Practical course using the  software

# Multivariate analysis of genetic markers as a tool to explore the genetic diversity: some examples

Thibaut Jombart

09/09/09

**Abstract**

This practical course aims at illustrating some possible applications of multivariate analyses to genetic markers data, using the `R` software [14]. Although a basic knowledge of the R language is assumed, most necessary commands are provided, so that coding should not be an obstacle. Two exercises are proposed, which go through different topics in genetic data analysis, respectively the study of spatial genetic structures, and the coherence of information coming from different markers. After going through the first section ('Let's start'), you should feel free to get to the exercise you want, as these are meant to be independent. This practical course uses mostly the *adegenet* [10] and *ade4* packages [4, 7, 6], but others like *adehabitat* [2, 1], *genetics* [15] and *hierfstat* [8] are also used.

# Contents

# 1 Let's start

## 1.1 Loading the packages

Before going further, we shall make sure that all we need is installed on the computer. Launch R, and make sure that the version being used is greater than 2.8.1 by typing:

```
> R.version.string
```

```
[1] "R version 2.9.2 (2009-08-24)"
```

The next thing to do is check that relevant packages are installed. To load an installed package, use `library` instruction; for instance:

```
> library(adegenet)
```

loads *adegenet* if it is installed (and issues an error otherwise). To get the version of a package, use:

```
> packageDescription("adegenet", fields = "Version")
```

```
[1] "1.2-3"
```

*adegenet* version should read 1.2-3.

In case a package would not be installed, you can install it by using `install.packages`. To install all the required dependencies, specify `dep=TRUE`. For instance, the following instruction should install *adegenet* with all its dependencies (it can take up to a few minutes, so don't run it unless *adegenet* is not installed):

```
> install.packages("ape", dep = TRUE)
```

Using the previous instructions, load (and install if required) the packages *adegenet*, *ade4*, *spdep*, *genetics*, and *hierfstat*.

## 1.2 How to get information?

There are several ways of getting information about R in general, and about *adegenet* in particular. Function `help.search` is used to look for help on a given topic. For instance:

```
> help.search("Hardy-Weinberg")
```

replies that there is a function `HWE.test.genind` in the *adegenet* package, and functions `HWE.chisq`,`HWE.exact` and `HWE.test` in *genetics*. To get help for a given function, use `?foo` where 'foo' is the function of interest. For instance:

```
> `?`(spca)
```

will open the manpage of the spatial principal component analysis [11]. At the end of a manpage, an 'example' section often shows how to use a function. This can be copied and pasted to the console, or directly executed from the console using `example`. For further questions concerning R, the function `RSiteSearch` is a powerful tool to make an online research using keywords in R's archives (mailing lists and manpages).

*adegenet* has a few extra documentation sources. Information can be found from the website (`http://adegenet.r-forge.r-project.org/`), in the 'documents' section, including two tutorials and a manual which includes all manpages of the package. To open the website from R, use:

```
> adegenetWeb()
```

The same can be done for tutorials, using `adegenetTutorial` (see manpage to choose the tutorial to open).

You will also find a listing of the main functions of the package typing:

```
> `?`(adegenet)
```

Note that you can also browse help pages as html pages, using:

```
> help.start()
```

```
If '/usr/bin/firefox' is already running, it is *not* restarted, and
    you must switch to its window.
Otherwise, be patient ...
```

To go to the *adegenet* page, click 'packages', 'adegenet', and 'adegenet-package'.

Lastly, several mailing lists are available to find different kinds of information on R; to name a few:

- R-help (`https://stat.ethz.ch/mailman/listinfo/r-help`): general questions about R
- R-sig-genetics (`https://stat.ethz.ch/mailman/listinfo/r-sig-genetics`): genetics in R
- adegenet forum (`https://lists.r-forge.r-project.org/cgi-bin/mailman/listinfo/adegenet-forum`): adegenet and multivariate analysis of genetic markers

# 2 Spatial genetic structure of the chamois in the Bauges mountains

The chamois (*Rupicapra rupicapra*) is a conserved species in France. The Bauges mountains is a protected area in which the species has been recently

studied. One of the most important questions for conservation purpose relates to whether individuals from this area form a single reproductive unit, or whether they are structured into sub-groups, and if so, what causes are likely to cause this structuring.

While field observations are very scarce and do not allow to answer this question, genetic data can be used to tackle the issue, as departure from panmixia should result in genetic structure. The dataset *rupica* contains 335 georeferenced genotypes of Chamois from the Bauges mountains for 9 microsatellite markers, which we propose to analyse in this exercise.

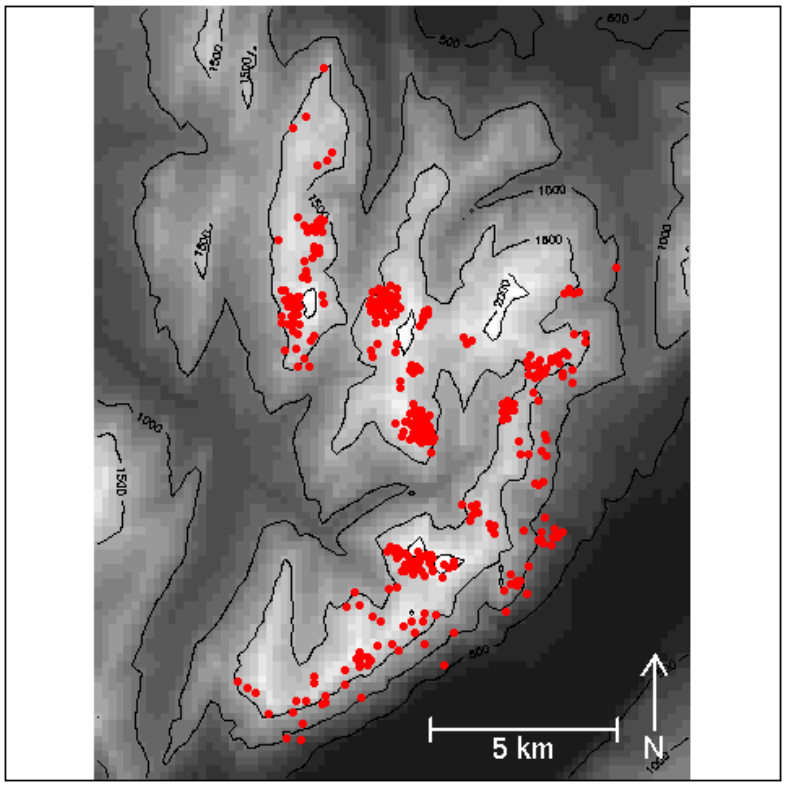
## 2.1 An overview of the data

We first load the data:

```
> data(rupica)
> rupica

   ######################
   ### Genind object ###
   ######################
- genotypes of individuals -

S4 class:  genind
@call: NULL

@tab:  335 x 55 matrix of genotypes

@ind.names: vector of  335 individual names
@loc.names: vector of  9 locus names
@loc.nall: number of alleles per locus
@loc.fac: locus factor for the  55 columns of @tab
@all.names: list of  9 components yielding allele names for each locus
@ploidy:  2
@type:  codom

Optionnal contents:
@pop:  - empty -
@pop.names:  - empty -

@other: a list containing: xy  mnt  showBauges
```

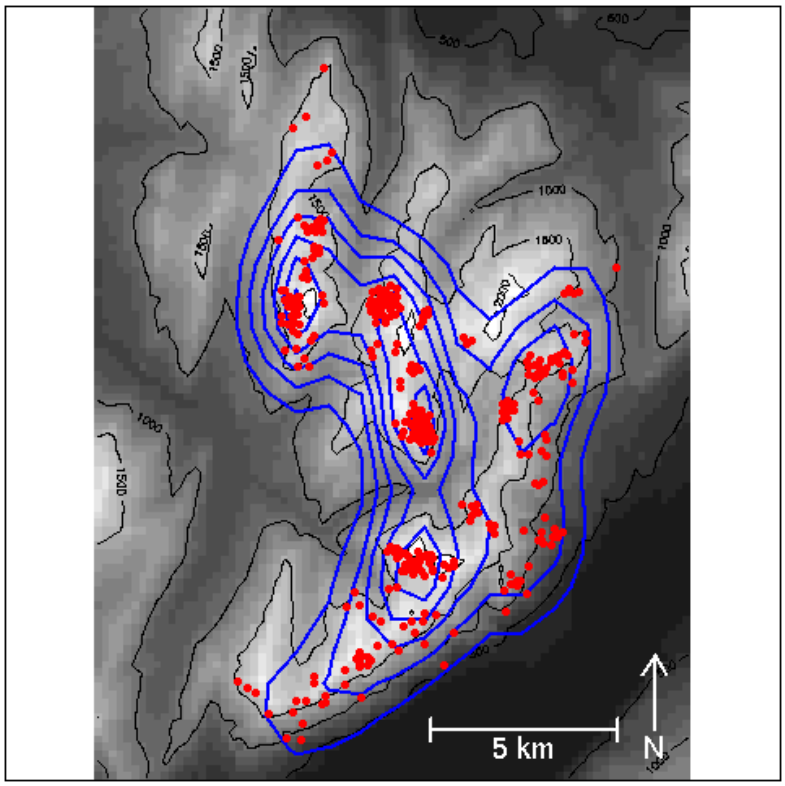
`rupica` is a typical `genind` object, which is the class of objects storing genotypes (as opposed to population data) in *adegenet*. `rupica` also contains topographic information about the sampled area, which can be displayed by calling `rupica$other$showBauges`. For instance, the spatial distribution of the sampling can be displayed as follows:

```
> rupica$other$showBauges()
> points(rupica$other$xy, col = "red", pch = 20)
```




This spatial distribution is clearly not random, but arranged into loose clusters; this can be confirmed by superimposing a kernel density curve (in blue) on the previous figure:

```
> rupica$other$showBauges()
> s.kde2d(rupica$other$xy, add.plot = TRUE)
> points(rupica$other$xy, col = "red", pch = 20)
```




However, this spatial clustering is not strong enough to assign safely all genotypes to a given geographic group. Hence, further analyses would have to be performed on individuals rather than groups of individuals.

