GBIF data for species distribution modelling: A case study of Lithobius erythrocephalus (Chilopoda: Lithobiidae)

Zan Kuralt

- 1 Load packages
- 2 Getting and cleaning occurrence data
  - 2.1 Download occurrence data from GBIF
  - 2.2 Plot occurence data on map
  - 2.3 Remove duplicate rows
  - 2.4 Remove occurrences with faulty coordinates.
  - 2.5 Remove occurences anchored to country centroids
  - 2.6 Apply spatial thinning to occurrences
- 3 Getting and preparing environmental layers
  - 3.1 Get predictor variables from WorldClim and import downloaded Envirem layers
  - 3.2 Crop environmental layers to species occurrence extent
  - 3.3 Select envrionmental layers
  - 3.4 Test for multicollinearity using Variance Inflation Factor.
- 4 Modeling part
  - 4.1 Sample background points
  - 4.2 Create models across a range of settings
  - 4.3 Look at the results
  - 4.4 Select best model
  - 4.5 Evaluation plots
  - 4.6 Plot response curves
  - 4.7 Take a look at model prediction
  - 4.8 Check prediction for Slovenia

1 Load packages

```r
library(ENMeval)
library(rgbif)
library(maptools)
library(rgeos)
library(HH)
library(tidyverse)
library(rgdal)
library(scales)
library(spThin)
```

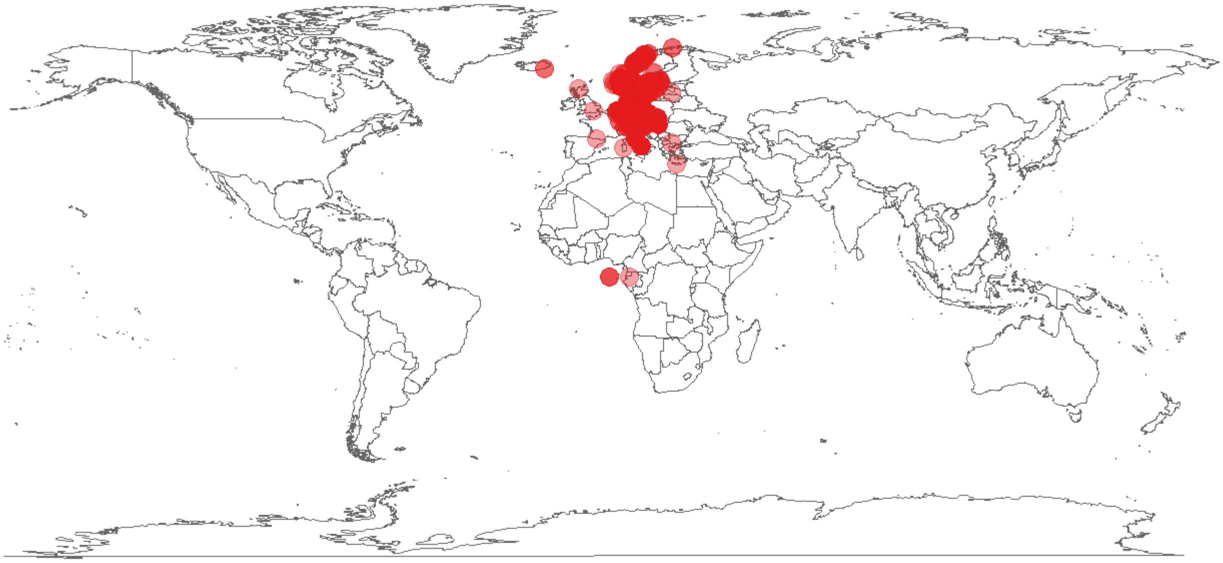
2 Getting and cleaning occurrence data

2.1 Download occurrence data from GBIF

```
occurrences <- as.data.frame(occ_data(scientificName = "Lithobius erythrocephalus", limit = 1500)[[2]])
```

2.2 Plot occurence data on map

```
## Rendering map...plotting 818 points
```



2.3 Remove duplicate rows

```
recs.dups <- duplicated(occurrences %>% dplyr::select(decimalLongitude, decimalLatitude))
occurrences <- occurrences[!recs.dups, ]
```

2.4 Remove occurrences with faulty coordinates.

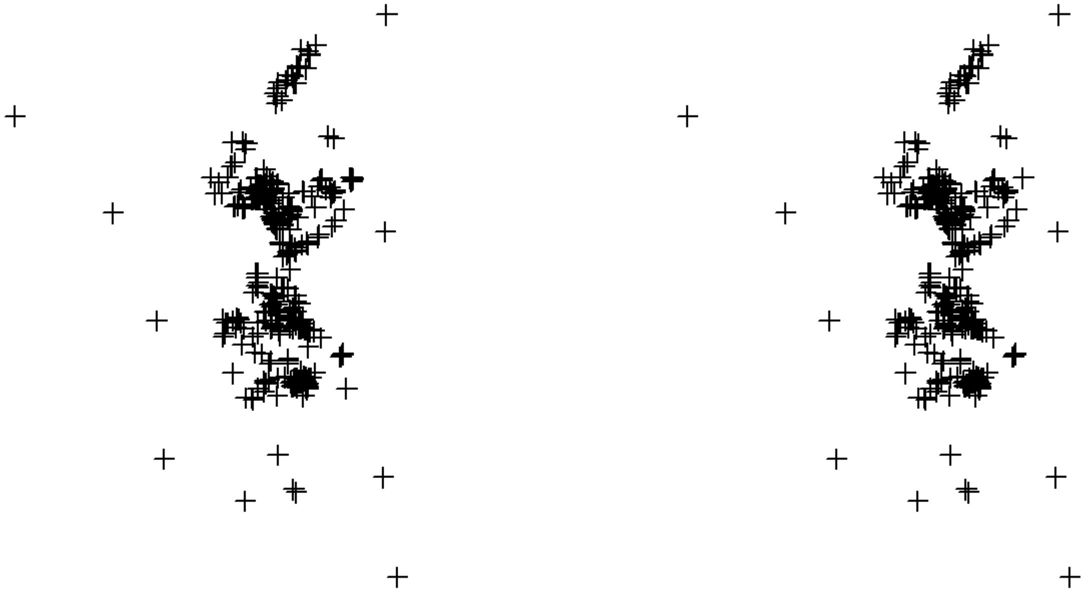
```
occurrences <- occurrences[occurrences$decimalLatitude > 22, ]
```

