# Step 5.2 Computing habitat metrics

Flat and Hierarchical modeling strategies

Diane ESPEL

2025-06-20

## Contents



## 1 Objectives

This script aims to calculate a set of diversity and landscape structure indices from predicted habitat maps, enabling a quantitative assessment of habitat composition and spatial configuration over time and across classification levels.

## 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

```
library(sf)  # For handling spatial vector data with simple features framework
library(dplyr)  # For data manipulation and transformation (filter, select, mutate, etc.)
library(nngeo)  # For nearest neighbor analysis in spatial data
library(stringr)  # For consistent and convenient string manipulation functions
```

## 2.3 Create functions

## 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 satellite name for multispectral imagery
- the `"maxTypoLevel"`: the maximum typology level

```
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
maxTypoLevel = 4  # maximum typology level
```

## 2.5 Set working directory

You must define a general root directory (`"localscratch"`) that serves as the base path for your input and output data. This directory should point to the local environment where:

- input spatial predictions (maps) are located under `"results/Predictions"`
- outputs metrics will be saved under `"results/Landscape_metrics"`

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

# Path to open spatial predictions
open_predictions_path = paste0(localscratch, "results/Predictions")

# Path to save metrics
save_metrics_path = paste0(localscratch, "results/Landscape_metrics")
```

### ##Compute habitat metrics for each classification level

For each modeling strategy (e.g., "FLAT", "HIERARCHICAL"), - the script detects **all available years** by scanning shapefiles stored in the prediction results directory.

- for each year and typology level, it constructs the **expected file path** for the corresponding shapefile.
- If the shapefile exists, it is loaded, and the relevant habitat column is renamed to a standard name (`"Hab"`).
- The script calculates **global diversity indices** from the habitat polygons, based on the **surface area of each habitat type**. These include:
  - **Shannon diversity index**, which measures both the richness and evenness of habitat types and is sensitive to the presence of rare classes;

  - **Simpson's diversity index**, which gives more weight to dominant classes and reflects the probability that two randomly selected units belong to different classes;
  - **Pielou's evenness index**, which normalizes the Shannon index by the maximum possible diversity, giving an indication of how evenly habitat types are distributed. Together, these indices provide insights into the heterogeneity and dominance relationships between habitat types.

- In parallel,**landscape metrics** are computed to characterize the spatial structure of habitat patches. These include:

  - the **number of habitat patches** (providing information on habitat fragmentation),
  - the **mean, minimum, and maximum patch size** (reflecting habitat availability and potential for species occupancy),
  - the **proportion of the largest patch** (an indicator of habitat dominance and continuity),
  - the **mean (euclidian) distance to the nearest neighbor patch of the same class** (which informs on habitat connectivity and isolation). Such metrics are essential in landscape ecology and conservation planning as they help assess ecosystem integrity, fragmentation patterns, and potential habitat functionality for species movement and persistence.

All results are exported as CSV files, organized by year, habitat level, and modeling strategy.

```
# Define list of modeling strategy
type_model_list = c("FLAT", "HIERARCHICAL")


# Loop through model types
for (type_model in type_model_list) {

    # type='FLAT' #debug
    print(paste0("Modeling strategy: ", type_model))

    # Detect available years from shapefile names
    shp_files = list.files(path = open_predictions_path, pattern = ".shp$", recursive = TRUE,
        full.names = TRUE)

    # Extract years (4-digit numbers) from filenames
    all_years = unique(str_extract(shp_files, "\\d{4}"))
    all_years = all_years[!is.na(all_years)]

    # Loop through each available year
    for (Year in all_years) {

        print(paste0("Processing year ", Year))

        # Loop through each classification level
        for (l in seq(1:maxTypoLevel)) {

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

            # Path to folder for this classification level
            newFolder = paste0(open_predictions_path, "/", "Hab_L", l)

            # Build expected shapefile name for this year and level
            FILE1 = paste0(newFolder, "/", "Smoothed_simplified_final_map_RF_", type_model,
                "_model_", District, "_", Island, "_", Satellite1, "_", Year, "_level_",
                l, ".shp")

            # Check if the shapefile exists before proceeding
            if (!file.exists(FILE1)) {
```

```
    warning(paste0("Shapefile not found: ", FILE1))
    next
}

print("Opening final predicted habitat map")
final_map = st_read(FILE1)

# Rename the habitat column to 'Hab'
ihab = which(colnames(final_map) == paste0("Hab_L", l))
colnames(final_map)[ihab] = "Hab"

# Save folders
SaveFolder = paste0(save_metrics_path, "/", "Hab_L", l)
dir.create(SaveFolder, showWarnings = FALSE, recursive = TRUE)

# Diversity and heterogeneity metrics
# ----------------------------------------

print("Calculating total ROI surface")
ROI_surface = sum(final_map$Surface)

print("Calculating total surface per habitat class")
surface_table = final_map %>%
    group_by(Hab) %>%
    summarise(Surface_totale = sum(as.numeric(Surface))) %>%
    mutate(Proportion = Surface_totale/sum(Surface_totale))

# Shannon diversity index
shannon = -sum(surface_table$Proportion * log(surface_table$Proportion))

# Simpson diversity index
simpson = 1 - sum(surface_table$Proportion^2)

# Pielou's evenness index
richness = nrow(surface_table)
pielou = if (richness > 1)
    shannon/log(richness) else NA

# Stack all heterogeneity indices
indices_df = data.frame(Year_map = Year, Typo_level = l, Shannon_index = shannon,
    Simpson_index = simpson, Pielou_index = pielou)

FILE2 = paste0(SaveFolder, "/", "Diversity_indices_RF_", type_model,
    "_model_", District, "_", Island, "_", Satellite1, "_", Year, "_level_",
    l, ".csv")
write.table(indices_df, FILE2, sep = ";", dec = ".", row.names = FALSE)

# Landscape metrics by habitat classes
# ----------------------------------------

print("Computing landscape metrics by habitat class")

summary_table = final_map %>%
    group_by(Hab) %>%
    mutate(nearest_neighbor_distance = if (n() >= 2) {
      nearest_neighbors = st_nn(geometry, geometry, k = 2, returnDist = TRUE)
      distances = sapply(nearest_neighbors$dist, function(x) x[2])
    } else {
      NA
```

```
                }, Nb_total_polygons = n()) %>%
                summarise(Nb_patchs = n(), Relative_abundance = n()/unique(Nb_total_polygons),
                  Mean_patch_size_m2 = mean(Surface), Min_patch_size_m2 = min(Surface),
                  Max_patch_size_m2 = max(Surface), Total_patch_size_m2 = sum(Surface),
                  Largest_patch_index = max(Surface)/Island_surface, Mean_euclidian_distance =
                   ↪  mean(nearest_neighbor_distance,
                    na.rm = TRUE))

            FILE3 = paste0(SaveFolder, "/", "Landscape_metrics_RF_", type_model,
                "_model_", District, "_", Island, "_", Satellite1, "_", Year, "_level_",
                l, ".csv")
            write.table(summary_table, file = FILE3, sep = ";", dec = ".", row.names = FALSE)

        }  # End of typology level loop

    }  # End of year loop

}  # End of model type loop
```

This script enables temporal and hierarchical comparison of habitat patterns across the island landscape.