



Préparation du Plan d'échantillonnage

Version 3 Mai 2018

Historique de versions	
Version 9 Avril 2018	Version initiale qui inclut un changement dans le code afin de pouvoir ajouter le code de projet dans l'ID.
Version 3 Mai 2018	Ajoutées plus des précisions sur la section de la carte de stratification. Il faut vérifier le co-registrement des images et confirmer la projection. Ceci est important !!

Contents

1	Carte de stratification	1
2	Calcul des superficies par strate	2
3	Calcul du nombre d'échantillons	3
4	Sélection des échantillons.....	3

1 Carte de stratification

Tout d'abord on va faire la carte de stratification. On va prendre la carte de stratification

MADA_NationalStrat_171123.tif, qui est la carte nationale 2005-2016 et on va prendre seulement la zone d'intérêt en utilisant la fonction « extract by mask » de ArcGIS ou « clip raster with polygon » de QGIS.

Important : Il est très important qu'on soit sûres du SHP qu'on utilise pour la masque, e.g. pour Makira il faut utiliser le SHP qui corresponde à la zone de délégation de gestion.

Important : Confirmer le co-registrement de l'image résultante avec la carte nationale.

Important : La carte nationale est en UTM86S WGS84. Il faut que l'image résultante soit aussi en UTM WGS84

Ceci est importante pour assurer que les superficies utilisées pour les estimations sont correctes !!

2 Calcul des superficies par strate

- Dans le logiciel R, copier-coller et éditer le code suivant. Les parties en rouge doivent être définies par l'utilisateur.
- Les fichiers doivent être dans le dossier « working directory ».
- Le résultat se trouve dans un fichier .csv dans le dossier de travail.

```
#####PREMIERES ETAPES#####  
  
# ---Clear the workspace  
rm(list=ls())  
  
#Load libraries  
library(raster)  
library(rgdal)  
library(stringr)  
  
#####DEFINIR DES ENTREES#####  
  
#set working directory. Il faut definir ici le repertoire. Il faut mettre / au lieu de \  
wd <- "C:/Dropbox/MADA ERP/CAZ/"  
setwd(wd)  
  
# read a raster, GeoTiff or something. Il faut mettre le nom du raster de stratification. Il faut que le raster  
soit dans le working directory  
Raster <- raster('STR_CAZ.tif')  
  
##Resolution de la carte utilise  
Res <- 90  
  
##Fichier avec les superficies par strate
```

```

Output <- "MADA_ERPD_CAZ_Stratification.csv"

###NE TOUCHER CE QUI EST ECRIT CI
DESSUS#####

#####CALCUL DU NOMBRE DE PIXELS PER CLASSE DE LA
CARTE#####

t <- Raster@data@attributes
map_area <- as.data.frame(t)

map_area$Ha

map_area$Ha <- map_area$Count*Res*Res/10000

write.csv(map_area, file= Output, row.names= FALSE)

```

3 Calcul du nombre d'échantillons

En introduit les superficies par estrate dans la feuille de calcul « **MADA_xxx_AD_Sampling design_xxx_V0.xlsx** » (lignes 3 á 10).

Si on laisse les proportions et erreurs par défaut, on obtient le résultat d'échantillons par strate.

4 Sélection des échantillons

Tout d'abord il faut créer un fichier .csv avec deux colonnes, une colonne qui s'appelle "map_value" avec la valeur de la strate et une autre colonne "final" avec le nombre d'échantillons par strate. Il faut mettre le fichier dans le repertoire de travail.

Après il faut copier, coller et éditer en R le code suivant. Les parties en rouge doivent être définit par l'utilisateur. Les fichiers doivent être dans le dossier « working directory ». Le resultat se trouve dans un fichier .csv dans le dossier de travail.

La sortie est un SHP avec les échantillons ET un fichier .CSV qu'on pourra afficher sur Collect Earth.

Important : Confirmer que les points sont dans le centroïde des pixels de la carte de stratification.

Important : Il faut nommer les archives de sortie d'une façon qui permette l'organisation et archivage de l'information

Important : Chaque unité d'échantillonnage doit avoir un code unique

```

#####PREMIERES ETAPES#####

# ---Clear the workspace

```

```

rm(list=ls())

#Load libraries
library(raster)
library(rgdal)
library(stringr)

#####DEFINIR DES ENTREES#####
#set working directory. Il faut definir ici le repertoire. Il faut mettre / au lieu de \
wd <- "C:/Dropbox/MADA ERP/CAZ/"
setwd(wd)

#Set the project code
Proj <- "MAK2"

# read a raster, GeoTiff or something. Il faut mettre le nom du raster de stratification. Il faut que le raster
soit dans le working directory
Raster <- raster('STR_CAZ.tif')

##Resolution de la carte utilise
Res <- 90

#File name with number of samples per stratum. Fichier avec une colonne "map_value" avec la valeur de
la strate et une autre
##colonne "final" avec le nombre dechantillons
rp <- read.csv('Sampling.csv', header = T)

# Fichier de sortie avec les echantillons
output_file <- "test_plots.csv"

