Consultations statistiques avec le logiciel 

# Que faire avec un tableau croisant individus et sites microsatellites ?

## Résumé

La question est posée par Hadrien Vanthomme à propos de chats et du problème de l'hybridation entre Felis silvestris (chat sauvage) et Felis catus (chat domestique). On regroupe ici des réflexions induites par cette consultation.

## Plan

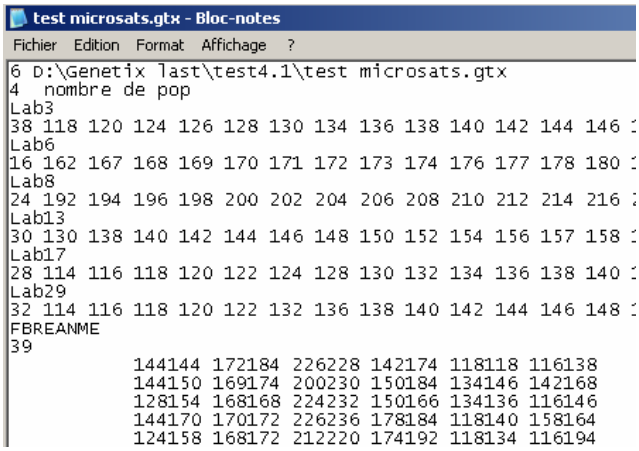
# 1. Lire les données

Généralement les données génétiques portent sur des individus initialement regroupés en populations (au sens commun du terme, il peut s'agir d'échantillons, de groupes, de races, de sites, de populations biologiques, …). On s'intéresse ici à la lecture d'un tableau de données dont les lignes sont des individus diploïdes, les colonnes des loci et les valeurs le type des deux allèles rencontré chez l'individu dans ce site. On appellera forme allélique une modalité observée en un locus et allèle une réalisation de cette modalité. De l'information supplémentaire peut être disponible par ailleurs (morphométrie, sexe, date, lieu, statut, …) mais les données génétiques proprement dite forment un tableau autosuffisant.

Nous utiliserons le format des données de Genetix :

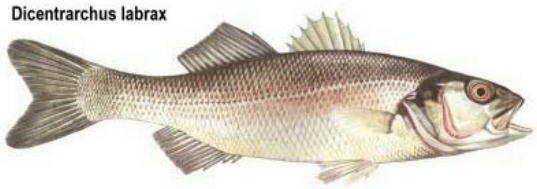
http://www.univ-montp2.fr/~genetix/genetix/genetix.htm

La version 4.05 propose un fichier d'exemple :



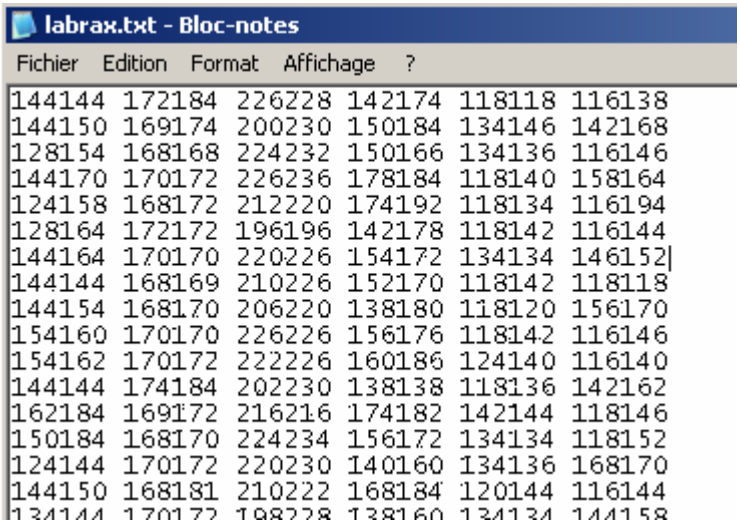

Ces données sont publiées dans Naciri et al. (1999).

http://www.echosmouche.com/IMG/jpg/dicentrachus_labrax.jpg




Il y a 6 sites microsatellites :

Labrax-3 , Labrax-6, Labrax-8 , Labrax-13, Labrax-17 et Labrax-29 définis par Leon et al. (1995). On a 4 groupes (39, 60, 26 et 50 individus) qu'on ignore provisoirement. On réduit le fichier à l'essentiel :




```
tab1=read.table("labrax.txt",as.is=T,sep=" ")
tab1[1:5,]
      V1     V2     V3     V4     V5     V6
1 144144 172184 226228 142174 118118 116138
2 144150 169174 200230 150184 134146 142168
3 128154 168168 224232 150166 134136 116146
4 144170 170172 226236 178184 118140 158164
5 124158 168172 212220 174192 118134 116194
```

Mettre des noms de variables :

```
names(tab1) = paste("lab",c(3,6,8, 13,17,29),sep="")
tab1
    lab3   lab6   lab8  lab13  lab17  lab29
1 144144 172184 226228 142174 118118 116138
2 144150 169174 200230 150184 134146 142168
...
```

La seule chose utile à connaître est le paramètre **as.is** dans **read.table** qui conserve les chaînes de caractères en évitant le passage automatique à la classe facteur. Un autre exemple est dans **casitas**.

```
data(casitas)
dim(casitas)
[1] 74 15
summary(casitas)
     Aat                Amy                Es1                Es2
 Length:74          Length:74          Length:74          Length:74
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
     Es10               Hbb                Gpd1               Idh1
 Length:74          Length:74          Length:74          Length:74
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
     Mod1               Mod2               Mpi                Np
 Length:74          Length:74          Length:74          Length:74
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
     Pgm1               Pgm2               Sod
 Length:74          Length:74          Length:74
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character

casitas[1:5,]
     Aat    Amy    Es1    Es2   Es10    Hbb   Gpd1   Idh1   Mod1   Mod2    Mpi
1 100100 080080 094094 100100 100100 120120 100100 100100 110110 100100 100100
2 100100 080100 094094 100100 100100 120120 100100 100125 110110 100100 100100
3 100100 080080 094094 100100 100100 120120 100100 100100 110110 100100 100100
4 100100 080080 094094 100100 100100 120120 100100 100125 100100 100100 100100
5 100100 080080 094094 100100 100100 120120 100100 100100 110110 100100 100100
      Np   Pgm1   Pgm2    Sod
1 100100 100100 100100 100100
```

```
2 100100 100100 100100 100100
3 100100 100100 100100 100100
4 100100 100100 100100 100100
5 100100 100100 100100 100100
```

Les données sont proposées dans le logiciel Genetix et publiées dans Orth et al. (1998). Les données manquantes sont codées `0`. On peut utiliser aussi `000000`.

```
tab1[apply(tab1,1, function(x) any(x=="0")),]
     lab3   lab6   lab8  lab13  lab17  lab29
29 158162 168168      0 160168 118122 144152
32 162172 168172 204230 156162      0 140152
37 162162 170172 210222 152158      0 116154
44 144154 172172 222228 138152      0 158160
50 160162 170170      0 156160 132134 146164
51 142160 168170      0 146156 118136 118144
52 144150 172172      0 142144 118146 118144
```

La notation `96094` pour `096094` est tolérée, la fonction de lecture ramenant toute chaîne à 6 caractères en ajoutant des `0` en tête. On suppose qu'on a des individus complètement typés (`100110`) ou complètement non typés (`0` ou `000000`)

Il peut y avoir des en-têtes et des noms d'individus dans le fichier :

vanthomme.txt

| chat | Fca8 | Fca26 | Fca31 | Fca43 | Fca58 | Fca77 | Fca78 | Fca80 | Fca90 | Fc |
|---|---|---|---|---|---|---|---|---|---|---|
| Fs0101 | 144144 | | 140142 | | 238240 | | 120128 | | 000000 | |
| Fs0102 | 136142 | | 144144 | | 234234 | | 118118 | | 222232 | |
| Fs0104 | 136142 | | 136156 | | 234238 | | 118118 | | 000000 | |
| Fs0105 | 140140 | | 144146 | | 234238 | | 118118 | | 230232 | |
| Fs01x1 | 142142 | | 146150 | | 234240 | | 118118 | | 222224 | |
| Fs01x2 | 136136 | | 142144 | | 238244 | | 118118 | | 214222 | |
| Fs01x3 | 1421 42 | | 144150 | | 218218 | | 118130 | | 222222 | |
| Fs0303 | 126142 | | 152156 | | 230238 | | 000000 | | 000000 | |
| Fs0306 | 128144 | | 142146 | | 236240 | | 118118 | | 230232 | |
| Fs0307 | 138138 | | 144148 | | 000000 | | 118118 | | 000000 | |

```
vanthomme = read.table("vanthomme.txt", h=T,r=1,as.is=T)
vanthomme[1:4,]
         Fca8  Fca26  Fca31  Fca43  Fca58  Fca77  Fca78  Fca80  Fca90  Fca96
Fs0101 144144 140142 238240 120128      0 151159 195203 246256  94102      0
Fs0102 136142 144144 234234 118118 222232 141145 197203      0 114114 209213
Fs0104 136142 136156 234238 118118      0 141145 195203 250266 104108 223225
Fs0105 140140 144146 234238 118118 230232 139149 197197 250250 114118 207213
...
dim(vanthomme)
[1] 261  13
```

Ce tableau contient 261 chats typés sur 13 sites (recherche en cours).