2.5 Remove occurences anchored to country centroids

```r
occs <- na.omit(occurrences[, c("decimalLatitude", "decimalLongitude")])
coordinates(occs) <- c("decimalLongitude", "decimalLatitude")
proj4string(occs) <- CRS("+init=epsg:4326")

boundaries <- readOGR(dsn = "NATIONAL BOUNDARIES", layer = "euro_boundaries")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\zanku\Documents\Work\Strige\Workshop_on_Soil_Zoology_2018\lithobi
us_erythrocephalus_sdm\NATIONAL BOUNDARIES", layer: "euro_boundaries"
## with 69 features
## It has 70 fields
## Integer64 fields read as strings:  OBJECTID ID_0
```

```r
centroids <- gCentroid(spgeom = boundaries, byid = TRUE)

initcrs <- CRS("+init=epsg:4326")
proj4string(centroids) <- initcrs
centroids <- sp::spTransform(centroids, CRSobj = CRS("+init=epsg:3035"))
buff_centro <- gBuffer(centroids, width = 20000) # buffer of 20 km
buff_centro <- sp::spTransform(buff_centro, CRSobj = initcrs)

no.centro <- gDifference(occs, buff_centro) # Remove points at country centroids
par(mfrow=c(1,2))
plot(occs)
plot(no.centro)
```

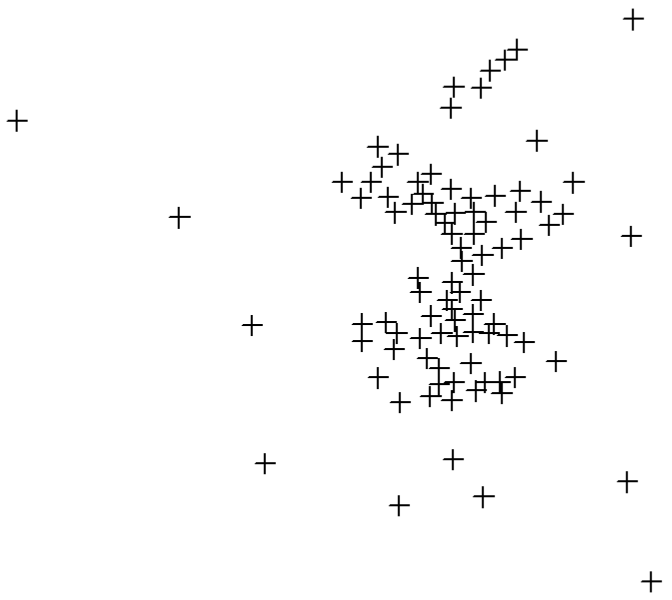


2.6 Apply spatial thinning to occurrences

```
dat <- as.data.frame(x = no.centro, row.names = 1:length(no.centro))

dat$species <- "L.erythrocephalus"
thinned <- thin(loc.data = dat, lat.col = "y", long.col = "x",
                spec.col = "species",
                thin.par = 70,
                reps = 20,
                out.dir = "thinned",
                out.base = "litho_erythro_70",
                write.log.file = TRUE,
                log.file = "thinning.txt",
                max.files = 1)
```

```
## *********************************************
##  Beginning Spatial Thinning.
##  Script Started at: Thu Aug 30 09:08:28 2018
## lat.long.thin.count
## 88 89 90 91
##  1 10  5  4
## [1] "Maximum number of records after thinning: 91"
## [1] "Number of data.frames with max records: 4"
## [1] "Writing new *.csv files"
## [1] "Writing file: thinned/litho_erythro_70_thin1.csv"
```

