance inter avec **bca()**, alors que c'est le rapport variance inter/totale en discriminante).

☛ L'analyse inter-intra classe peut être considérée comme un exemple d'analyse à K-tableaux (= les K classes) (cf. n°20) où les classes peuvent avoir des nombres inégaux d'individus. Les applications potentielles en écologie sont très nombreuses.

Références

Thorpe R.S. 1988. Multiple group principal component analysis and population differentiation. *J. Zool. London* 216: 37-40.

Cazes, P., Chessel, D., and Dolédec, S. 1988. L'analyse des correspondances internes d'un tableau partitionné : son usage en hydrobiologie. *Revue de Statistique Appliquée* 36:39–54.

ANALYSES A DEUX TABLEAUX

En écologie, il est fréquent que l'un des tableaux concerne un système biologique (Y), et l'autre son environnement (X). On cherche alors à expliquer Y par X (analyses avec variables instrumentales), ou bien à mettre simplement en évidence la structure commune à Y et X (Analyse de Coinertie).

☛ Les analyses précédentes (n° 10 à 12) peuvent être vues comme un cas particulier où le deuxième tableau est réduit à une seule variable catégorielle.

Une référence introductive :

Prodon R., Lebreton J.D. 1994. Analyses multivariées des tableaux espèces-milieu: structure et interprétation écologique. *Vie et Milieu* 44 : 69-91.

13) L' AFCVI (OU CCA)

L'AFC avec Variables Instrumentales (AFCVI), ou AFC sous contrainte linéaire, est appelée *Canonical Correspondence Analysis* (CCA) par les anglophones. Dans cette analyse, les coordonnées des individus-lignes dans l'AFC du tableau de contingence *Y* (généralement un tableau espèces × sites) sont contraintes d'être fonction linéaire des variables-colonnes du tableau environnemental *X*. Les $1^{ères}$ sont donc dépendantes, les 2^{ndes} indépendantes.

L'AFCVI peut être vue comme une AFC des prédictions du tableau Y obtenues par régressions multiples sur les variables du tableau *X* (ce qui suppose des relations au moins approximativement linéaires entre ces dernières et les valeurs du tableau *Y*).

L'utilisation de l'AFCVI en écologie se justifie par son lien étroit avec la théorie de la niche et l'approximation des réponses unimodales des espèces aux gradients environnementaux. Elle permet de trouver une combinaison des variables environnementales (gradient) qui maximise la séparation des niches des espèces : une espèce est positionnée à la moyenne des conditions environnementales où elle est présente, et on maximise la séparation de ces moyennes (Dray, adelist).

La fonction correspondante (qui génère un objet de classe dudi) est :

> **cca(*sitsp*, *sitvar*, scannf = TRUE, nf = 2)**

avec :

- ***sitsp*** = tableau de contingence (data frame), généralement *n* sites contenant *m* espèces
- ***sitvar*** = tableau des variables environnementales : *n* sites et *p* variables
- **scannf** et **nf**, optionnels, ont les mêmes significations que pour l'ACP ou l'AFC

Tables créées lors de l'AFCVI (avec ***x*** = nom donné à l'analyse) :

***x*$eig** : valeurs propres de l'analyse
***x*$lw** : poids des lignes
***x*$cw** : poids des colonnes (selon AFC)
***x*$Y** : matrice des variables dépendantes (des espèces)
***x*$X** : matrice des variables explicatives
***x*$tab** : données modifiées (variables projetées)
***x*$c1** : scores des colonnes (des variables dépendantes, les espèces, de variance 1, d'où le 1)
***x*$co** : scores prédit des colonnes (variables dépendantes, avec une norme lambda)
***x*$as** : axes principaux
***x*$ls** : scores des lignes-échantillons (à la moyenne pondérée de leurs espèces)
***x*$li** : scores prédit des lignes-échantillons

***x*$fa** : poids canoniques des variables explicatives normalisées dans la constitution des coordonnées de variance 1
***x*$l1** : scores canoniques des lignes (combinaison linéaire des variables explicatives, norme 1)
***x*$cor** : corrélations des variables explicatives avec les axes de l'AFCVI.

☛ S'il n'y a qu'un seul facteur dans le tableau ***var***, la fonction est équivalent au **bca()** (cf. n°12).

Graphiques de l'AFCVI

Un ensemble de graphiques issus de l'AFCVI est donné par **plot(*x*)**, où *x* est le nom donné à l'AFCVI (on peut choisir ses axes avec **xax=** et **yax=**) :

– [*haut, gauche*] contributions (*loading* ; ***x*$fa**) des variables explicatives aux axes de l'analyse. En cas de colinéarité des variables, ces coefficients sont instables (comme dans une régression multiple) et il vaut mieux privilégier les corrélations ci-après pour l'interprétation écologique.

– [*milieu, gauche*] corrélations ***x*$cor** entre les axes et les variables explicatives.

– [*bas, gauche*] | inertie projetée sur chaque axe (***x*$as**)

– [*haut, droite*] scores ***x*$ls** des individus-lignes (par *averaging* des espèces présentes; pointes des flèches) et leurs prédictions ***x*$li** (par régression sur les variables de milieu; départs des flèches). Les flèches sont d'autant plus courtes que la composition spécifique du relevé est bien prédite à partir des variables environnementales. De longues flèches signalent des échantillons dont la composition en espèces n'obéit pas au modèle général espèce-milieu révélé par l'analyse.

– [*bas, milieu*] les scores des variables-espèces (***x*$co**) sur les axes

– (*bas, droite*] les valeurs propres (***x*$eig**) des axes.

Chacun des trois principaux graphiques que l'on donne généralement à la suite d'une AFCVI (qu'on appelle parfois le « **triplot** ») peut aussi être obtenus individuellement :

- le graphique du nuage de points des *n* échantillons est obtenu avec **s.label(*x*$li)** ou **(*x*$ls)**
- le graphique du nuage de points des *m* espèces est obtenu avec **s.label(*x*$co)**
- le cercle de corrélation montrant les corrélations des *p* variables de milieux avec les axes F1 et F2 de l'analyse est obtenu avec **s.corcircle(*x*$cor)**

Les *écarts* entre les valeurs initiales et les valeurs prédites peuvent aussi être représentés (par une flèche) avec la fonction **s.match()** [voir aussi plus loin], il suffit de préciser dans la parenthèse les deux jeux de données contenant les valeurs initiales **(*x*$ls)** et prédites **(*x*$li)**. Plus courtes les flèches, meilleure la prédiction.

Interprétation d'une AFCVI

– La commande **summary(*x*)** donne l'inertie totale, les valeurs propres, et l'inertie projetée sur chaque axe.

