Multivariate analysis of genetic data: investigating spatial structures

Thibaut Jombart[*]

Imperial College London

MRC Centre for Outbreak Analysis and Modelling

August 19, 2016

Abstract

This practical provides an introduction to the analysis of spatial genetic structures using `R`. Using an empirical dataset of microsatellites sampled from wild goats in the Bauges mountains (France), we illustrate univariate and multivariate tests of spatial structures, and then introduce the spatial Principal Component Analysis (sPCA) for uncovering spatial genetic patterns. For a more complete overview of spatial genetics methods, see the basics and sPCA tutorials for *adegenet*, which you can open using `adegenetTutorial("basics")` and `adegenetTutorial("spca")` (with an internet connection) or from the adegenet website.

[*] `tjombart@imperial.ac.uk`

Contents

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 purposes 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 induce 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 structuring. 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.

1 An overview of the data

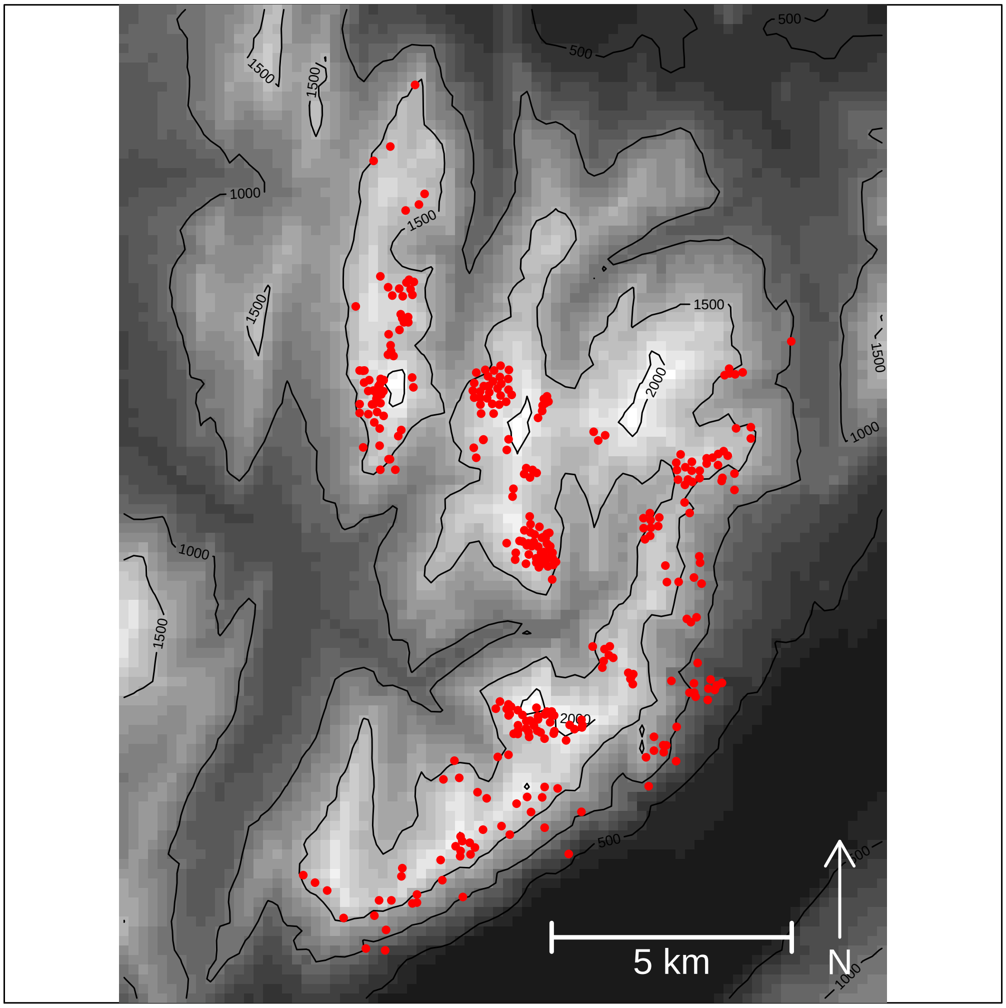
We first load some required packages and the data:

```
library(spdep)
library(adehabitat)
```

```
library(ade4)
library(adegenet)

##
##    /// adegenet 2.0.1 is loaded ////////////
##
##    > overview:  '?adegenet'
##    > tutorials/doc/questions:  'adegenetWeb()'
##    > bug reports/feature requests:  adegenetIssues()

data(rupica)
rupica

## /// GENIND OBJECT /////////
##
##  // 335 individuals; 9 loci; 55 alleles; size: 217.2 Kb
##
##  // Basic content
##    @tab:  335 x 55 matrix of allele counts
##    @loc.n.all: number of alleles per locus (range: 4-10)
##    @loc.fac: locus factor for the 55 columns of @tab
##    @all.names: list of allele names for each locus
##    @ploidy: ploidy of each individual  (range: 2-2)
##    @type:  codom
##    @call: NULL
##
##  // Optional content
##    @other: a list containing: xy  mnt  showBauges
```

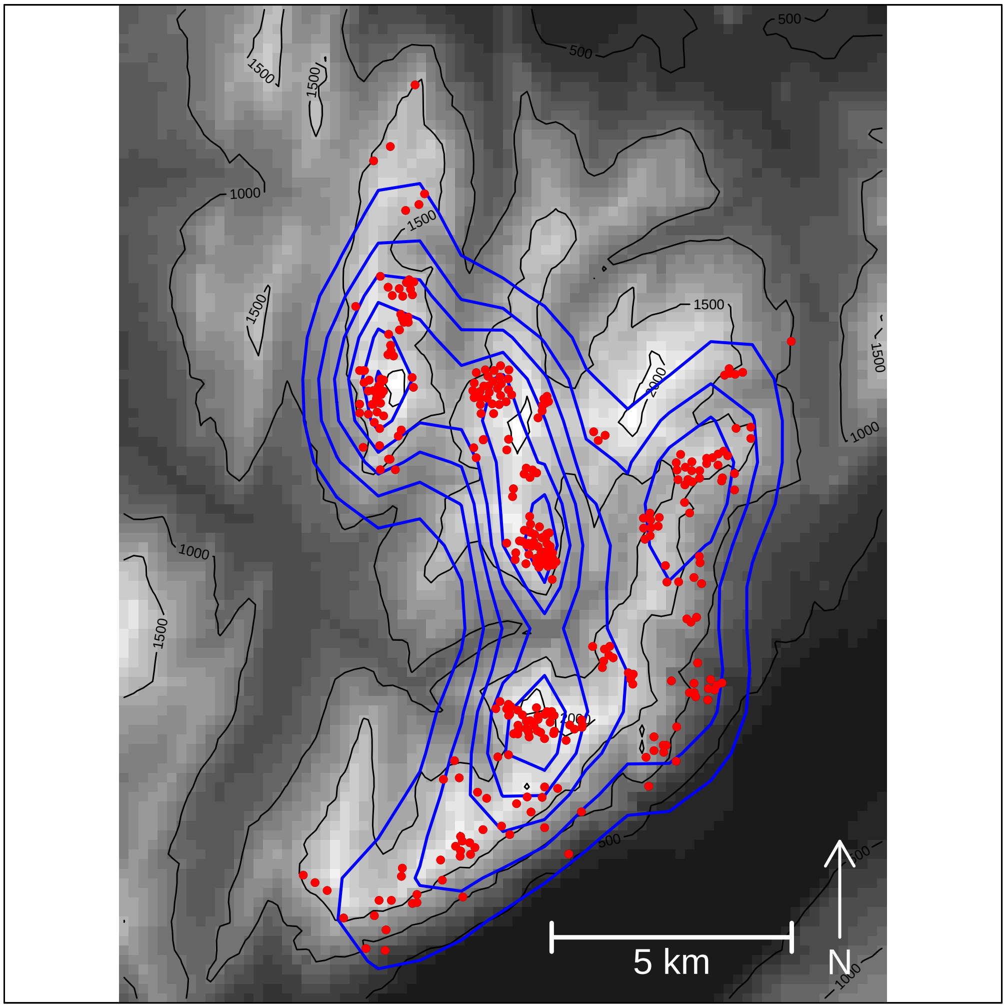
`rupica` is a typical `genind` object, which is the class of objects storing genotypes (as opposed to population data) in *adegenet*. `rupica` also contains topographic information about the sampled area, which can be displayed by calling `rupica$other$showBauges`. For instance, the spatial distribution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