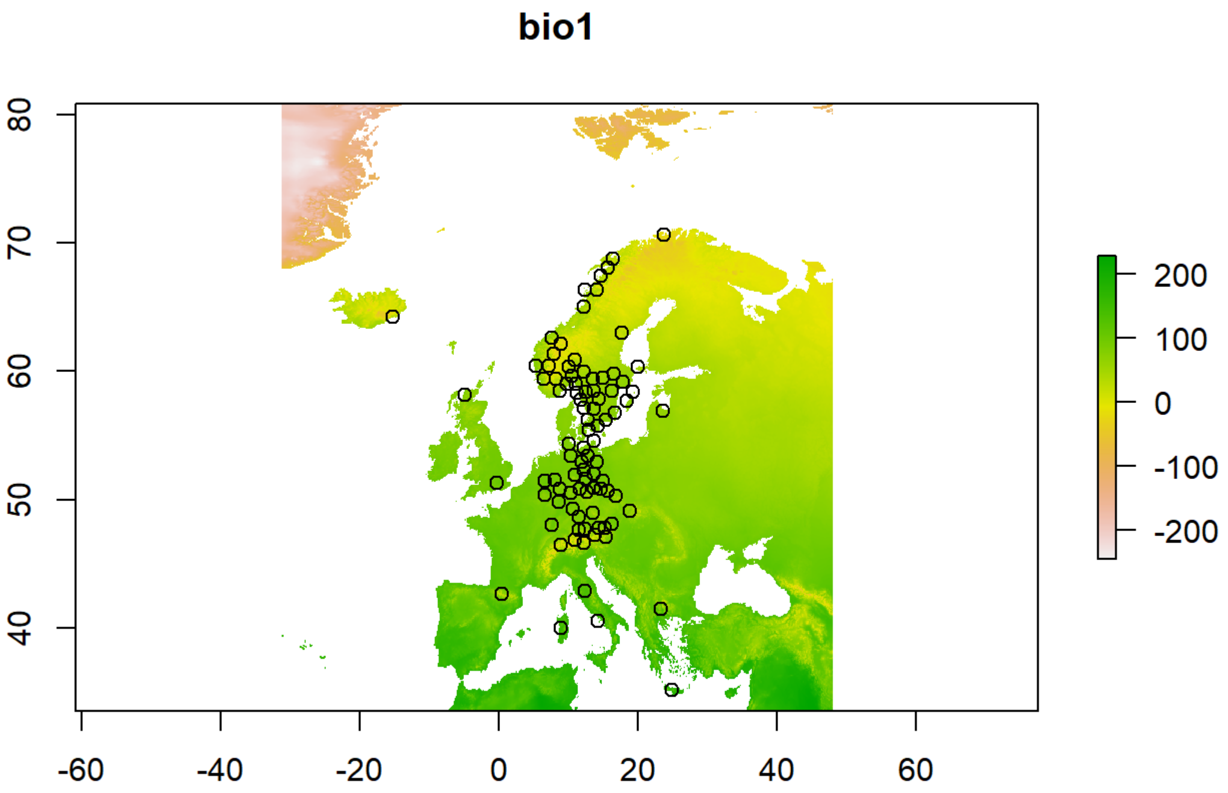
```
thn <- read.csv(file = "thinned/litho_erythro_70_thin1.csv")
coordinates(thn) <- ~ x + y

plot(thn)
```



3 Getting and preparing environmental layers

3.1 Get predictor variables from WorldClim and import downloaded Envirem layers

```r
# bioclimatic variables for current conditions directly from worldclim
bioclim <- getData(name = "worldclim", var = "bio", res = 2.5)

# envirem dataset downloaded from http://envirem.github.io/
xy <- sapply(list.files("envirem2.5/", full.names = TRUE),
             FUN = raster)
envirem <- stack(xy)
# rename layers
envinames <- list.files("envirem2.5/", full.names = FALSE)
envinames <- unlist(strsplit(x = envinames, split = ".tif"))
names(envirem) <- envinames

# altitude related layers downloaded from http://envirem.github.io/
alt <- sapply(list.files("alt/", full.names = TRUE),
              FUN = raster)
altitude <- stack(alt)
# rename layers
alts <- list.files("alt2.5/", full.names = FALSE)
alts <- unlist(strsplit(x = alts, split = ".tif"))
names(altitude) <- alts

bioclim <- raster::resample(bioclim, envirem)
alts <- raster::resample(altitude, envirem)
covars <- stack(bioclim, envirem, alts)

# Plot first raster in the stack, bio1
plot(covars[[1]], main=names(covars)[1])

# Add points for all the occurrence points onto the raster
points(thn)
```



3.2 Crop environmental layers to species occurrence extent

```
# Make a SpatialPoints object
occs.sp <- SpatialPoints(thn)

# Get the bounding box of the points
bb <- bbox(occs.sp)

# Add 5 degrees to each bound by stretching each bound by 10, as the resolution is 0.5
degree.
bb.buf <- extent(bb[1]-10, bb[3]+10, bb[2]-10, bb[4]+10)

# Crop environmental layers to match the study extent
envs.backg <- crop(covars, bb.buf)
```

3.3 Select envrionmental layers

```
cvrs <- envs.backg[[c(3, 10, 17, 23, 31, 37)]]
```

3.4 Test for multicollinearity using Variance Inflation Factor.

Variables with VIF > 10 should be excluded.

```
data.frame(vif(as.data.frame(cvrs)))
```

```
##                                      vif.as.data.frame.cvrs..
## bio3                                                 4.683363
## bio10                                                2.500806
## bio17                                                1.751838
## current_2.5arcmin_continentality                     3.614712
## current_2.5arcmin_PETDriestQuarter                   2.952672
## current_2.5arcmin_tri                                1.430960
```

4 Modeling part

4.1 Sample background points

```
bg <- randomPoints(envs.backg[[1]], n = 10000)
bg <- as.data.frame(bg)
```

4.2 Create models across a range of settings

```
occ <- as.data.frame(thn)[, 2:3]
mod <- ENMevaluate(occ = occ,
                     env = cvrs,
                     bg.coords = bg,
                     method = "block",
                     RMvalues = seq(from = 1, to = 4, by = 0.5),
                     fc = c("LTPHQ", "LTQH", "LQ", "LQH", "LPQ", "LT", "L", "LQHP"),
                     rasterPreds = TRUE,
                     parallel = TRUE)
```