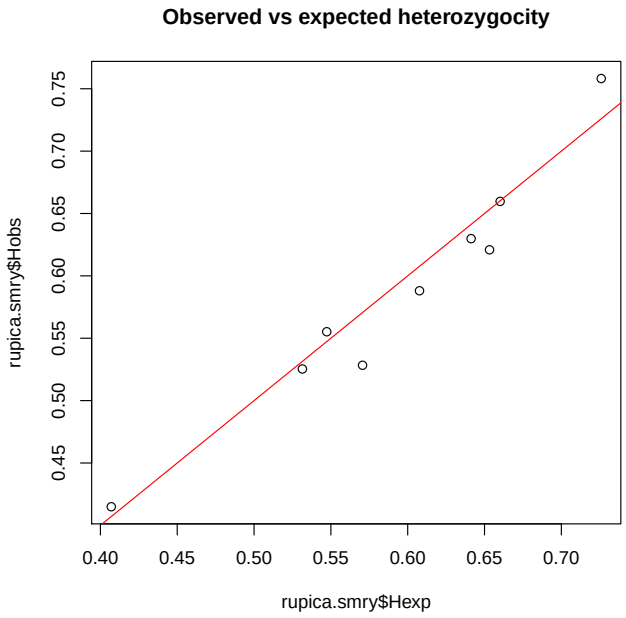
## 2.2 Standard analyses

As a prior clustering of genotypes is not known, we cannot employ usual $F_{ST}$-based approaches to detect genetic structuring. However, genetic structure could still result in a deficit of heterozygocity. The `summary` of `genind` objects provides expected and observed heterozygocity for each locus, which allows for a comparison:

```
> rupica.smry <- summary(rupica)

 # Total number of genotypes:  335

 # Population sample sizes:

335

 # Number of alleles per locus:
L1 L2 L3 L4 L5 L6 L7 L8 L9
 7 10  7  6  5  5  6  4  5

 # Number of alleles per population:
 1
```

```
55

 # Percentage of missing data:
[1] 0

 # Observed heterozygosity:
       L1        L2        L3        L4        L5        L6        L7        L8
0.5880597 0.6208955 0.5253731 0.7582090 0.6597015 0.5283582 0.6298507 0.5552239
       L9
0.4149254

 # Expected heterozygosity:
       L1        L2        L3        L4        L5        L6        L7        L8
0.6076769 0.6532517 0.5314591 0.7259657 0.6601604 0.5706082 0.6412742 0.5473112
       L9
0.4070709
```

```
> plot(rupica.smry$Hexp, rupica.smry$Hobs, main = "Observed vs expected heterozygocity")
> abline(0, 1, col = "red")
```



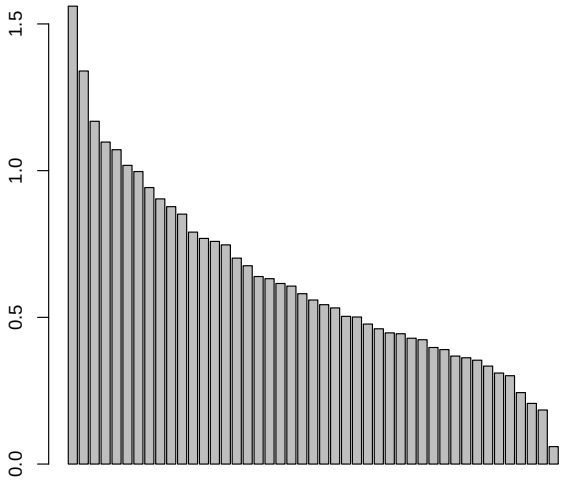

The red line indicate identity between both quantities. What can we say about heterozygocity in this population? The following test provides further insights to answer this question:

```
> t.test(rupica.smry$Hexp, rupica.smry$Hobs, paired = TRUE, var.equal = TRUE)

        Paired t-test

data:  rupica.smry$Hexp and rupica.smry$Hobs
t = 0.9461, df = 8, p-value = 0.3718
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
```

```
 -0.01025068  0.02451318
sample estimates:
mean of the differences
             0.00713125
```

We can seek a global picture of the genetic diversity among genotypes using a Principal Component Analysis (PCA, [13, 9], dudi.pca in ade4 package). The analysis is performed on a table of standardised alleles frequencies, obtained by scaleGen:

```
> rupica.X <- scaleGen(rupica, method = "binom")
> rupica.pca1 <- dudi.pca(rupica.X, cent = FALSE, scale = FALSE)
```



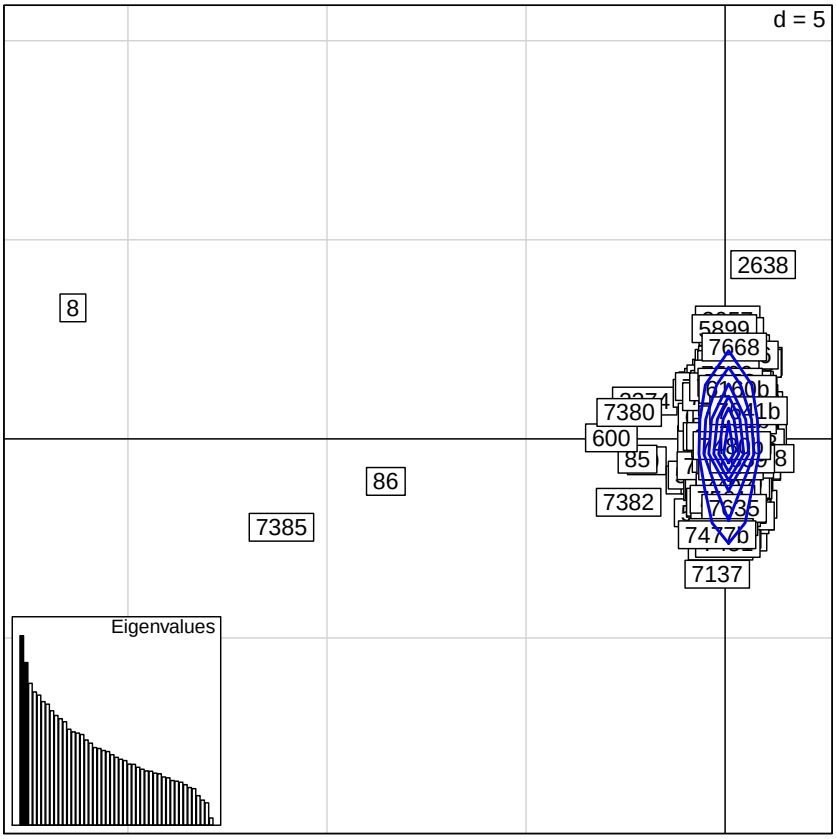

The function dudi.pca displays a barplot of eigenvalues and asks for a number of retained principal components. Eigenvalues represent the amount of genetic diversity (as measured by the multivariate method being used) represented by each principal component. An abrupt decrease in eigenvalues is likely to indicate the boundary between true patterns and non-interpretable structures. In this case, we shall examin the first two principal components (though nothing really clear emerges from the eigenvalues).

```
> rupica.pca1

Duality diagramm
class: pca dudi
$call: dudi.pca(df = rupica.X, center = FALSE, scale = FALSE, scannf = FALSE,
```

```
    nf = 2)

$nf: 2 axis-components saved
$rank: 45
eigen values: 1.561 1.34 1.168 1.097 1.071 ...
  vector length mode    content
1 $cw    55     numeric column weights
2 $lw    335    numeric row weights
3 $eig   45     numeric eigen values

  data.frame nrow ncol content
1 $tab       335  55   modified array
2 $li        335  2    row coordinates
3 $l1        335  2    row normed scores
4 $co        55   2    column coordinates
5 $c1        55   2    column normed scores
other elements: cent norm
```

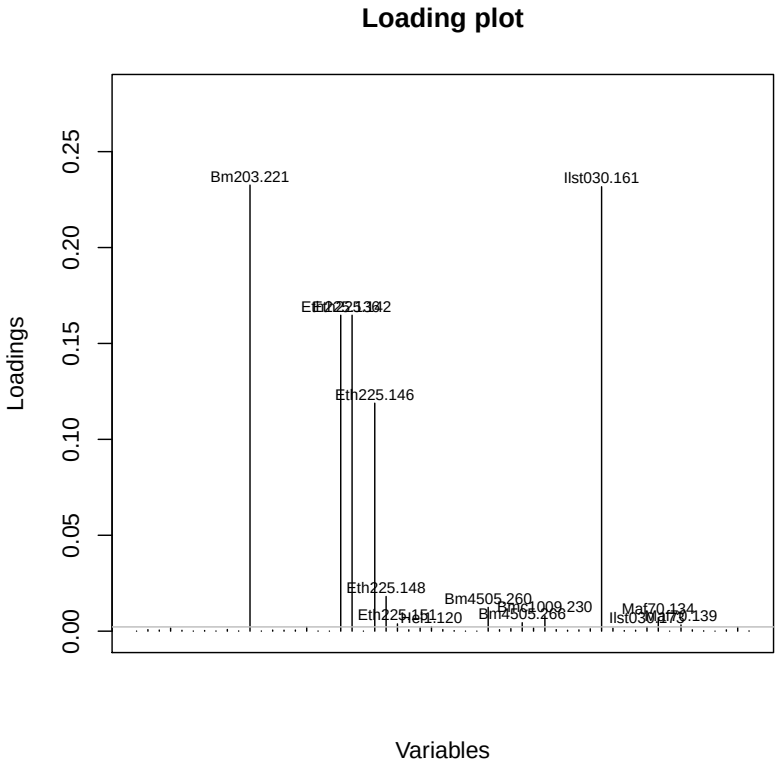
A `dudi` object contains various information; in the case of PCA, principal axes (loadings), principal components (synthetic variable), and eigenvalues are respectively stored in `$c1`, `$li`, and `$eig` slots. The function `s.label` can be used to display to two first components; a kernel density (`s.kde2d`) is used for a better assessment of the distribution of the genotypes onto the principal axes:

```
> s.label(rupica.pca1$li)
> s.kde2d(rupica.pca1$li, add.p = TRUE, cpoint = 0)
> add.scatter.eig(rupica.pca1$eig, 2, 1, 2)
```




What can we say about the genetic diversity among these genotypes as inferred by PCA? The function `loadingplot` allows to visualize the contribution of each allele, expressed as squared loadings, for a given principal component. This figure then gives further clues about the revealed structure:

```
> loadingplot(rupica.pca1$c1^2)
```




We can get back to the genotypes for the concerned markers (e.g., Bm203) to check whether the highlighted genotypes are indeed uncommon. `truenames` extracts the table of allele frequencies from a `genind` object:

```
> X <- truenames(rupica)
> class(X)

[1] "matrix"

> dim(X)

[1] 335  55

> bm203.221 <- X[, "Bm203.221"]
> table(bm203.221)

bm203.221
                  0 0.00597014925373134                   0.5
                330                   1                     4
```

Only 4 genotypes possess one copy of this allele (the second result corresponds to a replaced missing data). Which individuals are they?

