```r
# Annex II _ R code
#
# EWG-17-15 - GEOSPATIAL ANALYSIS TO IDENTIFY HAULS WITH SAMPLING STATIONS
# Authors: "Francesc Ordinas, Polona Pengal & Alessandro Mannini"

library(ecodist)
library(plyr)

#Calculation for the shallow stratum#
setwd("afolderinyourcomputer")
mydataAB<-read.csv("TA_cluster_AB.csv",header=T,sep=";")
dadesAB<-mydataAB[,4:5]
euclidea <- dist(dadesAB, method = "euclidean")
meuclusterAB <- hclust(euclidea, method="average")
ABGroup <- cutree(meuclusterAB, h=0.035)
(ABGroup2<-cbind(mydataAB,ABGroup))
write.csv(ABGroup2, "ABGroup.csv")
dim(dadesAB)

#Calculation for the medium stratum#
setwd("afolderinyourcomputer")
mydataC<-read.csv("TA_cluster_C.csv",header=T,sep=";")
dadesC<-mydataC[,4:5]
euclidea <- dist(dadesC, method = "euclidean") # distance matrix basat en
distàncies euclidees
meuclusterC <- hclust(euclidea, method="average")

CGroup <- cutree(meuclusterC, h=0.035) # si tenc "k" cut tree into x clusters, si
tenc "h" talla a una determinada distància (o similaritat)
(CGroup2<-cbind(mydataC,CGroup))
write.csv(CGroup2, "CGroup.csv")
dim(dadesC)

#Calculation for the deep (200-500) stratum#
setwd("afolderinyourcomputer")
mydataD<-read.csv("TA_cluster_D.csv",header=T,sep=";")
dadesD<-mydataD[,4:5]
euclidea <- dist(dadesD, method = "euclidean") # distance matrix basat en
distàncies euclidees
meuclusterD <- hclust(euclidea, method="average")# Aqui se pot especificar "ward",
"single", "complete", "average"...
DGroup <- cutree(meuclusterD, h=0.035)
(DGroup2<-cbind(mydataD,DGroup))
write.csv(DGroup2, "DGroup.csv")
dim(dadesD)

#Calculation for the deep (>500) stratum#
setwd("afolderinyourcomputer ")
mydataE<-read.csv("TA_cluster_E.csv",header=T,sep=";")
dadesE<-mydataE[,4:5]
euclidea <- dist(dadesE, method = "euclidean") # distance matrix basat en
distàncies euclidees
meuclusterE <- hclust(euclidea, method="average")
EGroup <- cutree(meuclusterE, h=0.035)
(EGroup2<-cbind(mydataE,EGroup))
write.csv(EGroup2, "EGroup.csv")
dim(dadesE)
```

```
GroupAB<-read.csv("ABGroup.csv")
(GroupAB$ID<-"AB")
GroupAB<-rename(GroupAB,c("ABGroup"="Group"))
GroupC<-read.csv("CGroup.csv")
(GroupC$ID<-"C")
GroupC<-rename(GroupC,c("CGroup"="Group"))
GroupD<-read.csv("DGroup.csv")
(GroupD$ID<-"D")
GroupD<-rename(GroupD,c("DGroup"="Group"))
GroupE<-read.csv("EGroup.csv")
(GroupE$ID<-"E")
GroupE<-rename(GroupE,c("EGroup"="Group"))
FinalResult<-rbind(GroupAB,GroupC,GroupD,GroupE)
write.csv(FinalResult,"FinalResult.csv")

# EWG-17-15 - HIGH DENSITY PERSISTENCE ANALYSIS AT SAMPLING STATIONS

remove(list=ls())
library(doBy)  # summaryBy
library(reshape)
library(data.table)
library(stringr)
library(plyr)
library(dplyr)

setwd("aplaceinyourcomputer")

# Specify gsa,gen, spec codes and recruits cutoff length (if needed!) ####
areacode=c("22","23") # use combine for many GSA
countrycode=c("GRC")# Uppercase use combine for pull together two or more countries
# if you are working with GSA18 add on countrycode ALB Albania and MTN Montenegro

state="Aegean" #assign country code to output file name (i.e ITA_HRV_ for 2