4.3 Look at the results

```
results <- mod@results
settings <- as.character(mod@results$settings)
setts <- t(as.data.frame(strsplit(x = settings, split = "_")))
colnames(setts) <- c("FC", "RM")
rownames(setts) <- 1:nrow(setts)
setts <- data.frame(setts)
setts$Mean.AUC <- results$Mean.AUC
setts$dAICc <- results$delta.AICc
setts <- filter(setts, dAICc < 2)
setts$scaled.AUC <- scale(setts$Mean.AUC)
setts$scaled.dAICc <- scale(setts$dAICc)
setts$combined.scaled <- setts$scaled.dAICc - setts$scaled.AUC
setts$scaled.rank <- rank(setts$combined.scaled)
setts$setting <- paste(setts$FC, "_", setts$RM, sep = "")
setts[order(setts$scaled.rank), ]
```

```
##      FC  RM  Mean.AUC    dAICc scaled.AUC scaled.dAICc combined.scaled
## 1 LTPHQ   2 0.8534714 0.000000  1.3867640   -1.4090076      -2.7957716
## 3 LTPHQ 3.5 0.8520726 1.033480 -0.1518214    0.2240036       0.3758250
## 4  LQHP 3.5 0.8519962 1.037455 -0.2358340    0.2302847       0.4661187
## 2 LTPHQ   3 0.8513023 1.495927 -0.9991086    0.9547193       1.9538279
##   scaled.rank   setting
## 1           1   LTPHQ_2
## 3           2 LTPHQ_3.5
## 4           3  LQHP_3.5
## 2           4   LTPHQ_3
```

4.4 Select best model

```
best.model <- min(setts$scaled.rank)
print(paste("Selected settings are: ", setts$setting[setts$scaled.rank == best.model],
sep = ""))
```

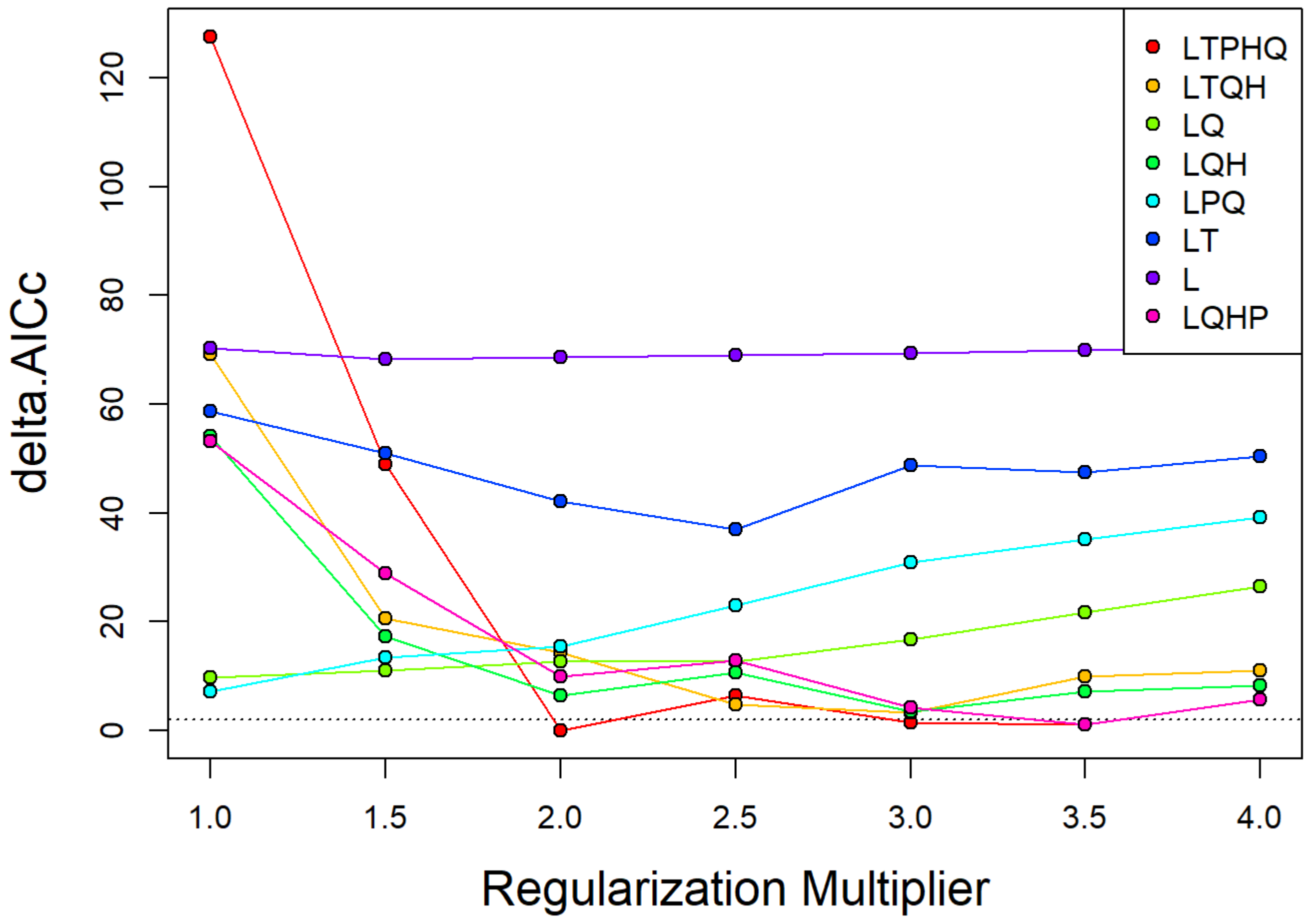
```
## [1] "Selected settings are: LTPHQ_2"
```

```
aic.opt <- mod@models[[which(setts$scaled.rank == best.model)]]
aic.opt
```