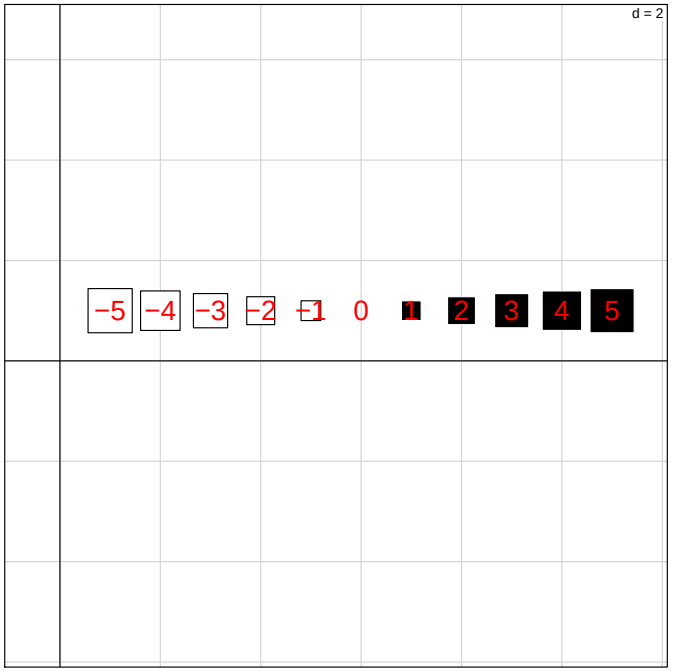
```
> rownames(X)[bm203.221 == 0.5]
```

```
  001    019    029    276
  "8"   "86"  "600" "7385"
```

Conclusion?

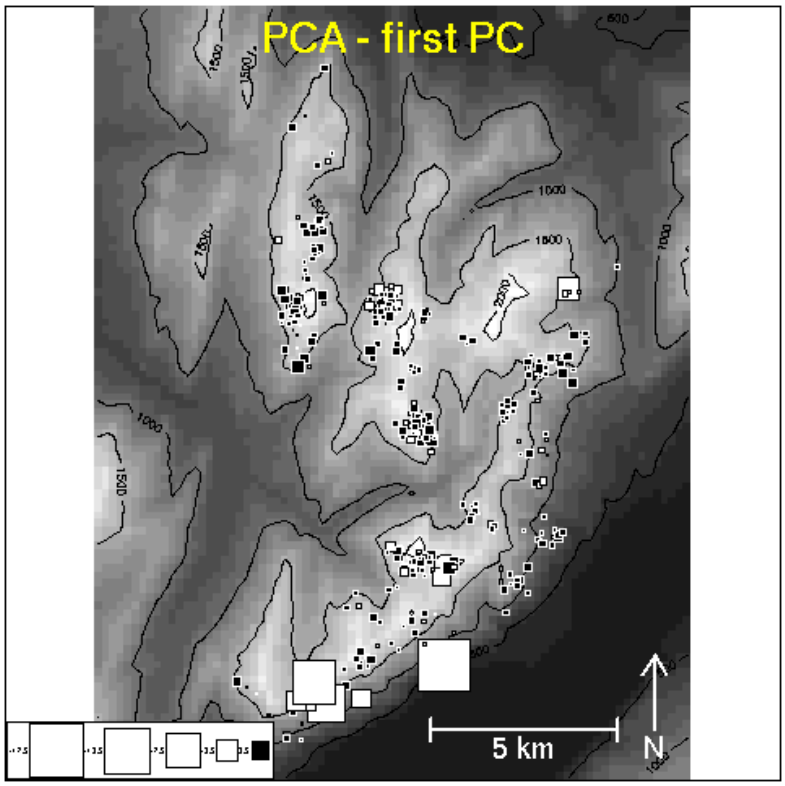
Just to make sure that this analysis shows no spatial pattern, we can map geographically the principal components. The function `s.value` is well-suited to do so, using black and white squares of variable size for positive and negative values. For instance:

```
> s.value(cbind(1:11, rep(1, 11)), -5:5, cleg = 0)
> text(1:11, rep(1, 11), -5:5, col = "red", cex = 1.5)
```

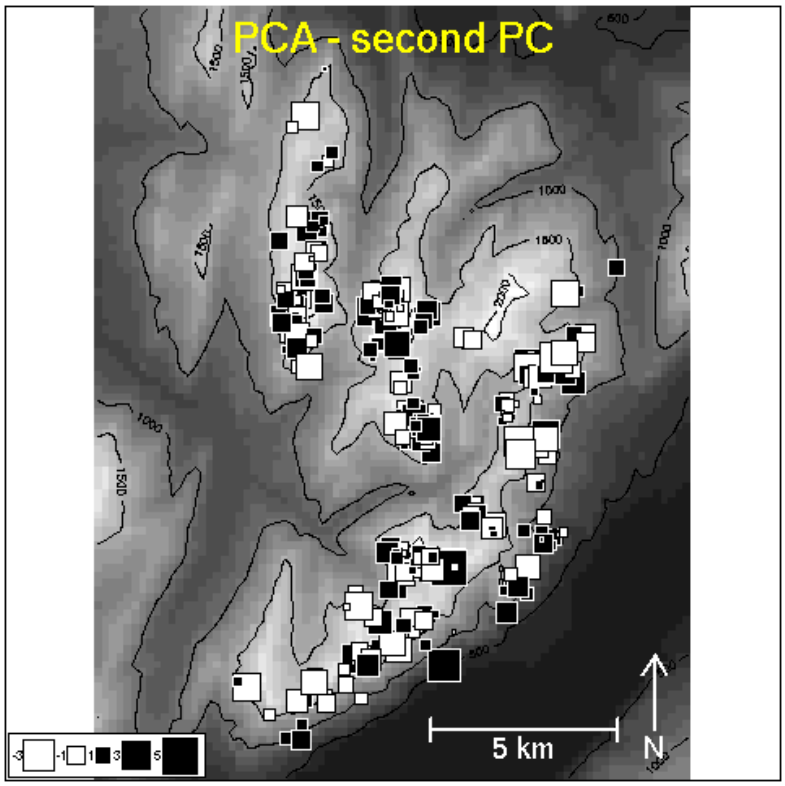



We can then apply this graphical representation to the first two principal components of the PCA:

```
> showBauges <- rupica$other$showBauges
> showBauges()
> s.value(rupica$other$xy, rupica.pca1$li[, 1], add.p = TRUE, cleg = 0.5)
> title("PCA - first PC", col.main = "yellow", line = -2, cex.main = 2)
```




```
> showBauges()
> s.value(rupica$other$xy, rupica.pca1$li[, 2], add.p = TRUE, csize = 0.7)
> title("PCA - second PC", col.main = "yellow", line = -2, cex.main = 2)
```




What can we say about spatial genetic structure as inferred by PCA?

## 2.3 spatial Principal Component Analysis

PCA did not reveal any kind of spatial genetic structure, but is not anyway meant to do so; most likely, it will fail to detect spatial genetic structures that are not associated with the strongest genetic differentiation. The spatial Principal Component Analysis (sPCA, [11]) has been developed to include spatial information in the analysis of genetic data. Although implemented in *adegenet*, sPCA needs spatial methods from the *spdep* package, which should thus be loaded:

```
> library(spdep)
```

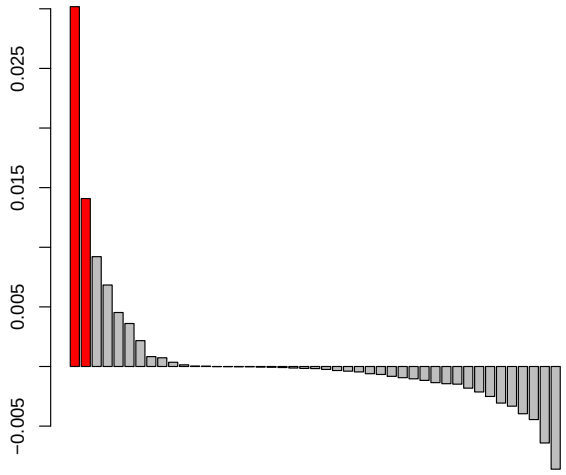
sPCA first requires the spatial proximities between genotypes to be modeled. The most convenient way to do so is to define geographic neighbours according to a given, preferably objective criterion. This amounts to constructing a spatial graph on which neighbours are linked by an edge. The function `chooseCN` proposes several spatial graphs (try `example(chooseCN)` for an example) that can be chosen interactively. In the case of the Chamois, we can use the intersection of home ranges as a criterion for neighbourhood; this amounts to considering as neighbours pairs of individuals separated by less than 2300 m.

Knowing that spatial coordinates of individuals are stored in `rupica$other$xy`, use `chooseCN` to build the corresponding spatial graph. Save the resulting object as `rupica.graph`; this object should look like this (displaying it may take a few seconds):

```
> rupica.graph

Neighbour list object:
Number of regions: 335
Number of nonzero links: 18018
Percentage nonzero weights: 16.05525
Average number of links: 53.78507


> plot(rupica.graph, rupica$other$xy)
> title("rupica.graph")
```


From there, we can use the `spca` function. Note that it would also be possible to specify the parameters of the spatial graph as arguments of `spca`.

```
> rupica.spca1 <- spca(rupica, cn = rupica.graph)
```




Like dudi.pca, spca displays a barplot of eigenvalues, but unlike in PCA, eigenvalues of sPCA can also be negative. This is because the criterion optimized by the analysis can have positive and negative values, corresponding respectively to positive and negative autocorrelation. In this case, only the principal components associated with the two first positive eigenvalues (in red) shall be retained.

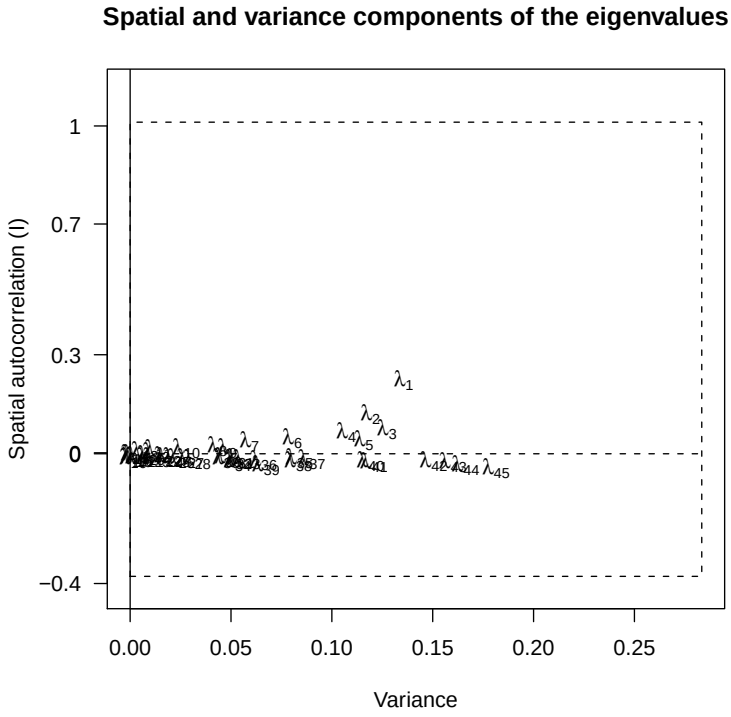
The printing of `spca` objects is more explicit than `dudi` objects, but named with the same conventions:

```
> rupica.spca1

        ########################################
        # spatial Principal Component Analysis #
        ########################################
class: spca
$call: spca(obj = rupica, cn = rupica.graph, scannf = FALSE, nfposi = 2,
    nfnega = 0)

$nfposi: 2 axis-components saved
$nfnega: 0 axis-components saved
Positive eigenvalues: 0.03018 0.01408 0.009211 0.006835 0.004529 ...
Negative eigenvalues: -0.008611 -0.006414 -0.004451 -0.003963 -0.003329 ...

  vector length mode    content
1 $eig   45     numeric eigenvalues

  data.frame nrow ncol content
1 $c1        55   2    principal axes: scaled vectors of alleles loadings
2 $li        335  2    principal components: coordinates of entities ('scores')
3 $ls        335  2    lag vector of principal components
4 $as        2    2    pca axes onto spca axes
```

```
$xy: matrix of spatial coordinates
$lw: a list of spatial weights (class 'listw')

other elements: NULL
```

