Preparing data for ENM/SDM

Emilio Berti

Species occurrence data

We selected and downloaded the occurrence data for the species *Podarcis muralis* (Laurenti, 1768) from GBIF: https://doi.org/10.15468/dl.x74f4b. I specified already some filters in the query from GBIF, namely:

- The coordinate uncertainty of the records must be $\leq 5km$.
- The year of the record must be ≥ 1970 and ≤ 2000.

The first filter removes records with high uncertainty, relatively to the spatial resolution of the climatic data we will use (ca. 12km). The second filter removes records outside the range of the climatic data we will use.

We load this data in R.

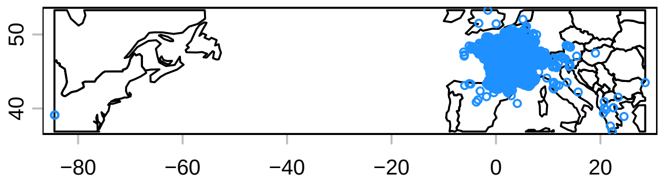
```r
library(terra)

gbif <- read.csv("../data/0002051-260120142942310.csv", sep = "\t")
```

> Note
>
> `sep = "\t"` specifies that the separator of the columns is a `TAB`, which is the standard used by GBIF.

This data frame has many columns that we do not need. We retain only the longitude and latitude columns and convert drop all duplicate coordinates.

```r
gbif <- gbif[, c("decimalLongitude", "decimalLatitude")]
gbif <- unique(gbif)

# load countries polygons
# this is from rnaturalearth package, which is required by CoordinateCleaner
```

```
country <- vect(rnaturalearth::countries110)

# plot gbif records
plot(crop(country, gbif))
points(gbif, cex = .5, col = "dodgerblue")
```



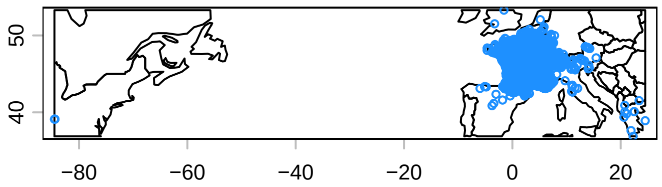
You can tell there is something wrong with this data, as this is a species found only in Europe, but we have points in the USA. This can be due to a variety of reasons, such as mistakes in the sampling and data preparation. For example, sometimes the reported coordinates are not where the species was found, but of the museum where the specimen is stored.

The package `CoordinateCleaner` performs several quality checks on GBIF data and flags potential inaccuracies.

```
library(CoordinateCleaner)

flags <- clean_coordinates(
  gbif,
  species = NULL,
  tests = c("capitals", "centroids", "equal", "gbif", "institutions", "seas", "zeros")
)
```

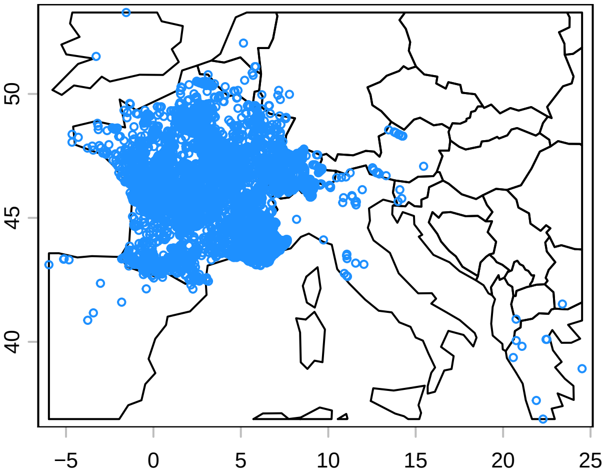
The data frame `flags` contains the column `.summary` with value `TRUE` if all tests did not find inaccuracies and `FALSE` if that data record failed at least one test. We use this to retain only GBIF records that have `.summary = TRUE`.

```
gbif <- gbif[flags$.summary, ]

plot(crop(country, gbif))
points(gbif, cex = .5, col = "dodgerblue")
```



There are still points in the USA, which we know cannot be correct. We can remove those manually.

```
gbif <- gbif[gbif$decimalLongitude >= -20, ]

plot(crop(country, gbif))
points(gbif, cex = .5, col = "dodgerblue")
```



We have now a data frame of cleaned occurrences from GBIF.

Pseudo-absences

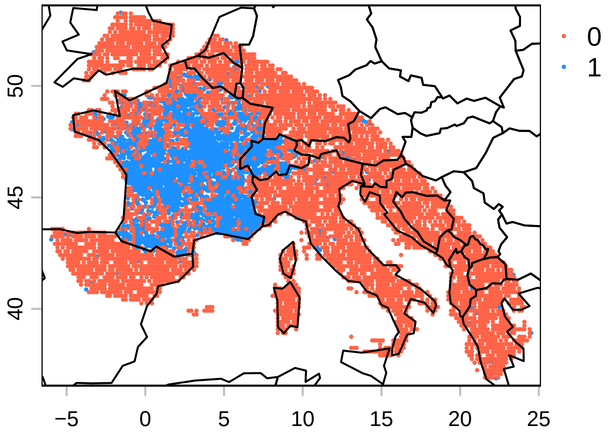
To be able to model the niche and the distribution of the species, we need also absences. We thus need to generate some *pseudo-absences*, i.e. simulated absences, and add them to the data frame. There are several ways to generate absences, but here we will focus only one one: randomly sampling the geographic area within the polygon inscribing all known occurrences. In doing so, however, we do not want to sample an absence in the same grid cell of a presence.