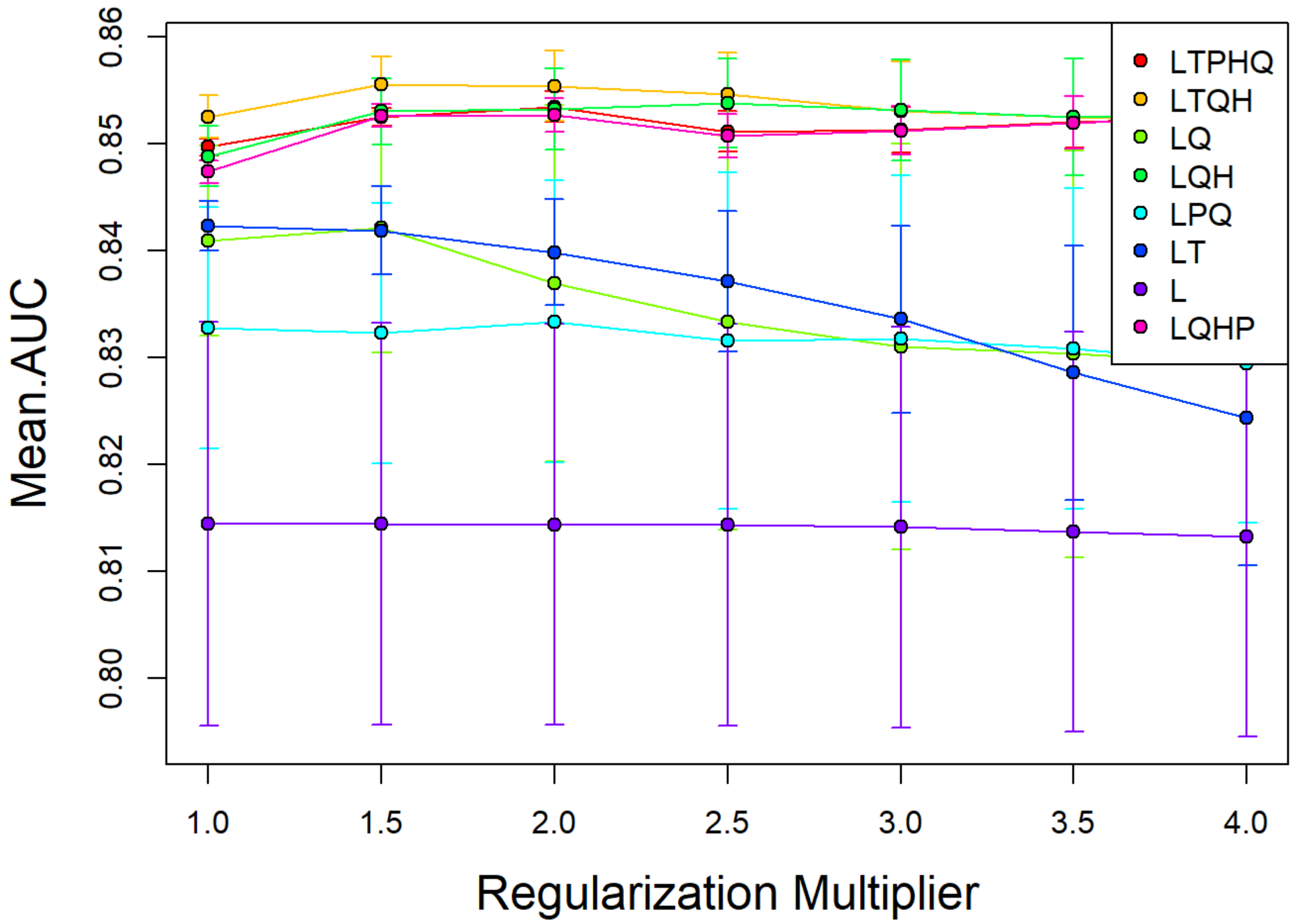
```
## class    : MaxEnt
## variables: bio3 bio10 bio17 current_2.5arcmin_continentality current_2.5arcmin_PETD
riestQuarter current_2.5arcmin_tri
## output html file no longer exists
```