countries)
gsa="22_23" #assign gsa code to output file name (i.e. 9_11_ for two GSAs)
GSAtable=read.csv("gsa_coordinates.csv") # use to bubble plot legend position
# if you already know Medits code
# gen<-"PAGE" # Uppercase genus MEDITS code
# spec<-"ERY"# Uppercase species MEDITS code

# if you instead prefer use 3alpha code
alpha_code="DPS"
mdts=fread("Sp_Medits.csv")
asfis=fread("ASFIS_2017.csv")
mdts=mdts[,c(2,3,7)]
asfis=asfis[,c(3,4)]
colnames(mdts)[2]="Scientific_name"
sp_list=merge(mdts,asfis,by="Scientific_name",all = T)
sp_list=subset(sp_list[!is.na(sp_list$code),])### NA values dismiss
sp_list=subset(sp_list[!is.na(sp_list$A_CODE),])### NA values dismiss
sp_list$gen=str_sub(sp_list$code,start=1,end=4)
sp_list$spec=str_sub(sp_list$code,start=5,end=7)
gen=unique(sp_list$gen[which(sp_list$A_CODE==alpha_code)])
spec=unique(sp_list$spec[which(sp_list$A_CODE==alpha_code)])

# Extract data ####
TAn <- fread("medits_TA.csv")
```

```
TAn <- as.data.frame(subset(TAn, (area %in% areacode) & (country %in%
countrycode)))
TAn=subset(TAn[TAn$validity=="V",])
#TCn <- fread("medits_tc.csv")
TCn <- read.csv("medits_tc.csv",sep=",") # to be used if fread crash
TCn <- as.data.frame(subset(TCn, (area %in% areacode) & (country %in% countrycode)
& (genus %in% gen) & (species %in% spec)))

## prepare the TA file for the next elaborations##
meandepth=(TAn$shooting_depth+TAn$hauling_depth)/2
sqkm=TAn$wing_opening/10000000*TAn$distance
id2=paste(TAn$area,TAn$country,TAn$year,TAn$haul_number,sep="_")
TAn["strata"]=NA
TAn=cbind(TAn,meandepth,sqkm,id2)
TAn$strata[]=TAn$nstrate[]

##Assign strata code based on the meandepth value and not on "codestrata"
for (i in 1:length(TAn$strata))
  if(TAn$meandepth[i]>0 & TAn$meandepth[i] < 51){TAn$strata[i]="A"}else{
    if(TAn$meandepth[i]>=51 & TAn$meandepth[i] < 101){TAn$strata[i]="B"}else{
      if(TAn$meandepth[i]>=101 & TAn$meandepth[i] < 201){TAn$strata[i]="C"}else{
        if(TAn$meandepth[i]>=201 & TAn$meandepth[i] <501){TAn$strata[i]="D"}else{
          TAn$strata[i]="E"}}}}
unique(TAn$strata)
table(TAn$strata)

dir_t=getwd()
dir.create(file.path(dir_t,"output"))

# Checking hauls positions ####
TA_lon=formatC(TAn$shooting_longitude, width = 4, format = "d", flag = "0")

TA_gr_lon=substr(TA_lon, 1, 2)
TA_gr_lon=as.integer(TA_gr_lon)
which(is.na(TA_gr_lon))
TA_gr_lon[is.na(TA_gr_lon)] <- 0

TA_mi_lon=substr(TA_lon, 3, 7)
TA_mi_lon=as.integer(TA_mi_lon)
which(is.na(TA_mi_lon))
TA_mi_lon[is.na(TA_mi_lon)] <- 0

#TA_se_lon=substr(TAn$shooting_longitude, 6, 7)
#TA_se_lon=as.integer(TA_se_lon)
#which(is.na(TA_se_lon))
#TA_se_lon[is.na(TA_se_lon)] <- 0

TA_lon_fin=TA_gr_lon+(TA_mi_lon/60)
which(is.na(TA_lon_fin))
TAn$TA_lon_fin=TA_lon_fin

TA_lat=formatC(TAn$shooting_latitude, width = 4, format = "d", flag = "0")

TA_gr_lat=substr(TA_lat, 1, 2)
TA_gr_lat=as.integer(TA_gr_lat)
which(is.na(TA_gr_lat))
TA_gr_lat[is.na(TA_gr_lat)] <- 0
```

```
TA_mi_lat=substr(TA_lat, 3, 7)
TA_mi_lat=as.integer(TA_mi_lat)
which(is.na(TA_mi_lat))
TA_mi_lat[is.na(TA_mi_lat)] <- 0

