Example for Species distribution models (random forest) based on environmental variables

The data as well as the modeling method is part of the TALE project (Towards multifunctional agricultural landscapes in Europe).

Step 1 – Prepare R

Install the packages 'dismo', 'caret' and 'randomForest' in R.

Step 2 – Collect the Data

The data used in the present example is already collected and can be found in the folder *"source"*.

The dataset *"SQ_presences.csv"* includes information on 18 environmental variables (land cover, soil, climate, linear elements and distance to water and to highways) for 271 presence observations. The dataset *"SQ_pseudoabs.csv"* includes information on the same environmental variables for 1055 pseudo-absences. We use pseudo-absences because true absence observations are (as it is often the case) not available for our example species. In the present case, presence observations for other species were used as pseudo-absences, excluding a radius of 500 m around the presences of the example species. The coordinates itself are missing from both datasets as the species is endangered and the presence coordinates are therefore not publicly available.

The file *"SQ_VariableMaps.RData"* contains rasters of all environmental variables used in this example and will be used for predicting the habitat of our example species.

Step 3 – Variable selection

The complete code of the variable selection process can be found in the R script *"Variable_Selection.R"*.

1. Preparation

The variable selection requires the packages ‘caret’ and ‘randomForest’. Do not forget to select your correct working directory!

The presence and the pseudo-absence datasets are imported and duplicate entries are removed.

```
> # Load the required packages for the variable selection.
>
> require (caret)
> require (randomForest)
> options (warn=-1)
>
> # Choose the working directory.
>
> setwd ("YourWorkingDirectory")
>
> # Load the provided species data.
> # Note: The datasets do not include coordinates because the species is
> # endangered and the presence coordinates are thus not for public use.
>
> SQ_presences <- read.csv ("source/SQ_presences.csv", row.names=1)
> SQ_pseudoabs <- read.csv ("source/SQ_pseudoabs.csv", row.names=1)
>
> # Duplicate entries are removed before continuing:
>
> SQ_presences <- SQ_presences[!duplicated (SQ_presences),]
> SQ_pseudoabs <- SQ_pseudoabs[!duplicated (SQ_pseudoabs),]
```

2. Feature selection

The datasets for presence and pseudo-absence include information on 18 environmental variables (8 land cover types, 4 soil characteristics, temperature, precipitation, linear elements and distance to water and to highways).

```
> colnames (SQ_presences)
 [1] "forest"       "cropland"     "orchard"      "pasture"      "wetland"      "settlement"
 [7] "novegetation" "water"        "lin_elements" "forest_edges" "temperature"  "precipitation"
[13] "soil_AWC"     "soil_BD"      "soil_CBN"     "soil_SolK"    "dist_water"   "dist_highway"
```

The presence points are joined with the pseudo-absences. Care should be taken that the ratio between occurrence and (pseudo-)absence data does not exceed 1:10 (Barbet-Massin et al., 2012). If the balance between occurrences and pseudo-absences is not given, the resulting models will focus on the much bigger data set and the prediction quality will be poor despite a possible good model quality. As we work with single species data with a minimum of 30 occurrence points, our standard size for pseudo-absence datasets is 300 pseudo-absences. In the example, we join 169 occurrences with 300 randomly chosen pseudo-absence points. The seed in R is set so that always the same random set of pseudo-absences is drawn from the dataset for comparability reasons.

```
> set.seed (10)
> SQ_jointdata1 <- rbind (SQ_presences, SQ_pseudoabs[sample(nrow(SQ_pseudoabs), 300),])
>
> # presence and pseudo-absence is then coded into the "occurrence" column
>
> SQ_jointdata1$occurrence <- c(rep (1,nrow(SQ_presences)),rep(0,300))
```

Then, the settings for the feature selection are chosen. We use the function *rfe ()* (‘*caret’* package) for feature selection that runs a recursive feature selection over our 18 environmental variables. The feature selection can be tuned by the function *rfeControl ()*. Furthermore, the subset steps that the feature selection should use are saved separately as *subsets*. The feature selection will thus start with all 18 environmental variables and will drop two in the first step, and so on.

```
> ctrl <- rfeControl(functions = rfFuncs,  method = "repeatedcv", repeats=5,
+                verbose = FALSE, saveDetails = TRUE,
+                returnResamp="all")
>
> # the 'subsets' steps are used for the selection of the variable sets
>
> subsets <- c (1,2,3,4,5,6,7,8,9,10,11,12,14,16,18)
```