– **La part de variabilité** du tableau de contingence (floristique ou faunistique) **expliquée** par les variables environnementales (= *Multivariate Correlation Ratio*) se mesure par rapport à l'AFC non contrainte :

```
> 100 * sum(cca$eig) / sum(coa$eig)        # où cca est l'AFCVI et coa l'AFC simple.
```

☛ Même si le % de variabilité expliquée peut paraître faible dans certains cas, l'AFCVI est capable d'extraire un message clair de données très "bruîtées" en ignorant la part de variabilité du tableau non liée aux variables environnementales (par exemple celle dues aux aléas de la détectabilité).

– La significativité de cette variabilité expliquée peut être testée par un test de permutation :

```
> rd <- randtest(x)                        # figure avec plot(rd)
```

Dans le **plot(*rd*)**, l'axe des abscisses correspond au coefficient RV (équivalent d'un R^2) qui varie entre 0 (pas de corrélation entre les tableaux *X* et *Y*) et 1 (prédiction parfaire de *Y* à partir de *X*).

Dans la pratique, le test est presque toujours significatif, les variables environnementales expliquant une part significative de la variance du tableau floro-faunistique…

☛ En raison de la régression sous-jacente, le nombre de relevés doit être bien supérieur (>10 fois) au nombre de variables explicatives pour que l'AFCVI ait un sens. Sinon, on tend en fait vers une AFC simple (sans contrainte) du tableau à expliquer. Lorsque le nombre d'échantillons est faible par rapport au nombre de variables, il faut préférer l'analyse de co-inertie (cf. n°17).

14) L' ACPVI (OU ANALYSE DE REDONDANCE)

L'Analyse en Composantes Principales avec Variables Instrumentales (ACPVI), ou ACP sous contrainte linéaire, appelée *Redundancy Analysis* (RDA) par les anglophones (ici *redundancy* signifie variance expliquée). Dans cette analyse, le tableau à expliquer (concernant ici un système biologique) n'est pas un tableau de contingence mais un tableau de variables quantitatives justifiable d'une ACP.

Dans cette analyse, les coordonnées des individus-lignes dans cette ACP sont contraintes d'être fonction linéaire des variables-colonnes du deuxième tableau (en général des variables environnementales). L'ACPVI peut être vue comme une ACP des prédictions du tableau Y obtenues par régressions multiples sur les variables du tableau X (ce qui suppose des relations au moins approximativement linéaires entre ces dernières et les valeurs du tableau Y).

Fonction correspondante sous ade4 (sans les commandes optionnelles):

> **pcaiv(*x*, *var*)**

où ***x*** est le nom de l'ACP du tableau à expliquer (c'est donc un objet de classe *dudi*)

var est le nom du tableau explicatif

On obtient une figure synthétique avec : **plot(*z*)**, si ***z*** est le nom donné à l'ACPVI.

Pour la légende des figures et les fichiers produits, cf. l'ACPVI. Mais ici, les scores des lignes-échantillons (***z*$ls**, pointes des flèches de la figure en haut à droite) ne sont pas calculés par *averaging*, mais d'après les coefficients (*loadings*) des variables du tableau Y (***z*$c1**, figure en bas). Les scores prédits (***z*$li**, bases des flèches) sont, comme dans l'AFCVI, calculés d'après les coefficients (*loadings*) des variables environnementales du tableau X (***z*$fa**, en haut à gauche)

Comme dans l'AFCVI, on peut calculer le % de variance du tableau Y expliqué par les variables environnementales de X:

> **100 * sum(*z*$eig) / sum(*x*$eig)**

☛ S'il n'y a qu'un seul facteur dans le tableau ***var***, la fonction est équivalent au **bca()** (cf. n°12).

15) ELIMINATION D'EFFETS : ACPVI OU AFCVI ORTHOGONALES

Il ne s'agit pas d'une analyse particulière, mais de l'utilisation des propriétés des analyses sous contrainte ci-dessus. Ici, on souhaite s'affranchir de l'action de (en anglais: *to factor out*) une ou plusieurs variables environnementales du tableau X. Autrement dit, on restreint l'analyse au résidu du tableau à expliquer une fois tenu compte de cette(ces) variable(s).

Fonction correspondante sous ade4 (sans les commandes optionnelles):

> pcaivortho(*x*, *var*)

où ***x*** est le nom de l'ACP, AFC, ACM, ANSC (ou tout autre analyse de classe *dudi*) du tableau à expliquer.

var est le nom de la variable ou du tableau de variables dont on veut éliminer l'effet.

On obtient une figure synthétique avec : **scatter(*z*)**, si ***z*** est le nom donné à l'analyse.

☛ S'il n'y a qu'un seul facteur dans le tableau ***var***, la fonction est équivalent au **wca()** (cf. n°12).

16) SEPARATION DES NICHES : LA METHODE OMI

Cette méthode, assez proche de l'AFCVI, permet une meilleure hiérarchisation des facteurs écologiques, une meilleure séparation des niches des espèces et une meilleure mesure de leur amplitude d'habitat en donnant un même poids à tous les échantillons, qu'ils soient riches ou pauvres en espèces et individus [répondant ainsi aux critiques de Legendre & Gallaguer (*Oecologia* 2001) sur la CCA]. Cette méthode est donc particulièrement adaptée aux cas où la pression d'échantillonnage n'est pas contrôlée. De plus, l'OMI ne fait pas d'hypothèse sur la forme (unimodale ou linéaire) des courbes de réponse des espèces, et est donc moins gênée par les troncatures aux extrémités du gradient. Le % de variance floro-faunistique expliquée par les variables environnementales s'avère plus grand que dans l'AFCVI ou ACPVI.

Un indice de marginalité ou de spécialisation (***Outlying Mean Index***) mesure la distance entre l'habitat moyen d'une espèce donnée et les conditions d'habitat moyennes de l'aire étudiée (correspondant à la distribution d'une espèce hypothétique uniformément distribuée en toutes conditions). L'analyse place les espèces sur les gradients d'habitat de façon à maximiser leurs OMIs.

Fonction correspondante sous ade4 (sans les commandes optionnelles):

> niche(*x*, *df*) # génère un objet de classe *dudi*

où ***x*** est le nom de l'ACP, AFC (ou une autre analyse de classe *dudi*) du tableau des variables environnementales, et ***df*** est le nom du tableau de contingence sites × espèces.