#TA_se_lat=substr(TAn$shooting_latitude, 6, 7)
#TA_se_lat=as.integer(TA_se_lat)
#which(is.na(TA_se_lat))
#TA_se_lat[is.na(TA_se_lat)] <- 0

TA_lat_fin=TA_gr_lat+(TA_mi_lat/60)
which(is.na(TA_lat_fin))
TAn$TA_lat_fin=TA_lat_fin
TAn$TA_lon_fin=TA_lon_fin

TAn$TA_lon_fin=ifelse(TAn$shooting_quadrant=="7",-TAn$TA_lon_fin,TAn$TA_lon_fin)
TA_lon_fin=TAn$TA_lon_fin

# Raising TC number by haul ####
raise=TCn$pfrac/TCn$pechan
nblonraise=TCn$nblon*raise
TCn=cbind(TCn,raise,nblonraise)
sum(TCn$nblonraise)
sum(TCn$nblon)
id2=paste(TCn$area,TCn$country,TCn$year,TCn$haul_number,sep="_")
TCn=cbind(TCn,id2)

#Calculating n°/km2
TATCn=merge(TAn,TCn,by=c("id2","year","haul_number"),all=T)
TATCn=TATCn[,c("year","haul_number","id2","area.x","sqkm","nblonraise","maturity")]
st=TATCn$nblonraise/TATCn$sqkm
TATC=cbind(TATCn,st)

#Subsetting for juveniles and adults and omitting NA values
juveniles<-subset(TATC, (maturity %in% c("0","1")))
juveniles=droplevels(juveniles)
adults<-subset(TATC,(maturity %in% c("2","3","4")))
adults=droplevels(adults)
#spawners<-subset(TATC,(maturity %in% c("3")))
#spawners=droplevels(spawners)

#Agregating (sum) by haul and subsetting for year, calculate 90th quartile
#and assigning 1 if above, 0 if below in column levels
juvenilesF=data.frame(aggregate(juveniles$st,list(juveniles$year,juveniles$haul_num
ber,juveniles$id2),sum))
names(juvenilesF)=c("Year","Haul","id2","N_sqkm")
adultsF=data.frame(aggregate(adults$st,list(adults$year,adults$haul_number,adults$i
d2),sum))
names(adultsF)=c("Year","Haul","id2","N_sqkm")
#spawnersF=data.frame(aggregate(spawners$st,list(spawners$year,spawners$haul_number
,spawners$id2),sum))
#names(spawnersF)=c("Year","Haul","id2","N_sqkm")

juvenilesF=na.omit(juvenilesF)
treshold=ddply(juvenilesF, "Year", summarise, q50 = quantile(N_sqkm, 0.50))
juvenilesF=merge(juvenilesF,treshold,by="Year")
juvenilesF$levels=NA
juvenilesF$levels <-ifelse(juvenilesF$N_sqkm <= juvenilesF$q50,"0","1")
```

```
adultsF=na.omit(adultsF)
treshold=ddply(adultsF, "Year", summarise, q50 = quantile(N_sqkm, 0.50))
adultsF=merge(adultsF,treshold,by="Year")
adultsF$levels=NA
adultsF$levels <-ifelse(adultsF$N_sqkm <= adultsF$q50,"0","1")

#spawnersF=na.omit(spawnersF)
#treshold=ddply(spawnersF, "Year", summarise, q50 = quantile(N_sqkm, 0.50))
#spawnersF=merge(spawnersF,treshold,by="Year")
#spawnersF$levels=NA
#spawnersF$levels <-ifelse(spawnersF$N_sqkm <= spawnersF$q50,"0","1")

#Merge selections with haul groups and create output file
haulGroups<-read.csv("dep_repaired.csv",sep=";")
haulGroups<-as.data.frame(subset(haulGroups, (area %in% areacode) & (country %in%
countrycode)))
persistJuv=merge(haulGroups,juvenilesF,by=("id2"),all=T)
persistAdu=merge(haulGroups,adultsF,by=("id2"), all=T)
#persistSpw=merge(haulGroups,spawnersF,by=("id2"), all=T)

setwd("C:/aplaceinyourcomputer")
write.csv(persistJuv,file=paste(alpha_code,"_",gsa,"_","juvenile",".csv"))
write.csv(persistAdu,file=paste(alpha_code,"_",gsa,"_","adult",".csv"))
#write.csv(persistSpw,file=paste(alpha_code,"_",gsa,"_","spawning",".csv"))
```