4.5 Evaluation plots

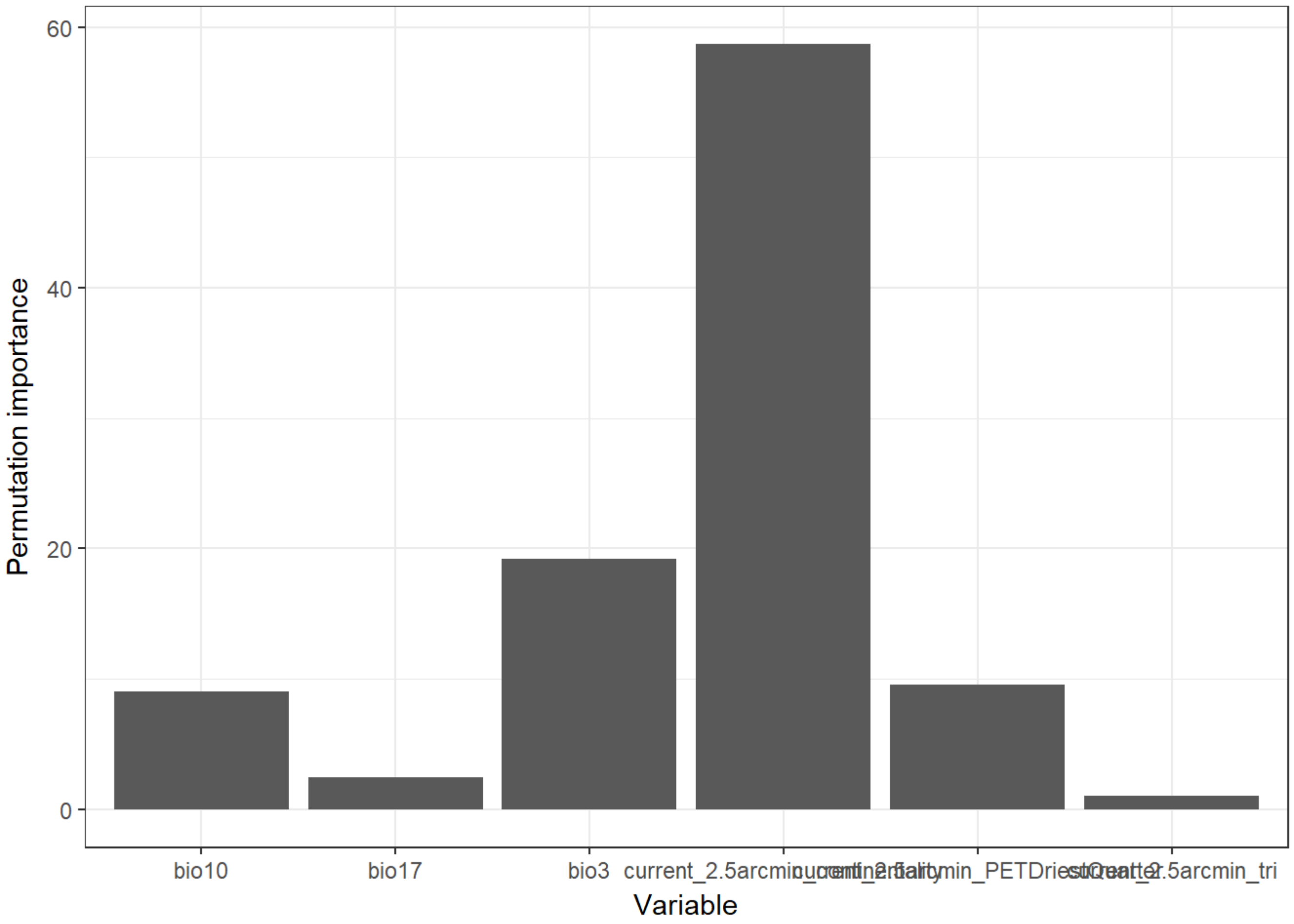
```
eval.plot(mod@results)
```



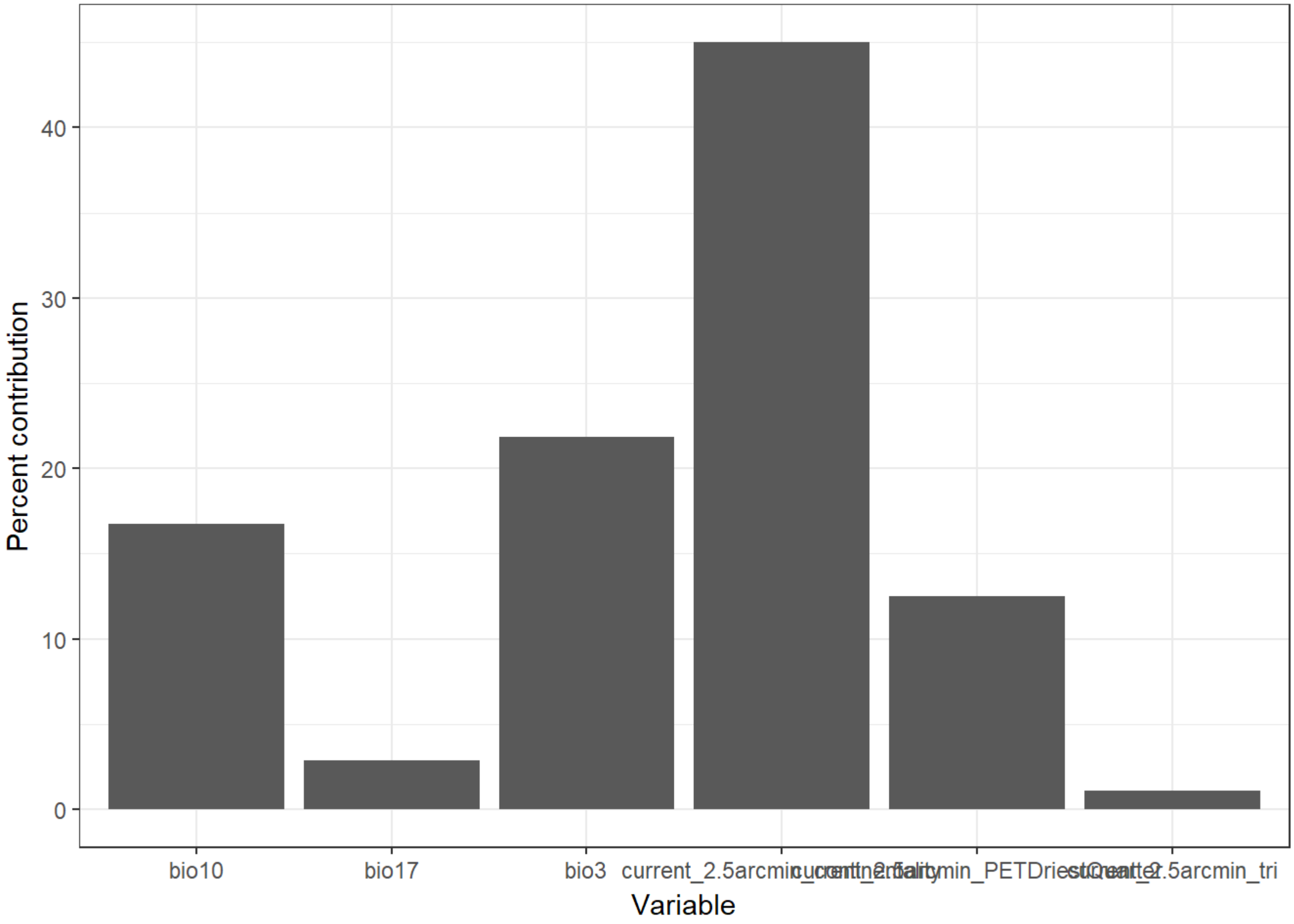
```
eval.plot(mod@results, 'Mean.AUC', var='Var.AUC')
```



```
df <- var.importance(aic.opt)

ggplot(df) +
  geom_col(aes(x = variable, y = permutation.importance)) +
  theme_bw() +
  xlab("Variable") +
  ylab("Permutation importance")
```