Unlike usual multivariate analyses, eigenvalues of sPCA are composite: they measure both the genetic diversity (variance) and the spatial structure (spatial autocorrelation measured by Moran's $I$). This decomposition can also be used to choose which principal component to interprete. The function `screeplot` allows to display this information graphically:

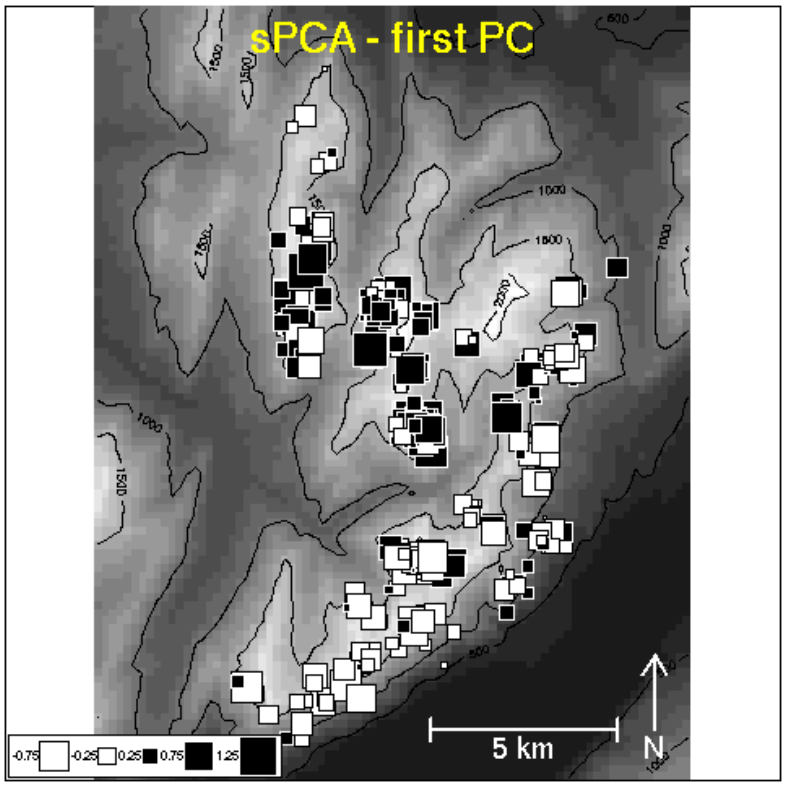
```
> screeplot(rupica.spca1)
```




While $\lambda_1$ indicates with no doubt a structure, the second eigenvalue, $\lambda_2$ is less clearly distinct from the successive values. Thus, we shall keep in mind this uncertainty when interpreting the second principal component of the analysis.

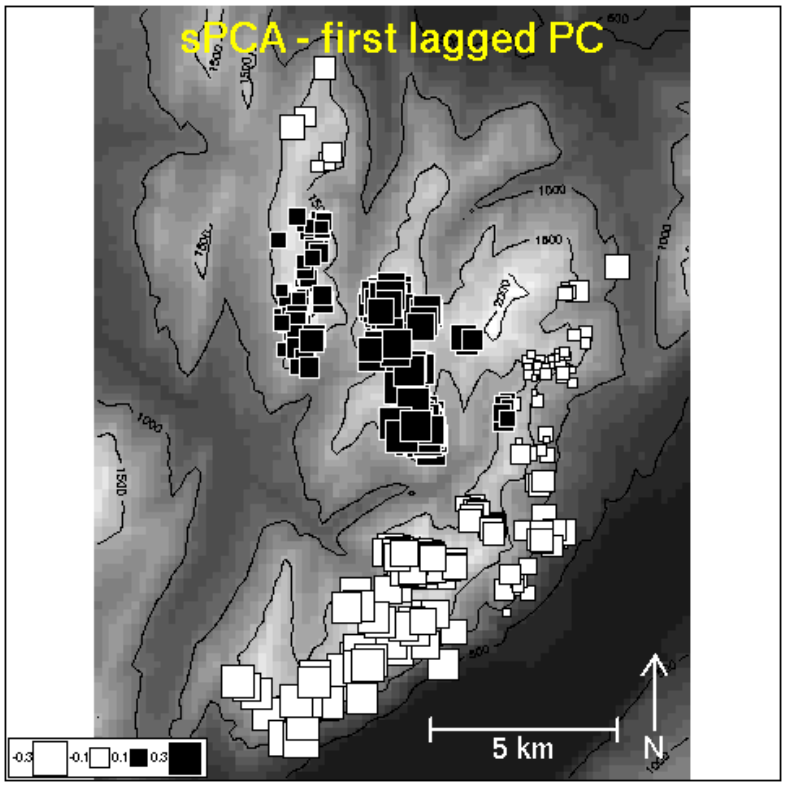
Let us now visualise the identified spatial structures, as we did for the PCA results:

```
> showBauges()
> s.value(rupica$other$xy, rupica.spca1$li[, 1], add.p = TRUE,
+     csize = 0.7)
> title("sPCA - first PC", col.main = "yellow", line = -2, cex.main = 2)
```




While the pattern is clear enough, we can still clarify the results using lagged scores, which allow a better perception of positively autocorrelated structures (by denoisifying data):

```
> showBauges()
> s.value(rupica$other$xy, rupica.spca1$ls[, 1], add.p = TRUE,
+     csize = 0.7)
> title("sPCA - first lagged PC", col.main = "yellow", line = -2,
+     cex.main = 2)
```




How would you interprete this result? How does it compare to results obtained by PCA? What likely inference can we make about the way the landscape influences this population of Chamois?

The second structure remains to be interpreted; using the same graphical representation as for the first principal component, try and visualise the second principal component. Some field observation suggest that it is not artefactual. How would you interprete this second structure?

To finish, you can try representing both structures at the same time using the color coding introduced by [3] (`?colorplot`).

# 3 Different pictures of biodiversity: African and French cattle breeds

The study of the genetic diversity for conservation purposes asks the question of which markers should be used for such studies. In the case of domestic cattle breeds, the FAO `http://www.fao.org/` recommended using a panel of 30 microstallites for conservation genetics studies. The dataset `microbov` provides the genotypes of 704 cattles structured in two species and 15 breeds for the 30 microsatellites recommended by the FAO.

One question of interest, which can be asked through this dataset, relates to whether all these markers provide the same information, and whether a smaller subset of markers could be used to achieve the same level of resolution.

## 3.1 An overview of the data - basic analyses

We first load the data:

```
> data(microbov)
> microbov

   ####################
   ### Genind object ###
   ####################
- genotypes of individuals -

S4 class:  genind
@call: genind(tab = truenames(microbov)$tab, pop = truenames(microbov)$pop)

@tab:  704 x 373 matrix of genotypes

@ind.names: vector of  704 individual names
@loc.names: vector of  30 locus names
@loc.nall: number of alleles per locus
@loc.fac: locus factor for the  373 columns of @tab
@all.names: list of  30 components yielding allele names for each locus
@ploidy:  2
@type:  codom

Optionnal contents:
@pop:  factor giving the population of each individual
@pop.names:  factor giving the population of each individual

@other: a list containing: coun  breed  spe
```

`microbov` is a typical `genind` object, which is the class of objects storing genotypes in *adegenet*. It also contains extra information (in `microbov$other`) relating to the origin (`coun`, Africa or France), the breed (`breed`), and the species (`spe`, *Bos taurus* or *Bos indicus*) of the individuals.

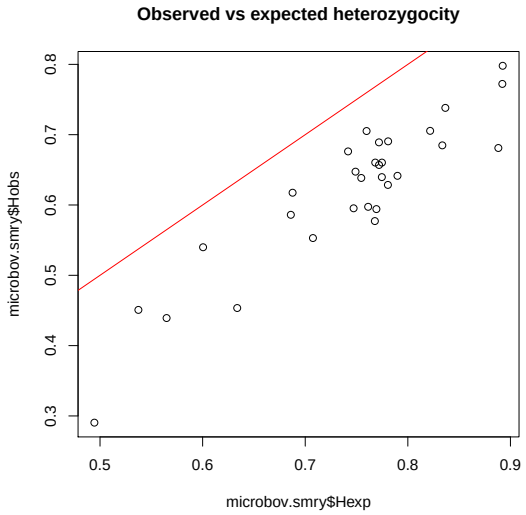
The function summary gives an overview of the data:

```
> microbov.smry <- summary(microbov)

 # Total number of genotypes:  704

 # Population sample sizes:
        Borgou            Zebu       Lagunaire           NDama           Somba
            50              50              51              30              50
        Aubrac        Bazadais BlondeAquitaine    BretPieNoire       Charolais
            50              47              61              31              55
        Gascon        Limousin      MaineAnjou     Montbeliard          Salers
            50              50              49              30              50

 # Number of alleles per locus:
L01 L02 L03 L04 L05 L06 L07 L08 L09 L10 L11 L12 L13 L14 L15 L16 L17 L18 L19 L20
  9   7  12   5  11   9   7  12  13   9  13  16  14  14  14  10  10  19  11  13
L21 L22 L23 L24 L25 L26 L27 L28 L29 L30
 17  12  16  13  12  15   8  22  21   9
```

```
 # Number of alleles per population:
 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15
251 235 143 179 194 212 146 196 176 200 213 186 191 168 188

 # Percentage of missing data:
[1] 2.320076

 # Observed heterozygosity:
      L01       L02       L03       L04       L05       L06       L07       L08
0.5530086 0.5399129 0.6905444 0.4508076 0.6415094 0.5974212 0.2904624 0.5860534
      L09       L10       L11       L12       L13       L14       L15       L16
0.6848306 0.5771429 0.6603221 0.7054598 0.5953079 0.7052023 0.7979943 0.6384505
      L17       L18       L19       L20       L21       L22       L23       L24
0.4534884 0.6396527 0.6474074 0.6285714 0.6603499 0.6569343 0.5941807 0.7381295
      L25       L26       L27       L28       L29       L30
0.6762178 0.7722063 0.6174785 0.6891117 0.6810730 0.4392387

 # Expected heterozygosity:
      L01       L02       L03       L04       L05       L06       L07       L08
0.7075198 0.6004379 0.7807931 0.5373943 0.7899071 0.7613320 0.4945057 0.6859640
      L09       L10       L11       L12       L13       L14       L15       L16
0.8336124 0.7678602 0.7747632 0.8217379 0.7471427 0.7597794 0.8924578 0.7546062
      L17       L18       L19       L20       L21       L22       L23       L24
0.6336998 0.7746696 0.7489997 0.7805834 0.7682354 0.7719260 0.7693717 0.8365613
      L25       L26       L27       L28       L29       L30
0.7417581 0.8921047 0.6876811 0.7718615 0.8882143 0.5648676
```