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



This spatial distribution is clearly not random, but seems arranged into loose clusters. However, superimposed samples can bias our visual assessment of the spatial clustering. Use a two-dimensional kernel density estimation (function `s.kde2d`) to overcome this possible issue.

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

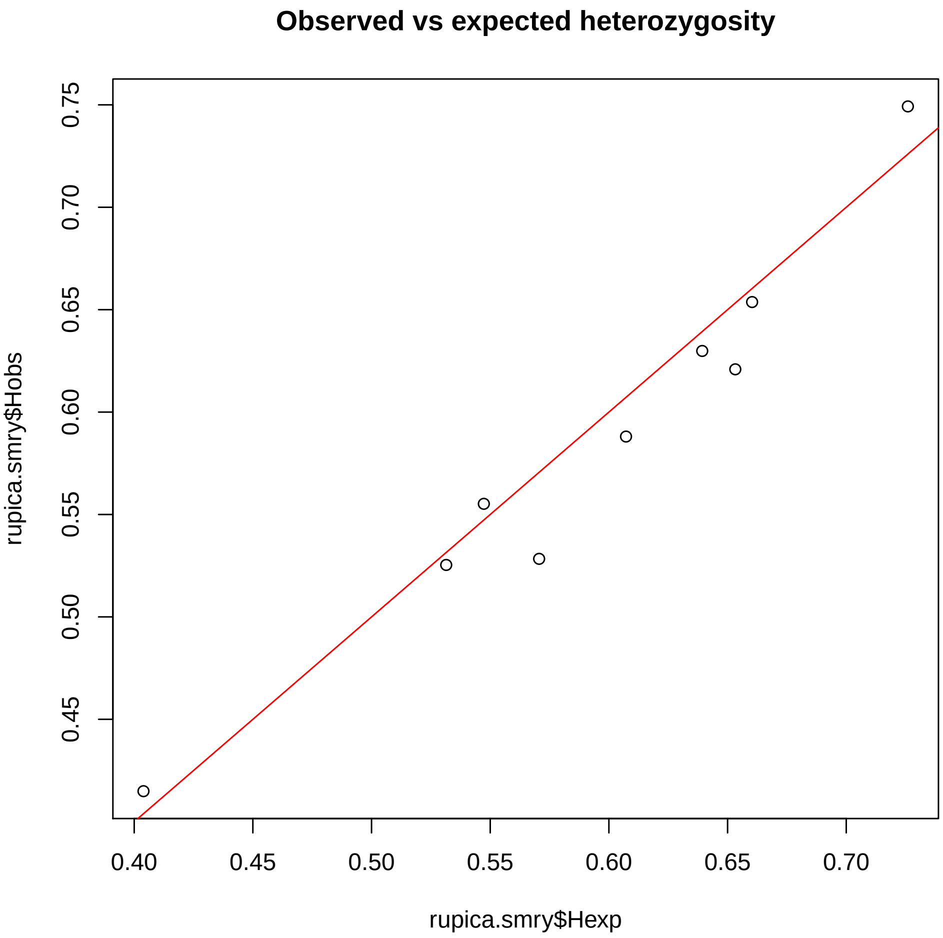


Is geographical clustering strong enough to assign safely each individual to a group? Accordingly, shall we analyse these data at individual or group level?

2 Summarising the genetic diversity

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 heterozygosity. Use the summary of genind objects to compare expected and observed heterozygosity:

```
rupica.smry <- summary(rupica)
plot(rupica.smry$Hexp, rupica.smry$Hobs, main="Observed vs expected heterozygosity")
abline(0,1,col="red")
```

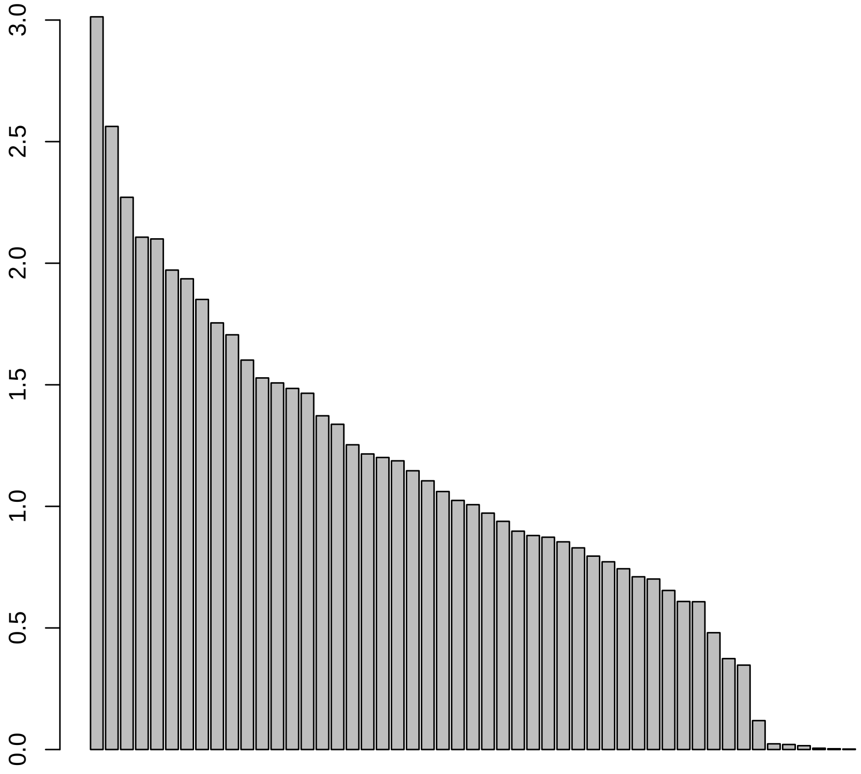


The red line indicate identity between both quantities. What can we say about heterozygosity in this population? How can this be tested? The result below can be reproduced using a standard testing procedure:

```
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 = 1.1761, df = 8, p-value = 0.2734
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.007869215  0.024249885
## sample estimates:
## mean of the differences
##             0.008190335
```

We can seek a global picture of the genetic diversity among genotypes using a Principal Component Analysis (PCA, [4, 2], dudi.pca in ade4 package). The analysis is performed on a table of allele frequencies, obtained by tab. The function dudi.pca displays a barplot of eigenvalues and asks for a number of retained principal components:

```
rupica.X <- tab(rupica, freq=TRUE, NA.method="mean")
rupica.pca1 <- dudi.pca(rupica.X, cent=TRUE, scannf=FALSE, nf=2)
barplot(rupica.pca1$eig)
```



