Appendix S1: example of pseudo-code for biodivMapR

**Definition of variables**

Define Image path

If mask exists

Define mask path

Define output directory

Define dimensions for spatial units of the diversity maps

**Radiometric filtering [Optional]**

If pixels < NDVI threshold then

Discard pixels

If pixels < NIR threshold then

Discard pixels

If pixels > BLUE threshold then

Discard pixels

Write updated mask

Update mask path

**Computation of the principal component analysis**

Randomly select a set of pixels

Exclude water absorption bands and additional unwanted spectral domains

If Continuum\_Removal == TRUE

Apply continuum removal to the selection

Perform PCA

If FilterPCA == TRUE

Exclude pixels out of the mean ±3 SD fo any of the first components

Randomly select a set of pixels

Exclude water absorption bands and additional unwanted spectral domains

If Continuum\_Removal == TRUE

Apply continuum removal to the selection

Perform PCA

Save PCA file and preprocessed pixel subset

**Component selection**

Select PCs from the PCA raster file based on visual interpretation

Save PC selection in SelectedComponents.txt file

**Spectral species mapping**

Randomly select a set of pixels from PCA raster file

Select relevant components

Split selected pixels into nb\_partitions subsets

For each pixel subset

Perform k-means clustering with nbclusters

Apply k-means to the full image: assign closest cluster centroid to all unmasked pixels

Save SpectralSpecies raster image

**α-diversity mapping**

For each spatial unit in SpectralSpecies file

For each repetition

Compute selected α-diversity index from spectral species distribution

Average selected α-diversity index for the spatial unit over nb\_partitions

Write α-diversity index map

**β-diversity mapping**

Select a subset of spatial units from the image (default value = 2000)

Compute dissimilarity matrix from this subset (only Bray-Curtis dissimilarity implemented)

Perform an ordination on dissimilarity matrix: project the dissimilarity matrix in a 3D space

Compute dissimilarity between each spatial unit in the image, and the subset of spatial units used to compute ordination

Identify the three nearest neighbors from each spatial unit among subset of spatial units used to compute ordination based on BC

Compute the coordinate of each spatial unit in the ordinated space as weighted mean distance with three nearest neighbors

Produce RGB representation of the three components of the ordination for all spatial units

Appendix S2: Script run with Cameroon example

library(raster)  
library(biodivMapR)  
################################################################################  
## DEFINE PARAMETERS FOR DATASET TO BE PROCESSED ##  
################################################################################  
# path (absolute or relative) for the image to process # expected to be in ENVI HDR Input\_Image\_File = system.file('extdata', 'RASTER', 'S2A\_T33NUD\_20180104\_Subset', package = 'biodivMapR')

# # convert the image using Convert.Raster2BIL if not in the proper format  
# Input.Image.File = raster2BIL(Raster\_Path=Input\_Image\_File, Sensor = 'SENTINEL\_2A',  
# Convert\_Integer = TRUE,Output\_Directory = '~/test')  
  
# path for the Mask raster corresponding to image to process  
# expected to be in ENVI HDR format, 1 band, integer 8bits  
# expected values in the raster: 0 = masked, 1 = selected  
# set to FALSE if no mask available  
Input\_Mask\_File = FALSE  
  
# relative or absolute path for the Directory where results will be stored  
# For each image processed, a subdirectory will be created after its name  
Output\_Dir = 'RESULTS'  
  
# SPATIAL RESOLUTION  
# resolution of spatial units for alpha and beta diversity maps (in pixels), relative to original image  
# if Res.Map = 10 for images with 10 m spatial resolution, then spatial units will be 10 pixels x 10m = 100m x 100m surfaces  
# rule of thumb: spatial units between 0.25 and 4 ha usually match with ground data  
# too small window\_size results in low number of pixels per spatial unit, hence limited range of variation of diversity in the image  
window\_size = 10  
  
# PCA FILTERING: Set to TRUE if you want second filtering based on PCA outliers to be processed. Slower  
FilterPCA = FALSE  
  
# type of PCA:  
# PCA: no rescaling of the data  
# SPCA: rescaling of the data  
TypePCA = 'SPCA'  
  
################################################################################  
## DEFINE PARAMETERS FOR METHOD ##  
################################################################################  
nbCPU = 2  
MaxRAM = 0.5  
nbclusters = 50  
  