This allows, for instance, to compare observed and expected heterozygocity at each locus:

```
> plot(microbov.smry$Hexp, microbov.smry$Hobs, main = "Observed vs expected heterozygocity")
> abline(0, 1, col = "red")
```

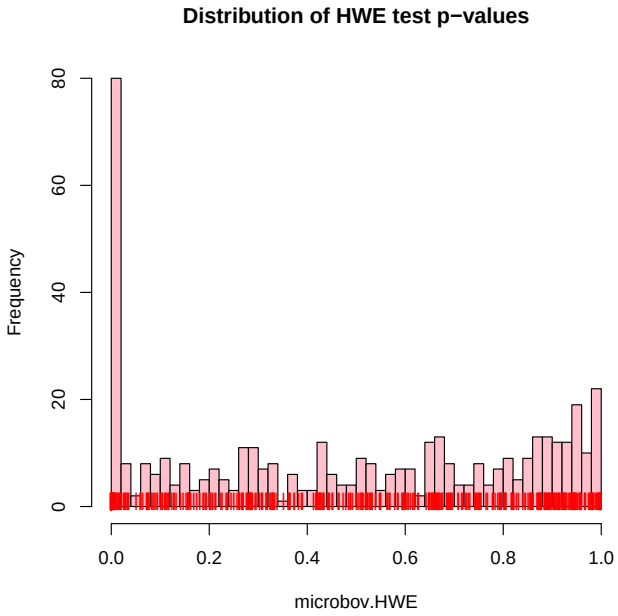



What can we tell about these populations? Is this result surprising?

To infer genetic differentiation using $F_{ST}$-based approaches, we have to check that populations are at Hardy-Weinberg equilibrium for each locus.

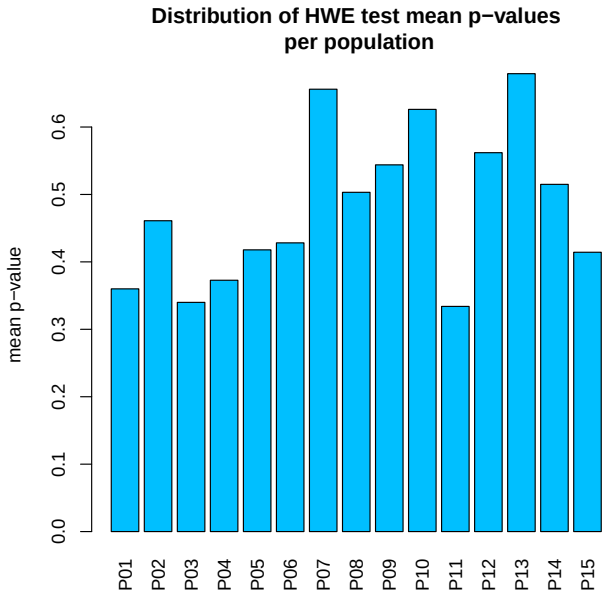
Given that we have 15 breeds for 30 loci to analyse, we have to perform 15x30=450 tests. Fortunately, the function `HWE.test.genind` does this job, returning either a list of detailed tests, or a matrix of p-values. In our case, interpreting each test and correcting for multiple testing would quickly become cumbersome. Rather, we shall describe how p-values are distributed across populations and across markers. We perform Hardy-Weinberg tests, asking for a matrix of p-values:

```
> microbov.HWE <- HWE.test.genind(microbov, res = "matrix")
> hist(microbov.HWE, col = "pink", main = "Distribution of HWE test p-values",
+     nclass = 60)
> points(as.vector(microbov.HWE), rep(1, 450), col = "red", pch = "|")
```



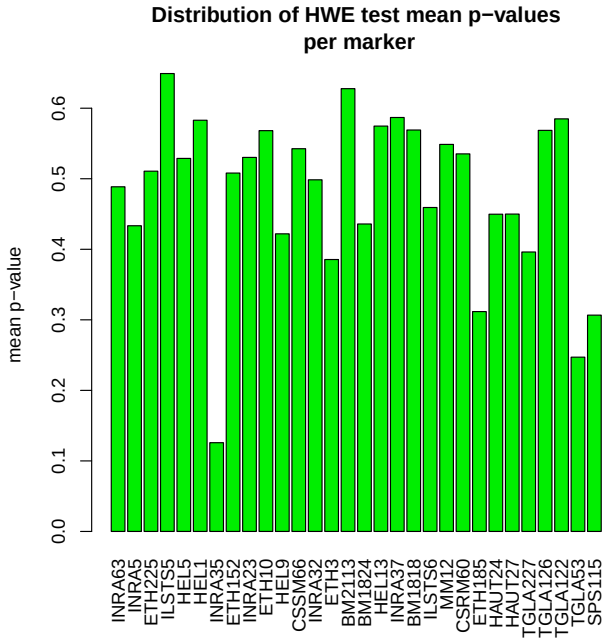

While a majority of tests do not indicate deviation from Hardy-Weinberg equilibrium, some exceptions seem to exist. Are these structured by populations?

```
> barplot(apply(microbov.HWE, 1, mean), col = "deepskyblue1", main = "Distribution of HWE te
+     ylab = "mean p-value", las = 3)
```




Are these structured by markers?

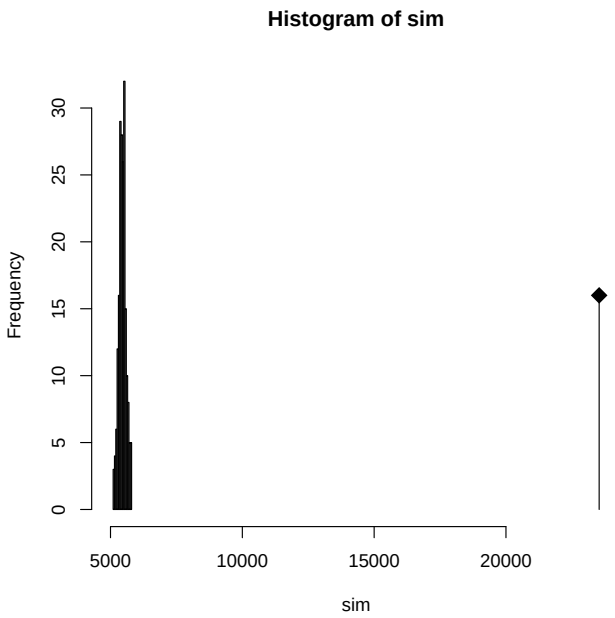
```
> barplot(apply(microbov.HWE, 2, mean), col = "green2", main = "Distribution of HWE test mea
+     ylab = "mean p-value", las = 3)
```



What would you conclude? Toward the end of this exercise, we shall remember that INRA35 seems to be a particular marker.

Genetic differentiation can be tested for multiallelic data using Goudet's $G$ test, implemented in *hierfstat*, and wrapped for `genind` objects by `gstat.randtest`. Basically, we can test the significance of the genetic differentiation between breeds, which is the default 'population' of the genotypes. For simplicity (and because it does not alter the results), all markers (including INRA35) are kept in this test:

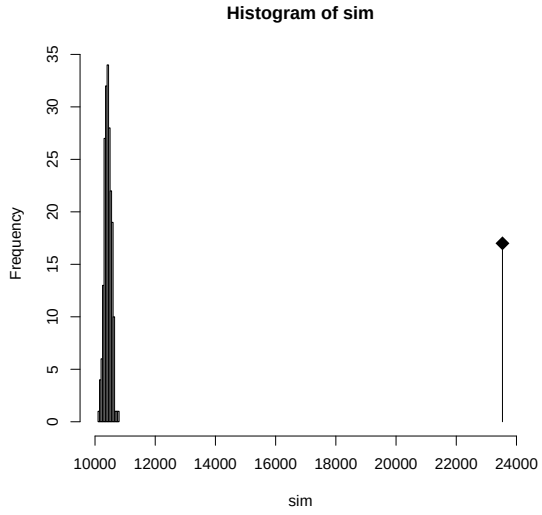
```
> microbov.gtest1 <- gstat.randtest(microbov, nsim = 199)
> microbov.gtest1

Monte-Carlo test
Call: gstat.randtest(x = microbov, nsim = 199)

Observation: 23534.67

Based on 199 replicates
Simulated p-value: 0.005
Alternative hypothesis: greater

    Std.Obs Expectation    Variance
    124.838    5480.433   20915.339

> plot(microbov.gtest1)
```




The histogram shows the distribution of the test statistic obtained by a Monte Carlo procedure (permutation of the group factor). The original

value of the statistic (on the right) being hugely superior to these values, there is no doubt that the genetic structuring is very significant. However, we can wonder if this structuration among breeds persists after accounting for the species differences. This can be tested using the same function:

```
> microbov.gtest2 <- gstat.randtest(microbov, nsim = 199, sup.pop = microbov$other$spe,
+     method = "within")
> microbov.gtest2

Monte-Carlo test
Call: gstat.randtest(x = microbov, method = "within", sup.pop = microbov$other$spe,
    nsim = 199)

Observation: 23534.67

Based on 199 replicates
Simulated p-value: 0.005
Alternative hypothesis: greater

    Std.Obs Expectation    Variance
  107.8852  10425.8523  14763.9883

> plot(microbov.gtest2)
```



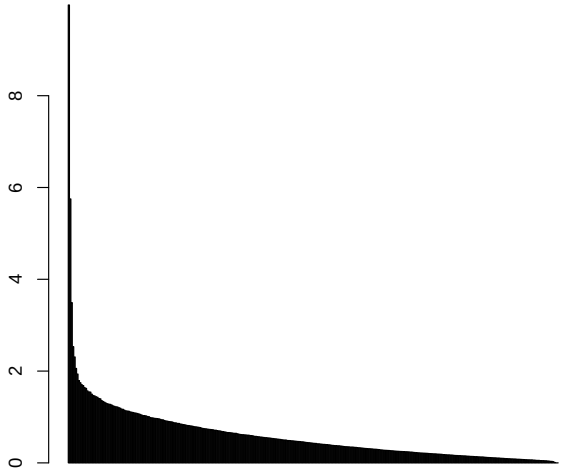

Is their a significant genetic differentiation between breeds once species differentiation has been partialled out?