Le fichier peut contenir d'autres variables mais le codage génétique seul doit être isolé :

BOVdata.txt

| race | num | INRA063 | INRA005 | ETH225 | ILSTS005 | HEL5 | HEL1 | INRA035 | ETH |
|---|---|---|---|---|---|---|---|---|---|
| Borgou | 1 | 183183 | 137141 | 147157 | 190190 | 149149 | | 103109 | 102 |
| Borgou | 2 | 181183 | 141141 | 139157 | 186186 | 151163 | | 105107 | 104 |
| Borgou | 3 | 177183 | 141141 | 139139 | 194194 | 151165 | | 103103 | 104 |
| Borgou | 4 | 183183 | 141141 | 141147 | 184190 | 167167 | | 103105 | 104 |
| Borgou | 5 | 177183 | 141141 | 153157 | 184186 | 155165 | | 103107 | 104 |
| Borgou | 6 | 177183 | 137143 | 149157 | 184186 | 155165 | | 103105 | 104 |
| Borgou | 7 | 177181 | 139141 | 147157 | 184190 | 165167 | | 103103 | 104 |
| Borgou | 8 | 183183 | 139141 | 155157 | 184186 | 165165 | | 103107 | 102 |
| Borgou | 9 | 177183 | 139141 | 139143 | 182190 | 165165 | | 103105 | 104 |
| Borgou | 10 | 183183 | 141141 | 157159 | 186186 | 167167 | | 103105 | 104 |
| Borgou | 11 | 177177 | 141141 | 147157 | 184190 | 165165 | | 107109 | 102 |
| Borgou | 12 | 183183 | 143143 | 139157 | 186186 | 155155 | | 101107 | 104 |

```
bovdata = read.table("BOVdata.txt",h=T,as.is=T)
names(bovdata)
 [1] "race"     "num"      "INRA063"  "INRA005"  "ETH225"   "ILSTS005"
 [7] "HEL5"     "HEL1"     "INRA035"  "ETH152"   "INRA023"  "ETH10"
[13] "CSSM66"   "INRA032"  "ETH3"     "BM2113"   "BM1824"   "HEL13"
[19] "INRA037"  "BM1818"   "ILSTS006" "MM12"     "CSRM60"   "ETH185"
[25] "HAUT24"   "HAUT27"   "TGLA227"  "TGLA126"  "TGLA122"  "TGLA53"
[31] "SPS115"

bovdata = bovdata[,-(1:2)]
bovdata[1:4,]

  INRA063 INRA005 ETH225 ILSTS005   HEL5   HEL1 INRA035 ETH152 INRA023  ETH10
1  183183  137141 147157   190190 149149 103109  102104 191195  215217 209211
2  181183  141141 139157   186186 151163 105107  104104 195195  199217 207217
3  177183  141141 139139   194194 151165 103103  104104 195197  201201 207211
4  183183  141141 141147   184190 167167 103105  104104 195195  201217 211211
...

dim(bovdata)
[1] 776  29
```

Ce tableau contient 776 bovins typés sur 29 sites (recherche en cours). Il existe des études où aucune classification a priori des individus ne sera disponible. Par exemple, il peut s'agir des arbres d'une espèce dans un site : on disposera alors de l'enregistrement des coordonnées spatiales (Smouse and Peakall 1999). Les données sont donc massivement multivariées (le nombre des variables est celui des formes alléliques et non celui des loci) et la question de la réduction du tableau pour l'étude de la typologie des individus demande de choisir une méthode d'ordination et des pratiques de dépouillement. ACP ? AFC ? ACM ? PCO ? Telle est la question posée.

# 2. Analyse de base : la plus simple est la meilleure

La réduction de tels tableaux n'est pas difficile mais pose quelques questions particulières. En général, on pose le problème au niveau des populations et on utilise directement un tableau de fréquences alléliques. On trouve alors des usages de l'ACP centrée classique (Menozzi et al. 1978), de l'AFC ou des variantes proches. On trouve l'expression "bidimensional representation of a 3-D correspondence analysis" pour désigner une analyse dont on a gardé 3 facteurs et utilisé deux sur la figure (Arnaiz-Villena et al. 2001). Il y a peu d'exemples d'utilisation d'une ordination multivariée sur des individus. Il faut dire que sous l'hypothèse d'Hardy-Weinberg un individu n'est rien d'autre qu'un assemblage aléatoire de deux allèles par locus tirés au hasard conformément aux fréquences alléliques et qu'à ce titre il n'est qu'un représentant sans personnalité d'un modèle aléatoire. On demande une analyse a priori, descriptive qui permettent de s'orienter vers des modèles plus complexes et des questions plus précises comme "Combien de populations ?". Dans Maudet et al. (2002) :

> Several studies have shown that microsatellites can be used to identify the population of origin of an individual (Paetkau *et al*. 1995; Cornuet *et al*. 1999). Several approaches have recently been proposed for identifying the origin of individuals using molecular makers (Paetkau *et al*. 1995; Rannala & Mountain 1997; Cornuet *et al*. 1999; Banks & Eichert 2000; Prichard *et al*. 2000). Population assignment tests might be very useful to fight against poaching by assigning an individual (trophy, carcass or animal product) to its population of origin (Manel *et al*. 2002). Poaching and illegal trafficking of wildlife products are among the most serious threats to the survival of many wildlife species. Ungulates are especially threatened by poaching because of their coveted trophy horns or antlers. Assignment tests are also useful for detecting dispersal and immigration (Rannala & Mountain 1997; Waser *et al*. 2001).

Une analyse de base demande un tableau transformé, une pondération des lignes et une pondération des colonnes. Les lignes, individus typés, ont tous la même importance et la pondération des lignes est nécessairement uniforme. Les colonnes-variables sont les formes alléliques. Un individu homozygote est codée 0010000, un hétérozygote est codé 0½00½00 et un inconnu est codé $f_1,...,f_k,...,f_m$ qui sont les fréquences alléliques calculées sur l'ensemble des données disponibles. Ainsi en la somme de chaque bloc individu-locus vaut 1 et le centrage à pondération uniforme remet les non typés à l'origine. Donc l'origine est l'individu moyen par définition. La fonction **fuzzygenet** fait cette opération.

```
data(casitas)
casi=fuzzygenet(casitas)
dim(casitas)
[1] 74 15 # tableau individus-loci
dim(casi)
[1] 74 38
names (casi)
 [1] "L01.1" "L01.2" "L02.1" "L02.2" "L03.1" "L03.2" "L04.1" "L04.2" "L04.3"
[10] "L05.1" "L05.2" "L06.1" "L06.2" "L07.1" "L07.2" "L07.3" "L08.1" "L08.2"
... Les locus et les allèles sont numérotés pour simplifier les éditions

attributes(casi)
$names
 [1] "L01.1" "L01.2" "L02.1" "L02.2" "L03.1" "L03.2" "L04.1" "L04.2" "L04.3"
...

$row.names
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
...

$class
[1] "data.frame"

$col.blocks
L01 L02 L03 L04 L05 L06 L07 L08 L09 L10 L11 L12 L13 L14 L15
  2   2   2   3   2   2   3   4   3   2   2   4   3   2   2
Important : vecteur du nombre d'allèles par locus

$all.names
     L01.1      L01.2      L02.1      L02.2      L03.1      L03.2      L04.1
 "Aat.080"  "Aat.100"  "Amy.080"  "Amy.100"  "Es1.094"  "Es1.100"  "Es2.095"
     L04.2      L04.3      L05.1      L05.2      L06.1      L06.2      L07.1
...

$loc.names
   L01    L02    L03    L04    L05    L06    L07    L08    L09    L10    L11
 "Aat"  "Amy"  "Es1"  "Es2" "Es10"  "Hbb" "Gpd1" "Idh1" "Mod1" "Mod2"  "Mpi"
   L12    L13    L14    L15
  "Np" "Pgm1" "Pgm2"  "Sod"
Important : loc.names sert d'étiquettes en clair

$row.w
 [1] 0.01351 0.01351 0.01351 0.01351 0.01351 0.01351 0.01351 0.01351 0.01351
...

$col.freq
   L01.1    L01.2    L02.1    L02.2    L03.1    L03.2    L04.1    L04.2
0.148649 0.851351 0.750000 0.250000 0.702703 0.297297 0.081081 0.195946
   L04.3    L05.1    L05.2    L06.1    L06.2    L07.1    L07.2    L07.3
0.722973 0.116438 0.883562 0.465753 0.534247 0.375000 0.534722 0.090278
   L08.1    L08.2    L08.3    L08.4    L09.1    L09.2    L09.3    L10.1
0.101351 0.006757 0.608108 0.283784 0.128378 0.797297 0.074324 0.689189
   L10.2    L11.1    L11.2    L12.1    L12.2    L12.3    L12.4    L13.1
0.310811 0.794521 0.205479 0.061644 0.089041 0.061644 0.787671 0.027397
   L13.2    L13.3    L14.1    L14.2    L15.1    L15.2
0.130137 0.842466 0.082192 0.917808 0.121622 0.878378

$col.num
 [1] Aat  Aat  Amy  Amy  Es1  Es1  Es2  Es2  Es2  Es10 Es10 Hbb  Hbb  Gpd1 Gpd1
[16] Gpd1 Idh1 Idh1 Idh1 Idh1 Mod1 Mod1 Mod1 Mod2 Mod2 Mpi  Mpi  Np   Np   Np
```

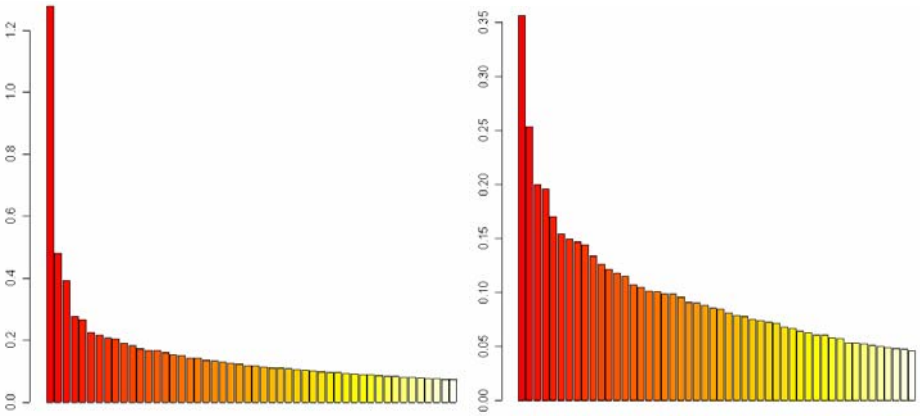
```
[31] Np   Pgm1 Pgm1 Pgm1 Pgm2 Pgm2 Sod  Sod
15 Levels: Aat Amy Es1 Es10 Es2 Gpd1 Hbb Idh1 Mod1 Mod2 Mpi Np Pgm1 ... Sod
```