The output produced by dudi.pca is a dudi object. 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. Here is the content of the PCA:

```
rupica.pca1

## Duality diagramm
## class: pca dudi
## $call: dudi.pca(df = rupica.X, center = TRUE, scannf = FALSE, nf = 2)
##
## $nf: 2 axis-components saved
## $rank: 51
## eigen values: 3.013 2.563 2.271 2.107 2.1 ...
##   vector length mode    content
## 1 $cw    55     numeric column weights
## 2 $lw    335    numeric row weights
## 3 $eig   51     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
```

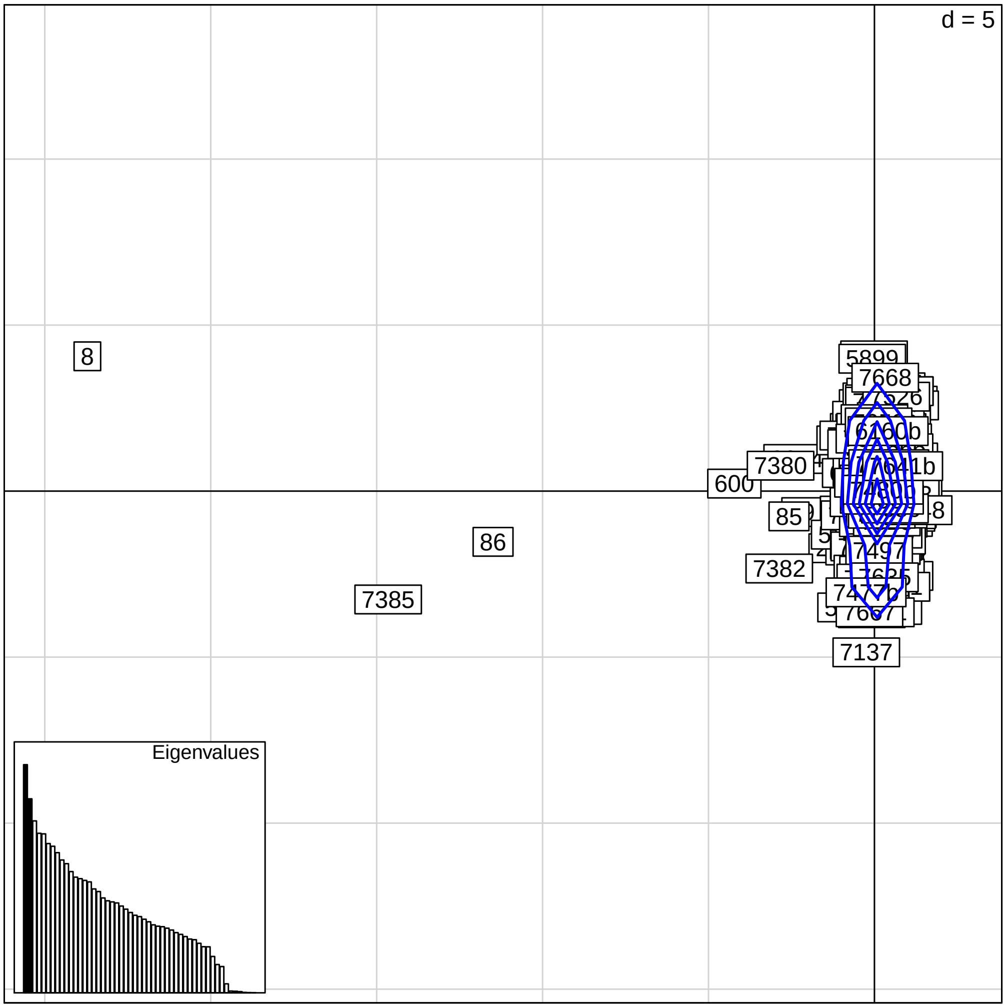
In general, eigenvalues represent the amount of genetic diversity — as measured by the multivariate method being used — represented by each principal component (PC). Verify that here, each eigenvalue is the variance of the corresponding PC.

```
head(rupica.pca1$eig)

## [1] 3.013193 2.562510 2.271093 2.107049 2.099738 1.971703

apply(rupica.pca1$li,2,var)*334/335

##    Axis1    Axis2
## 3.013193 2.562510
```

An abrupt decrease in eigenvalues is likely to indicate the boundary between true patterns and non-interpretable structures. In this case, how many PCs would you interprete?

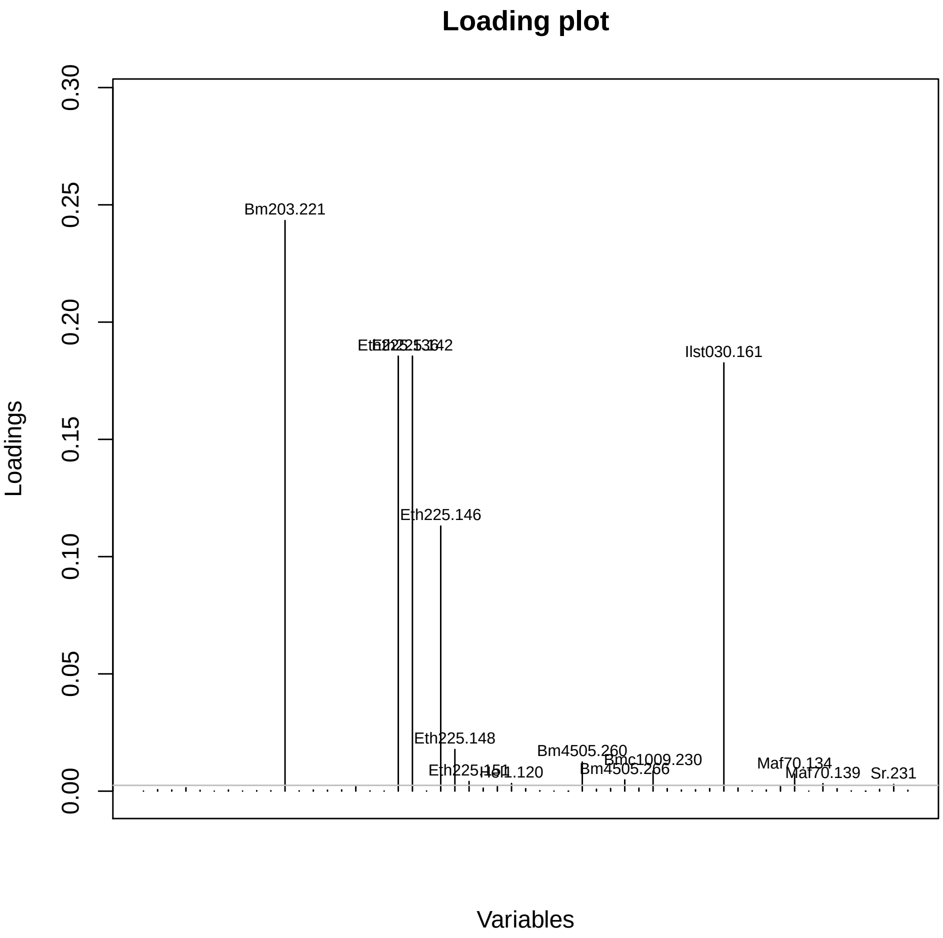
Use s.label to display to two first components of the analysis. Then, use a kernel density (s.kde2d) for a better assessment of the distribution of the genotypes onto the principal axes:

```
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. Using this function, reproduce this figure:

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



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