################################################################################  
## PROCESS IMAGE ##  
################################################################################  
# 1- Filter data in order to discard non vegetated / shaded / cloudy pixels  
NDVI\_Thresh = 0.5  
Blue\_Thresh = 500  
NIR\_Thresh = 1500  
print("PERFORM RADIOMETRIC FILTERING")  
ImPathShade = perform\_radiometric\_filtering(Input\_Image\_File,Input\_Mask\_File,Output\_Dir, NDVI\_Thresh = NDVI\_Thresh, Blue\_Thresh = Blue\_Thresh, NIR\_Thresh = NIR\_Thresh)  
  
# 2- Compute PCA for a random selection of pixels in the raster  
print("PERFORM PCA ON RASTER")  
PCA\_Output = perform\_PCA(Input\_Image\_File,Input\_Mask\_File,Output\_Dir,FilterPCA=FilterPCA,nbCPU=nbCPU,MaxRAM = MaxRAM)  
PCA\_Files = PCA\_Output$PCA\_Files

Pix\_Per\_Partition = PCA\_Output$Pix\_Per\_Partition

nb\_partitions = PCA\_Output$nb\_partitions

Input\_Mask\_File = PCA\_Output$ImPathShade

PCA\_model = PCA\_Output$PCA\_model

SpectralFilter = PCA\_Output$SpectralFilter

# 3- Select principal components from the PCA raster  
select\_PCA\_components(Input\_Image\_File,Output\_Dir,PCA.Files,File.Open = TRUE)  
  
################################################################################  
## MAP ALPHA AND BETA DIVERSITY ##  
################################################################################  
print("MAP SPECTRAL SPECIES")  
map\_spectral\_species(Input\_Image.File,Output\_Dir,PCA.Files, PCA\_model,SpectralFilter,Input\_Mask\_File,Pix\_Per\_Partition,nb\_partitions,nbCPU=nbCPU,MaxRAM=MaxRAM,nbclusters=nbclusters, TypePCA=TypePCA,CR=TRUE)  
  
print("MAP ALPHA DIVERSITY")  
Index\_Alpha = c('Shannon')  
map\_alpha\_div(Input\_Image\_File,Output\_Dir,window\_size,nbCPU=nbCPU,MaxRAM=MaxRAM,Index.Alpha = Index.Alpha,nbclusters=nbclusters)  
  
print("MAP BETA DIVERSITY")  
map\_beta\_div(Input\_Image\_File,Output\_Dir,window\_size,nbCPU=nbCPU,MaxRAM=MaxRAM,nbclusters=nbclusters)  
  
################################################################################  
## COMPUTE ALPHA AND BETA DIVERSITY FROM FIELD PLOTS ##  
################################################################################  
# location of the spectral species raster needed for validation  
Dir.Raster = file.path(Output.Dir,basename(Input.Image.File),TypePCA,'SpectralSpecies')  
Name.Raster = 'SpectralSpecies'  
Path.Raster = file.path(Dir.Raster,Name.Raster)  
  
# location of the directory where shapefiles used for validation are saved  
vect = system.file('extdata', 'VECTOR', package = 'biodivMapR')  
Shannon.All = list()  
  
# list vector data  
Path.Vector = list\_shp(vect)  
Name.Vector = tools::file\_path\_sans\_ext(basename(Path.Vector))  
  
# get alpha and beta diversity indicators corresponding to shapefiles  
Biodiv.Indicators = diversity\_from\_plots(Raster = Path.Raster, Plots = Path.Vector,NbClusters = nbclusters)  
# if no name  
Biodiv.Indicators$Name.Plot = seq(1,length(Biodiv.Indicators$Shannon[[1]]),by = 1)  
Shannon.RS = c(Biodiv.Indicators$Shannon)[[1]]  
  
####################################################  
# write RS indicators  
####################################################  
# write indicators for alpha diversity  
Path.Results = paste(Output.Dir,'/',basename(Input.Image.File),'/',TypePCA,'/VALIDATION/',sep='')  
dir.create(Path.Results, showWarnings = FALSE,recursive = TRUE)  
write.table(Shannon.RS, file = paste(Path.Results,"ShannonIndex.csv",sep=''), sep="\t", dec=".", na=" ", row.names= Biodiv.Indicators$Name.Plot, col.names= F,quote=FALSE)  
  