**Important : col.num est le facteur qui attribue chaque allèle au locus auquel il appartient.**

## 2.1. Analyse de base

L'ACP centrée de ce tableau est la plus simple des analyses de base. On utilisera pour les illustrations les exemples suivants :

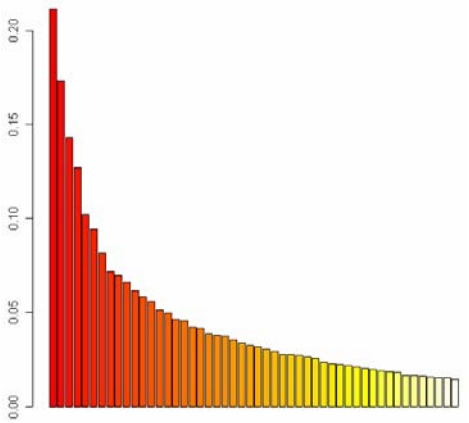
```
bov=fuzzygenet(bovdata)
bovpca=dudi.pca(bov,scale=F)
Select the number of axes: 5
barplot(bovpca$eig[1:50])

cha=fuzzygenet(vanthomme)
chapca=dudi.pca(cha,scale=F)
Select the number of axes: 5
barplot(chapca$eig[1:50])

lab=fuzzygenet(tab1)
labpca=dudi.pca(lab,scale=F)
Select the number of axes: 4
barplot(labpca$eig[1:50])

casi=fuzzygenet(casitas)
casipca=dudi.pca(casi,scale=F)
Select the number of axes: 3
barplot(casipca$eig[1:30])
```



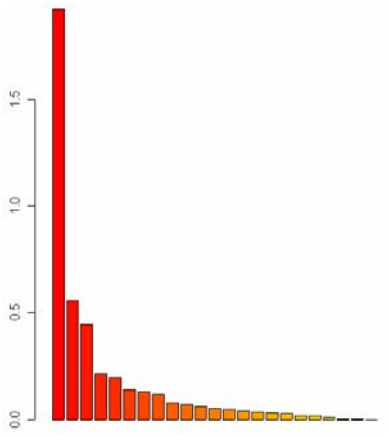
776 bovins et 378 allèles



261 chats et 204 allèles



175 poissons et 168 allèles

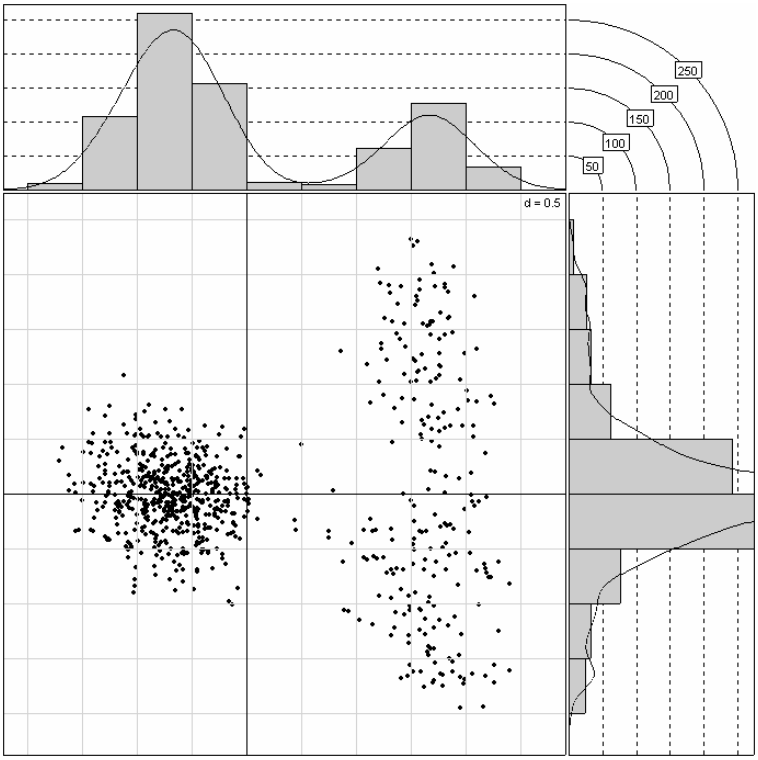


74 souris et 38 allèles

## 2.2. La carte des individus

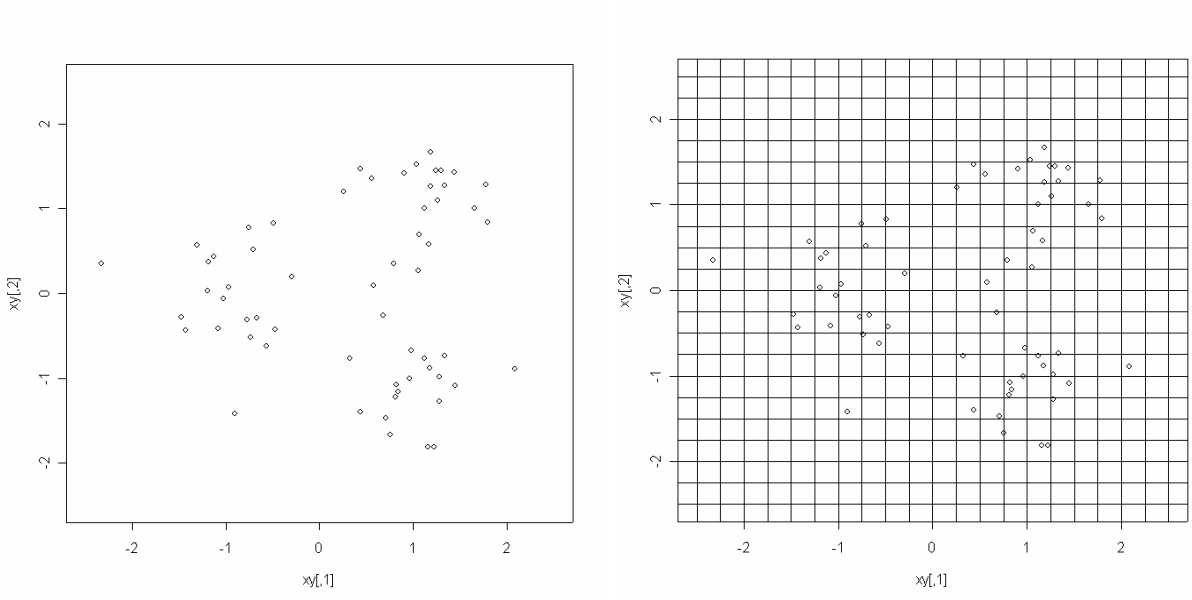
Elle est particulière et forme essentiellement un nuage de points. On cherche d'abord à savoir de quel type est la variabilité : groupement ou gradient ? C'est plus ou moins simple.

```
s.hist(bovpca$li,clab=0)
```




Pour faciliter la lecture on utilise la fonction **kde2d** de la librairie MASS. Soit un nuage de points :

```
a=mvrnorm(20,c(-1,0),S=diag(1,2)/5)
b=mvrnorm(20,c(1,-1),S=diag(1,2)/5)
c=mvrnorm(20,c(1,1),S=diag(1,2)/5)
xy=rbind(a,b,c)
plot(xy,xlim=c(-2.5,2.5),ylim=c(-2.5,2.5))
```

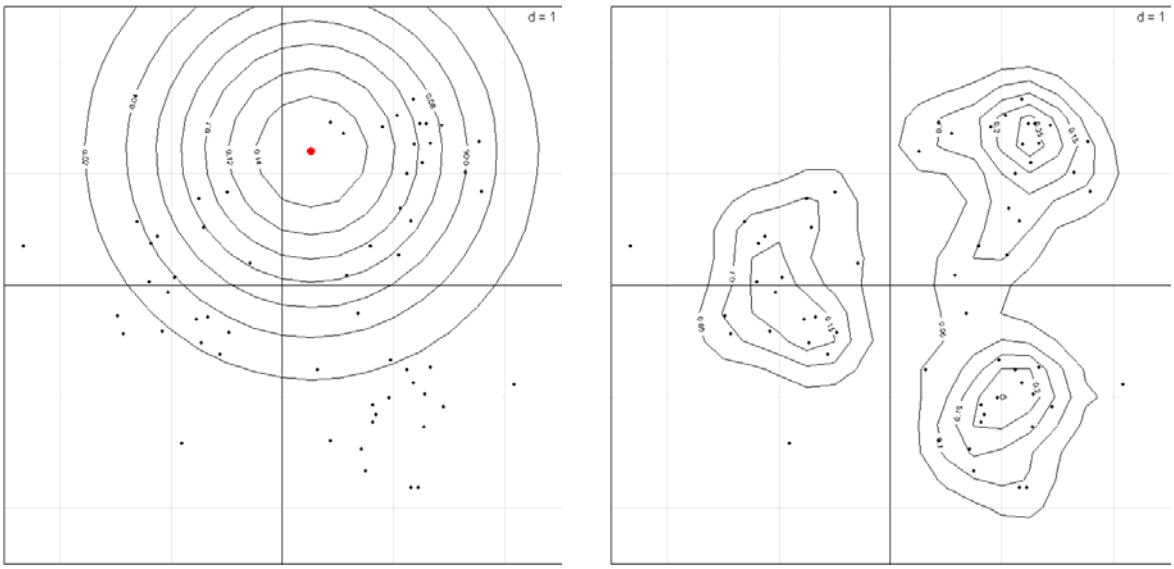



```
wx = seq(-2.5,2.5, by = 0.25)
wy = seq(-2.5,2.5, by = 0.25)
abline(v=wx)
```

```
abline(h=wy)
```

