Example script associated with Gibert et al. in Palaeontology

Corentin Gibert

Manuscript: A coherent biogeographic framework for Old World Neogene and Pleistocene mammals

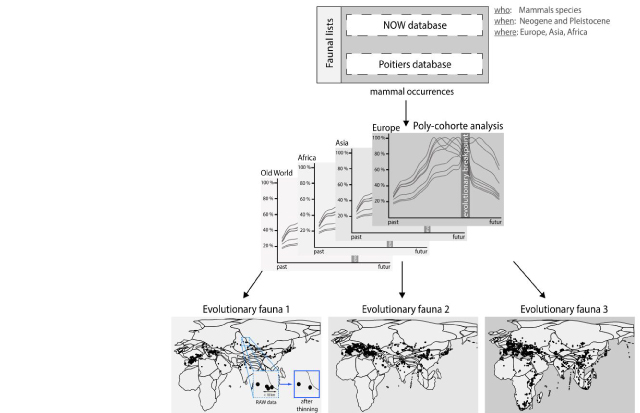
### The European Example :

Here is a Rmarkdown script designed to reproduce the temporal and spatial analysis computed in the main manuscript. Supplementary Figures **4A** and **4B** (**Appendix 1**) summarize the successive steps conducted in this script. In order to keep this script as short as possible, this example focus on the Old World and European scale, but the same analysis can be computed with this script at the Asian and African scales.

## First step: Cleaning of database

Example for cleaning dataset at the Old World scale. Please use the NOWBDD.txt file included in Supplementary Materials and place it in the working directory.

library(stringr)  
library(plyr)  
library(matlib)  
source("Personal\_function.R")  
  
##A##  
#Import of NOW Database, here NOWBDD.txt regroup all occurrences of mammals in the Old World during Miocene, Pliocene and Pleistocene.  
#Here we double the infos about locality names to keep them through the thinning process  
NOWBDD <- read.table("NOWBDD.txt", h = T)   
NomLoc <- NOWBDD$NAME  
NOWBDD <- cbind(NOWBDD, NomLoc)  
head(NOWBDD)  
  
##B##  
#Poorly time-constrained localities are removed from the analysis: the one lasting more than 5 Ma (i.e. the mean duration of a geological stage/age)  
wantTime <- which((NOWBDD$MAX\_AGE - NOWBDD$MIN\_AGE) > 5)  
NOWBDD <- NOWBDD[-wantTime,]  
if(length(which((NOWBDD$MAX\_AGE - NOWBDD$MIN\_AGE) > 5)) == 0)  
{  
 print("OK")  
}  
  
##C##  
#Thinning at 0.1° of latitude and longitude (i.e. close localities are regrouped into a unique locality at ~10km scale)  
Min\_LAT <- floor(min(NOWBDD$LAT, na.rm = TRUE))  
Max\_LAT <- ceiling(max(NOWBDD$LAT, na.rm = TRUE))  
Min\_LNG <- floor(min(NOWBDD$LONG, na.rm = TRUE))  
Max\_LNG <- ceiling(max(NOWBDD$LONG, na.rm = TRUE))  
Grain01.LAT <- seq(from = Min\_LAT, to = Max\_LAT, 0.1)  
Grain01.LNG <- seq(from = Min\_LNG, to = Max\_LNG, 0.1)  
Liste.GRAIN01 <- list()   
for(j in 1:(length(Grain01.LAT) - 1))  
{  
 for(k in 1:(length(Grain01.LNG) - 1))  
 {  
 want <- which(NOWBDD$LAT >= Grain01.LAT[j] & NOWBDD$LAT < Grain01.LAT[j+1]   
 & NOWBDD$LONG >= Grain01.LNG[k] & NOWBDD$LONG < Grain01.LNG[k+1])  
 if(length(want) != 0)  
 {  
 names.Grain01.dataframe.eu <- paste("WORLD", Grain01.LAT[j], Grain01.LNG[k], sep = "\_")  
 N <- assign(names.Grain01.dataframe.eu, NOWBDD[want,])  
 Liste.GRAIN01[[(length(Liste.GRAIN01)+1)]] <- N #   
 rm(names.Grain01.dataframe.eu)  
 }  
 }  
 print(paste(ceiling((j/(length(Grain01.LAT) - 1) \*100)), " % "))  
}  
  
save(Liste.GRAIN01, file = "ListeGRAIN01.RData") #Thinned dataset is extracted to save time (Thinning is a long process)  
load("ListeGRAIN01.RData") #Can be loaded after it.  
  
##D##  
#Grouped localities are renamed with thinned latitude\_longitude values as new names (e.g. 36.5\_5.7).  
for(i in 1:length(Liste.GRAIN01))  
{  
 titre <- paste(round(Liste.GRAIN01[[i]][1,]$LAT, digits = 1), round(Liste.GRAIN01[[i]][1,]$LONG, digits = 1), sep = "\_")  
 for(j in 1:length(Liste.GRAIN01[[i]]))  
 {  
 Liste.GRAIN01[[i]][j]$NAME <- titre  
 }  
}  
  
FULL\_GRAIN01 <- data.frame()  
for(i in 1:length(Liste.GRAIN01))  
{  
 FULL\_GRAIN01 <- join(FULL\_GRAIN01, Liste.GRAIN01[[i]], match="all", type="full")  
 print(paste(ceiling((i/(length(Liste.GRAIN01) - 1) \*100)), " % "))  
}  
str(FULL\_GRAIN01)  
  
##E##  
# Marine mammals, bats and indet. genus are removed from dataset  
table(FULL\_GRAIN01$FAMILY)  
  
wantOUT<-which(FULL\_GRAIN01$ORDER=="Chiroptera"|FULL\_GRAIN01$ORDER=="Cetacea"|FULL\_GRAIN01$FAMILY=="Phocidae"| FULL\_GRAIN01$FAMILY=="Vespertilionidae"| FULL\_GRAIN01$ORDER=="Indet"|FULL\_GRAIN01$GENUS=="indet."| FULL\_GRAIN01$FAMILY=="Balaenopteridae"|FULL\_GRAIN01$FAMILY=="Rhinolophidae"| FULL\_GRAIN01$FAMILY=="Dugongidae"|FULL\_GRAIN01$FAMILY=="Ziphiidae"| FULL\_GRAIN01$FAMILY=="Balaenidae"|FULL\_GRAIN01$FAMILY=="Eurhinodelphinidae"| FULL\_GRAIN01$FAMILY=="Odobenidae"|FULL\_GRAIN01$FAMILY=="Cetotheriidae"| FULL\_GRAIN01$FAMILY=="Squalodontidae"|FULL\_GRAIN01$FAMILY=="Physeteridae"| FULL\_GRAIN01$FAMILY=="Hipposideridae"|FULL\_GRAIN01$FAMILY=="Monodontidae"| FULL\_GRAIN01$FAMILY=="Megadermatidae"|FULL\_GRAIN01$FAMILY=="Tranatocetidae"| FULL\_GRAIN01$FAMILY=="Platanistidae"|FULL\_GRAIN01$FAMILY=="Kentriodontidae"|FULL\_GRAIN01$FAMILY=="Molossidae"|FULL\_GRAIN01$FAMILY=="Eschrichtiidae"|FULL\_GRAIN01$FAMILY=="Desmostylidae"|FULL\_GRAIN01$FAMILY=="Eoplatanistidae"|FULL\_GRAIN01$FAMILY=="Kogiidae"|FULL\_GRAIN01$FAMILY=="Phocoenidae"| FULL\_GRAIN01$FAMILY=="Pontoporiidae"|FULL\_GRAIN01$FAMILY=="Squalodelphinidae"|FULL\_GRAIN01$FAMILY=="Emballonuridae"|FULL\_GRAIN01$FAMILY=="Trichechidae"| FULL\_GRAIN01$FAMILY=="Dalpiazinidae"|FULL\_GRAIN01$FAMILY=="Acrodelphidae"| FULL\_GRAIN01$FAMILY=="Patriocetidae"|FULL\_GRAIN01$FAMILY=="Hyperoodontidae"| FULL\_GRAIN01$FAMILY=="Bohlininae"|FULL\_GRAIN01$GENUS=="Pinocetus"|FULL\_GRAIN01$GENUS=="Isocetus"| FULL\_GRAIN01$GENUS=="Scaldicetus"|FULL\_GRAIN01$GENUS=="Tagicetus"|FULL\_GRAIN01$GENUS=="Uranocetus"| FULL\_GRAIN01$GENUS=="Hoplocetus"|FULL\_GRAIN01$GENUS=="Aglaocetus"|FULL\_GRAIN01$GENUS=="Diorocetus"| FULL\_GRAIN01$GENUS=="Graamocetus"|FULL\_GRAIN01$GENUS=="Pelocetus"|FULL\_GRAIN01$GENUS=="Phococetus"| FULL\_GRAIN01$GENUS=="Gen.")  
  