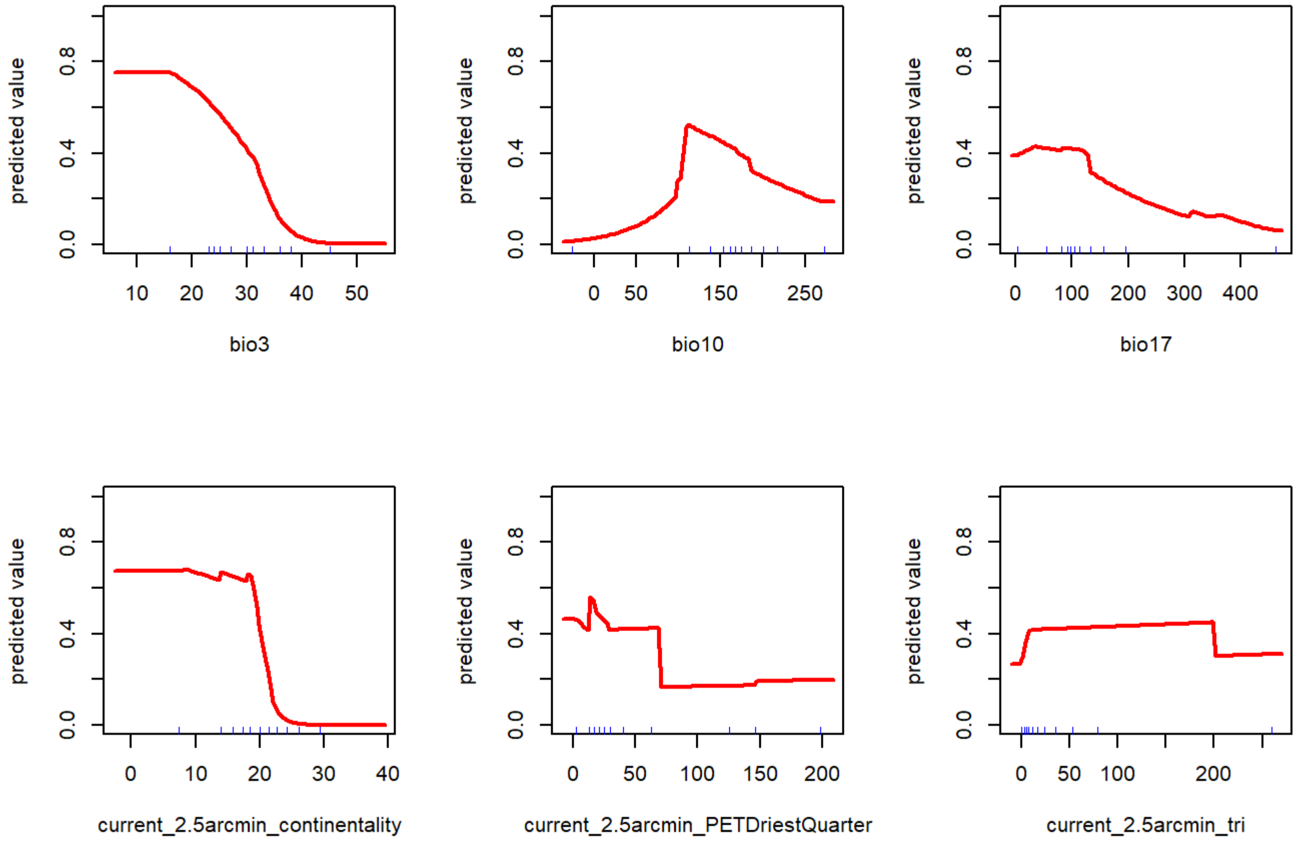


```
ggplot(df) +
  geom_col(aes(x = variable, y = percent.contribution)) +
  theme_bw() +
  xlab("Variable") +
  ylab("Percent contribution")
```