On obtient une figure synthétique avec : **plot(*z*)**, si ***z*** est le nom donné à l'analyse.
On obtient le biplot des espèces et des variables (en ajustant les limites du graphe, ici±3) par :

> s.arrow(*z*$c1, xlim=c(-3,3))

> s.label(*z*$li, add.plot=T)

Les niches (moyennes, amplitudes) des espèces sur un axe de l'analyse (ici le 1er) peuvent être visualisées par:

> sco.distri(*z*$ls[,1], *df*)

Un test de permutation : **rtest(*z*)** permet de savoir si la distribution de chaque espèce, ou de l'ensemble, diffère d'une distribution au hasard indépendamment des variables.

Référence : Dolédec, S., Chessel D., Gimaret-Carpentier C. 2000. Niche separation in community analysis : a new method. *Ecology* 81: 2914-2927.

17) L' ANALYSE DE COINERTIE

L'analyse de coinertie met en évidence la covariance entre deux tableaux dont les lignes, en nombre égal, correspondent aux mêmes individus statistiques. Contrairement à l'AFCVI et l'ACPVI, ces deux tableaux (ici *Y* et *X*) jouent un rôle symétrique : il n'y a pas de tableau à expliquer ni de tableau explicatif. L'analyse révèle seulement une éventuelle co-structure entre les deux tableaux en cherchant une ordination commune des deux tableaux qui soit de corrélation maximale avec les ordinations de chacun des deux tableaux séparés : les axes de co-inertie maximisent la covariance entre les coordonnées des projections des lignes de chacun des tableaux.

Il faut préalablement soumettre chacun des deux tableaux *Y* et *X* à une ordination. ☛ Il est indispensable que les lignes aient le même poids dans ces deux ordinations. En conséquence :

– si on associe deux ACP, pas de problème car la pondération des lignes y est (par défaut) uniforme

– si on associe l'AFC de *Y* et l'ACP de *X*, il faut (i) effectuer l'AFC de *Y*, (ii) recueillir le fichier du poids des lignes ***dudiY*$lw**, (iii) pondérer les lignes de l'ACP de *X* à l'aide de ce fichier:

> **dudiX <- dudi.pca(*X*, row.w=*dudiY*$lw,…)**

Dans tous les cas, la fonction correspondant à l'analyse de coinertie est (sans les options) :

> ***z* <- coinertia(*dudiY*, *dudiX*)**

où : ***dudiY*** est le nom de l'analyse du tableau *Y (les espèces, en général)*

dudiX est le nom de l'analyse du tableau *X (les variables environnementales)*

Outre plusieurs fichiers (dont on obtient la liste en tapant seulement le nom de l'analyse, ici ***z***), on obtient une figure synthétique avec **plot(*z*)** :

– [*haut, gauche*] corrélation des axes de l'analyse de *X* avec les axes de la coinertie (fichier **z$aX**).

– [*milieu, gauche*] corrélation des axes de l'analyse de *Y* avec les axes de la coinertie (**z$aY**).

Ces deux figures permettent de juger de la corrélation ± grande des axes de la coinertie avec les axes des analyses séparées (espèces *vs* variables) non contraintes.

– [*bas, gauche*] valeurs propres des axes de la coinertie (**z$eig**)

– [*haut, droite*] nuage des individus-lignes. Le départ des flèches est la position des individus décrites d'après leur composition en espèces (**z$mX**), la pointe des flèches leur position d'après leurs variables environnementales (**z$mY**). Plus courtes sont les flèches, meilleure est l'accord entre les deux ordinations.

– [*bas, milieu*] coefficients (*loadings*) des variables environnementales (**z$l1**).

– [*bas, droite*] coefficients des variables-espèces (**z$c1**).

☛ L'hypothèse de l'existence d'une costructure significative entre les deux tableaux peut être testée. Si chacun des deux tableaux est soumis à une ACP, et en utilisant les fichiers **–$tab** produits par ces deux analyses, on peut faire **RV.rtest(*dudiY*$tab, *dudiX*$tab)**. Plus approprié:

> ***rd* <- randtest(*z*)**

Dans le **plot(*rd*)**, l'axe des x correspond au coefficient RV qui varie entre 0 (aucune corrélation entre les structures des deux tableaux) et 1 (structures des deux tableaux parfaitement corrélées).

Rem) Dans le cas d'un couplage AFC-ACP, il faut il fixer le tableau d'AFC qui induit les poids des lignes et permuter uniquement le deuxième tableau : **randtest(*z*, fixed = 2)**.

☛ On peut obtenir les contributions (absolues et relatives) des variables, avec la fonction

> **inertia.dudi(*z*, row = T, col = T)**

La fonction **summary(*z*)** détaille les valeurs de covariance entre les deux tableaux, variances de l'un et l'autre, corrélations, etc.

☛ Basée sur la maximisation de la variance, la coinertie a le grand avantage pratique sur l'ACPVI et l'AFCVI d'être plus robuste, c'est-à-dire de supporter un plus faible nombre d'individus (échantillons) relativement au nombre de variables. Elle supporte aussi la colinéarité entre variables. Elle admet aussi des classes (voir la fonction **bca.coinertia()**)…

☛ L'analyse de coinertie est très souple et permet de coupler plusieurs types d'analyses (ACP, AFC, AFCM, Hille & Smith). Mais s'assurer que le poids des lignes est le même dans les deux analyses à coupler.

Références

Dolédec, S., Chessel D.. 1994. Co-inertia analysis : an alternative method for studying species-environment relationships. *Freshwater Biology* 31: 277-294.

Dray S., Chessel D., Thioulouse J. 2003. Co-inertia analysis and the linking of two ecological data tables. *Ecology* 84: 3078-3089.

ANALYSES A 3 TABLEAUX
ANALYSES RLQ

Ces méthodes d'analyse s'appliquent aux jeux de données constitués de trois tableaux :

dfL un *data frame* contenant les abondances de *p* espèces (colonnes) dans *n* sites (lignes)

dfR sun *data frame* avec les mesures de *m* variables environ. (colonnes) dans ces *n* sites (lignes)

dfQ un *data frame* avec *s* traits spécifiques (☛ en colonnes) pour ces *p* espèces (☛ en lignes)

Le but de ces méthodes est de mettre en évidence les liens entre les *s* traits spécifiques et les *m* variables environnementales au travers du tableau floro-faunistique.

☛ Les tableaux des variables environnementales et des traits spécifiques peuvent être en classes (ce qui est pertinent si on soupçonne que les relations entre eux ne sont pas linéaires).

Ces méthodes sont de deux sortes:

– on peut s'intéresser aux "traits moyens" pour l'ensemble de la communauté, laquelle est considérée comme l'unité d'observation. Ces traits moyens reflètent les conditions environnementales mesurées, et pourront donc changer avec elles (n°**18** ci-après).