Grain01.terre<-FULL\_GRAIN01[-wantOUT,]  
  
#We make sure that our selection in NOWdatabase of Old World, Neogene mammals was effective by removing potential Oligocene taxa.  
Grain01.terre <- subset(Grain01.terre, Grain01.terre$MIN\_AGE < 25 & Grain01.terre$MAX\_AGE < 29)  
Grain01.terre <- subset(Grain01.terre, Grain01.terre$LAT > -40 & Grain01.terre$LAT < 80 &   
 Grain01.terre$LONG > -20 & Grain01.terre$LONG < 180)   
  
##F##  
#Removal of indet. species for next poly-cohort analysis.   
ch <- str\_detect(Grain01.terre$SPECIES, "\_") #Allow to detect undefined taxa names, ex: sp.\_1   
Grain01.terre <- Grain01.terre[-which(ch == TRUE),]  
  
Data\_PolyFULL <- Grain01.terre  
  
want <- which(Data\_PolyFULL$SPECIES == "Indet."| Data\_PolyFULL$SPECIES == "indet." | Data\_PolyFULL$SPECIES == "indet" | Data\_PolyFULL$SPECIES == "Indet" | Data\_PolyFULL$SPECIES == "sp." | Data\_PolyFULL$SPECIES == "SP." | Data\_PolyFULL$SPECIES == "sp" | Data\_PolyFULL$SPECIES == "SP" | Data\_PolyFULL$GENUS == "indet." |  
Data\_PolyFULL$GENUS == "indet" | Data\_PolyFULL$GENUS == "Indet." | Data\_PolyFULL$GENUS == "Indet" | Data\_PolyFULL$GENUS == "Gen.")  
  
want  
Data\_PolyFULL <- Data\_PolyFULL[-want,]  
str(Data\_PolyFULL)  
  
##G##  
#Later a final step of cleaning is computed before spatial analysis by removing localities with less than 5 species

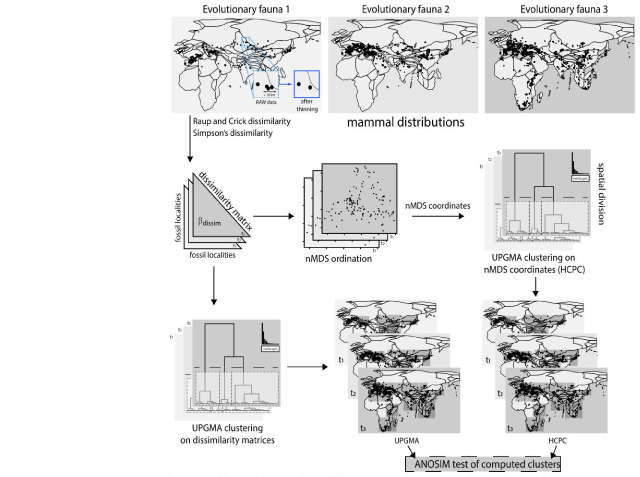


First part of Supplementary Figure 4A, description of First and Second steps of the analytical process described in this script. The temporal analysis

## Second step : Computation of Evolutionary Faunas via Poly-cohort analysis:

Here we focus on the Old World and European Scale, we compute poly-cohort analysis with two specific functions, one included in Supplementary Material (“Polycohorte\_matrix.R”) designed by me and one private function developed by Claude Monnet (“divCohorts.R”).  
source("divCohorts.R") #This function is not included in this script because it is an unpublished R function writed by macroevolution specialist  
#Claude Monnet (mail: Claude.Monnet@univ-lille1.fr). It has been imported to R from an original article of Escarguel and Legendre (2006) in  
#Strata. The functionning of this method (poly-cohort analysis) is described in Supplementaty Figure 3 (Appendix 1). Please ask Claude Monnet to obtain divCohort.R function.   
  
##A##  
#Looking at origination/extinction distribution in dataset  
temp\_analyse\_MN <- c(Grain01.terre$MAX\_AGE, Grain01.terre$MIN\_AGE)  
plot(table(temp\_analyse\_MN), xlab = "Origination/Extinction datum (Ma)", ylab = "Frequency")  
#MN(EQ) boundaries (See Supplementary Figure 1)  
AGE\_MN <- c(0, 1.95, 2.5, 3.55, 5, 5.3, 7.1, 7.6, 8.9, 9.9, 11.2, 12.85, 14.2, 16.4, 17.2, 19.5, 21.7, 23.8)  
NOM\_MN <- c("MN18","MN17","MN16","MN15","MN14","MN13","MN12","MN11","MN10","MN9","MN7/8","MN6","MN5","MN4","MN3","MN2","MN1")  
#divCohort.R use GENUS columns, in order to compute poly-cohort analysis on species level, we paste Genus and Species names  
Data\_PolyFULL$GENUS <- paste(Data\_PolyFULL$GENUS, Data\_PolyFULL$SPECIES, sep = "\_")  
  
##B##  
#Construction of matrix for polycohort analysis  
Poly\_FULL\_SP <- Polycohorte\_matrix(Data\_PolyFULL, AGE\_MN, NOM\_MN)  
head(Poly\_FULL\_SP) #This matrix has species names in rows and MN(EQ) in columns  
wantOUT <- c() #Looking for species outside the temporal range (Miocene+Pliocene+Pleistocene+Recent), they are removed.  
for(i in 1:length(Poly\_FULL\_SP[,1]))  
{  
 if(sum(Poly\_FULL\_SP[i,]) == 0)  
 {  
 wantOUT <- c(wantOUT, i)  
 }  
}  
Poly\_FULL\_SP <- Poly\_FULL\_SP[-wantOUT,]  
str(Poly\_FULL\_SP)  
  
#We reverse matrix column order in order to have the older MN(EQ) in first columns.  
Poly\_FULL\_SPinv <- matrix(nrow = 5097, ncol = 17, data = 0)   
for(i in 1:length(Poly\_FULL\_SP[1,]))  
{  
 Poly\_FULL\_SPinv[,i] <- Poly\_FULL\_SP[,((length(Poly\_FULL\_SP[1,])+1)-i)]  
}  
NOM\_rev <- rev(NOM\_MN)  
RowName\_rev <- rownames(Poly\_FULL\_SP)  
dimnames(Poly\_FULL\_SPinv) <- list(RowName\_rev, NOM\_rev)  
  
##C##  
#Poly-cohort analysis for origination and extinction.  
SURVI.FULL\_SP <- divCohorts(Poly\_FULL\_SPinv)  
PRENA.FULL\_SP <- divCohorts(Poly\_FULL\_SPinv, way = "backward")  
  
#divCohorts.R function being private (please ask Claude Monnet to obtain this function), the two output files below are included in Supplementary Material  
write.table(SURVI.FULL\_SP, file = "Poly\_SURVI\_WORLD\_SP\_NOW.txt", sep = " ", row.names = TRUE, col.names = TRUE)  
write.table(PRENA.FULL\_SP, file = "Poly\_PRENA\_WORLD\_SP\_NOW.txt", sep = " ", row.names = TRUE, col.names = TRUE)  
  
########################################################################################  
################################# ANALYSIS at the European Scale #######################  
########################################################################################  
  
#Select European occurrences  
FULLeurope <- subset(Grain01.terre, Grain01.terre$LAT < 80 & Grain01.terre$LAT > 30 & Grain01.terre$LONG > - 20 & Grain01.terre$LONG < 50)  
FULLeurope$GENUS <- paste(FULLeurope$GENUS, FULLeurope$SPECIES, sep = "\_")