## 3.2 A first glance: Principal Component Analysis

Now that we know that strong genetic structures exists among the considered breeds, we can try to get a picture of it. Principal Component Analysis (PCA [13, 9]) is well suited for a first glance at the data. PCA is implemented

in the dudi.pca function of the *ade4* package. The analysis is performed on a table of standardised alleles frequencies, obtained by scaleGen (which also replaces missing values adequately):

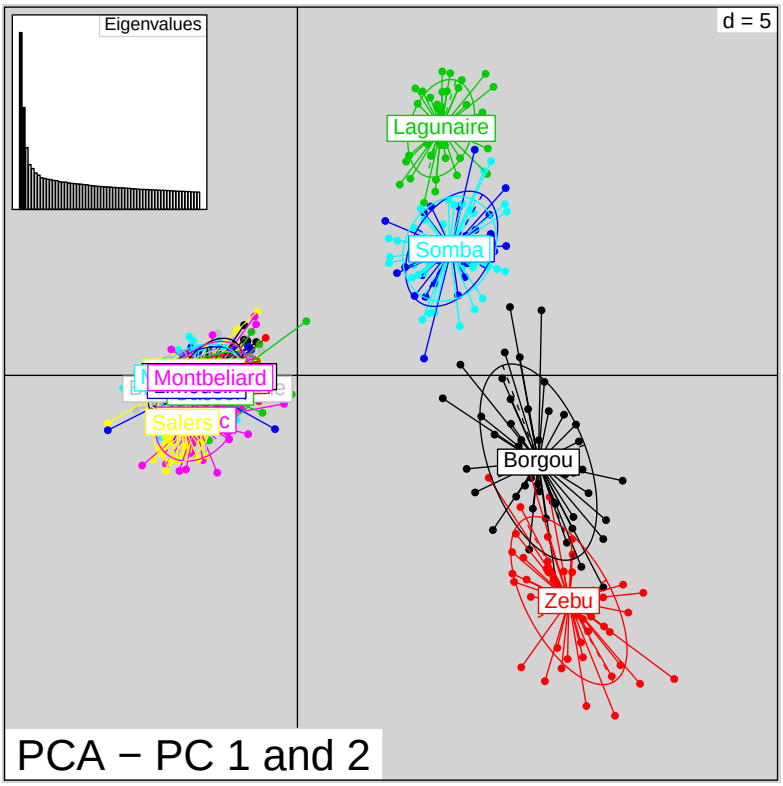
```
> microbov.X <- scaleGen(microbov, method = "binom")
> microbov.pca1 <- dudi.pca(microbov.X, cent = FALSE, scale = FALSE)
```



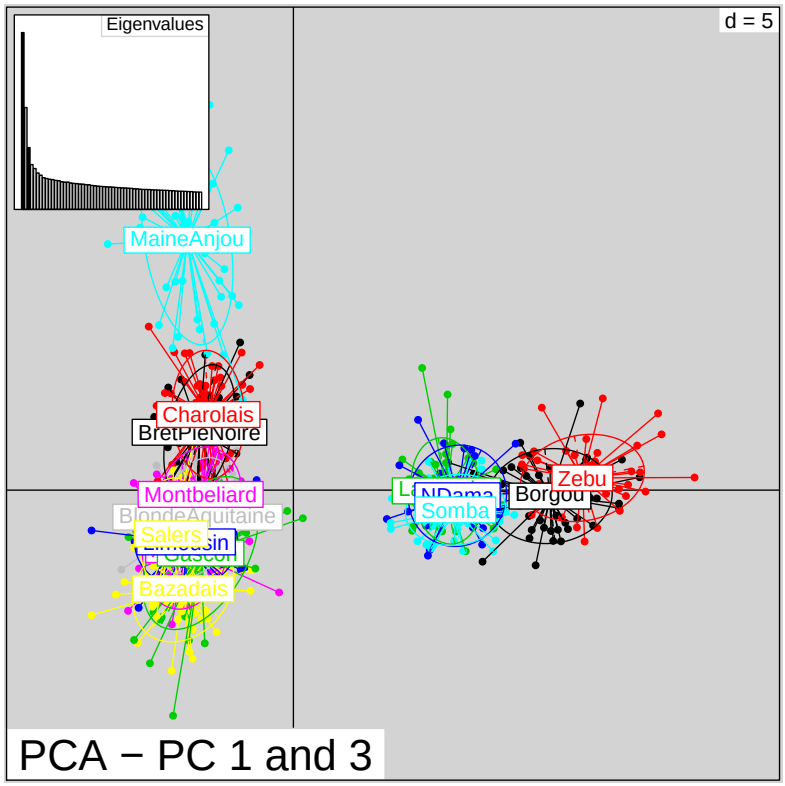

The function dudi.pca displays a barplot of eigenvalues and asks for a number of retained principal components. Eigenvalues represent the amount of genetic diversity (as measured by the multivariate method being used) contained in each principal component. An abrupt decrease in eigenvalues is likely to indicate the boundary between strong and non-interpretable structures. In this case, the first three eigenvalues clearly indicate strong structures; the first three principal components are thus retained.

A `dudi` object contains various information; in the case of PCA, principal axes (loadings), principal components (synthetic variable), and eigenvalues are respectively stored in `microbov.pca1$c1`, `microbov.pca1$li`, and `microbov.pca1$eig`. The function s.class can be used to display to two first principal components, while grouping genotypes by populations:

```
> par(bg = "lightgrey")
> palette <- rainbow(50)
> s.class(microbov.pca1$li, pop(microbov), col = 1:15, sub = "PCA - PC 1 and 2",
+     csub = 2)
> add.scatter.eig(microbov.pca1$eig[1:60], 3, 1, 2, posi = "top")
```



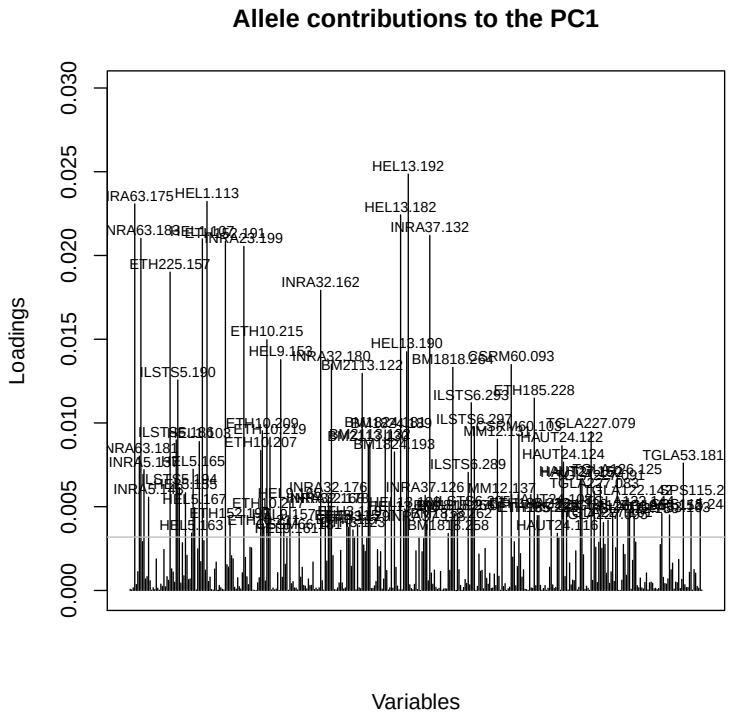

```
> par(bg = "lightgrey")
> s.class(microbov.pca1$li, xax = 1, yax = 3, pop(microbov), col = 1:15,
+     sub = "PCA - PC 1 and 3", csub = 2)
> add.scatter.eig(microbov.pca1$eig[1:60], 3, 1, 3, posi = "top")
```




These figures display the 'best' picture of genetic variability among the genotypes achievable in three dimensions. How would you interprete the resulting structures?

Now that clear patterns have been identified, we can ask how each marker contributes to showing these structures. The contribution of each marker (measured as squared loadings) can be displayed using `loadingplot`:

```
> loadingplot(microbov.pca1$c1^2, main = "Allele contributions to the PC1")
```




From this picture, could you tell if some markers play a more important role in the analysis than others? This was the contribution of alleles to the first principal component. Using the same function and the argument `axis`, try to obtain the same figure for the second and third principal components. Are the conclusions any different (if yes, how)?

## 3.3 A deeper look: Multiple Co-Inertia Analysis

PCA is not the most appropriate tool to compare the information provided by different markers about the populations (*i.e.*, breeds). Indeed, it only seeks principal axes of maximum genetic variability from all alleles, while a more appropriate approach would seek different principal components for each marker separately, and then compare them. The Multiple Co-Inertia Analysis (MCOA, [5, 12]) is especially devoted to this task. It performs separate analyses for each marker, and then coordinates these analyses so as to highlight the common information they provide about populations. From these coordinated analyses, it builds a compromise, that is, a typology of population reflecting the consensus information provided by the markers. It also provides a direct measure of the contribution of each marker to this consensus information.