– on peut chercher plutôt à mettre en évidence les combinaisons de traits spécifiques – et les dissemblances fonctionnelles entre espèces – qui coexistent dans les conditions mesurées. L'unité d'observation est ici l'espèce (n°**19** et **20** ci-après).

☛ Référence générale et revue des toutes les méthodes, plus couplages ordinations-classifications (avec programmes et fichiers en annexe):

Kleyer M., Dray S., de Bello F., Leps J., Pakeman R.J., Strauss B., Thuiller W, Lavorel S. 2012. Assessing species and community functional responses to environmental gradients: which multivariate methods? *Journal of Vegetation Science* 23: 805–821.

18) ACPVI DES MOYENNES PONDEREES DES TRAITS

Dans cette ACPVI-MPC (où MPC = Moyennes Pondérées des traits pour la Communauté ; en anglais CWM-RDA: *redundancy analysis of community weighted mean traits*) on n'individualise pas les espèces. Les résultats mesurent et illustrent les relations entre les traits moyens de la communauté et les variables d'environnement. Les étapes de l'analyse sont les suivantes :

multiplication matricielle du tableau des abondances des espèces "***dfL***" et de celui des traits spécifiques "***dfQ***" (→ moyennes pondérées des traits de toutes les espèces, par échantillon).

```
> tab.mpc <- prop.table(as.matrix(dfL),1) %*% as.matrix(scale(dfQ))
```

ACPVI (= RDA) du tableau résultant sur les variables environnementales du tableau "***dfR***" :

```
> acp. mpc <- dudi.pca(tab. mpc,  scannf=FALSE)
> rda. mpc <- pcaiv(acp. mpc,  dfR,  scannf=FALSE)
```

pourcentage de variation des traits expliqué par les variables environnementales :

```
> sum(rda. mpc $eig) / sum(acp. mpc $eig) * 100
```

figure (biplot) des relations entre les traits moyens et les variables environnementales :

```
> s.arrow(rda. mpc $c1,  xlim=c(-1,1),  boxes = FALSE)
> s.label(rda. mpc $cor[ , ],  add.plot=T,  clab=1.5)
```

La liste de tous les fichiers produits est dans ***rda.mpc*** (qui est un objet de classe dudi).

19) LA METHODE DU « 4EME COIN »

On cherche ici à mettre en évidence les liens entre les *s* traits spécifiques et les *m* variables environnementales en calculant un quatrième tableau contenant les corrélations entre ces *s* traits spécifiques et *m* variables environnementales. Leurs significativités sont calculées grâce à divers tests de permutation.

☛ Il peut y avoir une seule variable environnementale, dont on recherche les liens avec tel trait.

La commande est simple (☛ attention à bien respecter l'ordre des trois tableaux) :

> ***x* <- fourthcorner(*dfR*, *dfL*, *dfQ*, modeltype = 1, nrepet = 999)**

Consultation et visualisation des résultats :

x	donne toutes les corrélations entre traits et variables, avec leurs significativités
summary(*x*)	idem, sous forme d'un tableau
plot(*x*)	tableau des corrélations entre les *s* traits (lignes) et *m* variables (colonnes), avec significativités représentées par des grisés (foncé : positivement significatif ; moyen : négativement significatif ; blanc : non significatif)

Ce qui est moins simple, ce sont les différentes procédures de test de significativité (l'argument "modeltype" de la commande ci-dessus), car elles correspondent à différentes hypothèses nulles. Le test direct des corrélations entre les *df R* et *Q* n'est pas possible, d'où une batterie de tests alternatifs qui donnent des résultats différents.

Ces tests sont les suivants, avec l'interprétation de leur significativité (en cas de rejet de H_0) :

modeltype=1 il existe une relation entre R et L et/ou entre L et Q.
R et Q sont supposés "fixes" (ce sont des données biblio, ou des mesures préliminaires), L est aléatoire (résultat de l'échantillonnage)
[La permutation à l'intérieur de chaque colonne de L détruit le lien entre L et R et entre L et Q]

modeltype=2 il existe une relation entre R et L
R et Q sont supposés "fixes"
[La permutation des lignes dans L détruit le lien entre L et R, mais garde le lien entre L et Q]

modeltype=3 il existe une relation entre R et L et/ou entre L et Q
[La permutation à l'intérieur de chaque ligne de L détruit le lien entre L et R et entre L et Q]
Il semble que ça soit l'option par défaut si on ne précise pas le modèle dans la commande

modeltype=4 il existe une relation entre L et Q
[La permutation des colonnes de L détruit le lien entre L et Q, mais garde le lien entre L et R]

modeltype=5 il existe une relation entre R et L et/ou entre L et Q
[La permutation des colonnes et des lignes de L détruit le lien entre L et R et entre L et Q]

En fait, pour tester les relations entre R-L-Q, il est recommandé de tester les modèles 2 et 4 à la fois, en effectuant d'abord les deux analyses correspondantes *x2* et *x4*, puis en les combinant (d'une façon conservative, ce sont alors les probabilités les plus élevées qui sont retenues) :

> ***xcomb* <- combine.4thcorner(*x2*,*x4*)**
> **plot(*xcomb*)**

Pour le choix des différent modèles et la puissance des tests correspondants, cf. Dray et Legendre (*Ecology* 89: 3400-3412, 2008). Pour augmenter la chance de détecter des relations traits-environnement, il est recommandé d'augmenter le nombre d'échantillons et d'espèces considérées. Voir aussi : Ter Braak, C., Cormont, A., et Dray, S. 2012. Improved testing of species traits-environment relationships in the fourth corner problem. *Ecology*, 93:1525-1526

☛ La fonction **fourthcorner2()**, dont la syntaxe est la même que fourthcorner(), mesure les liaisons entre variables avec des coefficients différents (voir la documentation R), et se rapproche de l'analyse RLQ. En pratique, ses tests (analogues à ci-dessus) semblent plus conservatifs, c'est-à-dire mettent en évidence un plus petit nombre de corrélations significatives.

20) L'ANALYSE RLQ

S'applique aussi aux jeux de données constitués de trois tableaux variables-espèces-traits. Comme l'analyse de coinertie, l'analyse RLQ lie les analyses multivariées des trois tableaux en révélant leur co-structure. Le tableau espèces-sites L étant soumis à AFC, l'analyse RLQ cherche des combinaisons linéaires de traits Q et de variables environnementales R maximisant la coinertie. Cette méthode maximise donc les covariances alors que la méthode du 4ème coin considère les corrélations.