##D##  
#Cleaning dataset at species scale  
want <- which(FULLeurope$SPECIES == "Indet."| FULLeurope$SPECIES == "indet." | FULLeurope$SPECIES == "indet"  
 | FULLeurope$SPECIES == "Indet" | FULLeurope$SPECIES == "sp." | FULLeurope$SPECIES == "SP." |  
 FULLeurope$SPECIES == "sp" | FULLeurope$SPECIES == "SP" | FULLeurope$GENUS == "indet." |  
 FULLeurope$GENUS == "indet" | FULLeurope$GENUS == "Indet." | FULLeurope$GENUS == "Indet" | FULLeurope$SPECIES == "Gen.")  
FULLeurope <- FULLeurope[-want,]  
  
##E##  
#Construction of matrix for polycohort analysis  
FULLeurope\_SP <- Polycohorte\_matrix(FULLeurope, AGE\_MN, NOM\_MN)  
  
#Looking for species outside the temporal range (Miocene+Pliocene+Pleistocene+Recent), they are removed.  
wantOUT <- c()  
for(i in 1:length(FULLeurope\_SP[,1]))  
{  
 if(sum(FULLeurope\_SP[i,]) == 0)  
 {  
 wantOUT <- c(wantOUT, i)  
 }  
}  
FULLeurope\_SP <- FULLeurope\_SP[-wantOUT,]  
  
  
FULLeurope\_SPinv <- matrix(nrow = 2801, ncol = 17, data = 0) #We reverse matrix column order in order to have the older MN(EQ) in first columns.  
for(i in 1:length(FULLeurope\_SP[1,]))  
{  
 FULLeurope\_SPinv[,i] <- FULLeurope\_SP[,((length(FULLeurope\_SP[1,])+1)-i)]  
}  
NOM\_rev <- rev(NOM\_MN)  
RowName\_rev <- rownames(FULLeurope\_SP)  
dimnames(FULLeurope\_SPinv) <- list(RowName\_rev, NOM\_rev)  
  
##F##  
#Poly-cohort analysis for origination and extinction.  
SURVI.EUROPE\_SP <- divCohorts(FULLeurope\_SPinv)  
PRENA.EUROPE\_SP <- divCohorts(FULLeurope\_SPinv, way = "backward")  
  
##G##  
#Saving of poly-cohorts results  
#divCohorts.R function being private (please ask Claude Monnet to obtain this function),   
#the two output files below are included in Supplementary Material  
write.table(SURVI.EUROPE\_SP, file = "Poly\_SURVI\_EUROPE\_SP\_NOW.txt", sep = " ", row.names = TRUE, col.names = TRUE)  
write.table(PRENA.EUROPE\_SP, file = "Poly\_PRENA\_EUROPE\_SP\_NOW.txt", sep = " ", row.names = TRUE, col.names = TRUE)

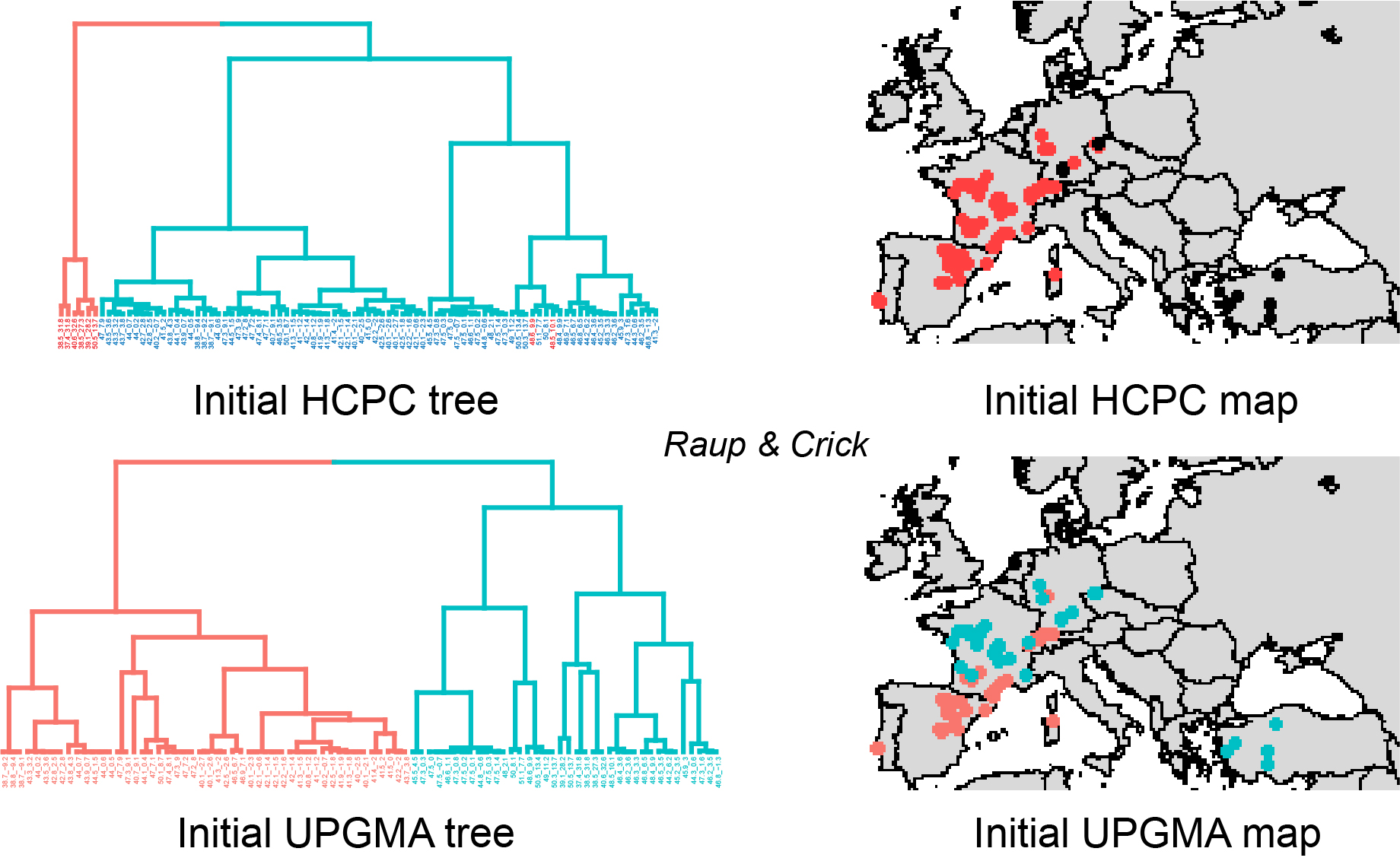


Second part of Supplementary Figure 4A, description of Third and Fourth step of the analytical process described in this script below. The spatial analysis

## Third step : Dissimilarity matrix, nMDS ordination, and UPGMA clustering:

We focus here at the European scale, same analysis can be conducted with Old World, Asian and African scale with similar script. Evolutionary faunas MN(EQ) must be changed. Evolutionary faunas are analysed one by one, we give here the script to compute spatial analysis on Evolutionary Fauna 1 (EF1).

library(vegan)  
library(factoextra)  
library(fossil)  
  
FULLeurope <- subset(Grain01.terre, Grain01.terre$LAT < 80 & Grain01.terre$LAT > 30 & Grain01.terre$LONG > - 20 & Grain01.terre$LONG < 50)  
  
##A##  
#Cleaning of European dataset  
want <- which(FULLeurope$SPECIES == "Indet."| FULLeurope$SPECIES == "indet." | FULLeurope$SPECIES == "indet"  
 | FULLeurope$SPECIES == "Indet" | FULLeurope$SPECIES == "sp." | FULLeurope$SPECIES == "SP." |  
 FULLeurope$SPECIES == "sp" | FULLeurope$SPECIES == "SP" | FULLeurope$GENUS == "indet." |  
 FULLeurope$GENUS == "indet" | FULLeurope$GENUS == "Indet." | FULLeurope$GENUS == "Indet" | FULLeurope$SPECIES == "Gen.")  
FULLeurope <- FULLeurope[-want,]  
  
##B##  
#Selection of the first evolutionary fauna (EF1)  
#EF1 starts at the beginning of MNEQ2, fade away at the end of MNEQ3.  
#Taxa that lived during the EF1 are included even if they appeared before or disappeared after EF1.  
#To analyse other evolutionary fauna bioregion distributions, please change lower and upper limits.  
Faune1.EU <- data.frame()  
wantFaune1 <- which(17.2 >= FULLeurope$MIN\_AGE & 17.2 < FULLeurope$MAX\_AGE & 21.7 >= FULLeurope$MAX\_AGE  
 | 17.2 <= FULLeurope$MIN\_AGE & 21.7 > FULLeurope$MIN\_AGE & 21.7 <= FULLeurope$MAX\_AGE  
 | 17.2 >= FULLeurope$MIN\_AGE & 21.7 <= FULLeurope$MAX\_AGE  
 | 17.2 < FULLeurope$MIN\_AGE & 21.7 > FULLeurope$MAX\_AGE)   
  