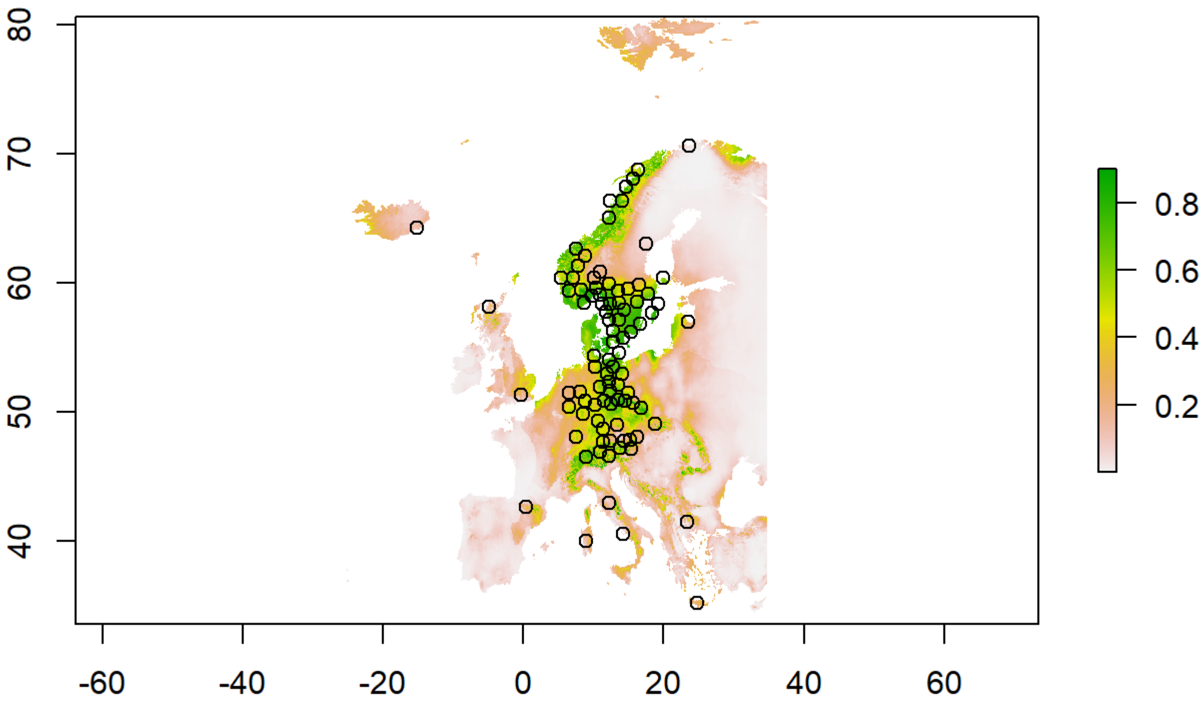
4.6 Plot response curves

```
response(aic.opt)
```



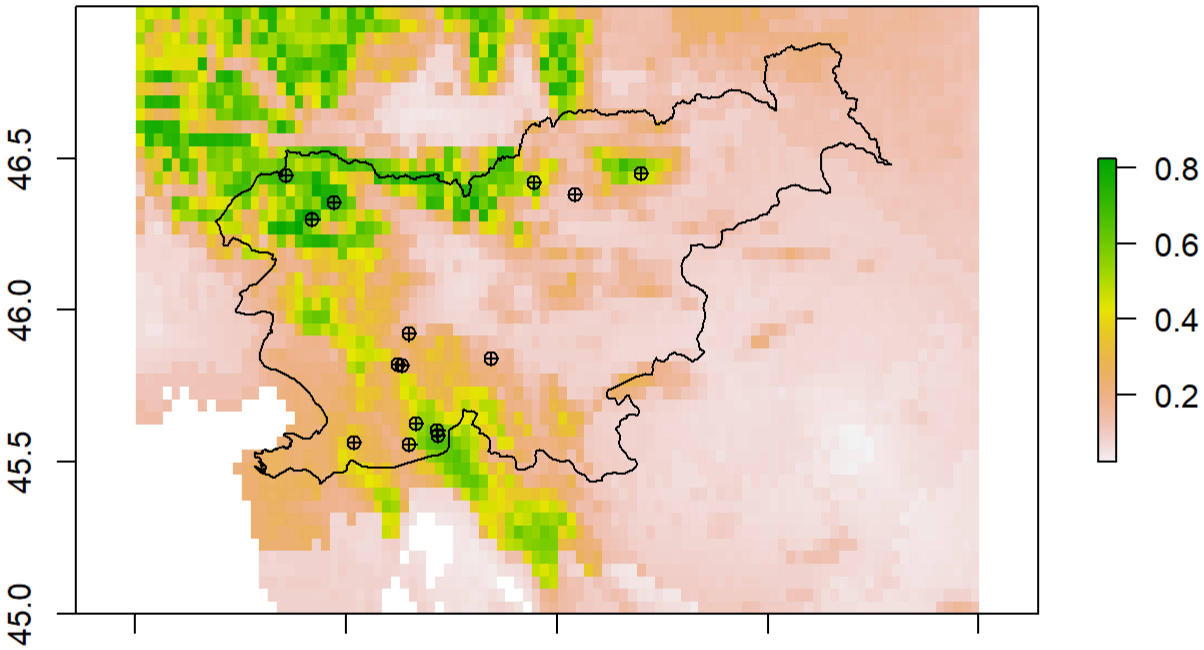
4.7 Take a look at model prediction

```
predictions <- raster::predict(object = cvrs, model = aic.opt)
plot(predictions)
points(occs.sp)
```



4.8 Check prediction for Slovenia

```
slo_bound <- getData(name = "GADM", country = "SVN", level = 0)

#Lets take a closer look
slov <- raster()
extent(slov) <- c(13, 17, 45, 47)
slovenija <- crop(x = predictions, y = slov)
plot(slovenija)
plot(slo_bound, add = TRUE)

slo <- read.csv(file = "litho_erythro_slo.csv", header = TRUE, fileEncoding = "UTF-8")
# colnames(slo) <- c("species", "x", "y", "locality", "habitat", "leg.")
coordinates(slo) <- ~ y + x

slocrs <- CRS("+init=epsg:3912")
finalcrs <- CRS("+init=epsg:4326")
proj4string(slo) <- slocrs
slo_erythro <- sp::spTransform(slo, CRSobj = finalcrs)

points(slo_erythro, pch = 10)
```



46.5
46.0
45.5
45.0
13
14
15
16
17
0.8
0.6
0.4
0.2