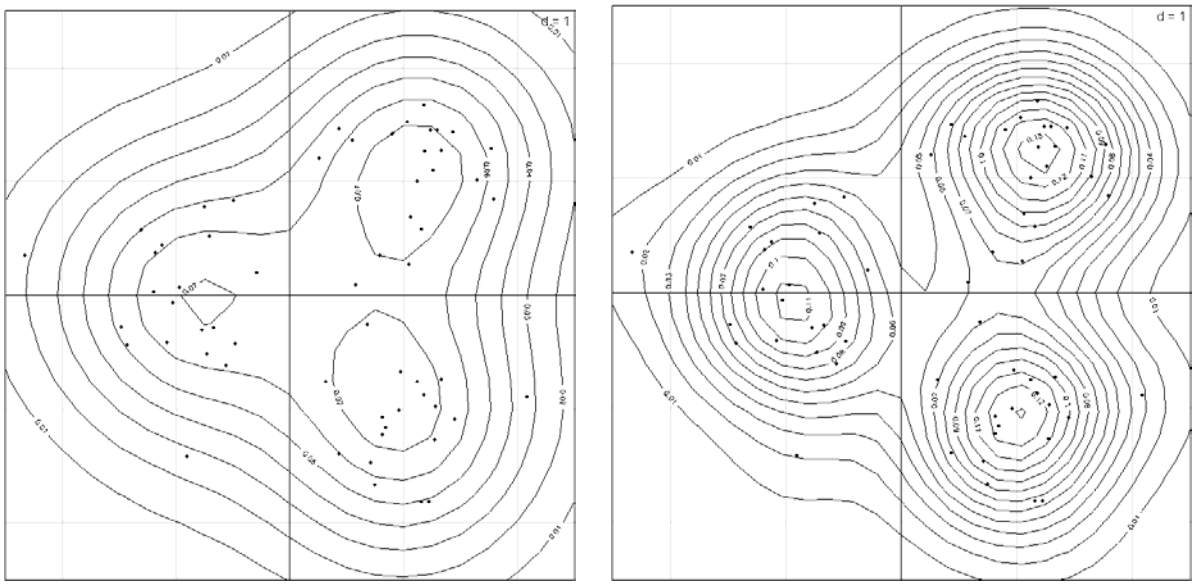
On a défini une grille sur le plan (par défaut 25x25, ici 21x21 pour faire simple). On définit alors une densité normale bivariée de moyenne nulle, de corrélation nulle et de variance unité. En chaque point de la grille la valeur calculée pour chaque point est moyennée et la surface ainsi définie donne des courbes de niveaux.

```
s.label(xy,xlim=c(-2.5,2.5),ylim=c(-2.5,2.5),clab=0)
xpo = xy[44,1] ; ypo = xy[44,2] ; points(xpo,ypo ,col="red",pch=20,cex=2)
ax = (wx - xpo)/1 ; ay = (wy - ypo)/1 ; az = dnorm(ax)%*%dnorm(t(ay))
contour(wx, wy ,az ,add=T, xlim=c(-2.5,2.5),ylim=c(-2.5,2.5))
```




```
s.label(xy,xlim=c(-2.5,2.5),ylim=c(-2.5,2.5),clab=0)
w=kde2d(xy[,1],xy[,2],h=c(1,1),n=21,lims=c(-2.5,2.5,-2.5,2.5))
contour(w,add=T, xlim=c(-2.5,2.5),ylim=c(-2.5,2.5))
```

On dit qu'on a fait une estimation non paramétrique par noyau de la densité locale des points (**kde** kernel estimation density). **h**, le paramètre de lissage contrôle l'écart-type de la loi de Gauss locale (le noyau). La valeur par défaut est la meilleure.




```
s.label(xy,xlim=c(-2.5,2.5),ylim=c(-2.5,2.5),clab=0)
w=kde2d(xy[,1],xy[,2],h=c(3,3),n=21,lims=c(-2.5,2.5,-2.5,2.5))
contour(w,add=T, xlim=c(-2.5,2.5),ylim=c(-2.5,2.5))

s.label(xy,xlim=c(-2.5,2.5),ylim=c(-2.5,2.5),clab=0)
```

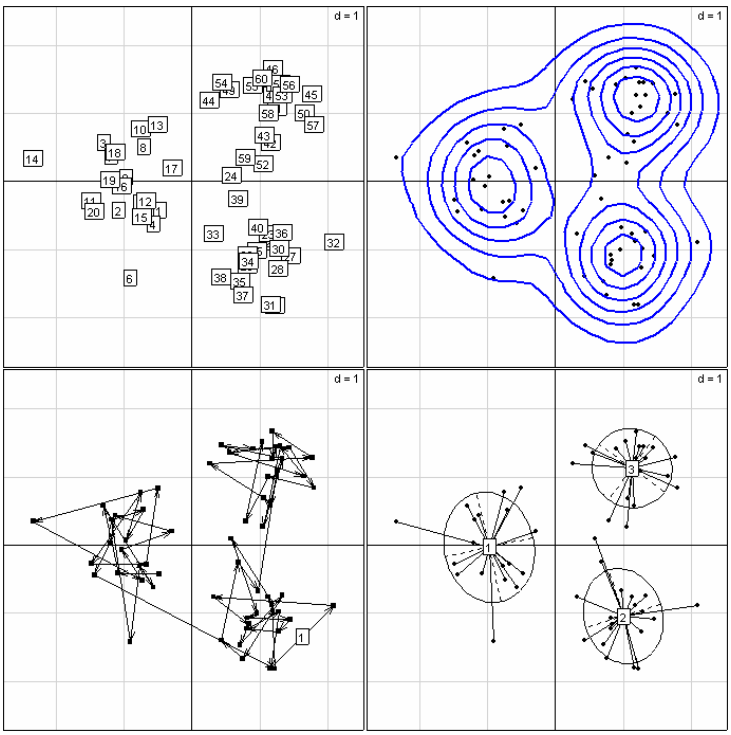
```
w=kde2d(xy[,1],xy[,2], lims=c(-2.5,2.5,-2.5,2.5))
contour(w,add=T, xlim=c(-2.5,2.5),ylim=c(-2.5,2.5))
```

*A méditer par les amateurs, la concision (10 lignes !) de la fonction* **kde2d** *de Venables et Ripley :*

```
function (x, y, h, n = 25, lims = c(range(x), range(y)))
{
    nx <- length(x)
    if (length(y) != nx) stop("Data vectors must be the same length")
    gx <- seq(lims[1], lims[2], length = n)
    gy <- seq(lims[3], lims[4], length = n)
    if (missing(h)) h <- c(bandwidth.nrd(x), bandwidth.nrd(y))
    h <- h/4
    ax <- outer(gx, x, "-")/h[1]
    ay <- outer(gy, y, "-")/h[2]
    z <- matrix(dnorm(ax), n, nx) %*% t(matrix(dnorm(ay), n, nx))/(nx * h[1] * h[2])
    return(list(x = gx, y = gy, z = z))
}
<environment: namespace:MASS>
```

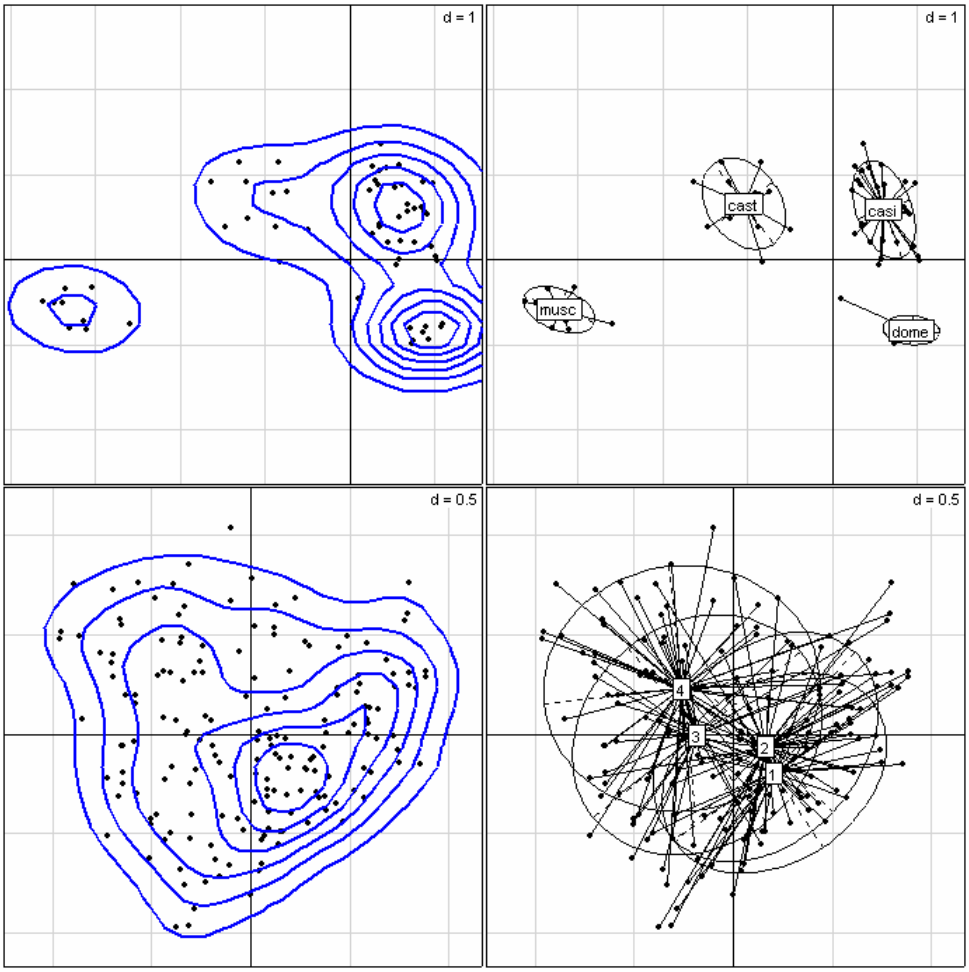
Elle est directement utilisée avec les paramètres par défaut dans **s.kde2d**.

```
par(mfrow=c(2,2))
s.label(xy)
s.kde2d(xy)
s.traject(xy)
s.class(xy,gl(3,20))
```

