Step 7 Temporal monitoring

Analyzing time series of habitat metrics

Diane ESPEL

2025-06-20

Contents

1 Objectives **1**

2 Script explanation **1**

2.1 **Clean environment and graphics** 1

2.2 **Load required packages** 1

2.3 **Define Global Variables** 2

2.4 **Set working directory** 2

2.5 **Plot Metrics over Years** 2

1 Objectives

This script analyzes and visualizes the temporal dynamics of habitat diversity indices and landscape metrics derived from predicted habitat maps. The goal is to assess ecological changes over time according to different classification levels and modeling strategies (FLAT, HIERARCHICAL).

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(ggplot2)  # For creating elegant and customizable graphics using the grammar of graphics
library(dplyr)  # For data manipulation: filtering, selecting, grouping, summarizing, etc.
library(tidyr)  # For reshaping and tidying data: e.g., pivoting longer/wider, handling missing
↪  values
```

```r
library(stringr)  # For consistent and simple string operations (pattern matching, extraction,
 ↪  etc.)
library(viridis)  # For color palettes optimized for perceptual uniformity and colorblind-friendly
 ↪  visualization
```

2.3 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

```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
maxTypoLevel = 4  # Define the maximum classification level
```

2.4 Set working directory

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

- input time series of landscape metrics are located under `"results/Landscape_metrics"`
- output temporal monitoring results will be saved under `"results/Landscape_metrics"`

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

# Path to open time series of landscape metrics
open_metrics_path = paste0(localscratch, "results/Landscape_metrics")

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

2.5 Plot Metrics over Years

For each modeling strategy and each hierarchical typology level, diversity indices (Shannon, Simpson, Pielou) and landscape metrics (mean patch size, relative abundance, mean Euclidean distance, etc.) are loaded, aggregated, and visualized to track their dynamics over time. The outputs are saved as .png and .svg plots, providing clear and reproducible visualizations of spatiotemporal changes.

```r
# 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))


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

    # Define the folder corresponding to the typology level
    LevelFolder=paste0(open_metrics_path,"/","Hab_L",l)

    # Diversity indices over time -------------------------------------------------------

    # Load diversity indices
    pattern_csv <- paste0("^Diversity_indices_RF_", type_model, "_model_.*\\.csv$")
    div_files <- list.files(path = LevelFolder, pattern = pattern_csv, full.names = TRUE)
    print(div_files)

    # Read and combine all diversity index files
    div_data <- div_files %>%
      lapply(read.csv, sep = ";", dec = ".") %>%
      bind_rows() %>%
      mutate(Year_map = as.integer(as.character(Year_map)))


    # Plot Diversity Indices Over Time (.png)

    NOMpng=paste0(LevelFolder,"/","Diversity_Indices_Over_Time_RF_",type_model,"_model_",District,"_",Island,"_",Sa
    png(file = NOMpng)
    div_plot <- ggplot(div_data, aes(x =as.factor(Year_map))) +
      geom_point(aes(y = Shannon_index, color = "Shannon"),size=2.5) +
      geom_point(aes(y = Simpson_index, color = "Simpson"),size=2.5) +
      geom_point(aes(y = Pielou_index, color = "Pielou"),size=2.5) +
      labs(title = "Diversity indices over time",
           x = "Year",
           y = "Index Value",
           color = "Index") +
      theme_minimal() +
      #scale_color_manual(values = c("Shannon" = "darkgreen", "Simpson" = "steelblue", "Pielou" = "darkorange"))
      scale_color_viridis_d(option = "D")  # discrete viridis

    print(div_plot)
    dev.off()


    # Plot Diversity Indices Over Time (.svg)

    NOMsvg=paste0(LevelFolder,"/","Diversity_Indices_Over_Time_RF_",type_model,"_model_",District,"_",Island,"_",Sa
    svg(file = NOMsvg)
    print(div_plot)
    dev.off()
```

```r
# Landscape metrics over time  --------------------------------------------------

# Load landscape files
pattern_csv2 <- paste0("^Landscape_metrics_RF_", type_model, "_model_.*\\.csv$")
landscape_files <- list.files(path = LevelFolder,
                              pattern =pattern_csv2,
                              full.names = TRUE)

# Read and combine all landscape metrics files
landscape_data <- landscape_files %>%
  lapply(function(f) {
    df <- read.csv(f, sep = ";", dec = ".")
    # Extract year and typology level from filename
    year <- str_extract(f, "\\d{4}")
    level <- str_extract(f, "level_(\\d)") %>% str_remove("level_")
    df$Year <- as.integer(year)
    df$Typo_level <- as.integer(level)
    return(df)
  }) %>%
  bind_rows()


# Reshape before plotting
metrics_long <- landscape_data %>%
  pivot_longer(cols = c(
    Relative_abundance,
    Mean_patch_size_m2,
    Min_patch_size_m2,
    Max_patch_size_m2,
    Total_patch_size_m2,
    Largest_patch_index,
    Mean_euclidian_distance
  ),
  names_to = "Metric",
  values_to = "Value")


# Plot landscape metrics Over Time  (.png)

↪ NOMpng=paste0(LevelFolder,"/","Landscape_Metrics_Over_Time_RF_",type_model,"_model_",District,"_",Island,"_",Sa
png(file = NOMpng)
landscape_plot <- ggplot(metrics_long, aes(x = as.factor(Year), y = Value, color = Hab, group =
↪ Hab)) +
  geom_point(size=2.5) +
  facet_wrap(~ Metric, scales = "free_y") + # to create a sub graph by metric
  scale_color_viridis_d(option = "D") +  # Palette viridis discrète
  labs(title = "Temporal evolution of landscape metrics by habitat",
       x = "Year", y = "Metric value", color = "Habitat") +
  theme_minimal()+
  theme(legend.position = "bottom")

print(landscape_plot)
dev.off()


# Plot landscape metrics Over Time  (.svg)

↪ NOMsvg=paste0(LevelFolder,"/","Landscape_Metrics_Over_Time_RF_",type_model,"_model_",District,"_",Island,"_",Sa
```

```
    svg(file = NOMsvg)
    print(landscape_plot)
    dev.off()

  }

} # End of model type loop
```

This analysis offers a detailed assessment of ecological dynamics, both in terms of composition and spatial structure, across different levels of habitat typology complexity.