Faune1.EU <- FULLeurope[wantFaune1,]   
head(Faune1.EU)  
  
##C##  
#Construction of the Presence, absence matrix + final step of cleaning: only localities with > 5 species are retained  
MatP.A.Faune1.EU <- Presence\_Absence\_matrix(Faune1.EU, type = "Species", singletons = TRUE, min5 = 5)  
#Two localities from middle east are removed.  
MatP.A.Faune1.EU <- MatP.A.Faune1.EU[-c(1,2),]  
  
##D##  
#Raup&Crick Index  
#Computation of Distance matrix  
Dist.Faune1.EU <- raupcrick(MatP.A.Faune1.EU) #Distance matrix for Raup&Crick index  
mds.Faune1.EU.RC <- cmdscale(Dist.Faune1.EU, eig = TRUE, x.ret= TRUE) #PCA to look at variation partitioning  
plot((mds.Faune1.EU.RC$eig/(sum(mds.Faune1.EU.RC$eig)))\*100, ylab = "Percentage of variance associated with component", xlab = "Axes/Components (EUROPE EF 1)") #Plot of eigenvalues  
  
#Two components explain 67 % of total variance. Five components explain, individually, more than 10 % of variance.  
plot(mds.Faune1.EU.RC$point)  
  
##E##  
#nMDS computation  
#K = 5 (more homogeneous distribution of localities inside clusters in comparison with K = 2, specifically in Anatolian region)  
mds.Faune1.EU.RC\_K5 <- metaMDS(Dist.Faune1.EU, try = 1000, k = 5, maxit = 2000, trace = 2, sratmax = 0.99999999)  
plot(mds.Faune1.EU.RC\_K5$points)  
text(x = mds.Faune1.EU.RC\_K5$points[,1], y = mds.Faune1.EU.RC\_K5$points[,2],   
 labels = names(mds.Faune1.EU.RC\_K5$points[,1]), cex = 0.7)  
stressplot(mds.Faune1.EU.RC\_K5)  
text(x = 0.4, y = 0.65, labels = paste("Stress: ",mds.Faune1.EU.RC\_K5$stress))  
  
##F##  
#Simpson Index  
#Computation of Distance matrix  
MatP.A.Faune1.EU\_Inv <- t(MatP.A.Faune1.EU)  
Dist.Faune1.EU.Simp <- ecol.dist(MatP.A.Faune1.EU\_Inv, method = simpson, type = "dis")  
mds.Faune1.EU.Simp <- cmdscale(Dist.Faune1.EU.Simp, eig = TRUE, x.ret= TRUE)  
plot((mds.Faune1.EU.Simp$eig/(sum(mds.Faune1.EU.Simp$eig)))\*100, ylab = "Percentage of variance associated with each component", xlab = "Axes/Components (EUROPE EF 1)") #Plot of eigenvalues  
plot(mds.Faune1.EU.Simp$point)#Four components explain more than 75 % of total variance.  
  
##G##  
#Simpson index nMDS  
#K = 4  
mds.Faune1.EU.SIMP\_K5 <- metaMDS(Dist.Faune1.EU.Simp, try = 1000, k = 4, maxit = 2000, trace = 2, sratmax = 0.99999999)  
plot(mds.Faune1.EU.SIMP\_K5$points)  
  
##H##  
#Save of nMDS (computation can be long)  
save(mds.Faune1.EU.SIMP\_K5, file = "nMDS\_EU1\_SIMPSON.RData")  
save(mds.Faune1.EU.RC\_K5, file = "nMDS\_EU1\_RAUPCRICK.RData")

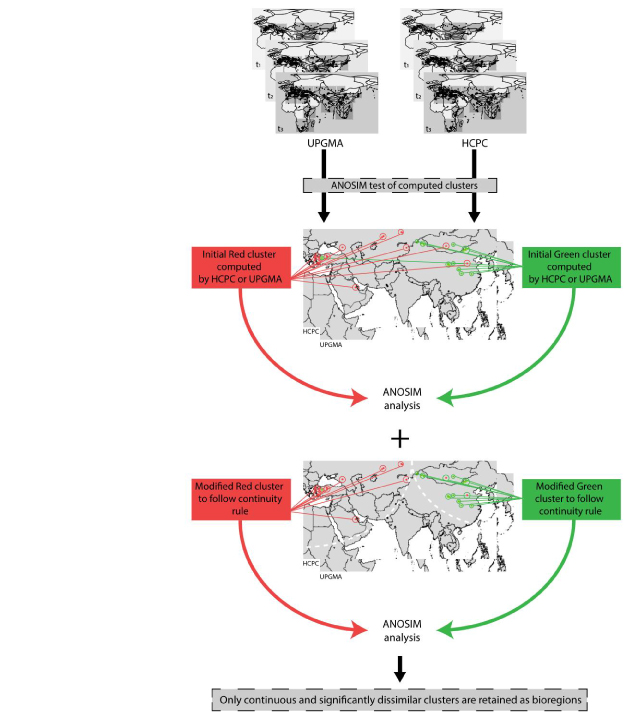


Initial HCPC and UPGMA trees associated with modern maps illustrating the distribution of fossil localities from the first evolutionary fauna (EF1) in bioregions at the european scale

## Fourth step : ANOSIM test on ‘initial’ dataset:

Here, we compute Hierarchical Clustering on Principal Component (HCPC) on nMDS. After this first step of clustering, we use ANOSIM on the computed clusters by both methods (named ‘Initial’ clusters in Supplementary Figure 4). ANOSIM is used at pairwise and overall scale, for the N computed clusters (i.e. 2, 3, 4, etc.). In the next step, the ‘initial’ clusters are modified to follow continuity rule, and ANOSIM is rerun on modified clusters.

library(vegan)  
library(factoextra)  
library(FactoMineR)  
library(betapart)  
  
##A##  
#ANOSIM with Raup & Crick Index on HCPC  
#Minimal number of clusters (2)  
#Hierarchical Clustering on Principal Component computed from FactoMineR package  
hcpc.EU1 <- HCPC(as.data.frame(mds.Faune1.EU.RC\_K5$points), nb.clust = 2)  
#ANOSIM for two 'initial' (unmodified) clusters  
ANOS\_EU1 <- anosim(MatP.A.Faune1.EU, grouping = hcpc.EU1$data.clust$clust, permutations = 1000, distance = "raup") #R 0.46 ; (p) 0.000999  
#A data.frame is created to stock all R and p-value scores.  
ANOSIM\_EU1 <- data.frame(signif = c(ANOS\_EU1$signif), Rstat = c(ANOS\_EU1$statistic))  
dimnames(ANOSIM\_EU1)[[1]][1] <- "EU1\_2clust"  
  
#Incremental approach, 3, 4, 5, 6 and 7 clusters, overall and pairwise tests  
for(i in 3:7){  
hcpc.EU1 <- HCPC(as.data.frame(mds.Faune1.EU.RC\_K5$points), nb.clust = i)  
ANOS\_EU1 <- anosim(MatP.A.Faune1.EU, grouping = hcpc.EU1$data.clust$clust, permutations = 1000, distance = "raup")  
ANOSIM\_EU1 <- rbind(ANOSIM\_EU1, c(ANOS\_EU1$signif, ANOS\_EU1$statistic))  
dimnames(ANOSIM\_EU1)[[1]][length(ANOSIM\_EU1[,1])] <- paste("EU1\_", i, "clust", sep = "\_")  
  
AllPairs <- combn(unique(hcpc.EU1$data.clust$clust), 2)  
 for(j in 1:ncol(AllPairs))  
 {  
 SplittedMatrix <- split(as.data.frame(MatP.A.Faune1.EU), hcpc.EU1$data.clust$clust)  
 JoinMatrix <- rbind(SplittedMatrix[[AllPairs[1,j]]], SplittedMatrix[[AllPairs[2,j]]])  
 JoinMatrix <- JoinMatrix[,which(apply(JoinMatrix, 2, sum) != 0)]  
 GroupPairs <- c(hcpc.EU1$data.clust$clust[hcpc.EU1$data.clust$clust == AllPairs[1,j]],  
 hcpc.EU1$data.clust$clust[hcpc.EU1$data.clust$clust == AllPairs[2,j]])  
 ANOS\_EU1 <- anosim(JoinMatrix, grouping = GroupPairs, permutations = 1000, distance = "raup")  
 ANOSIM\_EU1 <- rbind(ANOSIM\_EU1, c(ANOS\_EU1$signif, ANOS\_EU1$statistic))  
 dimnames(ANOSIM\_EU1)[[1]][length(ANOSIM\_EU1[,1])] <- paste("EU1\_", i, "clust", AllPairs[1,j], AllPairs[2,j], sep = "\_")  
   
 }  
}  
ANOSIM\_EU1 #This data.frame contains all R and p-value scores of ANOSIM for HCPC clusters with Raup&Crick index.  
  