Cette analyse, réellement multivariée, met en évidence d'une façon plus synthétique et plus graphique que le "4ème coin" les liaisons entre variables (R) et traits (Q), mais moins analytique (pas de tests variable par variable et trait par trait). Elle peut permettre une comparaison, basée sur des traits, de systèmes ayant des cortèges floro-faunistiques différents.

Exemple avec des variables environnementales et des traits quantitatifs :

– le *data frame* **L** des *p* espèces (colonnes) dans *n* sites (lignes) est soumis à AFC :

> ***afc*** **<- dudi.coa(*L*)**

– le *data frame* **R** des *m* variables environ. (colonnes) dans *n* sites (lignes) est soumis à ACP :

> ***acpvar*** **<- dudi.pca(*R*, row.w=*afc*$lw)**

– le *data frame* **Q** des *s* traits spécifiques (colonnes) pour les *p* espèces (lignes) est soumis à ACP :

> ***acptrait*** **<- dudi.pca(*Q*, row.w=*afc*$cw)**

La syntaxe est alors :

> ***x*** **= rlq(*acpvar*, *afc*, *acptrait*)**

On obtient la liste des fichiers produits en tapant le nom ***x***, et une figure synthétique par : **plot(*x*)**

[*en haut*] ordination des relevés (à gauche) et des espèces (à droite) ; les coordonnées correspondantes sont dans les fichiers ***x*$lR** et ***x*$lQ**, respectivement.

[*en bas*] les contributions des variables (à gauche) et des traits (à droite); les coordonnées correspondantes sont dans les fichiers ***x*$l1** et ***x*$c1**, respectivement. ☛ Ces graphiques visualisent de façon très synthétique les relations entre R et Q *via* L.

Le biplot synthétisant les corrélations entre traits et variables peut être obtenu par:

> **s.arrow(*x*$c1, xlim=c(-1,1), boxes =FALSE)**
> **s.label(*x*$li, add.plot=T)**

Les valeurs propres et inerties associées aux différents axes et tableaux sont données par :

> **summary(*x*)**

La significativité de la liaison globale entre R et Q peut être testée par un test de permutation :

> **randtest.rlq(*x*)**

☛ On peut soumettre les tableaux R et Q à différentes méthodes d'ordination, par ex. deux AFCM lorsque les variables environnementales et les traits sont en classes, ou des analyses de Hill & Smith, etc., mais les poids des lignes dans ces analyses doivent toujours être ceux de l'analyse de L (cf. n° 17).

☛ Après une analyse rlq, on peut tester les effets inter- et intra-classe (par exemple un éventuel effet-année) avec l'analyse inter-intra classe (cf. n°12). Dans la commande **wca()**, ***x*** est alors le nom de l'analyse rlq.

Référence

Dolédec, Chessel, Ter Braak, Champely 1996. Matching species traits to environmental variables: a new three-table ordination method. *Environmental & Ecological Statistics* 3: 143-166.

ANALYSES A K TABLEAUX

Dans l'analyse des données multi-tableaux, la principale difficulté est la maîtrise des objectifs poursuivis. Une autre est le choix des points de vue et des espaces de représentation. Il existe en effet plusieurs méthodes, avec chacune de nombreuses sorties graphiques, permettant l'analyse simultanée de plusieurs tableaux de données (*k* tableaux ou *k* fois deux tableaux). Quelques références pour s'orienter :

Blanc, L. 2000. *Données spatio-temporelles en écologie et analyses multitableaux : examen d'une relation*. Thèse Université Lyon-1. [http://pbil.univ-lyon1.fr/R/pdf/these_lb.pdf]

Dray, A., Dufour B., Chessel D. 2007. The ade4 Package— II: Two-table and K-table Methods. *R News* 7: 47-52.

Thioulouse, J., Chessel, D. 1987. Les analyses multi-tableaux en écologie factorielle. I De la typologie d'état à la typologie de fonctionnement par l'analyse triadique. *Acta Œcologica, Œcologia Generalis* 8: 463-480.

Thioulouse, J. 2011. Simultaneous analysis of a sequence of paired ecological tables : a comparaison of several methods. *The Annals of Applied Statistics* 5: 2300-2325.

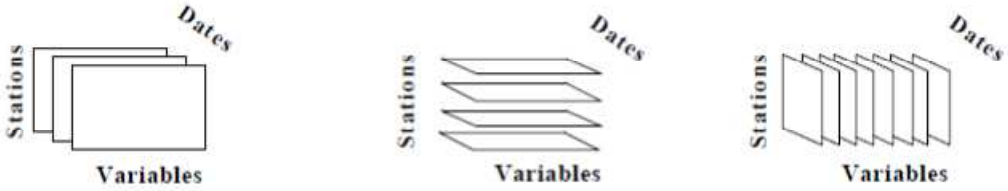
En rappelant que les analyses inter-intra peuvent rentrer dans ce cadre (cf. ci-dessus n°**12**), on donne ci-après un bref aperçu du *modus operandi* de deux méthodes à *k* fois un tableau.

21) L'AFC DE FOUCART

Cette méthode s'applique à ce qu'on appelle un "cube" de données. Ce "cube" (ses 3 dimensions ne sont en général pas égales) résulte typiquement de l'échantillonnage floro-faunistique répété d'un réseau de stations. Cet échantillonnage génère un tableau à 3 dimensions :

espèces × stations × dates.

Il ya 3 points de vue possibles pour décomposer la variabilité de ce cube en *k* sous-tableaux. Comme pour les autres méthodes à *k* tableaux, cette décomposition et la présentation des données qui en résulte est une étape fondamentale qui oriente toute l'analyse et son interprétation.



A B C

Rentrer indépendamment (**read.table**) les *k* data frames: *df1*, *df2*, *df3* correspondant chacun à un des sous-tableaux selon le découpage choisi (A, B ou C), puis définir un vecteur liste et le remplir :

> *cube* <- vector("list")
> *cube*[[1]] <- *df1*
> *cube*[[2]] <- *df2*
…

☛ Attention à ce que les noms des lignes et des colonnes soient bien les mêmes sur tous les *df*.

☛ La fonction **split(*df*, *fac*)** découpe un tableau *df* et génère une liste dans laquelle chaque élément correspond à un sous tableau associé à une classe du facteur *fac*.

La fonction correspondant à l'AFC de Foucart est la suivante (sans les options) :

> *x* <- foucart(*cube*)

La fonction **plot(*x*)** donne des ordinations différentes selon le type de décomposition du cube :

A : si chacun des *df* correspond à une date (*df* est donc un tableau espèces × sites à cette date), on obtient une ordination-compromis des espèces et des stations (ordination du "tableau moyen", construit en faisant la moyenne simple des tableaux).

