Step 6.2 Identifying misclassifications at the site scale

6.2.2 Rasterizing new polygon-based observed map

Diane ESPEL

2025-06-20

Contents

1 Objectives

This script aims to automate the rasterization of a corrected polygonal habitat map (see previous step 6.2.1) across multiple typology 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(terra)  # For raster and vector spatial data handling
library(sf)  # For reading and manipulating vector data (Simple Features)
library(dplyr)  # For data manipulation (tidyverse)
```

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 `"Year1"` : the acquisition year of the multispectral imagery
- the `"Res1"` : the spatial resolution of the 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
Year1 = "2022"  # acquisition year of multispectral imagery
Res1 = "50cm"  # spatial resolution of multispectral imagery
maxTypoLevel = 4  # Define maximum typology level
```

2.4 Set working directory

You must define two root directory (`"localHOME"` and `"localscratch"`) that serve as base path for your input and output data. These directories should point to the local environment where:

- input observed map (vector) with new typology is located under `"data/vector/Observed_map/NewTypo"`
- input reference raster is located under `"data/raster/Cut_image"`
- output raster of observed map will be saved under `"data/raster/Observed_map/NewTypo"`

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

# Path to open observed map (vector) with new typology
open_NewObsMap_vector_path = paste0(localscratch, "data/vector/Observed_map/NewTypo")

# Path to open raster layer
open_cut_raster_path = paste0(localscratch, "data/raster/Cut_image")

# Path to save rasterized observed map
save_NewObsMap_raster_path = paste0(localscratch, "data/raster/Observed_map/NewTypo")
```

::: {align="justify"} ## # **Load the polygons map and raster stack final**

```
# open shape file
print(paste0("on ouvre la carte vectorielle des habitats observés"))
FILE1 = paste0(open_NewObsMap_vector_path, "/", "Corrected_observed_map_NewTypo_",
    District, "_", Island, "_", Satellite1, "_", Year1, "_EPSG32739.shp")
polygons_map <- st_read(FILE1)

# open raster
print("on ouvre le raster final stack")
```

```
FILE2 = paste0(open_cut_raster_path, "/", District, "_", Island, "_Final_raster_stack_",
    Year1, "_", Res1, "_cut.TIF")
raster_map <- rast(FILE2)
```

##Rasterize each habitat level

This section of the script automates the rasterization of a corrected polygonal habitat map across multiple typology levels.

For each level of classification:

- The script first transforms categorical habitat codes into unique numeric identifiers to ensure compatibility with raster formats. This is achieved by generating a correspondence table (`correspondence_table`) mapping each unique habitat class to a numeric value, which is then joined to the spatial polygon dataset.
- The correspondence tables are optionally saved for traceability and future reference.
- An empty raster template (`raster_template`) is created for each level, matching the spatial extent, resolution, and coordinate reference system (CRS) of a predefined reference raster.
- Using this template, the polygons are rasterized based on the numeric values of the current typology level, with the `rasterize()` function. This step effectively converts vector habitat classifications into raster layers (`PolysToRaster`), enabling pixel-wise spatial analysis.
- Each resulting raster is saved as a separate GeoTIFF file, named to reflect the corresponding typology level and metadata such as island, satellite, and year.
- Memory is cleared after each iteration to optimize performance for large spatial datasets.

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


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

  # Convert habitat classes (characters) to numeric values----------------------
  print("Converting habitat classes from character to numeric codes")

  col <- paste0("Hab_L", l)
  valnum_col <- paste0("ValNum_Hab_L", l)


  # Create numeric correspondence table for current level
  unique_classes <- sort(unique(polygons_map[[col]]))
  correspondence_table <- data.frame(
    !!col := unique_classes,
    !!valnum_col := seq_along(unique_classes)
  )

  # Join numeric values to polygon map
  polygons_map <- polygons_map %>% left_join(correspondence_table, by = col)

  # OPTIONAL: Save the correspondence table
  file_table <- paste0(save_NewObsMap_raster_path, "/Correspondence_table_", col, "_NumValues.csv")
  write.table(correspondence_table, file_table, sep = ";", dec = ".", row.names = FALSE)
```

```
  # Create empty raster with same extent/resolution/CRS as reference raster ---

  print("Creating empty raster template")

  raster_template <- rast(ext(raster_map), # Define spatial extent of the future raster
                    res=res(raster_map)[1], # Define spatial resolution
                    crs=crs("+init=EPSG:32739")) # Define crs of the future raster


  # Rasterize: convert polygons to raster using the numeric values for this level ----------

  print("Rasterizing the polygon map")
  PolysToRaster <- rasterize(x = polygons_map,
                             y = raster_template,
                             field = paste0("ValNum_Hab_L", l))

  # Save the raster to file ------------------------------------------------


  ↪  NOMtiff=paste0(save_NewObsMap_raster_path,"/","Corrected_observed_map_NewTypo_",District,"_",Island,"_",Satelli
  ↪  Res1,"_level_",l,"_EPSG32739.TIF")
  writeRaster(PolysToRaster,NOMtiff,overwrite=T)

  rm(PolysToRaster) # free memory

} # End of typology levels loop
```