```
X <- tab(rupica)
class(X)

## [1] "matrix"

dim(X)

## [1] 335  55

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

## bm203.221
##   0   1
## 331   4
```

Only 4 genotypes possess one copy of the allele 221 of marker bm203 (the second result corresponds to a replaced missing data). Which individuals are they?

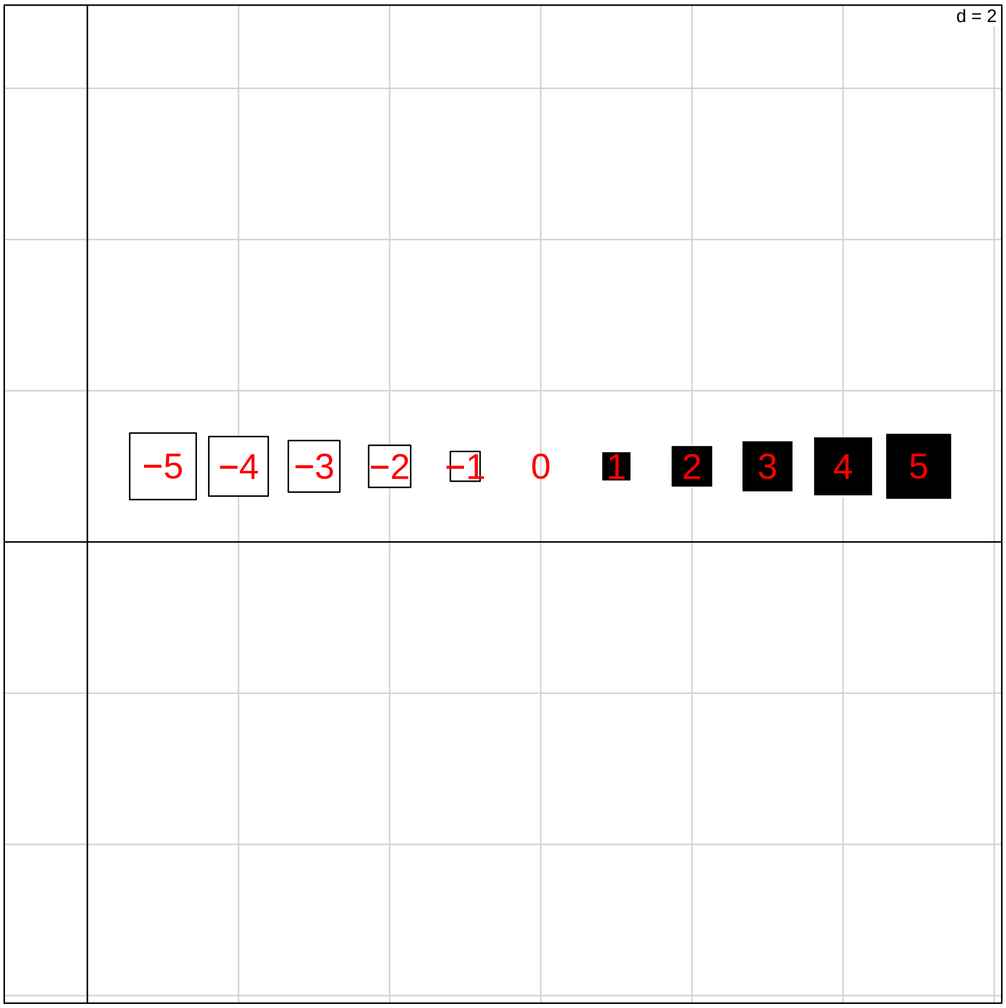
```
rownames(X)[bm203.221>0]

## [1] "8"    "86"   "600"  "7385"
```

Conclusion?

3 Mapping and testing PCA results

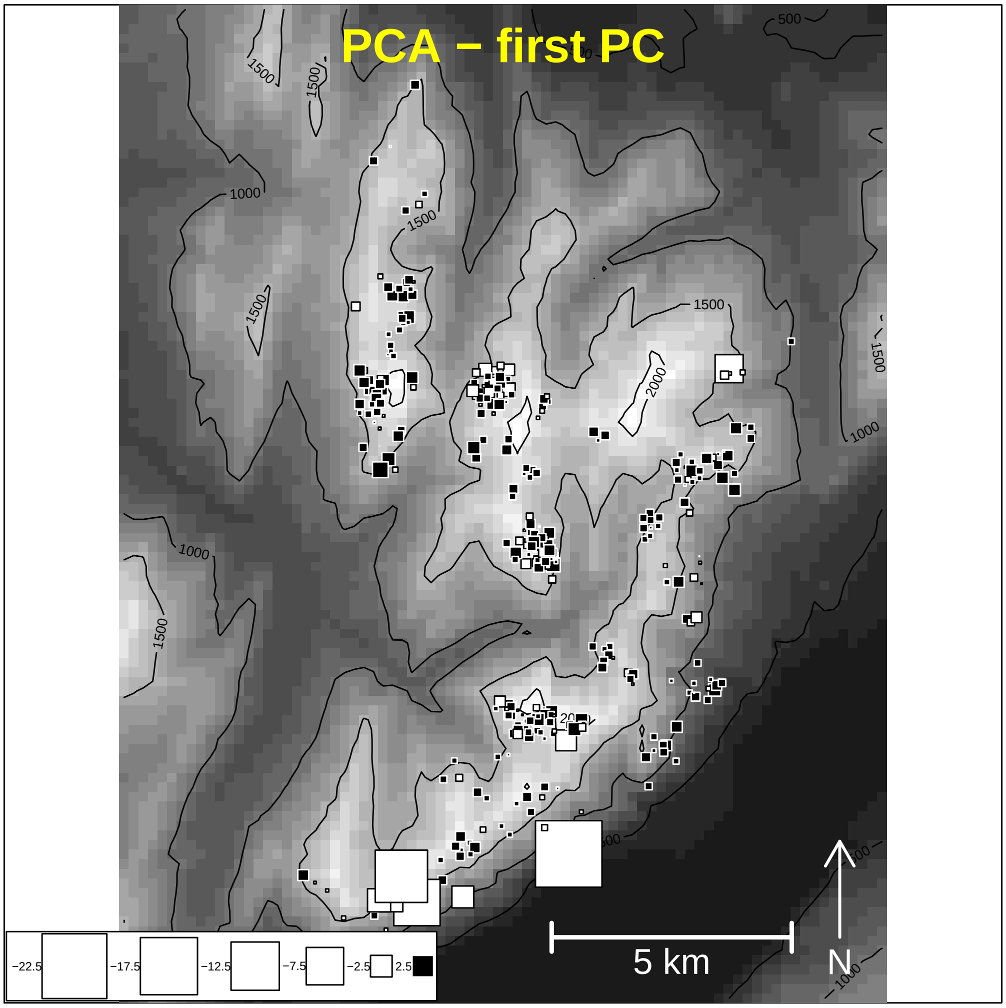
A frequent practice in spatial genetics is mapping the first principal components (PCs) onto the geographic space. The function `s.value` is well-suited to do so, using black and white squares of variable size for positive and negative values. To give a legend for this type of representation:

```
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)
```

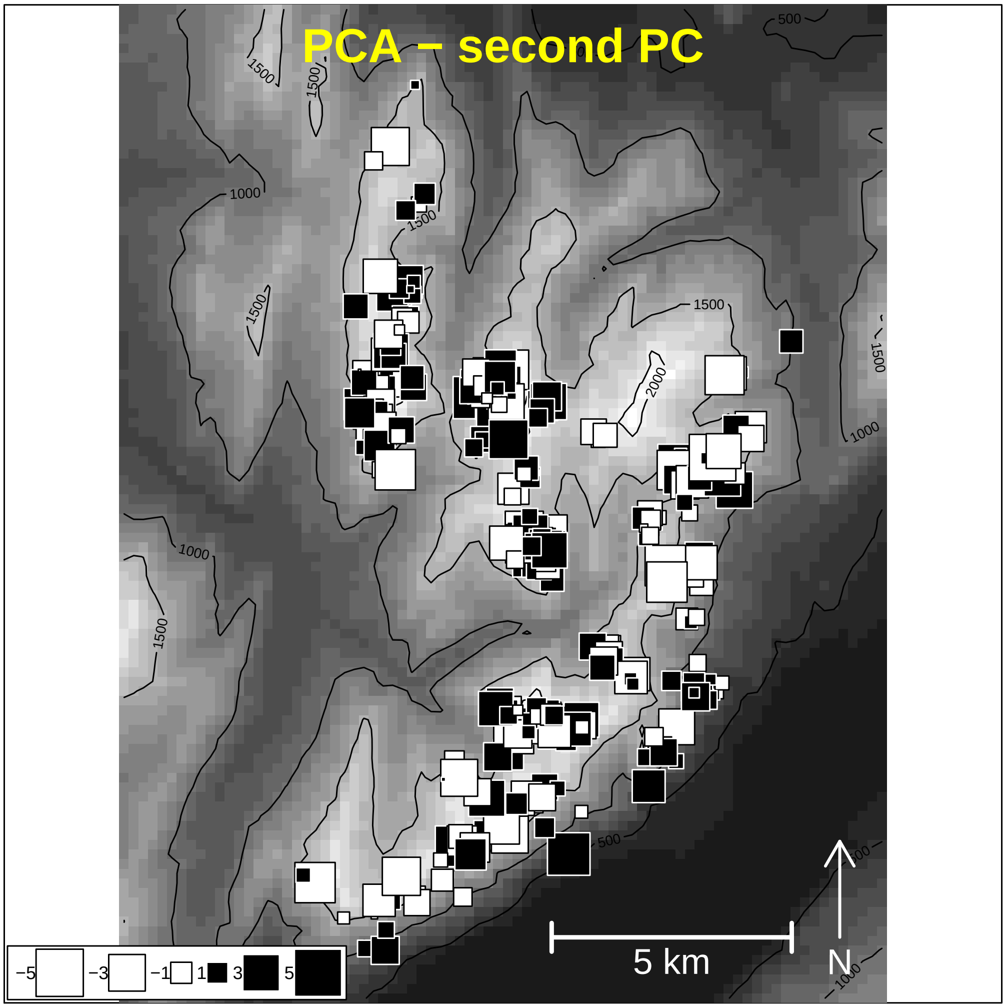


Apply this graphical representation to the first two PCs of the PCA:

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



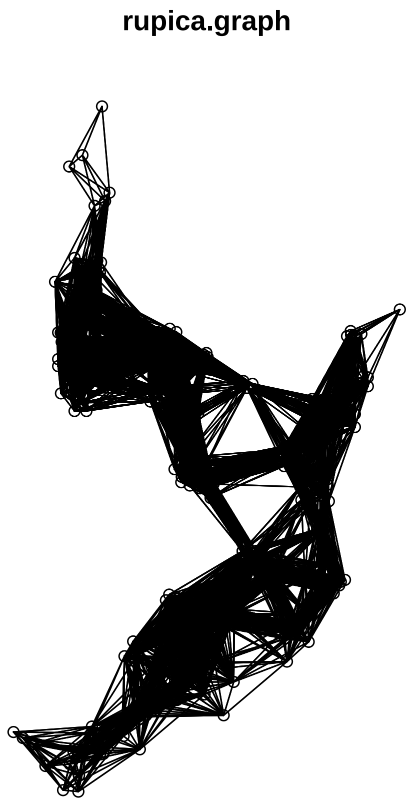
What can we say about spatial genetic structure as inferred by PCA? This visual assessment can be complemented by testing the spatial autocorrelation in the first PCs of PCA. This can be achieved using Moran's I test. Use the function `moran.mc` in the package *spdep* to perform these tests. You will need first to define the spatial connectivity between the sampled individuals. For these data, spatial connectivity is best defined as the overlap between home ranges of individuals. Home ranges will be modelled as disks with a radius of 1150m. Use `chooseCN` to create a connection network based on distance range ("neighbourhood by distance"). What threshold distance do you choose for individuals to be considered as neighbours?

```
rupica.graph <- chooseCN(other(rupica)$xy,type=5,d1=0,d2=2300, plot=FALSE,
                         res="listw")
```

The connection network should ressemble this:

```
rupica.graph

## Characteristics of weights list object:
## Neighbour list object:
```

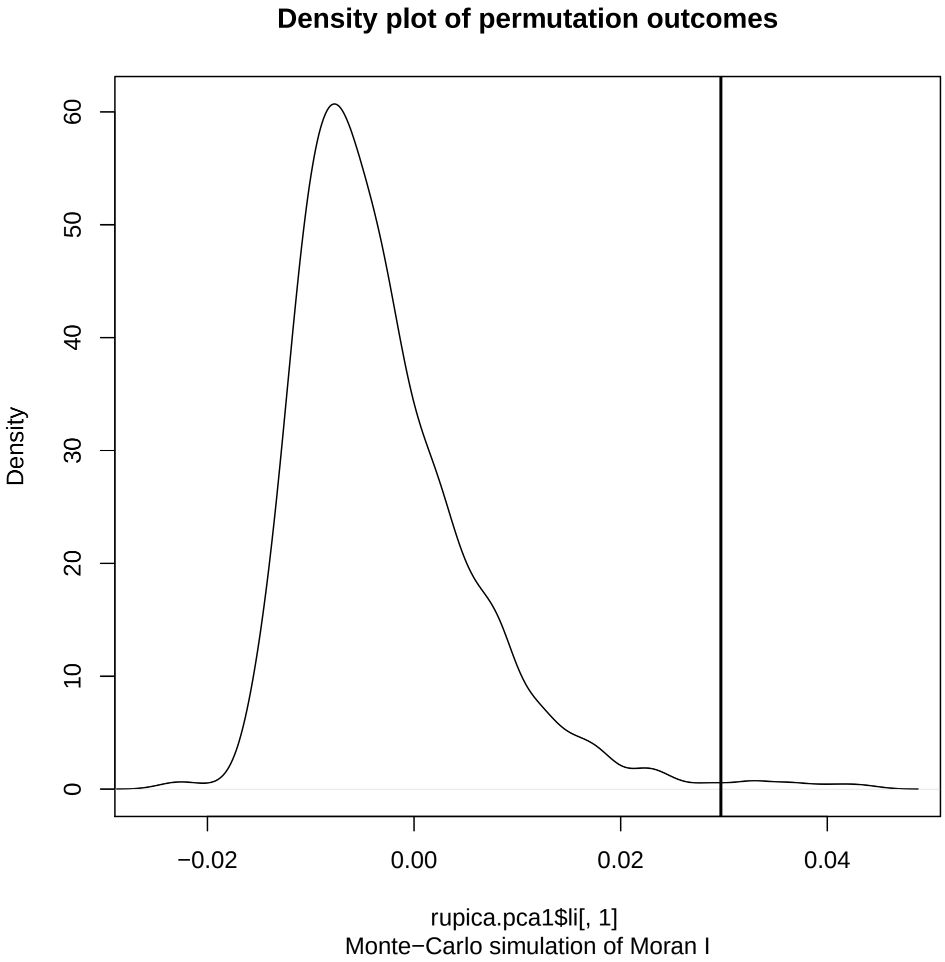
```
## Number of regions: 335
## Number of nonzero links: 18018
## Percentage nonzero weights: 16.05525
## Average number of links: 53.78507
##
## Weights style: W
## Weights constants summary:
##     n     nn  S0       S1      S2
## W 335 112225 335 15.04311 1352.07