Subsequently we start the feature selection (only run if you’ve got a few minutes):

```
> RFEresult1 <- rfe (SQ_jointdata1[,1:18], SQ_jointdata1$occurrence, sizes = subsets,rfeControl = ctrl)
> RFEresult1

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold, repeated 5 times)

Resampling performance over subset size:

 Variables   RMSE Rsquared  RMSESD RsquaredSD Selected
         1 0.3308   0.5287 0.04287    0.12545
         2 0.3142   0.5709 0.03615    0.10415
         3 0.2697   0.6830 0.04225    0.10017
         4 0.2550   0.7214 0.03804    0.08879
         5 0.2476   0.7393 0.04152    0.09420
         6 0.2426   0.7421 0.04494    0.09347        *
         7 0.2430   0.7436 0.04319    0.09107
         8 0.2468   0.7381 0.04052    0.08867
         9 0.2450   0.7393 0.04002    0.08656
        10 0.2456   0.7392 0.04009    0.08729
        11 0.2463   0.7389 0.03839    0.08566
```

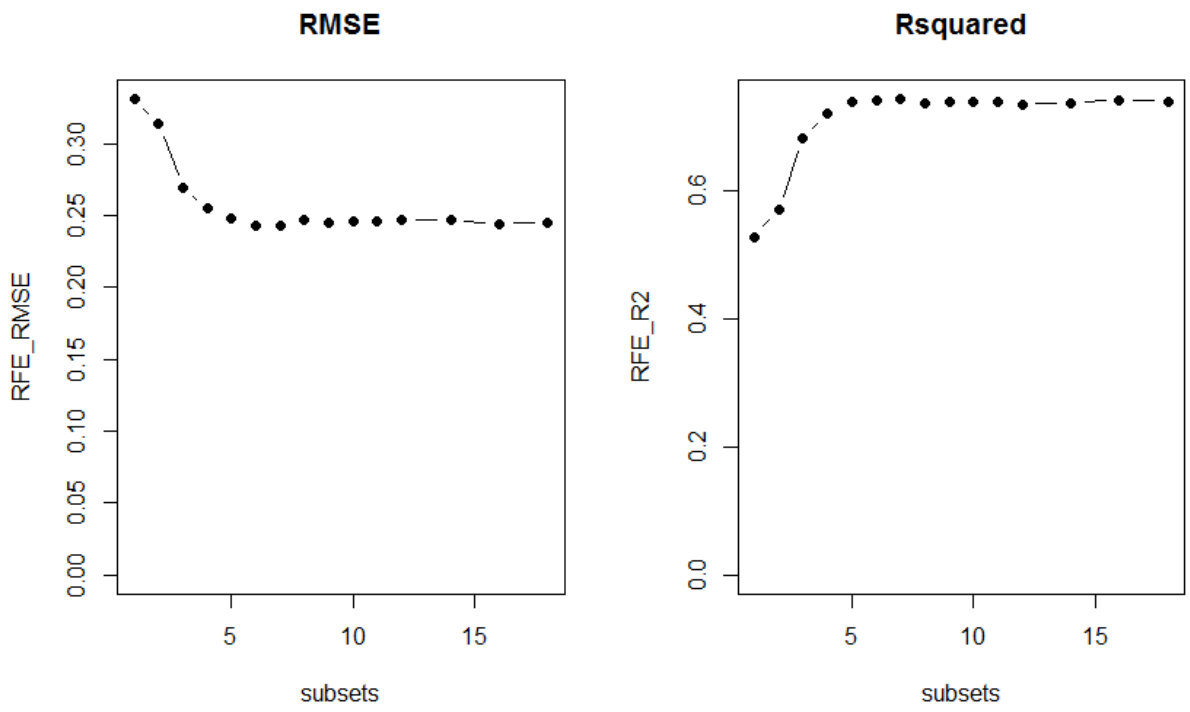
```
      12 0.2468   0.7363 0.03845    0.08579
      14 0.2465   0.7384 0.03919    0.08801
      16 0.2442   0.7431 0.03918    0.08662
      18 0.2449   0.7408 0.03978    0.08732

The top 5 variables (out of 6):
  dist_water, water, precipitation, forest_edges, temperature
```

Now we see an overview table over the results of the variable selection with statistical values (RMSE and pseudo-R2) for each of the subsets sizes. We also get an automatic recommendation for the best variable set size (6 variables).