##B##  
#Computation of UPGMA (Unweighted pair group method with arithmetic mean) on dissimilarity matrix  
ExtractUPGMA <- HCPC\_UPGMA\_Home(mds.Faune1.EU.RC\_K5, Dist.Faune1.EU, 2, "EU", sortie = FALSE, IllustrationMAP = FALSE) #Personal function,  
#included in Supplementary Materiel, work as a wrapper function to picture HCPC and UPGMA Map, UPGMA and HCPC Tree as well. Here it's used to  
#extract the cluster identify of localities (e.g. Loc A - Clust 1, Loc B - Clust 2, etc.) for UPGMA Tree and ANOSIM on UPGMA clusters. nMDS,  
#Distance Matrix, number of clusters to build, identification tag, automatic output file TRUE/FALSE and picture of map in R TRUE/FALSE are the  
#respective arguments.  
  
##C##  
#ANOSIM for two 'initial' (unmodified) clusters  
ANOS\_EU1\_UPGMA <- anosim(MatP.A.Faune1.EU, grouping = ExtractUPGMA$ClusterUPGMA, permutations = 1000, distance = "raup") #R 0.46 ; (p) 0.000999  
#A data.frame is created to stock all R and p-value scores.  
ANOSIM\_EU1\_UPGMA <- data.frame(signif = c(ANOS\_EU1\_UPGMA$signif), Rstat = c(ANOS\_EU1\_UPGMA$statistic))  
dimnames(ANOSIM\_EU1\_UPGMA)[[1]][1] <- "EU1\_2clust"  
  
##D##  
#ANOSIM on incremental approach, 3, 4, 5, 6 and 7 clusters, overall and pairwise tests  
for(i in 3:7){  
ExtractUPGMA <- HCPC\_UPGMA\_Home(mds.Faune1.EU.RC\_K5, Dist.Faune1.EU, i, "EU", sortie = FALSE, IllustrationMAP = FALSE)  
ANOS\_EU1\_UPGMA <- anosim(MatP.A.Faune1.EU, grouping = ExtractUPGMA$ClusterUPGMA, permutations = 1000, distance = "raup")  
ANOSIM\_EU1\_UPGMA <- rbind(ANOSIM\_EU1\_UPGMA, c(ANOS\_EU1\_UPGMA$signif, ANOS\_EU1\_UPGMA$statistic))  
dimnames(ANOSIM\_EU1\_UPGMA)[[1]][length(ANOSIM\_EU1\_UPGMA[,1])] <- paste("EU1\_", i, "clust", sep = "\_")  
  
AllPairs <- combn(unique(ExtractUPGMA$ClusterUPGMA), 2)  
 for(j in 1:ncol(AllPairs))  
 {  
 SplittedMatrix <- split(as.data.frame(MatP.A.Faune1.EU), ExtractUPGMA$ClusterUPGMA)  
 JoinMatrix <- rbind(SplittedMatrix[[AllPairs[1,j]]], SplittedMatrix[[AllPairs[2,j]]])  
 JoinMatrix <- JoinMatrix[,which(apply(JoinMatrix, 2, sum) != 0)]  
 GroupPairs <- c(ExtractUPGMA$ClusterUPGMA[ExtractUPGMA$ClusterUPGMA == AllPairs[1,j]],  
 ExtractUPGMA$ClusterUPGMA[ExtractUPGMA$ClusterUPGMA == AllPairs[2,j]])  
 ANOS\_EU1\_UPGMA <- anosim(JoinMatrix, grouping = GroupPairs, permutations = 1000, distance = "raup")  
 ANOSIM\_EU1\_UPGMA <- rbind(ANOSIM\_EU1\_UPGMA, c(ANOS\_EU1\_UPGMA$signif, ANOS\_EU1\_UPGMA$statistic))  
 dimnames(ANOSIM\_EU1\_UPGMA)[[1]][length(ANOSIM\_EU1\_UPGMA[,1])] <- paste("EU1\_", i, "clust", AllPairs[1,j], AllPairs[2,j], sep = "\_")  
   
 }  
}  
ANOSIM\_EU1\_UPGMA #This data.frame contains all R and p-value scores of ANOSIM for HCPC clusters with Raup&Crick index.  
  
##E##  
#Computation of ANOSIM on Simpson Index  
#Minimal number of clusters (2)  
#Hierarchical Clustering on Principal Component computed from FactoMineR package  
hcpc.EU1\_Simp <- HCPC(as.data.frame(mds.Faune1.EU.SIMP\_K5$points), nb.clust = 2)  
#ANOSIM for two 'initial' (unmodified) clusters  
ANOS\_EU1\_Simp <- anosim(MatP.A.Faune1.EU, grouping = hcpc.EU1\_Simp$data.clust$clust, permutations = 1000, distance = "raup") #R 0.41 ; (p) 0.000999  
#A data.frame is created to stock all R and p-value scores.  
ANOSIM\_EU1\_Simp <- data.frame(signif = c(ANOS\_EU1\_Simp$signif), Rstat = c(ANOS\_EU1\_Simp$statistic))  
dimnames(ANOSIM\_EU1\_Simp)[[1]][1] <- "EU1\_2clust"  
  
##F##  
#ANOSIM on incremental approach, 3, 4, 5, 6 and 7 clusters, overall and pairwise tests  
for(i in 3:7){  
print(i)  
hcpc.EU1\_Simp <- HCPC(as.data.frame(mds.Faune1.EU.SIMP\_K5$points), nb.clust = i)  
ANOS\_EU1\_Simp <- anosim(MatP.A.Faune1.EU, grouping = hcpc.EU1\_Simp$data.clust$clust, permutations = 1000, distance = "raup")  
ANOSIM\_EU1\_Simp <- rbind(ANOSIM\_EU1\_Simp, c(ANOS\_EU1\_Simp$signif, ANOS\_EU1\_Simp$statistic))  
dimnames(ANOSIM\_EU1\_Simp)[[1]][length(ANOSIM\_EU1\_Simp[,1])] <- paste("EU1\_", i, "clust", sep = "\_")  
  
AllPairs\_Simp <- combn(unique(hcpc.EU1\_Simp$data.clust$clust), 2)  
 for(j in 1:ncol(AllPairs\_Simp))  
 {  
 SplittedMatrix <- split(as.data.frame(MatP.A.Faune1.EU), hcpc.EU1\_Simp$data.clust$clust)  
 JoinMatrix <- rbind(SplittedMatrix[[AllPairs\_Simp[1,j]]], SplittedMatrix[[AllPairs\_Simp[2,j]]])  
 JoinMatrix <- JoinMatrix[,which(apply(JoinMatrix, 2, sum) != 0)]  
 GroupPairs <- c(hcpc.EU1\_Simp$data.clust$clust[hcpc.EU1\_Simp$data.clust$clust == AllPairs\_Simp[1,j]],  
 hcpc.EU1\_Simp$data.clust$clust[hcpc.EU1\_Simp$data.clust$clust == AllPairs\_Simp[2,j]])  
 ANOS\_EU1\_Simp <- anosim(JoinMatrix, grouping = GroupPairs, permutations = 1000, distance = "raup")  
 ANOSIM\_EU1\_Simp <- rbind(ANOSIM\_EU1\_Simp, c(ANOS\_EU1\_Simp$signif, ANOS\_EU1\_Simp$statistic))  
 dimnames(ANOSIM\_EU1\_Simp)[[1]][length(ANOSIM\_EU1\_Simp[,1])] <- paste("EU1\_", i, "clust", AllPairs\_Simp[1,j], AllPairs\_Simp[2,j], sep = "\_")  
   
 }  
}  
ANOSIM\_EU1\_Simp #This data.frame contains all R and p-value scores for 'initial' nMDS+HCPC clusters  
  
##G##  
#Computation of UPGMA (Unweighted pair group method with arithmetic mean) on dissimilarity matrix  
ExtractUPGMA\_Simp <- HCPC\_UPGMA\_Home(mds.Faune1.EU.SIMP\_K5, Dist.Faune1.EU.Simp, 2, "EU", sortie = FALSE, IllustrationMAP = FALSE) #Personal function, included in Supplementary Materiel, work as a wrapper function to picture HCPC and UPGMA Map, UPGMA and HCPC Tree as well. Here it's used to extract the cluster identify of localities (e.g. Loc A - Clust 1, Loc B - Clust 2, etc.) for UPGMA Tree and ANOSIM on UPGMA clusters. nMDS, Distance Matrix, number of clusters to build, identification tag, automatic output file TRUE/FALSE and picture of map in R TRUE/FALSE are the respective arguments.  
  