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



rupica.graph

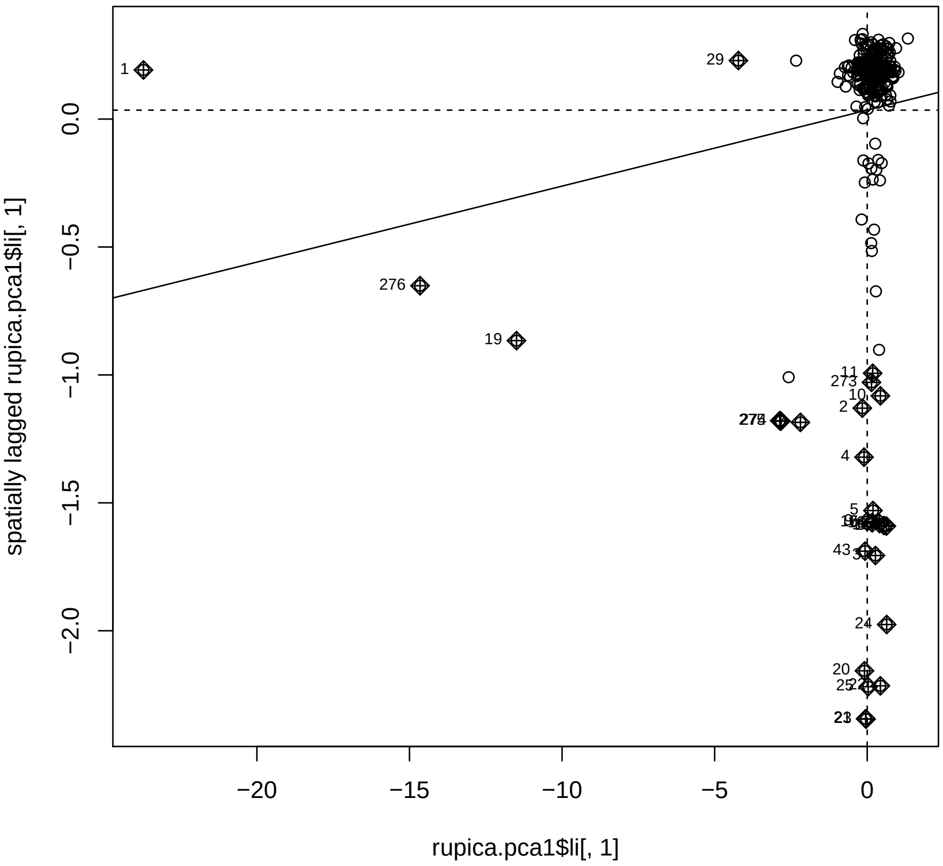
Perform Moran's test for the first two PCs, and plot the results. The first test should be significant:

```
pc1.mctest <- moran.mc(rupica.pca1$li[,1], rupica.graph, 999)
plot(pc1.mctest)
```



Compare this result to the mapping of the first PC of PCA. What is wrong? When a test gives unexpected results, it is worth looking into the data in more details. Moran's plot (`moran.plot`) plots the tested variable against its lagged vector. Use it on the first PC:

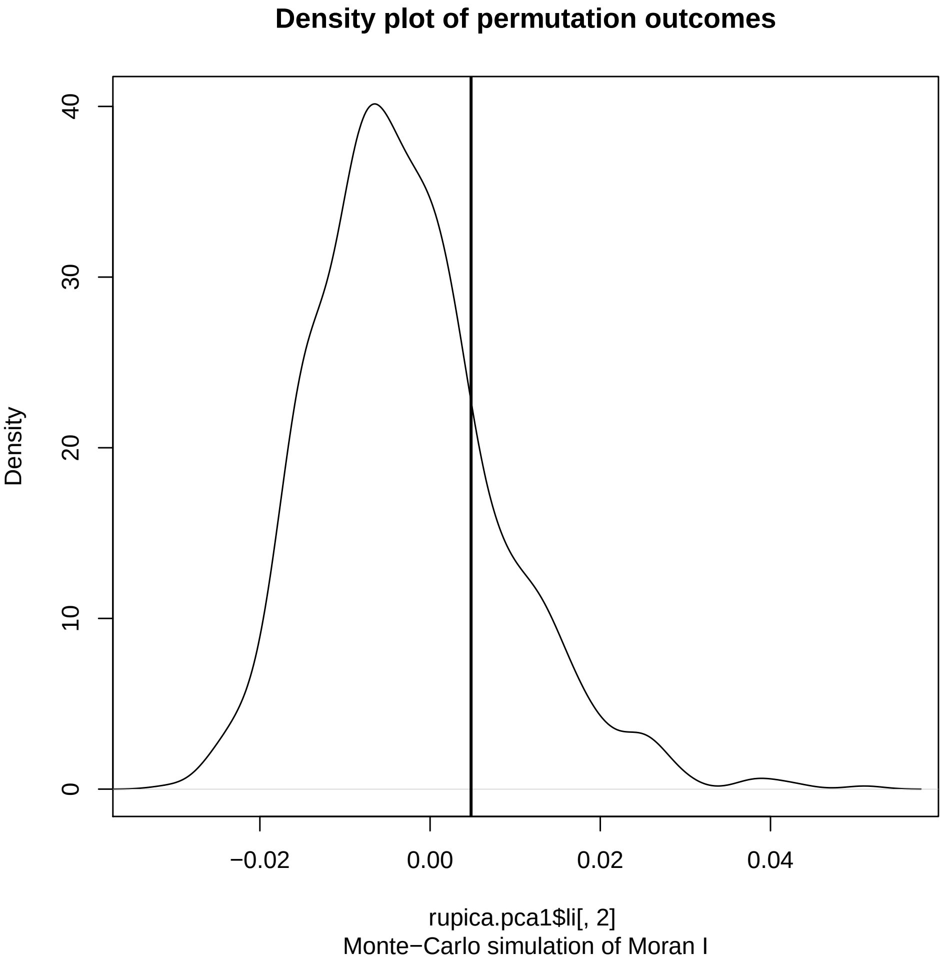
```
moran.plot(rupica.pca1$li[,1], rupica.graph)
```



Actual positive autocorrelation corresponds to a positive correlation between a variable and its lag vector. Is it the case here? How can we explain that Moran's test was significant?

Repeat these analyses for the second PC. What are your conclusions?