First of all, given that within-breed variability seems negligible compared to between-breed variability, we reduce data to counts of alleles per populations (losing the distinction between individuals). Objects storing population data in *adegenet* are `genpop` objects. This transformation is achieved by `genind2genpop`:

```
> bov <- genind2genpop(microbov)

 Converting data from a genind to a genpop object...

...done.

> bov

       ######################
       ### Genpop object ###
       ######################
- Alleles counts for populations -

S4 class:  genpop
@call: genind2genpop(x = microbov)

@tab:  15 x 373 matrix of alleles counts

@pop.names: vector of  15 population names
@loc.names: vector of  30 locus names
@loc.nall: number of alleles per locus
@loc.fac: locus factor for the  373 columns of @tab
@all.names: list of  30 components yielding allele names for each locus
@ploidy:  2
@type:  codom

@other: a list containing: coun  breed  spe
```

Data are then separated by marker using `seploc`, and only tables of allele counts are retained for further analysis:

```
> lbov <- seploc(bov)
> lX <- lapply(lbov, truenames)
> class(lX)

[1] "list"

> names(lX)

 [1] "INRA63"  "INRA5"   "ETH225"  "ILSTS5"  "HEL5"    "HEL1"    "INRA35"
 [8] "ETH152"  "INRA23"  "ETH10"   "HEL9"    "CSSM66"  "INRA32"  "ETH3"
[15] "BM2113"  "BM1824"  "HEL13"   "INRA37"  "BM1818"  "ILSTS6"  "MM12"
[22] "CSRM60"  "ETH185"  "HAUT24"  "HAUT27"  "TGLA227" "TGLA126" "TGLA122"
[29] "TGLA53"  "SPS115"

> lX$INRA63
```

```
                INRA63.167 INRA63.171 INRA63.173 INRA63.175 INRA63.177
Borgou                   0          0          0          4         27
Zebu                     0          1          0          7         16
Lagunaire                1          0          0         16         44
NDama                    0          0          0          2         39
Somba                    0          0          0         12         42
Aubrac                   0          0          0         80          0
Bazadais                 0          0          0         54         28
BlondeAquitaine          0          0          0         54         52
BretPieNoire             0          0          1         39         18
Charolais                0          0          5         46         37
Gascon                   0          0          0         77          1
Limousin                 0          0          1         45         52
MaineAnjou               0          1          0         46         48
Montbeliard              0          0          0         25         25
Salers                   0          0          0         70          0
                INRA63.179 INRA63.181 INRA63.183 INRA63.185
Borgou                   1          7         60          1
Zebu                     4         19         47          6
Lagunaire                0          2         16         23
NDama                    5         11          3          0
Somba                    3          8         34          1
Aubrac                  20          0          0          0
Bazadais                 0          0         10          0
BlondeAquitaine          7          0          7          0
BretPieNoire             2          0          2          0
Charolais               15          0          1          0
Gascon                  11          0          8          3
Limousin                 2          0          0          0
MaineAnjou               1          0          0          0
Montbeliard              4          0          6          0
Salers                  25          0          5          0
```

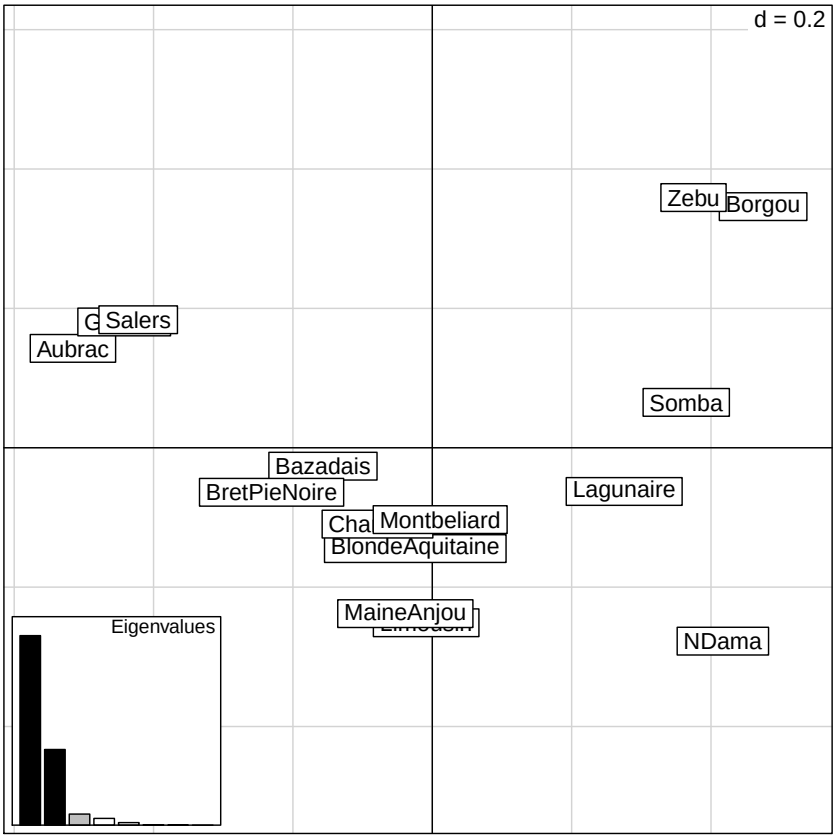
`kbov` contains counts of alleles per population separately for each marker. After turning these into allele frequencies, each table is analysed by a PCA. The method is applied to all 30 tables in a single command using `lapply`:

```
> lX <- lapply(lX, prop.table, 1)
> lPCA <- lapply(lX, dudi.pca, center = TRUE, scale = FALSE, scannf = FALSE,
+     nf = 3)
> class(lPCA)

[1] "list"

> names(lPCA)

 [1] "INRA63"  "INRA5"   "ETH225"  "ILSTS5"  "HEL5"    "HEL1"    "INRA35"
 [8] "ETH152"  "INRA23"  "ETH10"   "HEL9"    "CSSM66"  "INRA32"  "ETH3"
[15] "BM2113"  "BM1824"  "HEL13"   "INRA37"  "BM1818"  "ILSTS6"  "MM12"
[22] "CSRM60"  "ETH185"  "HAUT24"  "HAUT27"  "TGLA227" "TGLA126" "TGLA122"
[29] "TGLA53"  "SPS115"

> lPCA$INRA63

Duality diagramm
class: pca dudi
$call: FUN(df = X[[1L]], center = TRUE, scale = FALSE, scannf = FALSE,
    nf = 3)

$nf: 3 axis-components saved
```

```
$rank: 8
eigen values: 0.09829 0.03924 0.005741 0.003492 0.001314 ...
  vector length mode    content
1 $cw    9      numeric column weights
2 $lw    15     numeric row weights
3 $eig   8      numeric eigen values

  data.frame nrow ncol content
1 $tab       15   9    modified array
2 $li        15   3    row coordinates
3 $l1        15   3    row normed scores
4 $co        9    3    column coordinates
5 $c1        9    3    column normed scores
other elements: cent norm
```

To visualise the results of a given analysis (here, INRA63), one can use:

```
> s.label(lPCA$INRA63$li)
> add.scatter.eig(lPCA$INRA63$eig, 3, 1, 2)
```

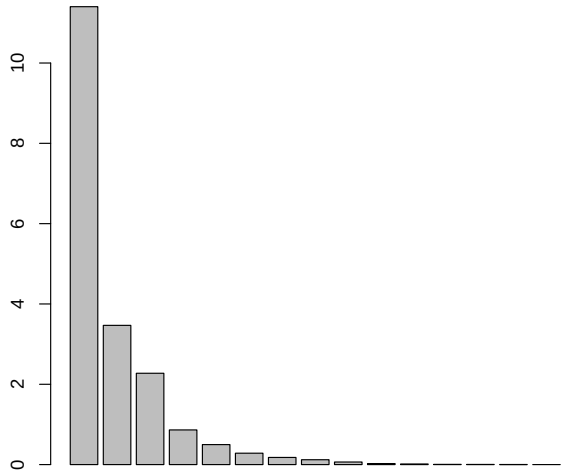



Now, using a `for` loop (or a `lapply`, or less elegantly several copy-paste operations), try and display results of other markers. Can you compare the information they provide? Note that the situation is complicated by the fact that the first principal component of one marker might ressemble best

the third of another marker, or even a mixture of several components.

Let us try coordinating these analyses using MCOA. The method is implemented as the function `mcoa` in the *ade4* package. It demands data to be stored as a `ktab` object, which we obtain by:

```
> bov.ktab <- ktab.list.dudi(lPCA)

> bov.mcoa1 <- mcoa(bov.ktab)
```




Proceed like in previous analyses to select the number of retained principal components.

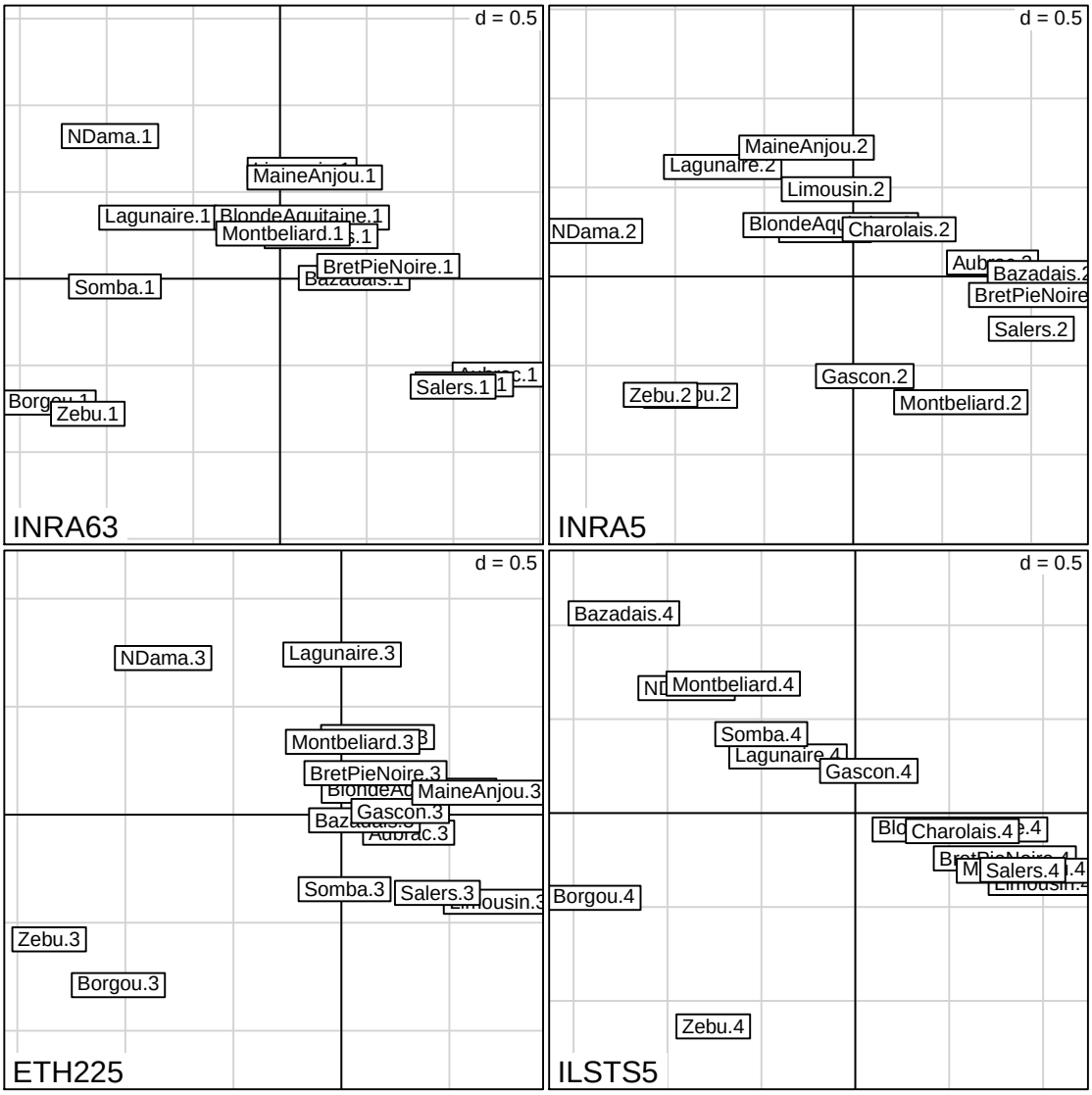
```
> bov.mcoa1

Multiple Co-inertia Analysis
list of class mcoa

$pseudoeig: 15 pseudo eigen values
11.4 3.467 2.274 0.8631 0.4978 ...

$call: mcoa(X = bov.ktab, scannf = FALSE, nf = 3)

$nf: 3 axis saved

   data.frame nrow ncol content
1  $SynVar    15   3    synthetic scores
2  $axis      373  3    co-inertia axis
3  $Tli       450  3    co-inertia coordinates
4  $Tl1       450  3    co-inertia normed scores
5  $Tax       120  3    inertia axes onto co-inertia axis
```

```
6  $Tco       373  3    columns onto synthetic scores
7  $TL        450  2    factors for Tli Tl1
8  $TC        373  2    factors for Tco
9  $T4        120  2    factors for Tax
10 $lambda    30   3    eigen values (separate analysis)
11 $cov2      30   3    pseudo eigen values (synthetic analysis)
other elements: NULL
```