When we plot the RMSE and the pseudo-R2 of the variable selection, we see that for the larger subsets, the RMSE and the pseudo-R2 only differ slightly at higher variable set sizes:

```
> RFE_RMSE <- RFEresult1$results$RMSE
> RFE_R2 <- RFEresult1$results$Rsquared
>
> par (mfrow =c(1,2))
> plot (RFE_RMSE~subsets, type="b", main="RMSE", pch=16, ylim=c(0,max(RFE_RMSE)))
> plot (RFE_R2~subsets, type="b", main="Rsquared", pch=16, ylim=c(0,max(RFE_R2)))
```



Therefore, the environmental variables with the lowest importance can be dropped without the loss of much model quality. If we set the acceptable quality drop for RMSE and pseudo-R2 to 5 %, we get

a recommended variable set of five environmental variables for the RMSE and of four variables for the pseudo-R2:

```
> bestset_RMSE <- which ( ( RFE_RMSE/min(RFE_RMSE) ) <=1.05 ) [1]
> bestset_RMSE
[1] 5
>
> bestset_R2 <- which ( ( RFE_R2/max(RFE_R2)*100 ) >=95 ) [1]
> bestset_R2
[1] 4
```

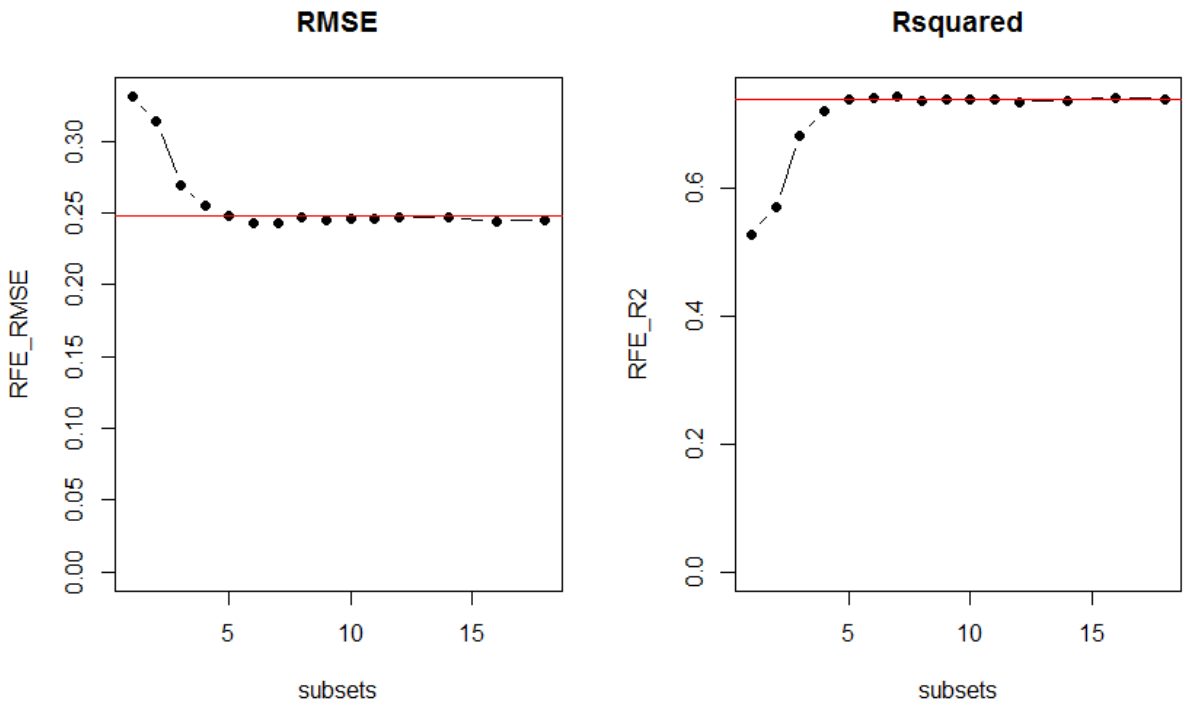
The smallest set with a maximum loss of 5% in both values is therefore the set with five environmental variables:

```
> chosenVars1 <- pickVars (RFEresult1$variables,max(bestset_RMSE,bestset_R2))
> chosenVars1
[1] "dist_water"    "water"         "precipitation" "forest_edges"  "temperature"
```