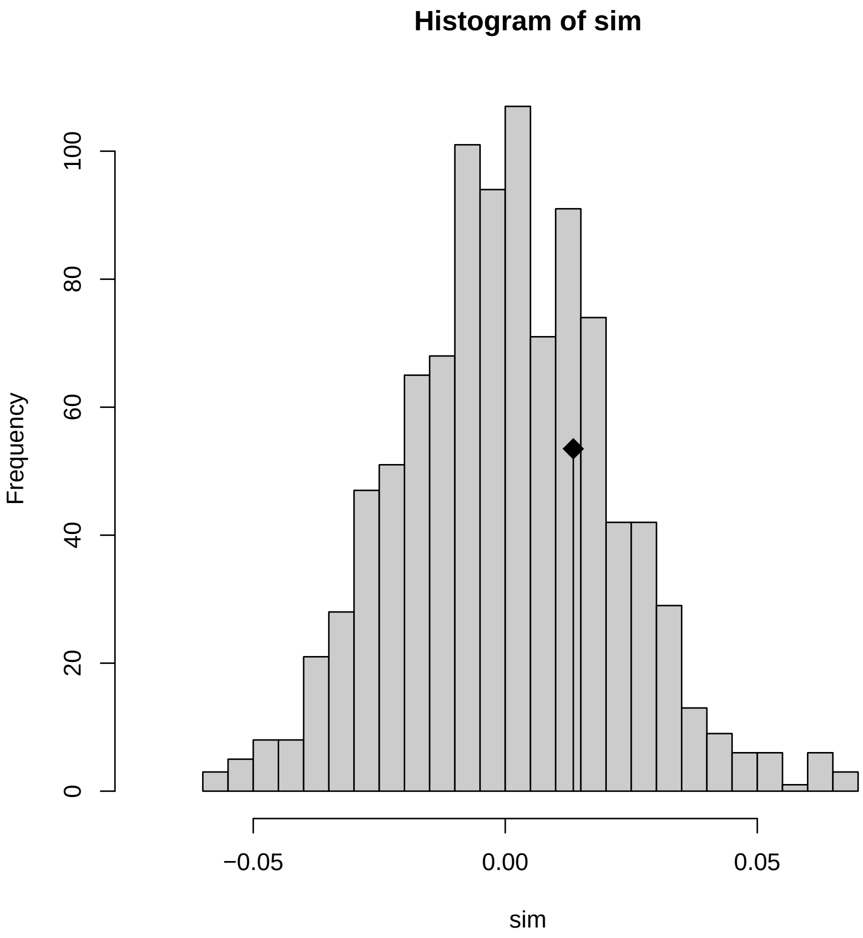
```
pc2.mctest <- moran.mc(rupica.pca1$li[,2], rupica.graph, 999)
pc2.mctest

##
##  Monte-Carlo simulation of Moran I
##
## data:  rupica.pca1$li[, 2]
## weights: rupica.graph
## number of simulations + 1: 1000
##
## statistic = 0.0048162, observed rank = 797, p-value = 0.203
## alternative hypothesis: greater

plot(pc2.mctest)
```



4 Multivariate tests of spatial structure

So far, we have only tested the existence of spatial structures in the first two principal components of a PCA of the data. Therefore, these tests only describe one fragment of the data, and do not encompass the whole diversity in the data. As a complement, we can use Mantel test (`mantel.randtest`) to test spatial structures in the whole data, by assessing the correlation between genetic distances and geographic distances. Pairwise Euclidean distances are computed using `dist`. Perform Mantel test, using the scaled genetic data you used before in PCA, and the geographic coordinates.

```
mtest <- mantel.randtest(dist(rupica.X), dist(other(rupica)$xy))
plot(mtest, nclass=30)
```



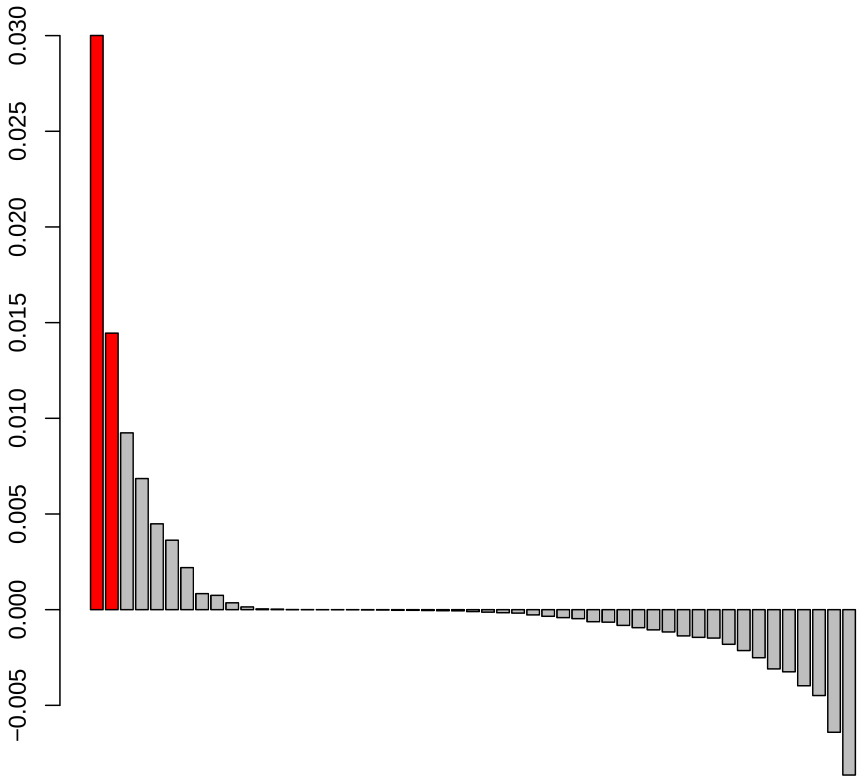
What is your conclusion? Shall we be looking for spatial structures? If so, how can we explain that PCA did not reveal them?

5 spatial Principal Component Analysis

The spatial Principal Component Analysis (sPCA, function spca [3]) has been especially developed to investigate hidden or non-obvious spatial genetic patterns. Like Moran's I test, sPCA first requires the spatial proximities between genotypes to be modeled. You will reuse the connection network defined previously using chooseCN, and pass it as the 'cn' argument of the function spca.

Read the documentation of spca, and apply the function to the dataset rupica. The function will display a barplot of eigenvalues:

```
rupica.spca1 <- spca(rupica, cn=rupica.graph,scannf=FALSE, nfposi=2,nfnega=0)
barplot(rupica.spca1$eig, col=rep(c("red","grey"), c(2,1000)) )
```



This figure illustrates the fundamental difference between PCA and sPCA. 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. Positive spatial autocorrelation correspond to greater genetic similarity between geographically closer individuals. Conversely, negative spatial autocorrelation corresponds to greater dissimilarity between neighbours. The spatial autocorrelation of a variable is measured by Moran's I, and interpreted as follows:

- $I_0 = -1/(n-1)$: no spatial autocorrelation (x is randomly distributed across space)
- $I > I_0$: positive spatial autocorrelation
- $I < I_0$: negative spatial autocorrelation

