Workshop on population genomics using R: studying population structure using genetic data

In this workshop, we will analyse an original dataset, consisting of genetic data collected for more than 60 Reunion harriers (*Circus maillardi*). This species is endangered, and endemic to the island of Reunion in the Indian Ocean. Today, we will go through a few analyses that were carried to elucidate the population structure of this bird. Harriers are philopatric: the young birds tend to live and reproduce quite close from where they are born, but they are also excellent at flying.

The main question that we will investigate here is: do we observe any significant population structure in this species? **Can you guess why this is an important question when examining an endangered species?**

You will find the files needed for this workshop in a folder on Moodle, containing the data and a R file containing information to plot a map of Reunion.

Use a text editor like NotePad to open the VCF file entitled “data_for_workshop.vcf”. Have a look at this Wikipedia page: https://en.wikipedia.org/wiki/Variant_Call_Format . **Once you are at this step, call the lecturer for a more guided explanation.**

Now that we have clarified what the dataset represents, it is time to start the analyses. We will use a set of functions made available through several packages in the **R software**. R is one of the most popular programming languages used by biologists, and mastering it is a great skill to have. You will find below a list of instructions and commands. It is important at each step that you try to understand the language and the various manipulations we are doing. As for any language, it is much easier to passively understand it than to actively use it. Once you get more familiar with the syntax, the functions, the oddities, you will start feeling like you can actually write your own commands. You will find on the Moodle archives resources for using R (Research Skills, L5), and on the Aspects Moodle you will find a PDF explaining how to install and run R.

Through the tutorial, we will discuss and explain some of the statistical analyses that we are running here. We will also talk about these concepts again during our last lecture together on conservation genetics.

First, we need to install and load the packages we will need for our analyses. Libraries contain specialist functions that do not come with the base version of R. They allow biologists to do almost everything in terms of analyses, which is why R is so popular. The commands that follow do that:

```
install.packages("vcfR")
install.packages("poppr")
install.packages("ape")
install.packages("RColorBrewer")
install.packages("ggplot2")


library(vcfR)
library(poppr)
library(ape)
library(RColorBrewer)
```

We can run the install.packages() commands only once. It will install the functions. However, to load the functions inside R, we need to use the library() function every time we need to call functions that are specific to said library.

Now, we need to load our raw data in R. **If you do not remember how to specify your working directory, call the lecturer!**

```
cmail.VCF <- read.vcfR("data_for_workshop.vcf")
```

We then need to convert our VCF database into a format that is understood by the functions we need. This is done like this:

```
gl.cmail <- vcfR2genlight(cmail.VCF)
```

Next, you need to execute this command. **Can you guess what it does?**

```
ploidy(gl.cmail) <- 2
```

Here, we start creating a tree based on genetic distance. I recommend you have a look at the help menu for the function aboot(). You can call this menu by typing: ?aboot(). **Can you explain what each option does here?**

```
tree <- aboot(gl.cmail, tree = "upgma", distance = bitwise.dist, sample = 1
00, showtree = F, cutoff = 50, quiet = T)
```

Execute each of these two commands, one after the other.

```
plot.phylo(tree, cex = 0.8, font = 2, adj = 0)
nodelabels(tree$node.label, adj = c(1.3, -0.5), frame = "n", cex = 0.8,font
= 3, xpd = TRUE)
```

We produced a phylogenetic representation of genetic distances between our individuals. But there are other ways to obtain an idea of population structure. A very fast category of methods includes the Principal Component Analysis (PCA). This method is a transformation of the raw data that summarizes the main axes of genetic variance and projects each individual along these axes.

First we need to run the PCA, and assess the proportion of variance explained by each of the axis the transformation produced.

```
cmail.pca <- glPca(gl.cmail, nf = 3)
barplot(100*cmail.pca$eig/sum(cmail.pca$eig), col = heat.colors(50), main="
PCA Eigenvalues")
title(ylab="Percent of variance\nexplained", line = 2)
title(xlab="Eigenvalues", line = 1)
```

Call the lecturer when you are done with this. We will explain what this means.