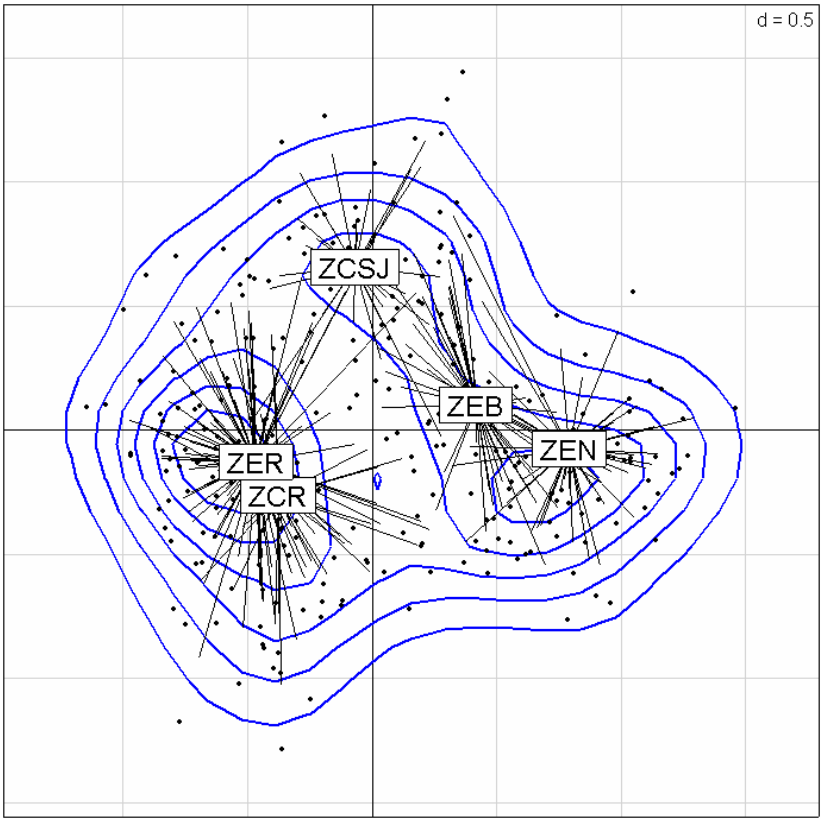


C'est un bon outil pour les données génétiques.

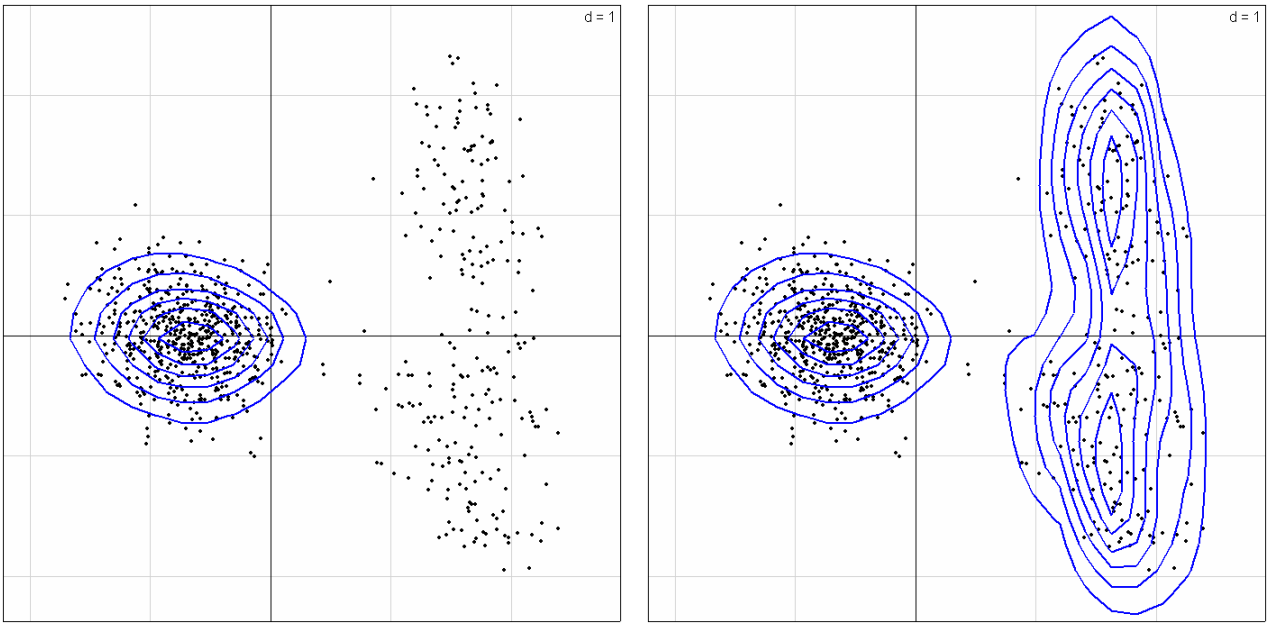
```
s.kde2d(casipca$li)
casitas.pop <- as.factor(rep(c("dome", "cast", "musc", "casi"), c(24,11,9,30)))
s.class(casipca$li,casitas.pop)

s.kde2d(labpca$li)
lab.pop <- as.factor(rep(1:4,c(39, 60,26,50)))
s.class(labpca$li,lab.pop)
```




```
s.kde2d(chapca$li)
s.class(chapca$li,morpho$zone,cell=0,csta=0.75,clab=1.5,add.p=T)
```

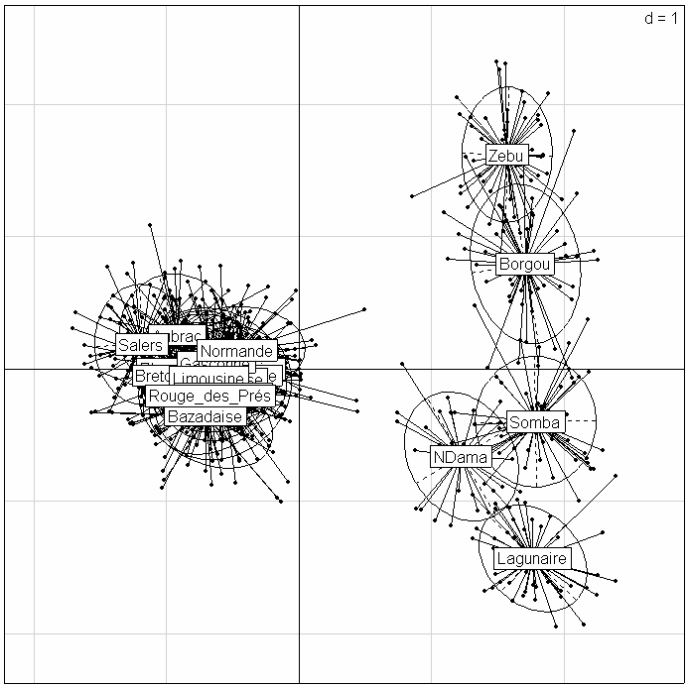


```
s.kde2d(bovpca$li)
s.kde2d(bovpca$li[bovpca$li[,1]>0.5,],add.p=T)
```



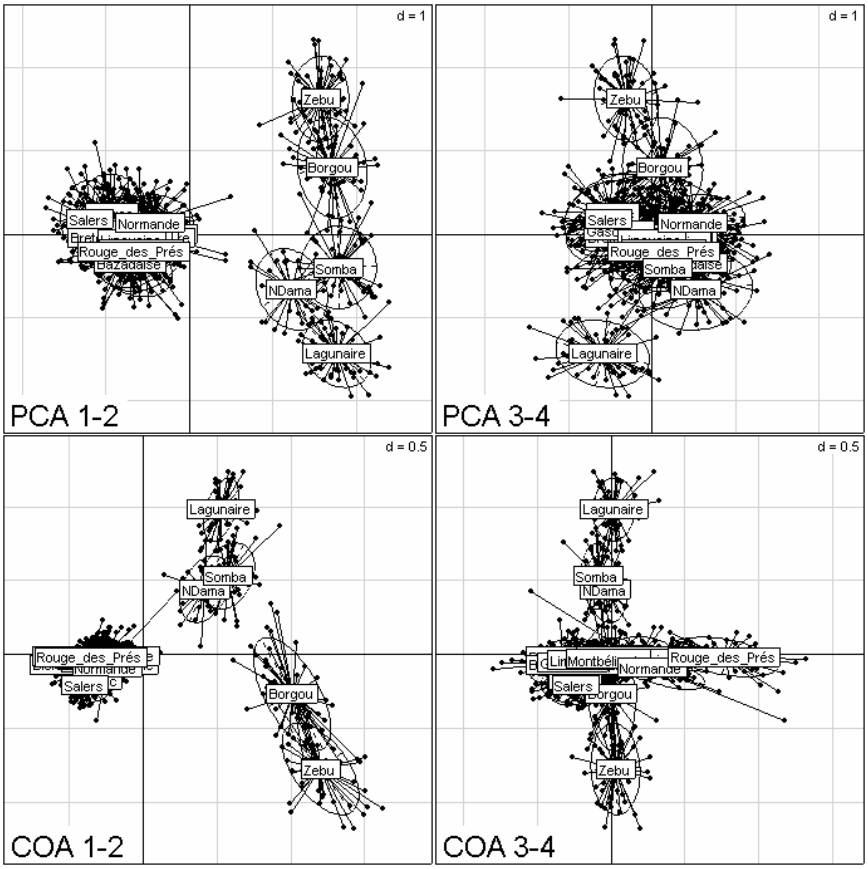

Attention : les différences de densité locale sont telles entre les deux parties du nuage que la procédure semble prise en défaut. En fait le facteur 1 a une telle importance (bovins africains, voir la première valeur propre) que l'analyse mélange plusieurs niveaux. Les deux familles de courbes de niveaux ne sont pas comparables.

```
s.class(bovpca$li,bov.race)
```