##H##  
#ANOSIM for two 'initial' (unmodified) clusters  
ANOS\_EU1\_UPGMA\_Simp <- anosim(MatP.A.Faune1.EU, grouping = ExtractUPGMA\_Simp$ClusterUPGMA, permutations = 1000, distance = "raup") #R 0.48 ; (p) 0.000999  
#A data.frame is created to stock all R and p-value scores.  
ANOSIM\_EU1\_UPGMA\_Simp <- data.frame(signif = c(ANOS\_EU1\_UPGMA\_Simp$signif), Rstat = c(ANOS\_EU1\_UPGMA\_Simp$statistic))  
dimnames(ANOSIM\_EU1\_UPGMA\_Simp)[[1]][1] <- "EU1\_2clust"  
  
##I##  
#Computation of ANOSIM on incremental approach, 3, 4, 5, 6 and 7 clusters, overall and pairwise tests  
for(i in 3:7){  
ExtractUPGMA\_Simp <- HCPC\_UPGMA\_Home(mds.Faune1.EU.SIMP\_K5, Dist.Faune1.EU.Simp, i, "EU", sortie = FALSE, IllustrationMAP = FALSE)  
ANOS\_EU1\_UPGMA\_Simp <- anosim(MatP.A.Faune1.EU, grouping = ExtractUPGMA\_Simp$ClusterUPGMA, permutations = 1000, distance = "raup")  
ANOSIM\_EU1\_UPGMA\_Simp <- rbind(ANOSIM\_EU1\_UPGMA\_Simp, c(ANOS\_EU1\_UPGMA\_Simp$signif, ANOS\_EU1\_UPGMA\_Simp$statistic))  
dimnames(ANOSIM\_EU1\_UPGMA\_Simp)[[1]][length(ANOSIM\_EU1\_UPGMA\_Simp[,1])] <- paste("EU1\_", i, "clust", sep = "\_")  
  
AllPairs <- combn(unique(ExtractUPGMA\_Simp$ClusterUPGMA), 2)  
 for(j in 1:ncol(AllPairs))  
 {  
 SplittedMatrix <- split(as.data.frame(MatP.A.Faune1.EU), ExtractUPGMA\_Simp$ClusterUPGMA)  
 JoinMatrix <- rbind(SplittedMatrix[[AllPairs[1,j]]], SplittedMatrix[[AllPairs[2,j]]])  
 JoinMatrix <- JoinMatrix[,which(apply(JoinMatrix, 2, sum) != 0)]  
 GroupPairs <- c(ExtractUPGMA\_Simp$ClusterUPGMA[ExtractUPGMA\_Simp$ClusterUPGMA == AllPairs[1,j]],  
 ExtractUPGMA\_Simp$ClusterUPGMA[ExtractUPGMA\_Simp$ClusterUPGMA == AllPairs[2,j]])  
 ANOS\_EU1\_UPGMA\_Simp <- anosim(JoinMatrix, grouping = GroupPairs, permutations = 1000, distance = "raup")  
 ANOSIM\_EU1\_UPGMA\_Simp <- rbind(ANOSIM\_EU1\_UPGMA\_Simp, c(ANOS\_EU1\_UPGMA\_Simp$signif, ANOS\_EU1\_UPGMA\_Simp$statistic))  
 dimnames(ANOSIM\_EU1\_UPGMA\_Simp)[[1]][length(ANOSIM\_EU1\_UPGMA\_Simp[,1])] <- paste("EU1\_", i, "clust", AllPairs[1,j], AllPairs[2,j], sep = "\_")  
   
 }  
}  
ANOSIM\_EU1\_UPGMA\_Simp #This data.frame contains all R and p-value scores of ANOSIM for HCPC clusters with Raup&Crick index.



Third part of Supplementary Figure 4B, description of Fifth step of the analytical process described in this script below. Initial clusters are modified to follow continuity rule. Spatial analysis

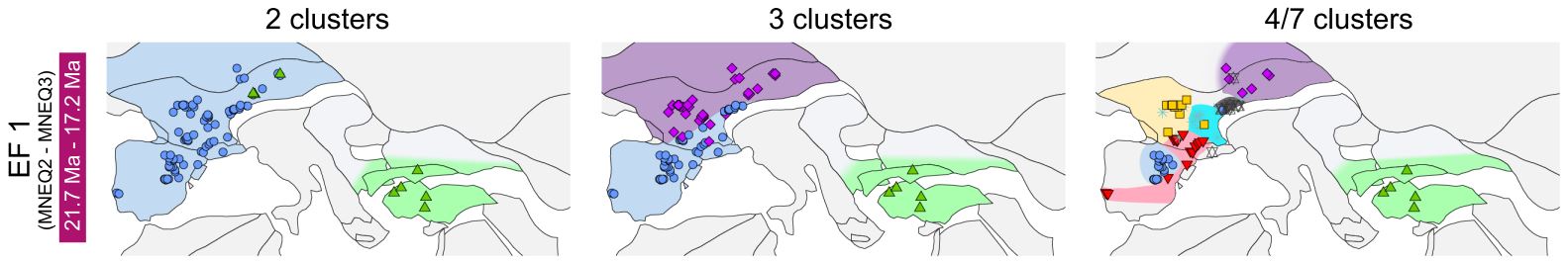
## Fifth step : Modification of clusters to follow continuity rule, new ANOSIM test on modified cluster:

After ANOSIM on ‘initial’ Simpson and Raup&Crick, UPGMA and HCPC Trees/Clusters, we choose Raup & Crick HCPC Tree as a base for ‘final’ clusters (i.e. clusters modified manually to follow continuity rules). HCPC and UPGMA, as well as Simpson and Raup&Crick clusters were very similar. HCPC/Raup&Crick is chosen because ‘initial’ clusters are more homogeneous and continuous. Here we compute ANOSIM analyses on

library(vegan)  
library(factoextra)  
library(FactoMineR)  
library(betapart)  
  
FULLeurope <- subset(Grain01.terre, Grain01.terre$LAT < 80 & Grain01.terre$LAT > 30 & Grain01.terre$LONG > - 20 & Grain01.terre$LONG < 50)  
  
##A##  
#Cleaning of dataset at species level  
#Similar step of cleaning as previously, it's added here in case of partial use of script (not from the beginning to the end)  
want <- which(FULLeurope$SPECIES == "Indet."| FULLeurope$SPECIES == "indet." | FULLeurope$SPECIES == "indet"  
 | FULLeurope$SPECIES == "Indet" | FULLeurope$SPECIES == "sp." | FULLeurope$SPECIES == "SP." |  
 FULLeurope$SPECIES == "sp" | FULLeurope$SPECIES == "SP" | FULLeurope$GENUS == "indet." |  
 FULLeurope$GENUS == "indet" | FULLeurope$GENUS == "Indet." | FULLeurope$GENUS == "Indet" | FULLeurope$SPECIES == "Gen.")  
FULLeurope <- FULLeurope[-want,]  
  