You can then use R to create a graphic, as follows:

```
cmail.pca.scores <- as.data.frame(cmail.pca$scores)

library(ggplot2)
set.seed(9)
p <- ggplot(cmail.pca.scores, aes(x=PC1, y=PC2))
p <- p + geom_point(size=2)
p <- p + geom_hline(yintercept = 0)
p <- p + geom_vline(xintercept = 0)
p <- p + theme_bw()

p

###An easier version with the R base package:

plot(cmail.pca.scores$PC1, cmail.pca.scores$PC2)
```

Principal Component Analyses are an interesting way to summarize the data, but they do not tell us much about whether we can detect discrete populations in our datasets. There are several methods that estimate which model of population structure is the most likely. One of these methods is the Discriminant Analysis of Principal Components (DAPC). It starts with a PCA, but then applies a series of clustering algorithms that aim at identifying discrete groups of individuals that are closely related. There is a R package which can implement this sort of analysis. You can find a complete explanation of the method here: https://adegenet.r-forge.r-project.org/files/tutorial-dapc.pdf

First we use a function that will tell us the most likely number of genetic groups that can be found in our dataset.

```
grp <- find.clusters(gl.cmail,max.n.clust=5,n.pca=100)
```

Call the lecturer to make sense of the graphic. Then, replace the "X" in the expression below by the most likely number of clusters.

```
grp2 <- find.clusters(gl.cmail,n.clust=X,n.pca=100)

dapctest <- dapc(gl.cmail, grp2$grp,n.da=2, n.pca=100)
temp1 <- optim.a.score(dapctest)
dapc2 <- dapc(gl.cmail,grp2$grp,n.da=2, n.pca=1)

myCol <- c("darkblue","red")
scatter(dapc2, col=myCol,scree.da=FALSE, bg="white", pch=20, cell=0, cstar=
0, solid=.4,cex=3,clab=0, leg=TRUE)
```

We now want to combine the results from the PCA and from DAPC, to check that the clusters we detect with DAPC are consistent with the results we obtained with PCA. This requires us to combine information using a function called "merge()". We will use this function to merge the different datasets that contain the coordinates of individuals in the PCA, and the assignation of each individual to a specific cluster.

```
cmail.pca.scores <- as.data.frame(cmail.pca$scores)
cmail.pca.scores$ind<-row.names(cmail.pca.scores)

tmp<-cbind(as.data.frame(rownames(dapc2$posterior)),as.data.frame(dapc2$pos
terior))
colnames(tmp)<-c("ind","Cluster_1","Cluster_2")

cmail.pca.scores<-merge(cmail.pca.scores,tmp,by="ind")
cmail.pca.scores$pop <- as.character(round(cmail.pca.scores$Cluster_1)+1)
cols <- brewer.pal(n = 3, name = "Dark2")


library(ggplot2)
set.seed(9)
p <- ggplot(cmail.pca.scores, aes(x=PC1, y=PC2, colour=pop))
p <- p + geom_point(size=2)
p <- p + geom_hline(yintercept = 0)
p <- p + geom_vline(xintercept = 0)
p <- p + theme_bw()
p <- p + scale_color_manual(values = cols)
p
```

What do you think of the consistency between the PCA and DAPC clustering?

Now, what can we say about the geographic structure here? We need now to have a look at where these samples fall on the map.

```
###We load the coordinates for our individuals
coord<-read.table("coordinates_harriers_workshop.txt",h=T)
library(maps)

load("REU_adm0.RData")
reu_obj<-gadm
tomap=merge(coord, cmail.pca.scores,by="ind")
```

```
plot(reu_obj)
points(tomap$LON,tomap$LAT,col=tomap$pop,pch=16,cex=2)

legend("topright", legend=c("Cluster 1", "Cluster 2"), col=levels(as.factor
(tomap$pop)), pch=16, cex=2)
```

Look on Google at a map of Reunion island. What could be a possible explanation for our observations?

If you want to learn more about spatially-explicit clustering analyses, go to https://bcm-uga.github.io/TESS3_encho_sen/articles/main-vignette.html and try the tutorial.

To go further with the analysis of genomic variants:
https://grunwaldlab.github.io/Population_Genetics_in_R/gbs_analysis.html