The content of a `mcoa` object is a bit more complicated than that of PCA (`dudi` object), but only bits are useful here. `bov.mcoa1$Tli` contains principal components of coordinated analyses for the different markers, while `bov.mcoa1$SynVar` contains the compromise, *i.e.* the typology of populations emerging as a consensus among the markers. `bov.mcoa1$cov2` gives the contribution of each marker to each structure of the compromise, and can be used to assess discrepancies in the information yielded by the different loci.

Coordinated analyses can be displayed like separated analyses:

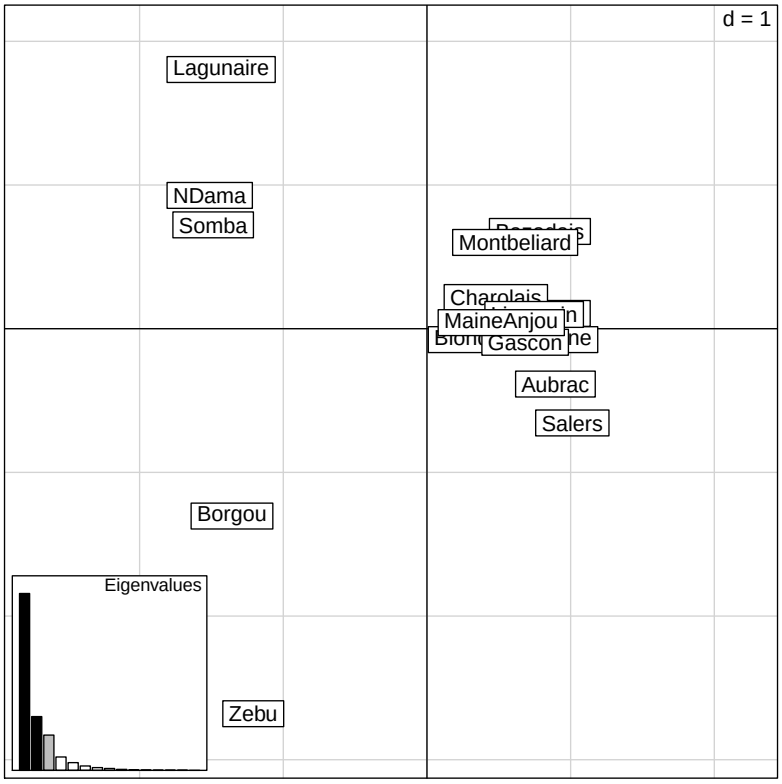
```
> newCoord <- split(bov.mcoa1$Tli, bov.mcoa1$TL[, 1])
> names(newCoord) <- locNames(bov)
> par(mfrow = c(2, 2))
> for (i in 1:4) {
+     s.label(newCoord[[i]], xax = 1, yax = 2, sub = names(newCoord)[i],
+         csub = 1.5)
+ }
```



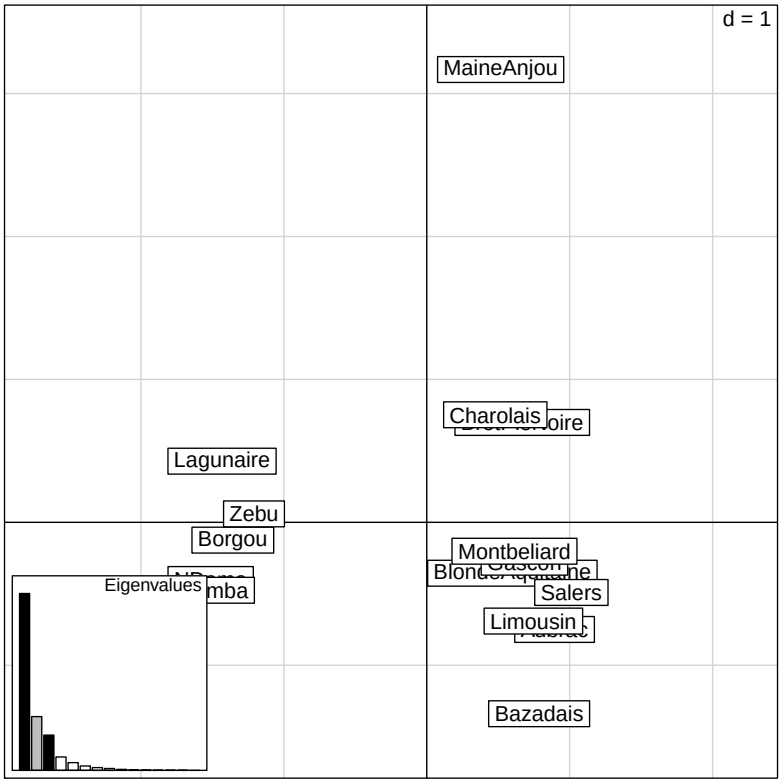

Use the commands above to plot results of different markers, making sure to visualise the plan of the first and third principal components as well. How does it compare to the results obtained with previous (uncoordinated) analyses?

The compromise between all these analyses is very similar to the usual PCA of all data:

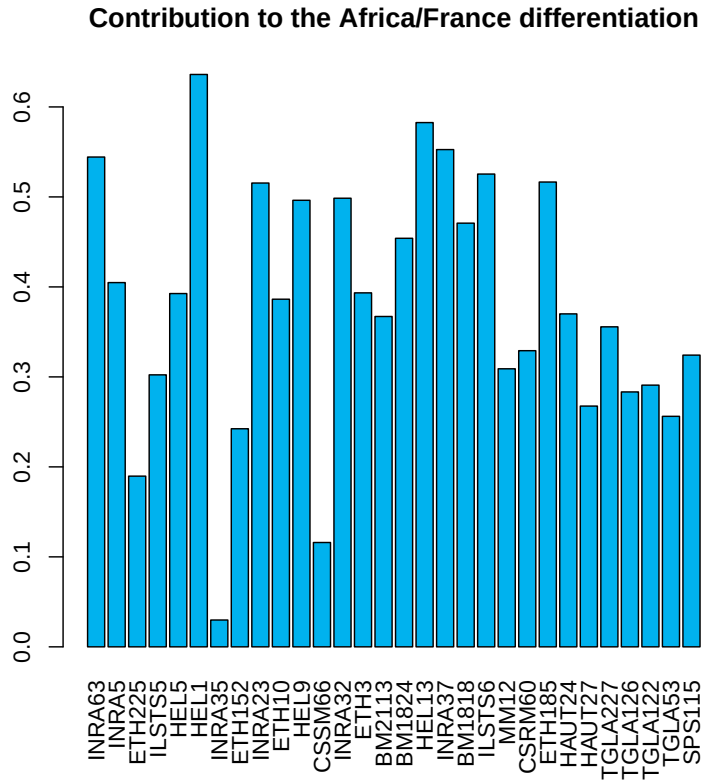
```
> s.label(bov.mcoa1$SynVar)
> add.scatter.eig(bov.mcoa1$pseudoeig, 3, 1, 2)
```




```
> s.label(bov.mcoa1$SynVar, xax = 1, yax = 3)
> add.scatter.eig(bov.mcoa1$pseudoeig, 3, 1, 3)
```




However, we now gained further information about how markers contribute to this figure. Try and represent graphically the marker contributions stored in `bov.mcoa1$cov2` for the three structures of the compromise; one example of result for the first structure would be:




What can we say about the general consistency of these markers? Are there redundant markers? Are there 'outlying' markers? Would it be possible to achieve the same structuring without using the full panel of 30 microsatellites recommended by the FAO?

# References

[1] C. Calenge. Exploring habitat selection by wildlife with adehabitat. *Journal of statistical software*, 22(6):1–19, 2007.

[2] C. Callenge. The package "adehabitat" for the r software: a tool for the analysis of space and habitat use by animals. *Ecological Modelling*, 197:516–519, 2006.

[3] L. L. Cavalli-Sforza, P. Menozzi, and A. Piazza. Demic expansions and human evolution. *Science*, 259:639–646, 1993.

[4] D. Chessel, A-B. Dufour, and J. Thioulouse. The ade4 package-I- one-table methods. *R News*, 4:5–10, 2004.

[5] D. Chessel and M. Hanafi. Analyses de la co-inertie de $K$ nuages de points. *Revue de statistique appliquée*, XLIV (2):35–60, 1996.

[6] S. Dray and A.-B. Dufour. The ade4 package: implementing the duality diagram for ecologists. *Journal of Statistical Software*, 22(4):1–20, 2007.

[7] S. Dray, A.-B. Dufour, and D. Chessel. The ade4 package - II: Two-table and $K$-table methods. *R News*, 7:47–54, 2007.

[8] J. Goudet. HIERFSTAT, a package for R to compute and test hierarchical F-statistics. *Molecular Ecology Notes*, 5:184–186, 2005.

[9] I. T. Joliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, second edition, 2004.

[10] T. Jombart. adegenet: a R package for the multivariate analysis of genetic markers. *Bioinformatics*, 24:1403–1405, 2008.

[11] T. Jombart, S. Devillard, A.-B. Dufour, and D. Pontier. Revealing cryptic spatial patterns in genetic variability by a new multivariate method. *Heredity*, 101:92–103, 2008.

[12] D. Laloë, T. Jombart, A.-B. Dufour, and K. Moazami-Goudarzi. Consensus genetic structuring and typological value of markers using multiple co-inertia analysis. *Genetics Selection Evolution*, 39:545–567, 2007.

[13] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[14] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.

[15] G. R. Warnes. The genetics package. *R News*, 3(1):9–13, 2003.