##B##  
#This example is set on EF1 (First Evolutionary Fauna)  
Faune1.EU <- data.frame()  
wantFaune1 <- which(17.2 >= FULLeurope$MIN\_AGE & 17.2 < FULLeurope$MAX\_AGE & 21.7 >= FULLeurope$MAX\_AGE  
 | 17.2 <= FULLeurope$MIN\_AGE & 21.7 > FULLeurope$MIN\_AGE & 21.7 <= FULLeurope$MAX\_AGE  
 | 17.2 >= FULLeurope$MIN\_AGE & 21.7 <= FULLeurope$MAX\_AGE  
 | 17.2 < FULLeurope$MIN\_AGE & 21.7 > FULLeurope$MAX\_AGE)   
  
Faune1.EU <- FULLeurope[wantFaune1,]   
head(Faune1.EU)  
#Two localities from middle east are removed.  
RemovedLocalities <- which(Faune1.EU$NAME == "30.2\_28.9" | Faune1.EU$NAME == "31\_35")  
Faune1.EU <- Faune1.EU[-RemovedLocalities,]  
  
##C##  
######## 2 Clusters #######  
WestEurope <- subset(Faune1.EU, Faune1.EU$LONG < 14)  
EastEurope <- subset(Faune1.EU, Faune1.EU$LONG > 14)  
EU1CLUST2 <- rbind(WestEurope, EastEurope)  
  
MatP.AEU1\_Clust2 <- Presence\_Absence\_matrix(EU1CLUST2, type = "Species", singletons = TRUE, min = 5)  
GroupEU1\_Clust2 <- c(rep(1,length(Presence\_Absence\_matrix(WestEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])),  
 rep(2,length(Presence\_Absence\_matrix(EastEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])))  
if(length(MatP.AEU1\_Clust2[,1]) == length(GroupEU1\_Clust2)){print("SAME SIZE")}else{print("FALSE")}  
EU1\_anos\_Clust2 <- anosim(MatP.AEU1\_Clust2, GroupEU1\_Clust2, distance = "raup") #R 0.62 p-value 0.001  
EU1\_anos\_Clust2 #ANOSIM statistics  
  
### Data.frame to stock R and p-values statistics.  
  
ANOSIM\_EU1\_Continuous <- data.frame(signif = EU1\_anos\_Clust2$signif, Rstat = EU1\_anos\_Clust2$statistic)  
dimnames(ANOSIM\_EU1\_Continuous)[[1]][length(ANOSIM\_EU1\_Continuous[,1])] <- "EU1\_2clust"  
ANOSIM\_EU1\_Continuous  
  
#Illustration on map  
HCPC\_UPGMA\_Home(mds.Faune1.EU.RC\_K5, Dist.Faune1.EU, 2, "EU", sortie = FALSE, IllustrationMAP = TRUE)  
  
##D##  
######## 3 Clusters #######  
  
SWEurope\_A <- subset(WestEurope, WestEurope$LAT < 45.65 & WestEurope$NAME != "44.8\_-0.6")  
SWEurope\_B <- subset(WestEurope, WestEurope$LAT > 46.3 & WestEurope$LAT < 47.6 & WestEurope$LONG > 6.3 & WestEurope$LONG < 10)  
SWEurope <- rbind(SWEurope\_A, SWEurope\_B)  
NWEurope <- subset(WestEurope, !(WestEurope$NAME %in% SWEurope$NAME))  
EU1CLUST3 <- rbind(EastEurope, SWEurope, NWEurope)  
  
MatP.AEU1\_Clust3 <- Presence\_Absence\_matrix(EU1CLUST3, type = "Species", singletons = TRUE, min = 5)  
GroupEU1\_Clust3 <- c(rep(1,length(Presence\_Absence\_matrix(EastEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])),  
 rep(2,length(Presence\_Absence\_matrix(SWEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])),  
 rep(3,length(Presence\_Absence\_matrix(NWEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])))  
if(length(MatP.AEU1\_Clust3[,1]) == length(GroupEU1\_Clust3)){print("SAME SIZE")}else{print("FALSE")}  
EU1\_anos\_Clust3 <- anosim(MatP.AEU1\_Clust3, GroupEU1\_Clust3, distance = "raup") #Overall ANOSIM analysis   
EU1\_anos\_Clust3 #R0.58 p-value 0.001  
  
ANOSIM\_EU1\_Continuous <- rbind(ANOSIM\_EU1\_Continuous ,c(EU1\_anos\_Clust3$signif, EU1\_anos\_Clust3$statistic))  
dimnames(ANOSIM\_EU1\_Continuous)[[1]][length(ANOSIM\_EU1\_Continuous[,1])] <- "EU1\_3clust"  
ANOSIM\_EU1\_Continuous  
  
AllPairs <- combn(unique(GroupEU1\_Clust3), 2) #Pairwise analysis  
for(j in 1:ncol(AllPairs))  
{  
 SplittedMatrix <- split(as.data.frame(MatP.AEU1\_Clust3), GroupEU1\_Clust3)  
 JoinMatrix <- rbind(SplittedMatrix[[AllPairs[1,j]]], SplittedMatrix[[AllPairs[2,j]]])  
 JoinMatrix <- JoinMatrix[,which(apply(JoinMatrix, 2, sum) != 0)]  
 GroupPairs <- c(GroupEU1\_Clust3[GroupEU1\_Clust3 == AllPairs[1,j]],  
 GroupEU1\_Clust3[GroupEU1\_Clust3 == AllPairs[2,j]])  
 EU1\_anos <- anosim(JoinMatrix, grouping = GroupPairs, permutations = 1000, distance = "raup")  
 ANOSIM\_EU1\_Continuous <- rbind(ANOSIM\_EU1\_Continuous, c(EU1\_anos$signif, EU1\_anos$statistic))  
 dimnames(ANOSIM\_EU1\_Continuous)[[1]][length(ANOSIM\_EU1\_Continuous[,1])] <- paste("EU1", "3 clust", AllPairs[1,j], AllPairs[2,j], sep = "\_")  
}  
ANOSIM\_EU1\_Continuous #Pairwise and overall ANOSIM Analysis statistics  
  
##E##  
######## 4 Clusters #######  
  
W\_NWEurope <- subset(NWEurope, NWEurope$LONG < 2.5)  
E\_NWEurope <- subset(NWEurope, NWEurope$LONG > 2.6)  
EU1CLUST4 <- rbind(EastEurope, SWEurope, W\_NWEurope, E\_NWEurope)  
  
MatP.AEU1\_Clust4 <- Presence\_Absence\_matrix(EU1CLUST4, type = "Species", singletons = TRUE, min = 5)  
GroupEU1\_Clust4 <- c(rep(1,length(Presence\_Absence\_matrix(EastEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])),  
 rep(2,length(Presence\_Absence\_matrix(SWEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])),  
 rep(3,length(Presence\_Absence\_matrix(W\_NWEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])),  
 rep(4,length(Presence\_Absence\_matrix(E\_NWEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])))  
if(length(MatP.AEU1\_Clust4[,1]) == length(GroupEU1\_Clust4)){print("SAME SIZE")}else{print("FALSE")}  
EU1\_anos\_Clust4 <- anosim(MatP.AEU1\_Clust4, GroupEU1\_Clust4, distance = "raup")  
EU1\_anos\_Clust4 #R 0.6038 (p-value) 0.001  
ANOSIM\_EU1\_Continuous <- rbind(ANOSIM\_EU1\_Continuous ,c(EU1\_anos\_Clust4$signif, EU1\_anos\_Clust4$statistic))   
dimnames(ANOSIM\_EU1\_Continuous)[[1]][length(ANOSIM\_EU1\_Continuous[,1])] <- "EU1\_4clust"  
ANOSIM\_EU1\_Continuous  
  
AllPairs <- combn(unique(GroupEU1\_Clust4), 2)  
for(j in 1:ncol(AllPairs))  
{  
 SplittedMatrix <- split(as.data.frame(MatP.AEU1\_Clust4), GroupEU1\_Clust4)  
 JoinMatrix <- rbind(SplittedMatrix[[AllPairs[1,j]]], SplittedMatrix[[AllPairs[2,j]]])  
 JoinMatrix <- JoinMatrix[,which(apply(JoinMatrix, 2, sum) != 0)]  
 GroupPairs <- c(GroupEU1\_Clust4[GroupEU1\_Clust4 == AllPairs[1,j]],  
 GroupEU1\_Clust4[GroupEU1\_Clust4 == AllPairs[2,j]])  
 EU1\_anos <- anosim(JoinMatrix, grouping = GroupPairs, permutations = 1000, distance = "raup")  
 ANOSIM\_EU1\_Continuous <- rbind(ANOSIM\_EU1\_Continuous, c(EU1\_anos$signif, EU1\_anos$statistic))  
 dimnames(ANOSIM\_EU1\_Continuous)[[1]][length(ANOSIM\_EU1\_Continuous[,1])] <- paste("EU1", "4 clust", AllPairs[1,j], AllPairs[2,j], sep = "\_")  
}  
ANOSIM\_EU1\_Continuous  
  