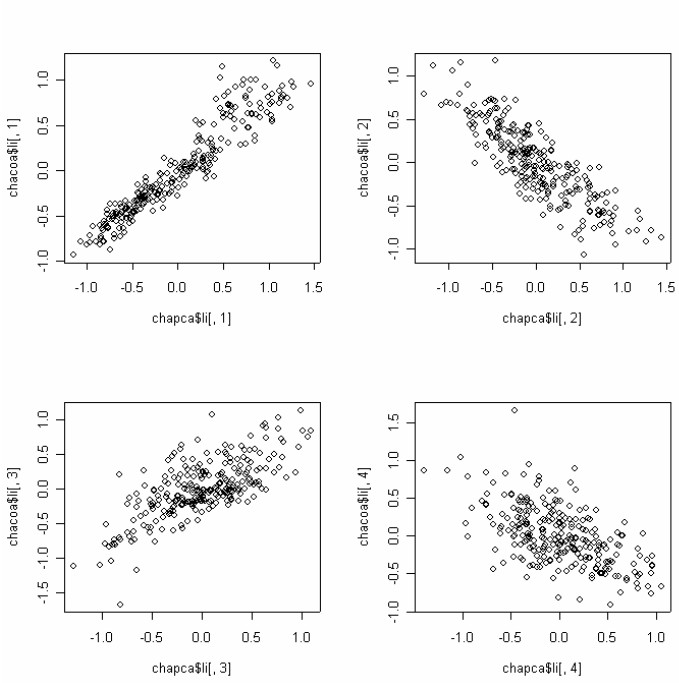

## 2.3. ACP contre AFC

C'est un vieux débat, un peu dépassé.




Quand on fait une AFC sur un tableau codé par individu en fréquences d'allèles, on n'obtient pas le même résultat. Ces résultats peuvent être sensiblement les mêmes ou présenter des nuances fortes qui ne se contredisent pas dans l'interprétation.

```
cor(chapca$li,chacoa$li)
         Axis1     Axis2   Axis3    Axis4
Axis1  0.94541 -0.154410  0.1312  0.05446
Axis2 -0.11826 -0.848931 -0.2557  0.04414
Axis3 -0.15334 -0.271270  0.7199 -0.18542
Axis4  0.08503  0.001785 -0.2010 -0.57411
Axis5 -0.03025 -0.072489  0.3031 -0.13686
```



En principe, l'AFC prend trop en compte les allèles rares et l'ACP n'en tient pas assez compte. On trouvera toujours des cas marginaux ou l'une des deux est préférable à l'autre. L'AFC est symétrique entre allèles et individus, mais ce n'est pas un argument : les loci font une typologie d'individus et non l'inverse. On est donc dans une logique d'analyse non symétrique des correspondances, mais à cause des pondérations uniformes dans les deux cas, l'ACP est une analyse des correspondances non symétriques. Une des meilleurs introduction est dans Kroonenberg and Lombardo (1999). Voir aussi Gimaret-Carpentier et al. (1998).

```
chansc=dudi.nsc(cha)
Select the number of axes: 4
 cor(chapca$li,chansc$li)
           Axis1      Axis2      Axis3      Axis4
Axis1  1.000e+00  1.906e-16 -1.517e-16 -8.574e-17
Axis2 -2.073e-16  1.000e+00  1.128e-16  2.056e-17
Axis3  4.314e-16  1.502e-16  1.000e+00  5.409e-15
Axis4  2.743e-17 -4.608e-17 -5.726e-15  1.000e+00
Axis5  1.451e-16  1.739e-16 -6.112e-16 -2.306e-15
```

*A contrario*, l'AFC du tableau est une AFC floue, très proche d'une Analyse des correspondances multiples (ACM), extension de l'ACM qui n'utilise que les codes disjonctifs complets : origine dans Chevenet et al. (Chevenet et al. 1994), dupliquée en génétique dans Guinand (1996).

```
chafca=dudi.fca(cha)
Select the number of axes: 4
 cor(chafca$li,chacoa$li)
          Axis1      Axis2      Axis3     Axis4
Axis1 1.000e+00  2.909e-16 -9.770e-17 2.766e-17
Axis2 2.768e-16  1.000e+00  2.147e-15 5.682e-16
Axis3 2.498e-17 -2.435e-15  1.000e+00 1.093e-15
Axis4 3.824e-17 -5.896e-16 -1.223e-15 1.000e+00
```

Si on cherche la principale différence entre les deux versions, on la trouvera dans la contrainte qui pèse sur les scores des colonnes. En ACP¨, on peut facilement démontrer que les coordonnées des colonnes sont centrées **par bloc (locus)** pour la pondération uniforme :

```
tapply(chapca$co[,1],attr(cha,"col.num"),sum)
    Fca124     Fca126      Fca26      Fca31      Fca43      Fca58     Fca668
 4.037e-17 -2.602e-18  1.952e-17 -3.578e-17 -5.557e-17  4.269e-17 -2.784e-17
     Fca77      Fca78       Fca8      Fca80      Fca90      Fca96
 1.821e-17 -5.898e-17  1.286e-16 -5.215e-17  5.760e-17 -3.914e-17
```

C'est faux pour l'AFC :

```
 tapply(chacoa$co[,1],attr(cha,"col.num"),sum)
 Fca124  Fca126   Fca26   Fca31   Fca43   Fca58  Fca668   Fca77   Fca78    Fca8
 4.8026  2.1823  3.8859  4.4784 -2.4158 -0.5806  0.6138 -0.7315  2.2793  1.0042
  Fca80   Fca90   Fca96
 0.2757  3.6227 -2.1754
```

Mais la même propriété est vraie avec la pondération issue des fréquences alléliques marginales :

```
 tapply(chacoa$co[,1]*chacoa$cw,attr(cha,"col.num"),sum)
    Fca124     Fca126      Fca26      Fca31      Fca43      Fca58     Fca668
 4.150e-20  1.057e-18 -1.206e-18  4.302e-18  3.185e-18 -4.320e-19  1.050e-18
     Fca77      Fca78       Fca8      Fca80      Fca90      Fca96
-3.632e-18 -1.247e-18 -1.595e-17 -1.355e-18  8.294e-18  5.773e-18
```

En bref, l'ACP centrée, la plus simple des versions d'analyse d'un tableau à deux normes diagonales, dans de nombreux cas, semble suffire.

# 3. Mesurer la valeur des marqueurs

Un marqueur est d'abord caractérisé par le nombre de formes alléliques rencontrées dans ce locus (polymorphisme).

```
chan=attr(cha,"col.blocks")
names(chan)=attr(cha,"loc.names")
chan
```

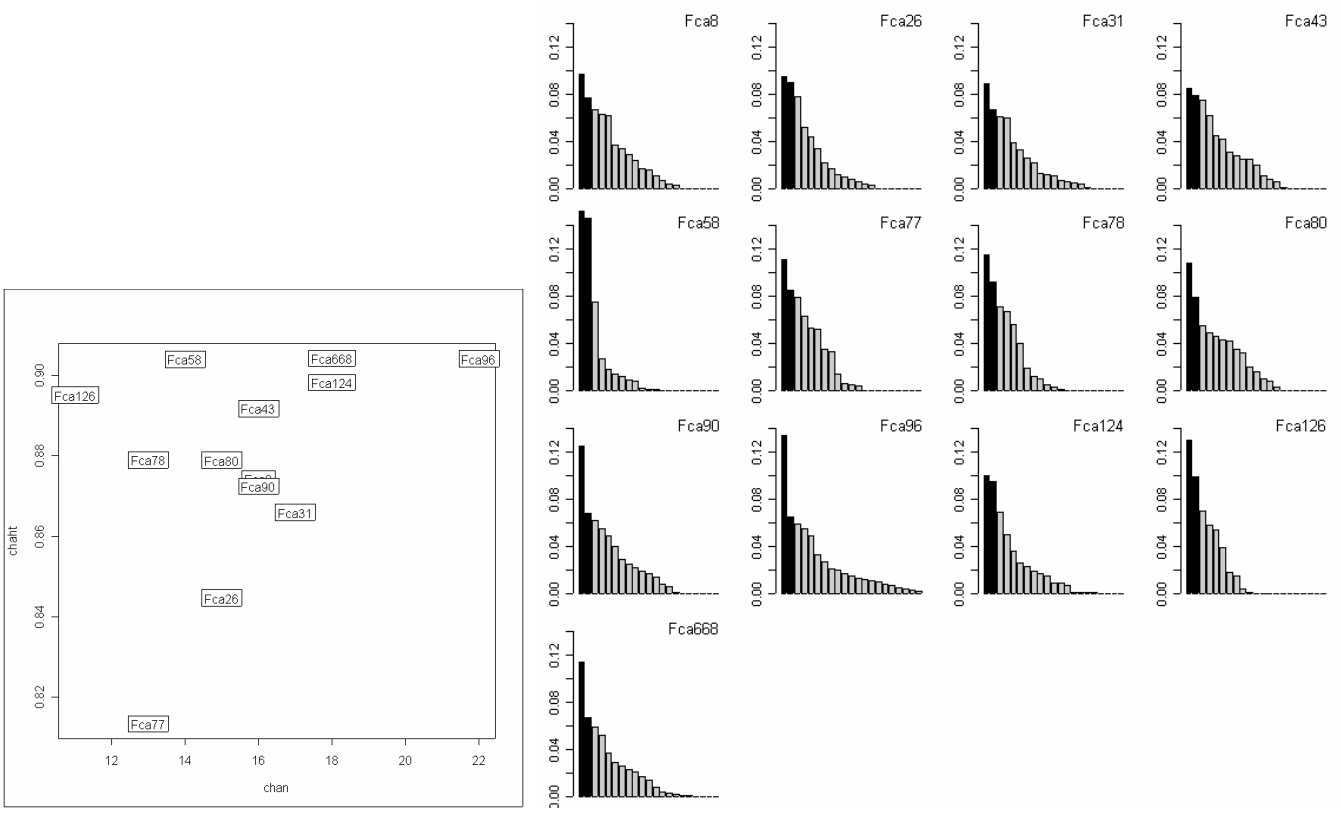
```
  Fca8  Fca26  Fca31  Fca43  Fca58  Fca77  Fca78  Fca80  Fca90  Fca96 Fca124
    16     15     17     16     14     13     13     15     16     22     18
Fca126 Fca668
    11     18
```