The code below show how to generate pseudo-absences following this approach.

```
library(terra)

# data frame as SpatVector
gbif <- vect(
  gbif,
  geom = c("decimalLongitude", "decimalLatitude"),
  crs = "EPSG:4326"
)

# (convex) hull inscribing all known occurrences
hull <- convHull(gbif)

# load one climate layer as template of the grid cell
grid <- rast("../data/wc2.1_10m_bio_1.tif") |> crop(hull)

# create a raster with
# - 0 if there is a gbif record in that cell
# - 1 if not
# - NA for sea cells
r <- rasterize(gbif, grid, fun = \(x) 0, background = 1)
r[is.na(grid)] <- NA

# remove areas outside the polygon inscribing all GBIF records
r <- mask(r, hull)

# sample absences
abs <- spatSample(
  r,
  length(gbif),       # n(abs) = n(pres)
  as.points = TRUE,   # return a SpatVector
  method = "weights", # trick to remove cells with a record (weight = r = 0)
  values = FALSE      # we do not care about the values of the grid template
)
```

We can now stitch the two SpatVector together.

```
library(terra)

gbif$occ <- 1 # presence
abs$occ <- 0  # absence

# combine into one SpatVector
p <- rbind(gbif, abs)

plot(
  p,
  "occ",                             # color by `occ`
  col = c("tomato", "dodgerblue"), # color palette
  cex = 0.3,
  fun = \() lines(country)          # outline of countries
)
```



```
d <- data.frame(
  lon = crds(p)[, 1],
  lat = crds(p)[, 2],
  occ = p$occ
)
```

We can now use the SpatVector `p` to extract climatic variables.

Climate data

In this course, we always use the WorldClim bioclimatic variables to model the niche of species. Bioclimatic variables are derived from temperature and precipitation data and are considered to have the strongest influence on the distribution of species. A list of all of them can be found at https://www.worldclim.org/data/bioclim.html.

Load eight bioclimatic variables.

```
library(terra)

# list of files of bioclimatic variables
ff <- list.files(
  "../data",          # where the files are
  pattern = "wc2.1", # wc = WorldClim
  full.names = TRUE   # full path
)
```

```r
# load them into memory
climate <- rast(ff)
climate
```

```
class       : SpatRaster
size        : 1080, 2160, 8  (nrow, ncol, nlyr)
resolution  : 0.1666667, 0.1666667  (x, y)
extent      : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (EPSG:4326)
sources     : wc2.1_10m_bio_1.tif
              wc2.1_10m_bio_12.tif
              wc2.1_10m_bio_13.tif
              ... and 5 more sources
names       : wc2.1~bio_1, wc2.1~io_12, wc2.1~io_13, wc2.1~io_14, wc2.1~io_15, wc2.1~bio_4,
min values  :   -54.72435,           0,           0,           0,      0.0000,       0.000,
max values  :    30.98764,       11191,        2381,         484,    229.0017,    2363.846,
```

Using `p`, we extract the values of the grid cells of `climate` for the occurrence data.

```r
d <- extract(climate, p, ID = FALSE, cell = TRUE)
```

Tip

`cell = TRUE` return also the ID of the cell of the raster where the records are found. This is useful to keep only one record per grid cell.

We assign the occurrence status (presence/absence) to this data frame.

```r
d$occ <- p$occ
```

Then, drop duplicate records, i.e. multiple records for the same grid cell.

```r
# drop rows with duplicated cells
d <- d[!duplicated(d$cell), ]

# drop the `cell` column
d <- d[, -which(names(d) == "cell")]

head(d)
```

```
  wc2.1_10m_bio_1 wc2.1_10m_bio_12 wc2.1_10m_bio_13 wc2.1_10m_bio_14
1       14.709969              510               60               20
2       11.454646              775               89               44
3        8.854438              933              106               60
4       10.599990              633               61               44
5       10.302510              632               58               45
6        2.760979              549               68               27
  wc2.1_10m_bio_15 wc2.1_10m_bio_4 wc2.1_10m_bio_5 wc2.1_10m_bio_6 occ
1        25.683231        686.3660        32.04650         2.11375   1
2        21.440331        589.9808        25.82825        -0.33375   1
3        18.089415        578.0707        22.92975        -2.12900   1
4         9.118695        567.0883        23.99325         0.50650   1
5         8.751935        560.5479        23.76875         0.34775   1
6        30.307224        617.9191        15.89150        -8.37175   1
```

Finally, make sure to have more or less the same number of presences and absences.

```r
table(d$occ) # not balanced
```

```
   0    1 
5016 1644 
```

```r
n <- table(d$occ)[["1"]]
index_pres <- which(d$occ == 1)
index_abs <- which(d$occ == 0)

# subsample
d <- d[c(index_pres, sample(index_abs, n)), ]

table(d$occ) # balanced
```

```
   0    1 
1644 1644 
```

We have now a data frame obtained from GBIF that is ready to be used for ENM/SDM.