##F##  
######## 7 Cluster #######  
EU1 <- HCPC\_UPGMA\_Home(mds.Faune1.EU.RC\_K5, Dist.Faune1.EU, i = 7, id = "Faune 1 EU", sortie = FALSE, IllustrationMAP = TRUE)  
  
CentralSpain\_A <- subset(SWEurope, SWEurope$LAT > 40.5 & SWEurope$LAT < 42.7 & SWEurope$LONG > -3   
 & SWEurope$LONG < 1)  
CentralSpain\_B <- subset(SWEurope, SWEurope$LAT > 39.5 & SWEurope$LAT < 40.5 & SWEurope$LONG > -3   
 & SWEurope$LONG < -1.8)  
CentralSpain <- rbind(CentralSpain\_A, CentralSpain\_B)  
SouthernCoast <- subset(SWEurope , !(SWEurope$NAME %in% CentralSpain$NAME) & SWEurope$LAT < 44.6 & SWEurope$NAME != "44.2\_6.2")  
EastFrance <- subset(Faune1.EU, Faune1.EU$LAT > 45.1 & Faune1.EU$LAT < 46.85 & Faune1.EU$LONG < 7 & Faune1.EU$LONG > 2.8   
 | Faune1.EU$NAME == "44.2\_6.2")  
Swiss <- subset(Faune1.EU, Faune1.EU$LAT > 46.949 & Faune1.EU$LAT < 47.6 & Faune1.EU$LONG < 9.3 & Faune1.EU$LONG > 6.9)  
NorthernEurope <- subset(Faune1.EU, Faune1.EU$LAT > 48.2)  
EU1CLUST7 <- rbind(EastEurope, NorthernEurope, Swiss, EastFrance, SouthernCoast, W\_NWEurope, CentralSpain)  
  
  
MatP.AEU1\_Clust7 <- Presence\_Absence\_matrix(EU1CLUST7, type = "Species", singletons = TRUE, min = 5)  
GroupEU1\_Clust7 <- c(rep(1,length(Presence\_Absence\_matrix(EastEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])),  
 rep(2,length(Presence\_Absence\_matrix(NorthernEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])),  
 rep(3,length(Presence\_Absence\_matrix(Swiss, type = "Species", singletons = TRUE, min5 = 5)[,1])),  
 rep(4,length(Presence\_Absence\_matrix(EastFrance, type = "Species", singletons = TRUE, min5 = 5)[,1])),   
 rep(5,length(Presence\_Absence\_matrix(SouthernCoast, type = "Species", singletons = TRUE, min5 = 5)[,1])),   
 rep(6,length(Presence\_Absence\_matrix(W\_NWEurope, type = "Species", singletons = TRUE, min5 = 5)[,1])),   
 rep(7,length(Presence\_Absence\_matrix(CentralSpain, type = "Species", singletons = TRUE, min5 = 5)[,1])))  
if(length(MatP.AEU1\_Clust7[,1]) == length(GroupEU1\_Clust7)){print("SAME SIZE")}else{print("FALSE")}  
EU1\_anos\_Clust7 <- anosim(MatP.AEU1\_Clust7, GroupEU1\_Clust7, distance = "raup")  
EU1\_anos\_Clust7  
ANOSIM\_EU1\_Continuous <- rbind(ANOSIM\_EU1\_Continuous ,c(EU1\_anos\_Clust7$signif, EU1\_anos\_Clust7$statistic)) # Overall analysis  
dimnames(ANOSIM\_EU1\_Continuous)[[1]][length(ANOSIM\_EU1\_Continuous[,1])] <- "EU1\_7clust"  
ANOSIM\_EU1\_Continuous  
  
  
AllPairs <- combn(unique(GroupEU1\_Clust7), 2)  
for(j in 1:ncol(AllPairs))  
{  
 SplittedMatrix <- split(as.data.frame(MatP.AEU1\_Clust7), GroupEU1\_Clust7)  
 JoinMatrix <- rbind(SplittedMatrix[[AllPairs[1,j]]], SplittedMatrix[[AllPairs[2,j]]])  
 JoinMatrix <- JoinMatrix[,which(apply(JoinMatrix, 2, sum) != 0)]  
 GroupPairs <- c(GroupEU1\_Clust7[GroupEU1\_Clust7 == AllPairs[1,j]],  
 GroupEU1\_Clust7[GroupEU1\_Clust7 == AllPairs[2,j]])  
 EU1\_anos <- anosim(JoinMatrix, grouping = GroupPairs, permutations = 1000, distance = "raup")  
 ANOSIM\_EU1\_Continuous <- rbind(ANOSIM\_EU1\_Continuous, c(EU1\_anos$signif, EU1\_anos$statistic))  
 dimnames(ANOSIM\_EU1\_Continuous)[[1]][length(ANOSIM\_EU1\_Continuous[,1])] <- paste("EU1", "7 clust", AllPairs[1,j], AllPairs[2,j], sep = "\_")  
}  
ANOSIM\_EU1\_Continuous

### Conclusion : maps of modified clusters for EF1 in Europe

After modification to follow continuity rule, the final step of ANOSIM is used to verify if modified clusters are significantly dissimilar. The significantly dissimilar clusters (and only those), produced with Hierarchical Clustering on Principal Component (i.e. on a nMDS computed with Raup & Crick indexes) are pictured in the main manuscript. UPGMA map and tree, HCPC map and tree with Simpson indexes can be found in Appendix 2.



Modified clusters (i.e. bioregions) illustrated on paleomap for the first evolutionary fauna (EF1) at the european scale

## Exploring the bioregion datasets and help to read Appendix 2 trees, nMDS, map, PCA

In this example, in order to keep this script short we produce poly-cohort, ‘initial’ clusters and ‘modified’ clusters, i.e. bioregions, solely at the European scale. List of locality and species names in each bioregions illustrated in the main manuscript, at continental and global scale, for EF1-5 are available as .RData files. They are included in GitHub and Dryad deposits:

### In Appendix 2, localities are refered to Latitude\_Longitude labels (e.g. 36.3\_45.8). Downloaded .RData files contain list() and data.frame() giving the corresponding names between Latitude\_Longitude labels and actual localities names found in NOW database.

Link : <https://github.com/Corentin-Gibert-Paleontology/A-coherent-biogeographic-framework-for-Old-World-Neogene-and-Pleistocene-mammals>

##A##  
setwd() #Set your working directory where .RData files named "SPATIALSCALE\_EF\_.RData" are stored (e.g. Europe\_EF4.RData)  
  
##B##  
#Here we show how to use four example of .RData files, associated with OLD WORLD, EUROPE, ASIA, AFRICA for EF2, EF3, EF4, EF5 respectively.  
load("WORLD\_EF2.RData")  
load("Europe\_EF3.RData")  
load("Asia\_EF4.RData")  
load("Africa\_EF5.RData")  
  
#How to explore these files  
  
##C##  
#First structure, bioregion configurations  
names(WORLD\_EF2)  
  
#They are lists, where the first structure correspond to the number of bioregions configurations #Here for the Old World during evolutionary fauna 2, three configurations are available  
#named "2 clusters" corresponding to the minimal number of bioregions, "3 clusters", and  
#finally "5 clusters" for the maximal number of dissimilar bioregions for this spatial/temporal configuration.  
  
##D##  
#Second structure, Sites/locality names and species names  
names(WORLD\_EF2$`2 clusters`) #North West Africa, Rest of the Old World  
names(WORLD\_EF2$`3 clusters`) #North West Africa, South Asia, Rest of the Old World  
names(WORLD\_EF2$`5 clusters`) #Europe, East and South Africa, North West Africa, South Asia, North East Asia  
  
##E##  
#Third structure, identify locality names with labels from Appendix 2.  
WORLD\_EF2$`3 clusters`$RestofOldWorld\_Sites\_names #Site/locality names  
head(WORLD\_EF2$`3 clusters`$RestofOldWorld\_Species\_names) #Species/taxa names