Step 3.3 Creating Detectable Typology

3.3.2 Computing spectral separability and refining typology

Diane ESPEL

2025-06-18

Contents

1 Objectives

The script aims to estimate and visualize the **number of sample plots available for each habitat class** across different classification levels. Understanding the distribution of sample plots is crucial for assessing the representativeness and robustness of habitat classifications in ecological studies

2 Script explanation

2.1 Clean environment and graphics

```
rm(list = ls())  # Clear all objects from the R environment to start fresh
graphics.off()  # Close all graphics devices (if any plots are open)
```

2.2 Load required packages

```r
library(ggplot2)  # For plotting
```

2.3 Create functions

We need to define a function that normalizes the nomenclature of habitats: here, adding dots to the toponymy (e.g., A11 –> A.1.1): "transform_chain()"

```r
# Function to insert dots between each character in a string (for formatting
# habitat codes)
transform_chain <- function(chain) {

    # Split string into individual characters
    divided_chain <- strsplit(chain, "")[[1]]

    # Join characters by inserting dots between them
    transformed_chain <- paste(divided_chain, collapse = ".")

    # Return the transformed string with dots
    return(transformed_chain)
}
```

2.4 Define global variables

Note: A global variable is a variable defined outside of any function. This means the variable is accessible from any part of the code, including inside functions. A global variable retains its value throughout the execution of the R script unless it is explicitly modified in the code.

It is important to define at a minimum:

- the "District": the archipelago of interest (e.g. “CRO” for Crozet archipelago)
- the "Island": the island within the archipelago of interest (e.g. “POS” for Possession island)
- the "Satellite1": the name of satellite providing multispectral imagery
- the "Year1": the acquisition year of the multispectral imagery
- the "maxTypoLevel": the maximum typology level

```r
District = "CRO"  # 3-letter code for archipelago (e.g. Crozet)
Island = "POS"  # 3-letter code for island (e.g. Possession)
Satellite1 = "Pleiades"  # satellite name of multispectral imagery
Year1 = "2022"  # acquisition year of multispectral imagery
maxTypoLevel = 4  # Define maximum typology level
```

2.5 Set working directory

To optimize memory, you must define one general root directory ("localscratch") that serves as the base path for your input and output data, respectively. This directory should point to the local environment where:

- input learning dataset is located under "data/Learning_data/NewTypo"
- output new typology will be saved under "data/Learning_data/NewTypo"

```
# Base local path (customize to your local environment)
localscratch = paste0("/scratch/despel/CARTOVEGE/")
# localscratch = paste0('your_local_path/')

# Path to open final learning data
open_learning_new_path = paste0(localscratch, "data/Learning_data/NewTypo")

# Path to save results
save_learning_new_path = paste0(localscratch, "data/Learning_data/NewTypo")
```

2.6 Load final learning data

The script reads a CSV file containing spectral and topographic data linked to learning plots. This dataset includes habitat classification labels across multiple levels and a variety of predictor variables derived from remote sensing data.

```
# Open vector data
FILE1 = paste0(open_learning_new_path, "/", "Final_learning_plots_", District, "_",
    Island, "_", Satellite1, "_", Year1, "_ALL_SOURCES_EPSG32739.csv")
learning_points <- read.csv(FILE1, sep = ";", dec = ".", stringsAsFactors = FALSE)  #
 ↪  `stringsAsFactors=F` ensures character strings don't import as factors
```

2.7 Analyze number of plots per habitat class

This step systematically computes the frequency of sample plots for each habitat class, stratified by the source of the data (field, high-frequency imagery, photo-interpretation, or all sources combined) and by classification level. For each combination, the script generates **bar plots** to visually represent the number of plots per habitat class and saves these results as both image files (PNG and SVG) and formatted CSV tables.

```
# List of plot types to consider
source_list=c("FIELD","HFI","PHOTO-INTERPRETATION","ALL")

# Loop over each plot source
for (s in source_list){

  print(paste0("Processing plot source: ", s))

  # Subset the dataset depending on the sample type, or use all if "ALL"

  if(s=="ALL"){
    learning_points=learning_points
  }else{
    learning_points=subset(learning_points,learning_points$Source==s)
  }

  # Loop over classification levels
  for (l in seq (1:maxTypoLevel)){

    print(paste0("Processing typology level: ", l))

    # Create an output folder for the current classification level
    LevelFolder=paste0(save_learning_new_path,"/","Hab_L",l)
    dir.create(LevelFolder,showWarnings=F) # ShowWarnings=F to remove warnings message if file
     ↪  already exists
```

```
    # Identify the column index for the corrected habitat class at current level
    ihab=which(colnames(learning_points)==paste0("Hab_L",l,"_corr"))

    # Compute the number of plots per habitat class
    frequency_hab <- as.data.frame(table(learning_points[,ihab]))
    frequency_hab[,1]=as.character(frequency_hab[,1])
    colnames(frequency_hab)=c(paste0("Hab_L",l), "Number_of_plots")

    # Save column of habitat types for plotting
    Type_habitat=frequency_hab[,1]

    # Bar plot of number of plots per habitat class (png format)

    NOMpng=paste0(LevelFolder,"/","Nb_quadrats_",District,"_",Island,"_",Year1,"_",s,"_SOURCES_level_",l,".png")
    png(file = NOMpng, width = 1000, height = 600)
    p <- ggplot(frequency_hab, aes(x = Type_habitat, y = Nombre_echantillons, fill = Type_habitat))
    +
      geom_bar(stat = "identity") + # Create a bar plot
      geom_text(aes(label = Nombre_echantillons), vjust = -0.5, color = "black", size = 4) + # Add
       data labels
      labs(title = paste0(sample," : Number of plots per habitat class - Level ", l), # Set plot
       titles and axis labels
           x = "Habitat classes",
           y = "Number of plots") +
      theme_classic() + # Apply classic theme
      theme(axis.text.x = element_text(size = 13, angle = 90, hjust = 1),  # Customize axis labels
            axis.text.y = element_text(size = 13),
            axis.title.x = element_text(size = 16),
            axis.title.y = element_text(size = 16),
            plot.title = element_text(size = 22,face = "bold"),  # Customize plot title
            legend.text = element_text(size = 14),  # Customize legend text
            legend.title = element_text(size = 16,face = "bold")) +  # Customize legend title
      scale_y_continuous(expand = c(0, 0))  +  # Adjust the y-axis range
      labs(fill = "Habitat classes")  # Rename the legend title
    print(p)
    dev.off()

    # Bar plot of number of plots per habitat class (svg format)
    NOMsvg <-
    paste0(LevelFolder,"/Nb_quadrats_",District,"_",Island,"_",Year1,"_",s,"_SOURCES_level_",l,".svg")
    svg(file = NOMsvg)
    print(p)
    dev.off()

    # Save frequency of habitat types in a dataframe
    frequency_hab[,1]<- sapply(frequency_hab[,1], transform_chain) # Format habitat codes: add dots
    between letters
    frequency_hab[,1] <- paste(paste0(District,"-"), frequency_hab[,1], sep = "") # # Format
    habitat codes:  prefix with 3-letters district code

     write.table(frequency_hab,file=paste0(LevelFolder,"/","Nb_quadrats_",District,"_",Island,"_",Year1,"_",s,"_
     = ";", dec = ".", row.names = FALSE)


  } # End of typology level loop

} # End of learning data source loop
```

::: {align="justify"} This script enables better evaluation of sampling effort and helps identify habitat classes that may be underrepresented or require additional sampling.

:::