On utilise ensuite le taux d'hétérozygotie théorique :

```
chaht=tapply(attr(cha,"col.freq"),attr(cha,"col.num"),function(x) sum(x*(1-x)))
names(chaht)=attr(cha,"loc.names")
chaht
```

```
  Fca8  Fca26  Fca31  Fca43  Fca58  Fca77  Fca78  Fca80  Fca90  Fca96 Fca124
0.8743 0.8446 0.8659 0.8918 0.9040 0.8134 0.8790 0.8789 0.8725 0.9041 0.8982
Fca126 Fca668
0.8951 0.9042
```

```
plot(chan,chaht)
s.label(cbind.data.frame(chan,chaht),add.p=T)
```

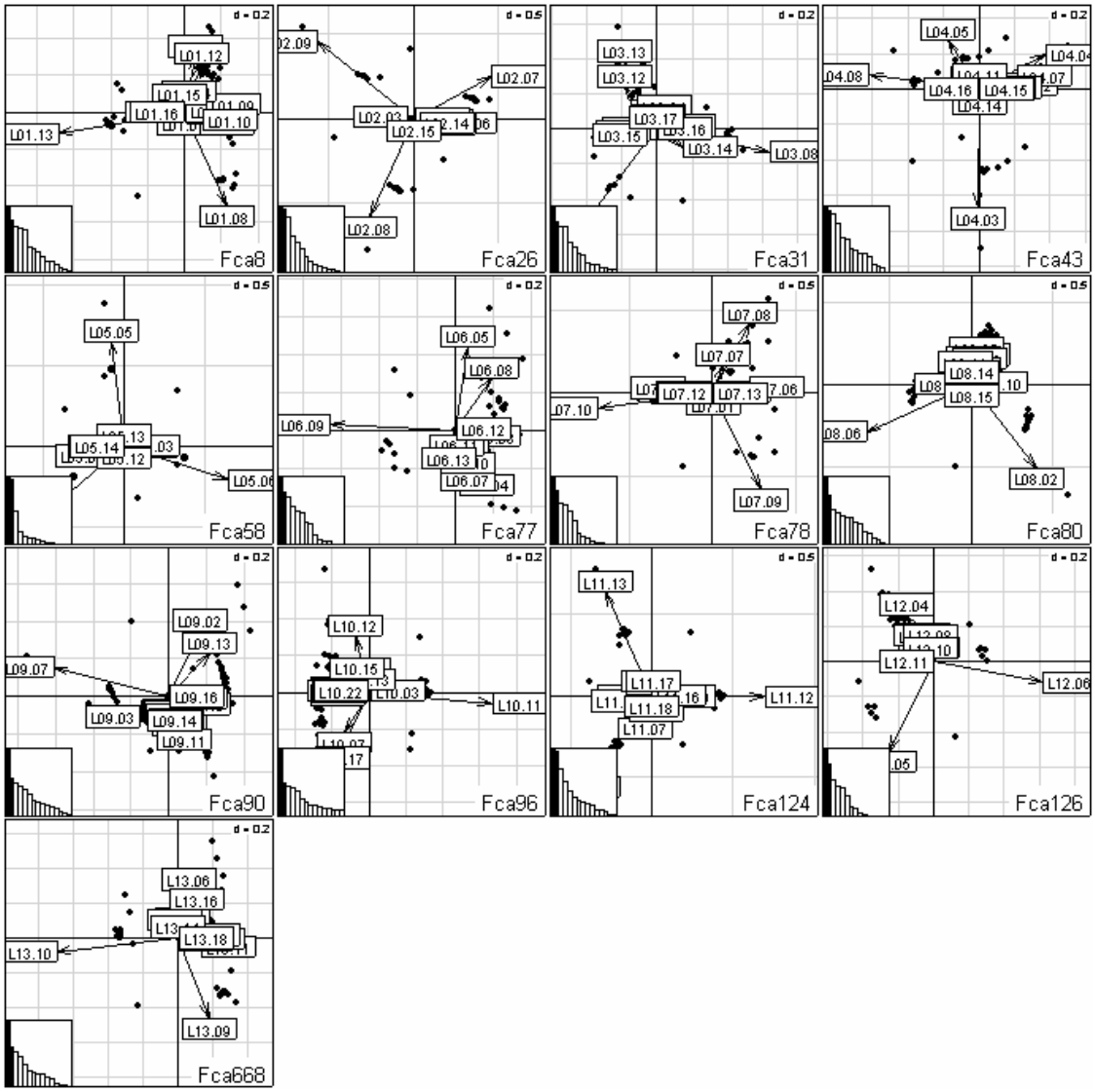



Ceci renseigne peu sur la cohérence entre ce que peut faire un marqueur et la typologie assurée globalement par l'ensemble des marqueurs, cohérence qu'on peut appeler *valeur typologique*. Il suffit de considérer alors le k-tableau associé à l'ACP :

```
kt=ktab.data.frame(chapca$tab, attr(cha,"col.blocks"))
tab.names(kt)=attr(cha,"loc.names")
plot(sepan(kt)) # ci-dessus à droite
```

On voit que l'ACP d'un sous-tableau associé à un seul locus a des propriétés variables. La figure suivante suggère la question d" la cohérence typologique : elle contient 17 biplots d'ACP sans relations entre eux.

```
kplot(sepan(kt),clab.r=0)
```



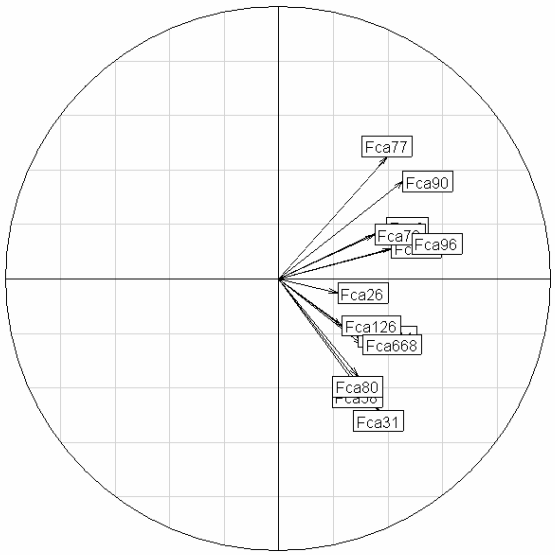

Quelles relations peut-il s'établir entre ces typologies ? Il n'est pas question ici de positionner un individu de plusieurs façon, la quantité d'information par individu au niveau d'un locus étant relativement faible, mais de comparer des capacités à disperser des individus. C'est typiquement un problème pour STATIS (Lavit 1988, Lavit et al. 1994).

```
chastatis=statis(chakt)
Select the number of axes: 3
```

```
round(chastatis$RV,2)
     L01  L02  L03  L04  L05  L06  L07  L08  L09  L10  L11  L12  L13
L01 1.00 0.02 0.06 0.07 0.04 0.08 0.06 0.04 0.09 0.09 0.04 0.04 0.05
L02 0.02 1.00 0.02 0.04 0.03 0.03 0.04 0.03 0.03 0.04 0.03 0.03 0.04
L03 0.06 0.02 1.00 0.05 0.09 0.03 0.06 0.10 0.04 0.07 0.06 0.04 0.05
L04 0.07 0.04 0.05 1.00 0.05 0.07 0.07 0.04 0.09 0.10 0.04 0.05 0.04
L05 0.04 0.03 0.09 0.05 1.00 0.02 0.02 0.04 0.04 0.07 0.03 0.03 0.05
L06 0.08 0.03 0.03 0.07 0.02 1.00 0.06 0.04 0.13 0.10 0.03 0.03 0.03
L07 0.06 0.04 0.06 0.07 0.02 0.06 1.00 0.03 0.07 0.08 0.03 0.04 0.04
L08 0.04 0.03 0.10 0.04 0.04 0.04 0.03 1.00 0.04 0.05 0.03 0.03 0.04
L09 0.09 0.03 0.04 0.09 0.04 0.13 0.07 0.04 1.00 0.11 0.05 0.03 0.05
L10 0.09 0.04 0.07 0.10 0.07 0.10 0.08 0.05 0.11 1.00 0.06 0.04 0.06
L11 0.04 0.03 0.06 0.04 0.03 0.03 0.03 0.03 0.05 0.06 1.00 0.03 0.06
L12 0.04 0.03 0.04 0.05 0.03 0.03 0.04 0.03 0.03 0.04 0.03 1.00 0.03
L13 0.05 0.04 0.05 0.04 0.05 0.03 0.04 0.04 0.05 0.06 0.06 0.03 1.00
```