Results = data.frame(Name.Vector, Biodiv.Indicators$Richness, Biodiv.Indicators$Fisher, Biodiv.Indicators$Shannon,Biodiv.Indicators$Simpson)  
names(Results) = c("ID\_Plot", "Species\_Richness", "Fisher", "Shannon", "Simpson")  
write.table(Results, file= paste(Path.Results,"AlphaDiversity.csv",sep=''), sep="\t", dec=".", na=" ", row.names = F, col.names= T,quote=FALSE)  
  
# write indicators for beta diversity  
BC\_mean = Biodiv.Indicators$BCdiss  
colnames(BC\_mean) = rownames(BC\_mean) = Biodiv.Indicators$Name.Plot  
write.table(BC\_mean, file= paste(Path.Results,"BrayCurtis.csv",sep=''), sep="\t", dec=".", na=" ", row.names = F, col.names= T,quote=FALSE)  
  
####################################################  
# illustrate results  
####################################################  
library(labdsv)  
# assign vegetation type to polygons in shapefiles  
nbSamples = c(6,4,7,7)  
vg = c('Forest high diversity', 'Forest low diversity', 'Forest medium diversity', 'low vegetation')  
Type\_Vegetation = c()  
for (i in 1: length(nbSamples)){  
 for (j in 1:nbSamples[i]){  
 Type\_Vegetation = c(Type\_Vegetation,vg[i])  
 }  
}  
  
# apply ordination unsing PCoA (same as done for map\_beta\_div)  
MatBCdist = as.dist(BC\_mean, diag = FALSE, upper = FALSE)  
BetaPCO = pco(MatBCdist, k = 3)  
  
# create data frame including alpha and beta diversity  
library(ggplot2)  
Results = data.frame('vgtype'=Type\_Vegetation,'pco1'= BetaPCO$points[,1],'pco2'= BetaPCO$points[,2],'pco3' = BetaPCO$points[,3],'shannon'=Shannon.RS)  
  
# plot field data in the PCoA space, with size corresponding to shannon index  
ggplot(Results, aes(x=pco1, y=pco2, color=vgtype,size=shannon)) +  
 geom\_point(alpha=0.6) +  
 scale\_color\_manual(values=c("#e6140a", "#e6d214", "#e68214", "#145ae6"))  
filename = file.path(Path.Results,'BetaDiversity\_PcoA1\_vs\_PcoA2.png')  
ggsave(filename, plot = last\_plot(), device = 'png', path = NULL,  
 scale = 1, width = 6, height = 4, units = "in",  
 dpi = 600, limitsize = TRUE)  
  
ggplot(Results, aes(x=pco1, y=pco3, color=vgtype,size=shannon)) +  
 geom\_point(alpha=0.6) +  
 scale\_color\_manual(values=c("#e6140a", "#e6d214", "#e68214", "#145ae6"))  
filename = file.path(Path.Results,'BetaDiversity\_PcoA1\_vs\_PcoA3.png')  
ggsave(filename, plot = last\_plot(), device = 'png', path = NULL,  
 scale = 1, width = 6, height = 4, units = "in",  
 dpi = 600, limitsize = TRUE)  
  
ggplot(Results, aes(x=pco2, y=pco3, color=vgtype,size=shannon)) +  
 geom\_point(alpha=0.6) +  
 scale\_color\_manual(values=c("#e6140a", "#e6d214", "#e68214", "#145ae6"))  
filename = file.path(Path.Results,'BetaDiversity\_PcoA2\_vs\_PcoA3.png')  
ggsave(filename, plot = last\_plot(), device = 'png', path = NULL,  
 scale = 1, width = 6, height = 4, units = "in",  
 dpi = 600, limitsize = TRUE)

Appendix S3: components produced from the PCA applied to the Sentinel-2 subset provided with biodivMapR. Components 1, 2, 5, 6 and 8 were selected after visual interpretation.

|  |
| --- |
|  |
|  |