Table of Contents

# SDMsel

**SDMsel** (Species Distribution Model selection) provides a framework that facilitates user in preparing data for analysis, train and evaluate models. It also implements functions for data driven variable selection and model tuning and includes some utilities to display results (at the moment it implements only MaxEnt models). **SDMsel** uses its own script to predict MaxEnt models that results to be much faster for large datasets than native prediction made using Java software. This reduce considerably the computation time when tuninig the model using the **AICc**.

## Installation

You can get the latest stable version from CRAN:

install.packages("SDMsel")

or latest development version from github:

devtools::install\_github("sgvignali/SDMsel")

## Example

Load required packages:

library(SDMsel)  
#> Loading required package: zeallot  
#> Loading required package: ggplot2  
#> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#> | \_\_\_\_ \_\_\_\_ \_\_ \_\_ \_ |  
#> | / \_\_\_| | \_ \ | \/ | \_\_\_ \_\_\_ | | |  
#> | \\_\_\_ \ | | | || |\/| |/ \_\_| / \_ \| | |  
#> | \_\_\_) || |\_| || | | |\\_\_ \| \_\_/| | |  
#> | |\_\_\_\_/ |\_\_\_\_/ |\_| |\_||\_\_\_/ \\_\_\_||\_| |  
#> |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  
#>   
library(dismo)  
#> Loading required package: raster  
#> Loading required package: sp

Obtain data for analysis, we use the dataset provided by dismopackage:

files <- list.files(path = paste(system.file(package = "dismo"), "/ex", sep = ""), pattern = "grd", full.names = T)  
predictors <- stack(files)  
file <- paste(system.file(package = "dismo"), "/ex/bradypus.csv", sep = "")  
bradypus <- read.table(file, header = TRUE, sep = ",")  
train\_coords <- bradypus[, 2:3]  
set.seed(25)  
bg\_coords <- randomPoints(predictors, 5000) # Extract random points

Prepare the data for the analisys with SDMsel:

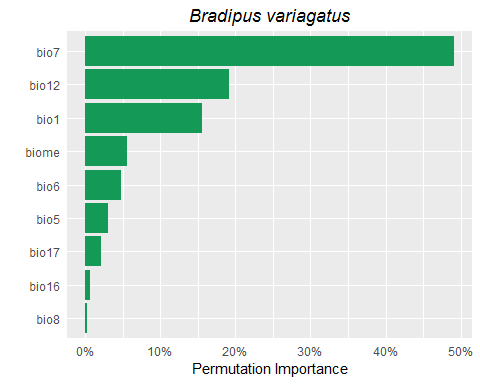
train <- prepareSWD(species = "Bradipus variagatus", coords = train\_coords, env = predictors, categoricals = "biome")  
#> Extracting environmental condition for Bradipus variagatus...  
bg <- prepareSWD(species = "Bradipus variagatus", coords = bg\_coords, env = predictors, categoricals = "biome")  
#> Extracting environmental condition for Bradipus variagatus...  
#> Warning: 5 locations are NA for some environmental variables, they will be discard!

Train a MaxEnt model:

model <- trainMaxent(presence = train, bg = bg, rm = 1, fc = "lqp", type = "cloglog", iter = 500)  
#> Loading required namespace: rJava

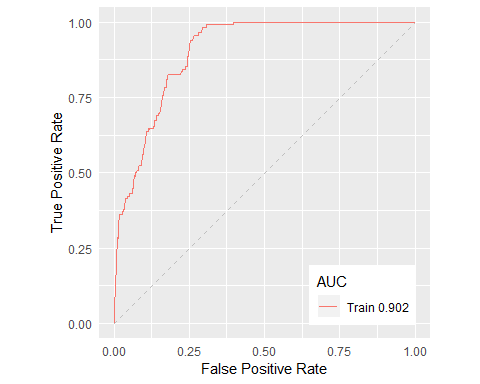
Plot variable importance:

plotVarImp(model, type = "permutation", color = "#159957")



Plot ROC curve:

plotROC(model)

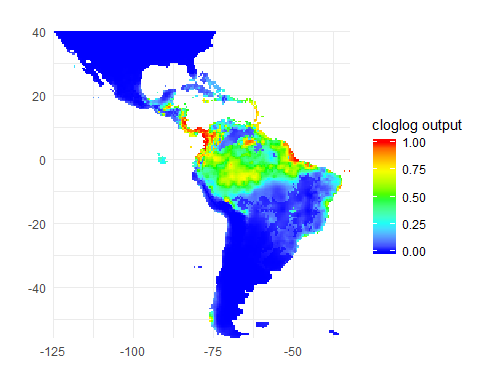


Make prediction:

pred <- predict(model, predictors)

Plot prediction:

plotPred(pred, lt = "cloglog output")



# Speed test

We test now the difference in speed between the predict function of **dismo** package that uses the Java software and the predict function of **SDMsel** package. We also test if the results are equivalent.

The following code creates a function to test the equivalence of the results:

my\_check <- function(values) {  
 error <- 1e-7   
 max\_error <- max(values(values[[1]]) - values(values[[2]]), na.rm = T)   
 max\_error < error  
}

If the results are different for more than 1e-7 the test will fail. The next function converts an object of class Maxent into an object of class MaxEnt (dismo pkg):

me\_model <- Maxent2MaxEnt(model)

Run the test with 10 replicates:

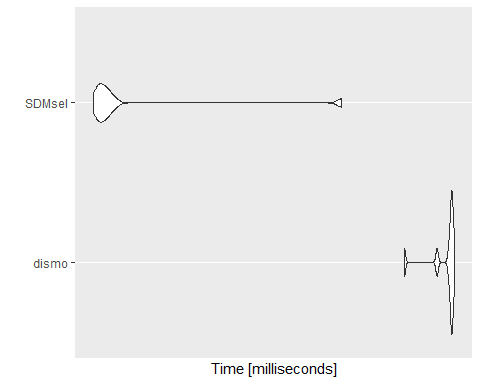
library(microbenchmark)  
res <- microbenchmark("dismo" = pred <- {predict(me\_model, predictors)},  
 "SDMsel" = pred <- {predict(model, predictors)},  
 times = 10,  
 check = my\_check)

Print results:

res  
#> Unit: milliseconds  
#> expr min lq mean median uq max neval  
#> dismo 550.0248 628.3019 622.6394 632.6830 634.8571 639.0053 10  
#> SDMsel 217.5260 220.5965 247.0092 223.8093 227.5432 455.2680 10

Plot results:

autoplot(res)  
#> Coordinate system already present. Adding new coordinate system, which will replace the existing one.



**SDMsel** is at least twice faster than native Java software and the speed improves much more for large datasets when using parallel computation (i.e. see details in predict function).