```

```

# The ISO country code. Code ISO du pays
countrycode <- 'MDG'

##Nom du SHP avec les points qu'on va creer
Shp <- "Points_echantillonnage"

####NE TOUCHER CE QUI EST ECRIT CI
DESSUS#####

####NE TOUCHER CE QUI EST ECRIT CI
DESSUS#####

#####CALCUL DU NOMBRE DE PIXELS PER CLASSE DE LA
CARTE#####

t <- Raster@data@attributes
map_area <- as.data.frame(t)
map_area$Ha
map_area$Ha <- map_area$Count*Res*Res/10000

#We paste the map_area in the rp
rp$map_area <- map_area$Count

#####SELECTION DES
ECHANTILLONS#####3

#beginCluster()

##### Generate 10x times the number of points from overall sample

rand_sample <- data.frame(sampleRandom(Raster,(sum(rp$final) *10 +
log((sum(rp$map_area))))),xy=TRUE))
names(rand_sample) <- c("x_coord","y_coord","map_value")

```

```

rand_sample$Sid <- row(rand_sample)[,1]
rp2 <- merge(rp,data.frame(table(rand_sample$map_value)),by.x="map_value",by.y="Var1",all.x=T)
rp2[is.na(rp2)]<-0

##### Create the list of classes that need to be specifically sampled
to_rtp <- rp2[rp2$Freq < rp2$final,]$map_value

##### Create the list of classes that are enough represented in the random sampling
to_spl <- rp2[rp2$Freq >= rp2$final,]$map_value

##### Sample points from the first class
i = 1

final <- rand_sample[ rand_sample$Sid %in%
  sample(rand_sample[rand_sample$map_value %in% c(to_spl[i],to_rtp[i]),]$Sid,
  rp2[rp2$map_value %in% c(to_spl[i],to_rtp[i]),]$final),]

##### Loop into the well represented classes, sample and append
if(length(to_spl) > 1){
for(i in 2:length(to_spl)){
tmp <- rand_sample[ rand_sample$Sid %in%
  sample(rand_sample[rand_sample$map_value == to_spl[i],]$Sid, rp2[rp2$map_value ==
to_spl[i],]$final),]
  final <- rbind(final,tmp)
}
}

```

```

##### Loop into the subrepresented classes, raster_to_point then append
if(length(to_rtp) > 0){
for(i in 1:length(to_rtp)){
  tmp_rtp <- as.data.frame(rasterToPoints(Raster,fun=function(rast){rast==to_rtp[i]}))
  names(tmp_rtp) <- c("x_coord","y_coord","map_value")
  tmp_rtp$Sid<-row(tmp_rtp)[,1]
  sampling <- min(rp2[rp2$map_value == to_rtp[i,]]$final,
                 rp2[rp2$map_value == to_rtp[i,]]$map_area)
  tmp<-tmp_rtp[tmp_rtp$Sid
              %in%
              sample(tmp_rtp[tmp_rtp$map_value == to_rtp[i,]]$Sid,
                    sampling
                    ),
                  ]
  final <- rbind(final,tmp)
}
}

#endCluster()

points <- final

#####
#####

##### Create vector layer with the points

sp_df<-SpatialPointsDataFrame(
  coords=points[,c(1,2)],

```

```
data=data.frame(points[,c(3)]),
proj4string=CRS(proj4string(Raster))
)

sp_df <- spTransform(sp_df,CRS("+proj=longlat +datum=WGS84"))
```

```
#####ON VA CREER LE CSV#####3
```

```
coord <- sp_df@coords
coord.sp <- SpatialPoints(coord)
coord.df <- as.data.frame(coord)
coord.spdf <- SpatialPointsDataFrame(coord.sp, coord.df)

#download province boundaries for the country
adm <- getData('GADM', country= countrycode, level=1)
#match the coordinate systems for the sample points and the boundaries
proj4string(coord.spdf) <-proj4string(adm)
adm1 <- over(coord.spdf, adm)
nsamples <- nrow(coord)

id <- matrix(sample(1:nsamples , nsamples , replace=F),nrow = nsamples , ncol =1, dimnames=
list(NULL,c("ID")))
id <- as.data.frame(id)
id$ID <- paste(Proj, "-",id$ID)
YCoordinate <- coord[,2]
XCoordinate <- coord[,1]
elevation <- getData("alt", country = countrycode)
slope <- terrain(elevation, opt = "slope")
aspect <- terrain(elevation, opt = "aspect")
```



```
elevation <- extract(elevation, cbind(coord[,1], coord[,2]))
slope <- extract(slope, cbind(coord[,1], coord[,2]))
aspect <- extract(aspect, cbind(coord[,1], coord[,2]))
map_code <- sp_df@data[, 1]

#write CSV file, this can be used directly in Collect Earth
m <- data.frame(id, YCoordinate, XCoordinate, elevation, slope, aspect, map_code) #, cadrillage)

write.csv(m, file= output_file, row.names= FALSE)

# write it out to a shapefile for further processing
writeOGR(obj = sp_df, dsn = Shp, layer = Shp, driver="ESRI Shapefile") # this is in geographical projection
```