B : si chacun des *df* correspond à un site, on obtient une ordination des espèces et des dates.

C : si chacun des *df* correspond à une espèce, on obtient une ordination des dates et des stations.

La fonction **kplot(*x*)** (identique à **kplot.foucart()**) projette en individus supplémentaires chacun des *k* data frames sur l'ordination-compromis correspondante.

Référence

Pavoine, S., Blondel J., Baguette M. & Chessel D. 2007. A new technique for ordering asymmetrical three-dimensional data sets in ecology. *Ecology* 88:512-523.

22) L'ANALYSE FACTORIELLE MULTIPLE (AFM)

☛ Ne pas confondre avec l'AFCM (cf. n°**6**).

Cette méthode s'applique à *k* tableaux ayant leurs lignes-échantillons en commun, chaque tableau correspondant à un groupe de différentes variables-colonnes. L'analyse met en évidence les relations entre ces *k* groupes de variables en projetant sur un plan moyen les ordinations des échantillons-lignes et des variables communes aux *k* tableaux. L'analyse inter-groupes met en évidence les rôles respectifs des *k* tableaux (sans distinguer les variables), les rapprochant s'ils ont le même rôle.

☛ Les variables peuvent être quantitatives ou qualitatives. Si elles sont toutes quantitatives, l'AFM peut être vue comme une variante de l'ACP, en trois étapes : ACPs des tableaux séparés et normalisations (cf. ci-après), réunion des tableaux normalisés en un seul soumis à une ACP non normée, puis projection de chacun des *k* tableaux sur l'analyse globale (= compromis, ou

consensus), mettant en évidence ses accords et les écarts avec celle-ci.

Pour uniformiser le rôle des tableaux dans l'analyse simultanée quel que soit leurs nombres de variables et leurs inerties, chaque tableau est multiplié par un poids (4 options ; par défaut, l'inverse de la première valeur propre) qui diminue l'importance des grands tableaux et augmente celle des petits.

En pratique, on commence à assembler les *k* tableaux (*df1* ; *df2* ,…*dfk*) en une liste, puis on transforme celle-ci en objet de classe ktab, que l'on soumet à l'analyse :

***listx* <- list(*df1,df2,…dfk*)**

***ktabx* <- ktab.list.df(*listx*)**

x* <- mfa(*ktabx*)** si ***x est le nom donné à l'analyse

Un ensemble de figures est obtenu par **plot(*x*)**

- En haut à droite : carte des variables (***x*$co**)
- En haut à gauche : carte des individus (***x*$li**)
- En bas à gauche : composantes principales du compromis
- En bas à droite : structure inter-groupes (typologie des tableaux dans un système d'axes communs à toutes les variables) ; les coordonnées d'un tableau (***x*$link**) s'interprètent en termes de liaison globale de ce tableau avec les facteurs communs aux *k* tableaux, les proximités signifiant de fortes corrélations entre tableaux.

Le rôle respectif des variables de chaque tableau (pris séparément) peut être apprécié par :

kplot(*x*)

Références

Abdi, H., Williams, L.J., Valentin, D. 2013. Multiple factor analysis: principal component analysis for multitable and multiblock data sets. *WIREs Comput Stat*. doi: 10.1002/wics.1246.

Escoffier, B., Pagès, J. 1998. *Analyses factorielles simples et multiples* (3ème édition). Dunod, Paris.

Et pour des exemples :

Alary, V., Messad, S., Tillard, E. 2001. Approche fonctionnelle de la diversité des systèmes d'élevage laitiers à l'Ile de la Réunion. Utilisation de l'AFM comme aide à l'interprétation de la variabilité inter et intra groupe. *Renc. Rech. Ruminants* 251-255.

Carlson, M.L., Flagstad, L.A., Gillet, F., Mitchell, E.A.D. 2010. Community development along a proglacial chronosequence: are above-ground and below-ground community structure controlled more by biotic than abiotic factors? *J. of Ecology* 98:1084–1095.

ANNEXE

Chargement et manipulation élémentaires de tableaux avec R

ade4 utilise en entrée, et produit en sortie, des tableaux qui doivent ou peuvent subir diverses manipulations (entrée, sauvegarde, examen, transposition, tri, fusions, transformations, etc.). Ci-après les fonctions **R** les plus utiles pour cela.

♦ Pour commencer

Se mettre immédiatement dans le bon répertoire de travail :

barre d'outils de la console / Fichier / Changer le répertoire courant

♦ Quelques fonctions numériques de R

```
> sqrt()              # racine carrée
> log()               # log népérien ;   ☛ décimal : log10() ; à base 2 : log2()
> exp()          # exponentielle
> sin() cos()         # sinus  ou  cosinus  (☛ avec angles en radians)
> sin(pi/180 * α)     # sinus               (avec angle α en degrés)
> asin()              # arcsin   (utile pour transformer des %)
```

♦ Gestion de la mémoire

On peut stocker en mémoire en tant qu'objet sous un nom quelconque (*alphanumérique, sans blancs, le premier caractère doit être une lettre*) : un nombre, une expression mathématique, un résultat numérique, un vecteur (c'est-à-dire une suite de nombres), un tableau (matrice ou dataframe), un modèle, une analyse… (consulter la doc. R pour les ≠ classes d'objets).

Pour connaître le contenu de la mémoire (accessible aussi par la barre d'outils de la console) :

```
>  ls()              # liste tous les objets en mémoire
```

Pour vider la mémoire :

Barre d'outils / Misc / effacer tous les objets

♦ Syntaxe R : noms, blanc, titres,

– Les blancs sont ignorés (ex: ligne(s) sautée(s) ou non ; A+B ou A + B ;.sont équivalents), mais il ne faut pas en mettre dans les noms d'objets (« mon_fichier » et non « mon fichier ») ou de variables (« strate1 » et non « strate 1 »).

– Dans les noms, éviter les caractères : « / », « - », et « **NA** ».

– ***R*** fait la différence entre majuscules et minuscules dans les noms d'objets ou de fonctions.

☛ Titres des variables-colonnes du tableau : alphanumériques commençant par une lettre. Titres des échantillons-lignes : peuvent être de simples numéros.

☛ Attention : R n'accepte pas de doublons dans les noms de lignes.

♦ Décimales

☛ Les décimales s'écrivent avec un point. Attention avec des données issues d'un tableur français, c'est une cause d'erreur fréquente.

Trois solutions : (i) changer l'option langue de français à anglais dans le panneau de configuration, (ii) remplacer les virgules par des points soit sur le tableur soit sur l'éditeur, (iii) ajouter **dec=","** dans la parenthèse **de read.table()** ; cf. ci-après.