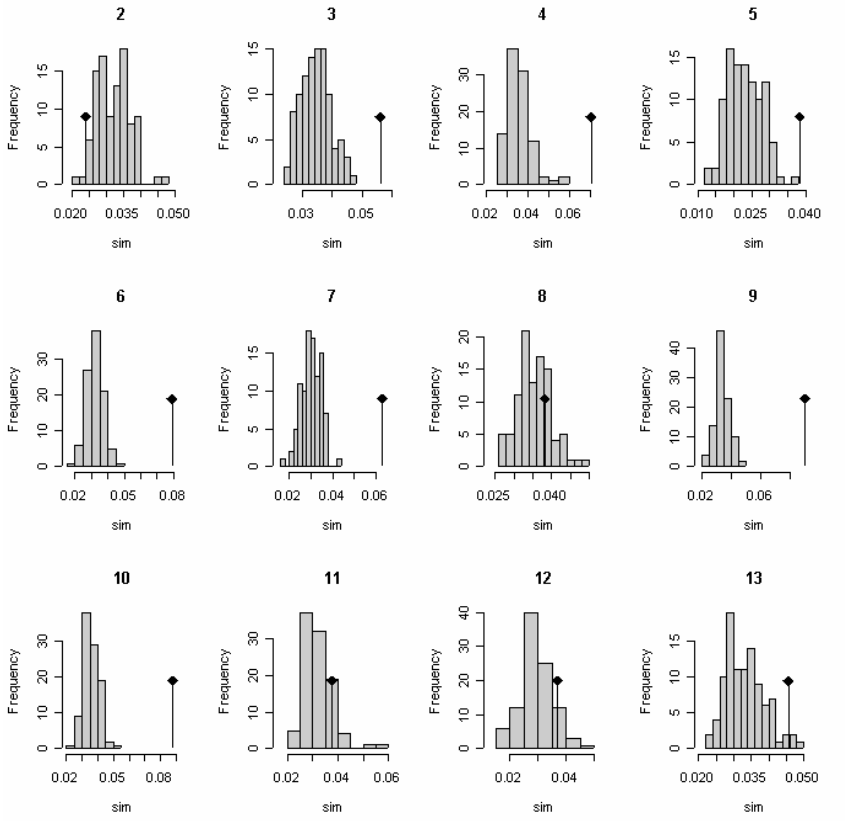
Ces coefficients ne dépassent pas 0.13, ce qui est déjà un résultat extraordinaire. La faiblesse de la représentation de l'image euclidienne des opérateurs est impressionnante :

```
s.corcircle(chastatis$RV.coo,lab=attr(cha,"loc.names"))
```




Il est pourtant assuré que ces valeurs sont hautement significative à partir de 0.06 (test de permutation (Heo and Gabriel 1998)) :

```
chastatis$RV[1,2:13]
    L02     L03     L04     L05     L06     L07     L08     L09     L10     L11
0.02409 0.05611 0.07052 0.03840 0.07931 0.06296 0.03822 0.09020 0.08767 0.03763
    L12     L13
0.03700 0.04560
par(mfrow=c(3,4))
for(k in 2:13) plot(RV.rtest(chakt[[1]], chakt[[k]]),main = as.character(k))
```

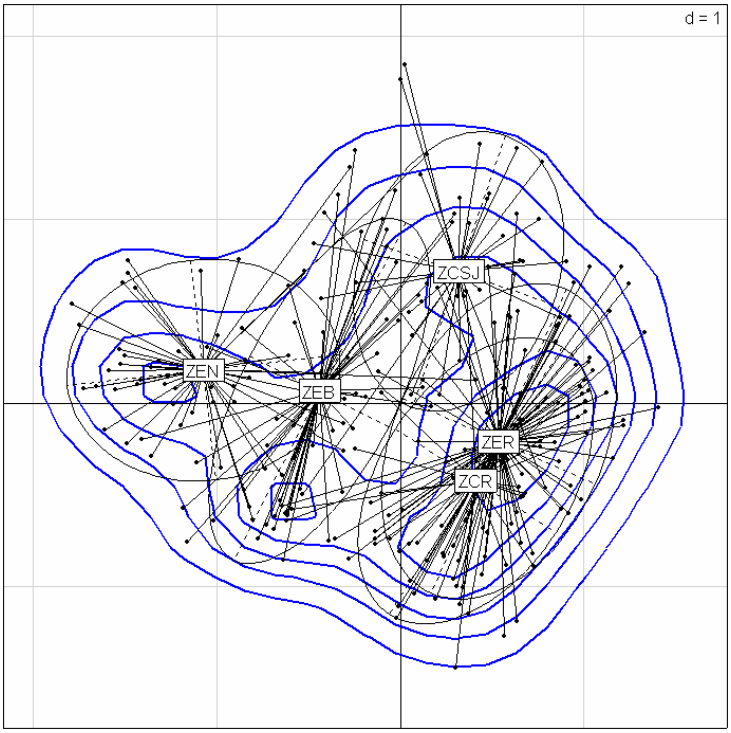



```
inertia=unlist(lapply(1:13,function(k) sum(svd(chakt[[k]])$d^2)))
```

```
cos2=chastatis$cos2
poids=chastatis$RV.tabw
index=cbind.data.frame(chan,chaht,inertia,poids,cos2)
index
```

| | chan | chaht | inertia | poids | cos2 | | chan | chaht | inertia | poids | cos2 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Fca8 | 16 | 0.8743 | 142.0 | 0.3078 | 0.20641 | Fca80 | 15 | 0.8789 | 144.2 | 0.2271 | 0.07092 |
| Fca26 | 15 | 0.8446 | 123.2 | 0.1691 | 0.03609 | Fca90 | 16 | 0.8725 | 141.6 | 0.3566 | 0.39408 |
| Fca31 | 17 | 0.8659 | 118.6 | 0.2866 | 0.18918 | Fca96 | 22 | 0.9041 | 149.8 | 0.3814 | 0.46629 |
| Fca43 | 16 | 0.8918 | 140.6 | 0.3231 | 0.22494 | Fca124 | 18 | 0.8982 | 125.6 | 0.2264 | 0.13531 |
| Fca58 | 14 | 0.9040 | 122.9 | 0.2260 | 0.18171 | Fca126 | 11 | 0.8951 | 128.0 | 0.1811 | 0.05820 |
| Fca77 | 13 | 0.8134 | 142.4 | 0.3108 | 0.26552 | Fca668 | 18 | 0.9042 | 124.7 | 0.2414 | 0.13869 |
| Fca78 | 13 | 0.8790 | 129.7 | 0.2756 | 0.18125 | | | | | | |

`chan` est le nombre de variable, `chaht` l'hétérozygotie, `inertia` la variance totale (à une constante près), `poids` le coefficient attribué au tableau dans la constitution du compromis et `cos2` l'ajustement entre la structure du tableau et celle du compromis. Les tableaux par locus sont sensiblement de même importance et sensiblement indépendants les uns des autres. Cela veut dire qu'il y a peu de différences entre l'analyse du tableau global (chaque sous-tableau compte à égalité) et STATIS (chaque sous-tableau compte pour son poids respectif). La nuance est *ici* sans intérêt :




```
s.kde2d(chastatis$C.li)
s.class(chastatis$C.li,zone,add.p=T)
```

L'analyse exploratoire des données génétiques sur marqueurs microsatellites n'est donc pas privée d'intérêt. Mais ces données ont des propriétés très particulières qui demandent quelques ajustements.

# 4. Références

Arnaiz-Villena, A., N. Elaiwa, C. Silvera, A. Rostom, J. Moscoso, E. Gómez-Casado, L. Allende, P. Varela, and J. Martínez-Laso. 2001. The origin of Palestinians and their genetic relatedness with other Mediterranean populations. Human Immunology **62**:889-900.

Chevenet, F., S. Dolédec, and D. Chessel. 1994. A fuzzy coding approach for the analysis of long-term ecological data. Freshwater Biology **31**:295-309.

Gimaret-Carpentier, C., D. Chessel, and J. P. Pascal. 1998. Non-symmetric correspondence analysis: an alternative for community analysis with species occurrences data. Plant Ecology **138**:97-112.

Guinand, B. 1996. Use of a multivariate model using allele frequency distributions to analyse patterns of genetic differentiation among populations. Biological Journal of the Linnean Society **58**:173-195.

Heo, M., and K. R. Gabriel. 1998. A permutation test of association between configurations by means of the RV coefficient. Communications in Statistics - Simulation and Computation **27**:843-856.

Kroonenberg, P. M., and R. Lombardo. 1999. Nonsymmetric correspondence analysis: a tool for analysing contingency tables with a dependence structure. Multivariate Behavioral Research **34**:367-396.

Lavit, C. 1988. Analyse conjointe de tableaux quantitatifs. Masson, Paris.

Lavit, C., Y. Escoufier, R. Sabatier, and P. Traissac. 1994. The ACT (Statis method). Computational Statistics and Data Analysis **18**:97-119.

Leon, F. J., J. Dallas, B. Chatain, M. Canone, J. Versini, and F. Bonhomme. 1995. Development and use of microsatellite markers in sea bass, *Dicentrarchus labrax* (Perciformes, Serranidae). Marine Molecular Biology and Biotechnology **4**:62-68.

Maudet, C., C. Miller, B. Bassano, C. Breitenmoser-Wursten, D. Gauthier, G. Obexer-Ruff, J. Michallet, P. Taberlet, and G. Luikart. 2002. Microsatellite DNA and recent statistical methods in wildlife conservation management: applications in Alpine ibex [*Capra ibex (ibex)*]. Molecular Ecology **11**:421-436.

Menozzi, P., A. Piazza, and L. L. Cavalli-Sforza. 1978. Synthetic maps of human gene frequencies in europeans. Science **201**:786-792.

Naciri, M., C. Lemaire, P. Borsa, and F. Bonhomme. 1999. Genetic study of the Atlantic / Mediterranean transition in sea bass, Dicentrarchus labrax. Journal of Heredity **90**:591-596.

Orth, A., T. Adama, W. Din, and F. Bonhomme. 1998. Hybridation naturelle entre deux sous espèces de souris domestique Mus musculus domesticus et Mus musculus castaneus près de Lake Casitas (Californie). Genome **41**:104-110.

Smouse, P. E., and R. Peakall. 1999. Spatial autocorrelation analysis of individual multiallele and multilocus genetic structure. Heredity **82**:561-573.