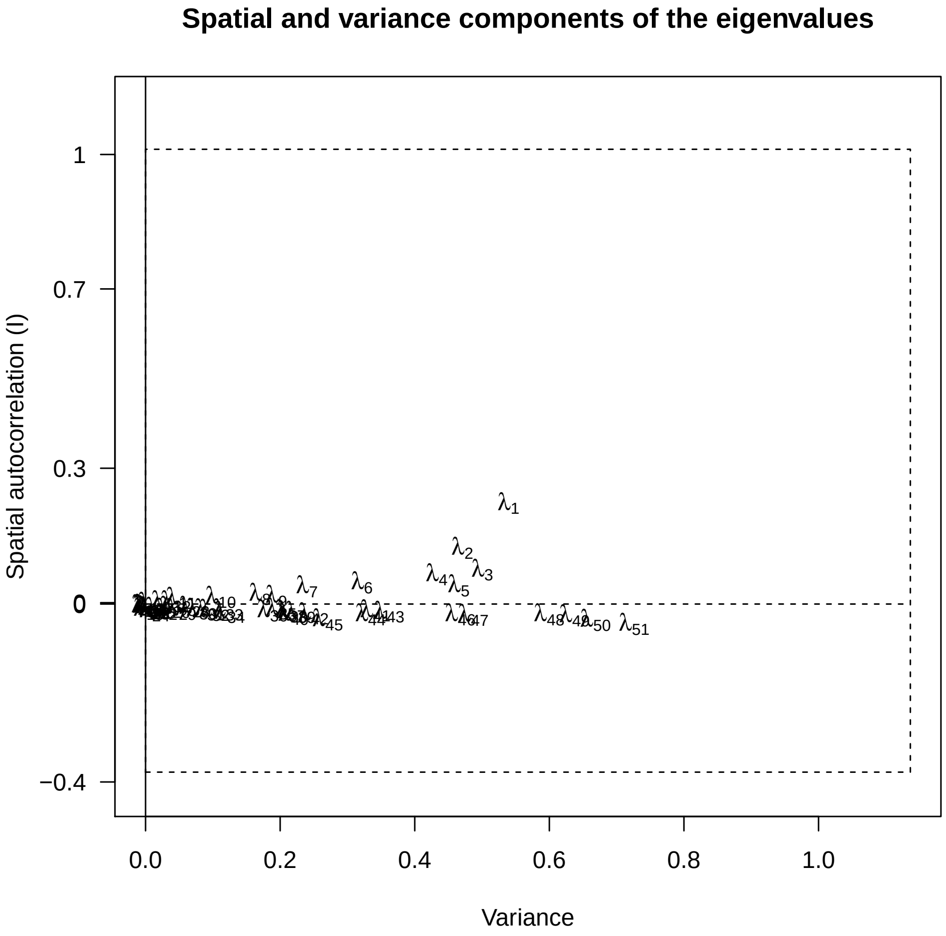
Principal components of PCA ensure that (ϕ referring to one PC) $var(\phi)$ is maximum. By contrast, sPCA provides PC which decompose the quantity $var(\phi)I(\phi)$. In other words, PCA focuses on variability only, while sPCA is a compromise between variability ($var(\phi)$) and spatial structure ($I(\phi)$).

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: 0.03001 0.01445 0.00924 0.006851 0.004485 ...
## Negative eigenvalues: -0.008643 -0.006407 -0.004488 -0.003977 -0.003248 ...
##
##   vector length mode    content
## 1 $eig   51     numeric eigenvalues
##
##   data.frame nrow ncol
## 1 $c1        55   2
## 2 $li        335  2
## 3 $ls        335  2
## 4 $as        2    2
##   content
## 1 principal axes: scaled vectors of alleles loadings
## 2 principal components: coordinates of entities ('scores')
## 3 lag vector of principal components
## 4 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 λ_1 indicates with no doubt a structure, the second eigenvalue, λ_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.

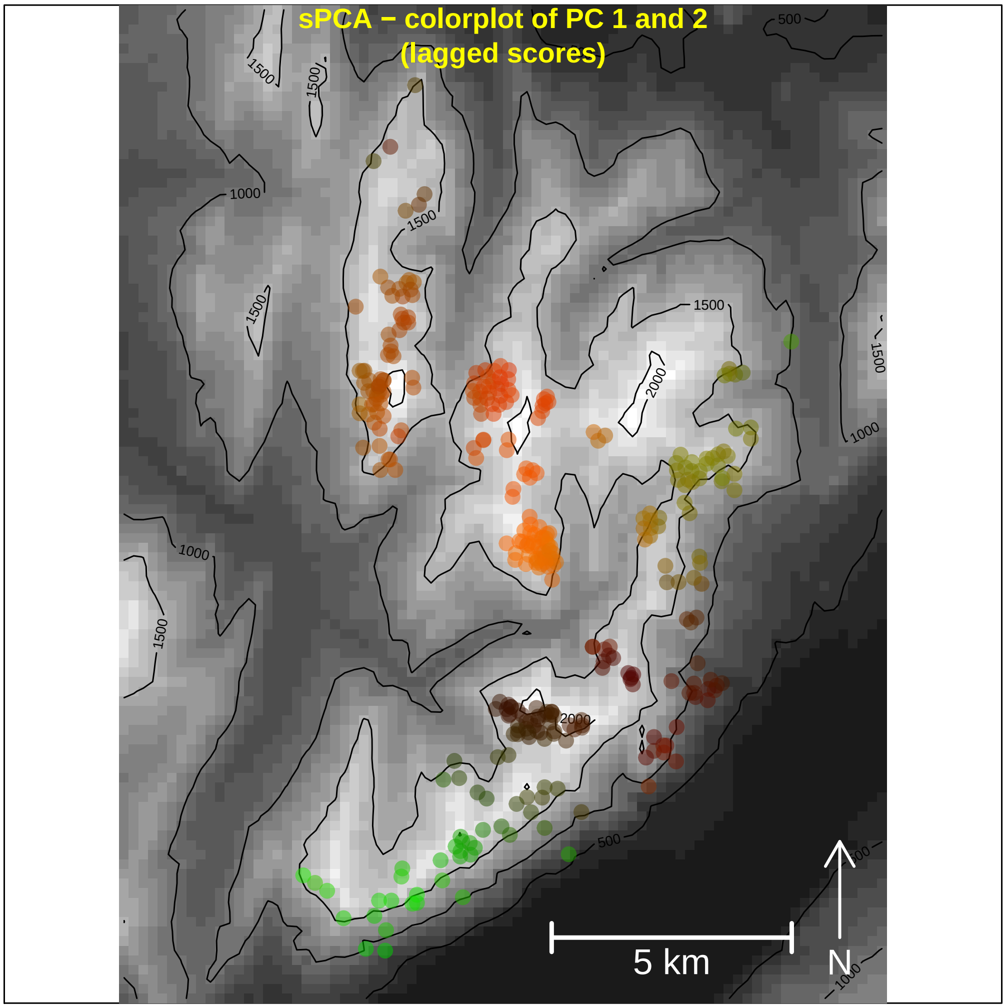
Try visualising the sPCA results as you did before with the PCA results. To clarify the possible spatial patterns, you can map the lagged PC ($ls) instead of the PC ($li), which are a 'denoisified' version of the PCs.

First, map the first principal component of sPCA. How would you interprete this result? How does it compare to the first PC of PCA? What inferrence can we make about the way the landscape influences gene flow in this population of Chamois?

Do the same with the second PC of sPCA. Some field observations suggest that this pattern 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 [1] (`?colorplot`). The final figure should ressemble this (although colors may change from one computer to another):

```
showBauges()
colorplot(other(rupica)$xy, rupica.spca1$ls, axes=1:2, transp=TRUE, add=TRUE,
          cex=2)
title("sPCA - colorplot of PC 1 and 2\n(lagged scores)", col.main="yellow",
      line=-2, cex=2)
```



6 To go further

More spatial genetics methods and are presented in the *basics tutorial* as well as in the tutorial dedicated to sPCA, which you can access from the *adegenet* website:
http://adegenet.r-forge.r-project.org/

or by typing:

```
adegenetTutorial("basics")
adegenetTutorial("spca")
```

References

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

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

[3] 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.

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