♦ Data frame et vecteurs

Voir les tutoriels R pour les différentes classes d'objets. Les analyses de **ade4** génèrent en sortie et

demandent en entrée des vecteurs (suites de nombres) et des *data frames* (valeurs numériques ou caractères).

Par défaut, la fonction read.table() transforme les variables codées en caractères en facteurs, ce qui est très pratique (il vaut mieux coder une variable *V* jouant le rôle de facteur en faible, moyen, fort ou en a, b, c, plutôt qu'en 1, 2, 3 ; dans ce dernier cas il faut la transformer en facteur par : *fac* = as.factor(*V*)

La plupart des analyses requièrent non une matrice (le message d'erreur correspondant est: "*operator is invalid for atomic vectors*" !) mais un « ***data frame*** », c'est-à-dire un tableau qui peut contenir des éléments de modes différents. Toutes les analyses peuvent être faites sur des *data frames*.

Pour vérifier la nature d'un fichier ***tab*** (*data.frame, factor, vector* ou autre) :

> class(*tab*)

Pour transformer un tableau *tab* en *data frame* :

> *df* <- as.data.frame(*tab*)

Entrer des données

♦ Générer un fichier de données .txt utilisable par R

En général on rentre les données à partir d'un tableur (Excel ou OpenOffice Calc).

Sauvegarder son tableau de données au format **.txt** avec séparateurs tabulations, ou au format **.prn** avec séparateurs blancs. (attention de bien être dans le répertoire de travail défini plus haut).

♦ Entrer un tableau de données et lui affecter un nom d'objet (par ex.: "*tab*")

Il y a quatre principales possibilité de fichiers texte, avec ou sans titres pour les lignes et les colonnes:

```
   (A)                (B)                (C)            (D)
   V1   V2   V3       NOM V1 V2 V3       V1  V2  V3
L1 1    2    3        L1  1   2   3      1   2   3      1  2  3
L2 4    5    6        L2  4   5   6      4   5   6      4  5  6
```

(A) Cas d'un fichier unique incluant les titres (*labels*) des colonnes et ceux des lignes, en laissant vide le coin en haut à gauche [la cellule A1] du tableur avant sauvegarde :

> *tab* <- read.table("*tableau.txt*")

(B) Cas où la cellule A1 n'est pas vide, il faut préciser :

> *tab* <- read.table("*tableau.txt*" , header=TRUE, row.names=1)

(C) Cas où il n'y a que des titres de colonnes :

> *tab* <- read.table("*tableau.txt*" , header=TRUE)

(D) Possible, mais pas recommandé : dans ce cas, read.table attribue par défaut les noms V1, V2, … à des variables-colonnes sans titres ; mais il faudra alors préciser les *index* des colonnes dans les fonctions.

☛ *Le plus rapide à partir d'un tableur* (sans passer par un fichier .txt) : sélectionner-copier votre tableau sous Excel ou OpenOffice Calc sous le format (***A***) ci-dessus, puis faire :

> *tab* <- read.table("clipboard")

Vérifier, manipuler, modifier des données

Vérifier un tableau de données (à faire dans tous les cas après un read.table !)

– Afficher tout le tableau :

> *tab* # défilement de tout le tableau (avec arrêt à la fin)
> edit (*tab*) # affichage du tableau dans une fenêtre

– Avec la commande edit, on peut faire des modifications à la main dans le tableau :
> *modif* <-edit(*tab*) # attention: attribuer un autre nom au tableau modifié

– Afficher seulement les 6 premières lignes du tableau (sinon il défile entièrement à l'écran) :
>head(*tab*)

– Afficher les 12 premières lignes et toutes les colonnes [☛ **l'*indexation*** est entre crochets] :
> *tab*[1 :12,] # ☛ noter la place de la virgule

– Afficher toutes les lignes, mais seulement telle ou telle variable VARj :
> *tab*[,VARj] # avec soit le nom de la variable, soit son n°

– Caractéristiques du tableau : cf. aussi min() et max() ; ex. : min(*tab*$VARj)
> dim(*tab*) # dimensions (lignes colonnes) du tableau
> rownames(), colnames() # noms des lignes, ou des colonnes
> summary(*tab*) # moyennes, médiane, extrêmes… des variables
> rowSums(*tab*) # sommes des lignes (S majuscule !); cf. colSums()
> max(*tab*) # valeur maximale du tableau (cf. aussi : min(*tab*)
> range(*tab*) # valeurs extrêmes du tableau

Transposer, concaténer, juxtaposer (ne pas oublier d'attribuer un nom d'objet <-)

> t(*tab*) # transposer (basculer un tableau de 90°)
> rbind(*tab*1, *tab*2) # concaténer (nombre de colonnes constant)
Si les 2 ou 3 tableaux ont les mêmes noms de lignes #, *R* ajoute 1,2,…aux noms de lignes du 2ème, 3ème tableau, etc.
> cbind(*tab*1, *tab*2) # juxtaposer (nombre de lignes constant)
Si les 2 tableaux ont des noms de lignes #, seuls les noms du premier sont retenus

Remplacer des données selon un critère

> *tab*[*tab*>0] <- 1 # ☛ convertir un tableau d'abondances en présences/absences

> *tab*[*tab*VAR_j$=="0", 4] <- 1 # ici remplace par 1 toutes les valeurs de la4éme colonne lorsque la VAR$_j$ = 0.

Transformer une (des) variable(s) (toute opération…)

> transform(*tab*, VAR$_j$=log(VAR$_j$)) # …ici transformation en log de VAR$_j$
> log(*tab*+1) # ☛ transforme en log (*x*+1) tout le tableau.

Répéter une opération sur toutes les lignes (si **1**) **ou colonnes** (si **2**)

> apply(*tab*, 1, mean) # ici moyennes de chaque ligne

Données manquantes dans un tableau

Les coder conventionnellement **NA** (not available) dans Excel avant de sauvegarder sous format .txt. Ne pas utiliser « NA » dans les fichiers ou les programmes pour autre chose que les données manquantes (éviter ce nom de var.). Cf. aussi: NaN: « not a number », par ex. 0/0)

Extraire des sous-tableaux (attribuer un nom d'objet à l'extrait avec <-)

> *extrait* <- subset(*tab*, select = c(VAR$_{j1}$,VAR$_{j2}$,…)) # sélection de var-colonnes
> *extrait* <- *tab*[1 :12,] # sélection des 12 premières lignes

– sélection de lignes selon une VAR$_j$ qui doit remplir certains critères ; exemples :
> *extrait* <- *tab*[*tab*VAR_j$ == "X",] # la var. doit avoir la chaîne « X »
> *extrait* <- *tab*[*tab*VAR_j$ > 0,] # la var. doit être > 0

– Elimination des lignes avec des données manquantes (NA) dans VAR$_{j1}$ ou dans VAR$_{j2}$:

> *extrait* <- *tab*[(!is.na(*tab*VAR_1$) & !is.na(*tab*$VAR$_2$)),]

NB) Quand il y a plusieurs conditions, les relier par : & (= et), | (= ou)
Par ailleurs, ! signifie not, et égale s'écrit == (et non =) car ici c'est un test.

Réordonnner un tableau selon une (ou deux) variable(s) (ici ordre croissant)

> *tri* <- *tab*[order(*tab*$ VAR$_j$, decreasing=F) ,] # ne pas oublier la virgule
> *tr i* <- *tab*[order(*tab*[,3], decreasing=F) ,] # idem, tri selon la 3ème variable
> *tri* <- *tab*[order(*tab*$ VAR$_1$, *tab*$ VAR$_2$) ,] # ici deux critères de tri

Visualiser les données d'un tableau

☛ ***Visualiser d'un coup toutes les valeurs d'un tableau*** *(pas trop grand !)*
Les valeurs sont remplacées par des carrés plus ou moins gros :

> table.value(*tab*, csize=1) # Rem : nécessite **library(ade4)**
Le csize règle la taille des carrés noirs

> table.cont(*tab*, csize=1) *idem* (carrés blancs) ; n'admet pas les valeurs négatives

Visualisation univariée ; histogramme

Nombre d'individus (lignes) dans chaque classe d'une variable (colonne) du tableau

> hist(*tab*[,"VAR$_1$"], nclass=10, labels=TRUE, xlab="Classes de VAR$_1$", ylab="Nombre d'observations", main="Exemple d'histogramme")

avec de nombreuses options (dont certaines communes à la fonction plot() ; cf. plus loin)

breaks=	on fixe le nombre de classes (ou : nclass=)
breaks=c(10,20,30,40,50,60)	on fixe e les limites des classes
labels= [booléen]	si TRUE, étiquette les barres
main ("chaîne de caractères")	titre principal du graphique
xlab ("chaîne de caractères")	titre de l'axe des abscisses
ylab ("chaîne de caractères")	titre de l'axe des ordonnées, …

Cf. aussi (pour une variable discontinue) la fonction barplot()

Visualisation bivariée : nuage de points x y

☛ ***Visualiser d'un coup tous les nuages xy possibles d'un tableau***

> plot(*tab*) # figure tous les nuages VAR$_{j1}$, VAR$_{j2}$

– Visualiser les deux variables VAR$_1$ (en *x*) et VAR$_2$ (en *y*) :

> plot(*tab*VAR_1$, *tab*$VAR$_2$)

Personnaliser son graphique xy. Un exemple :

> plot(VAR$_1$, VAR$_2$, xlab="Variable n°1 » , ylab="Variable n°2", xlim=c(0, 100), ylim=c(0, 100), pch=22, col="red", bg="yellow", bty="l", tcl=0.4,main="Comment personnaliser son graphique sous R ?", las=1, cex=1.5, lines(VAR1,VAR2))

xlim=c(), ylim=c() : valeurs extrêmes des axes
pch : taille des caractères cex : taille des symboles
col= : couleur des symboles bg= : couleur du remplissage
las=1 : graduations verticales sur y (=2: perpendiculaires aux deux axes)
bty=l : sinon le graphe est dessiné dans un carré tcl= taille des graduations
lines(VAR1,VAR2) : joint les points par une ligne continue, …

Pour légender les points avec les noms de lignes, ajouter après la fonction plot() :

> text(VAR_1, VAR_2, labels=fichier.des.noms.de.lignes$nom.de.la.colonne)

> text(VAR_1, VAR_2, labels=row.names(*tab*)) # noms des lignes

Les possibilités graphiques sont infinies: cf. « *An introduction to R* » *: Graphical procedures*.

Visualisation trivariées : nuage de points pondérés x y z

– Visualiser 3 variables VAR_1 (en ***x***), VAR_2 (en ***y***), et VAR_3 (en **z** par la taille des cercles) :

> plot(VAR_1, VAR_2, cex=VAR_3)

> plot(VAR_1, VAR_2, cex=0.5*VAR_3) # modifier la taille des cercles si nécessaire

– *idem* (en **z** par la taille des carrés) :

> xy <- cbind.data.frame(*tab*$$VAR_1$, y = *tab*$$VAR_2$)

> z <- *tab*$$VAR_3$

> s.value(xy, z, xax=1, yax=2) # Rem : nécessite library(ade4)

Cas d'un plan factoriel avec des points pondérés (nécessite library(ade4)**):**

> s.value() : après une analyse factorielle, permet de visualiser les valeurs d'une variable dans les points-relevés d'un plan factoriel (par des carrés plus ou moins gros, l'option par défaut). Cette fonction utilise le tableau *dfxy* contenant les coordonnées des individus sur les axes, et un vecteur *z* avec les valeurs de la variable dans ces individus. Il faut préciser les axes si ça n'est pas les 1 et 2. Par ex. :

> s.value(*df*xy, *z*, xax=1, yax=3, csize= *un_nombre*) # csize règle la taille des carrés.

idem avec des nuances de gris:

> s.value(*df*xy, *z*, method="greylevel")

Sauvegardes

♦ *Sauvegarder un graphique*

Par la barre-outil de la console :

fichier / sauvegarder sous…

On peut choisir le format vectoriel **.emf**, sans pixel à l'agrandissement, lu par Word.
Ou le format image **.png**, moins volumineux, mieux lu, et qui peut être retouché.
Ou le format **.pdf** de meilleure définition (notamment si les étiquettes des graphes sont en très petits caractères).

♦ *Sauvegarder un tableau* (sous format txt, par ex. pour utilisation dans un autre logiciel)

> write.table(*tab*,« tableau.txt »)

☛ dans ade4 sous R, les fichiers résultats numériques des analyses ne sont pas automatiquement sauvegardés ; il faut donc le faire de cette façon si on veut les conserver.

♦ *Sauvegarder une session de travail :* deux solutions :

Barre de commande/Fichier/Sauver l'historique des commandes : génère un fichier « .Rhistory » lisible avec un éditeur, et qui ne contient que les commandes, sans résultats numériques.

Barre de commande/Fichier/Sauver dans le fichier : génère un fichier « lastsave.txt » qui contient les commandes et les résultats numériques (seulement ceux qui ont été affichés à l'écran).

☛ Dans les deux cas, nettoyer le fichier de toutes les erreurs et le commenter avant archivage.