The feature selection thus reveals that for this species, the environmental variables that can be used to determine the breeding habitat are the distance to waterbodies, the amount of area covered by water, the length of forest edges as well as precipitation and temperature.

When we plot the RMSE and the pseudo-R2 of the chosen predictor set into the entire feature selection results, we see that there is no obvious loss of model quality:

```
> par (mfrow =c(1,2))
> plot (RFE_RMSE~subsets, type="b", main="RMSE", pch=16, ylim=c(0,max(RFE_RMSE)))
> lines (c(0,20),rep (RFE_RMSE[length (chosenVars1)],2), col="red")
> plot (RFE_R2~subsets, type="b", main="Rsquared", pch=16, ylim=c(0,max(RFE_R2)))
> lines (c(0,20),rep (RFE_R2[length (chosenVars1)],2), col="red")
```



3. Concluding remarks

This chosen variable set can now be used to create species distribution models. However, the dataset is with 300 pseudo-absences quite small. This can lead to suboptimal results when the range of the environmental variables is broader in habitat prediction than it is for modelling. It is therefore recommended to repeat modelling at least 10 times with different datasets (Barbet-Massin et al., 2012). For each species, information of the usual (breeding) habitat can also be found in numerous publications that should be used to cross-check the results of the feature selection.

The code for nine additional feature selections can be found in the appendix at the bottom of the R script *"Variable_Selection.R"*. The results are saved in the folder *"source"*. When the resulting variable sets are loaded, we see that the selected variables are similar in all runs:

```
> load ("source/RFEresults_chosenVars.RData")
>
> # The chosen variable sets of all 10 repeats are:
>
> chosenVars1
[1] "dist_water"   "water"       "precipitation" "forest_edges"  "temperature"
> chosenVars2
[1] "dist_water"   "water"       "precipitation"
> chosenVars3
[1] "dist_water"   "water"       "forest_edges"  "precipitation" "soil_BD"
```

```
> chosenVars4
[1] "dist_water"    "water"        "precipitation" "soil_BD"
> chosenVars5
[1] "dist_water"    "water"        "precipitation"
> chosenVars6
[1] "dist_water"    "water"        "precipitation" "temperature"
> chosenVars7
[1] "dist_water"    "water"        "precipitation" "soil_BD"       "temperature"
> chosenVars8
[1] "dist_water"    "water"        "precipitation" "temperature"
> chosenVars9
[1] "dist_water"    "water"        "precipitation" "temperature"
> chosenVars10
[1] "dist_water"    "water"        "precipitation" "temperature"   "soil_BD"
[6] "pasture"
>
```

In total the sets contain 7 distinct environmental variables that now can be used to create a species distribution model in the next step.

```
> chosenVars_allRFEs <-unique (c(chosenVars1,chosenVars2,chosenVars3,chosenVars4,chosenVars5,
chosenVars6,chosenVars7,chosenVars8,chosenVars9,chosenVars10))
> chosenVars_allRFEs
[1] "dist_water"    "water"        "precipitation" "forest_edges"  "temperature"
[6] "soil_BD"       "pasture"
```

Step 4 – Species distribution modelling with random forest

The complete code of the modeling process can be found in the R script *"SDM_modelling.R".* The method orients itself on the thorough and recommendable vignette about species distribution modelling by Hijmans & Elith (2017).

1. Preparation

The variable selection requires the packages 'dismo' and 'randomForest'. Do not forget to select your correct working directory!

The presence and the pseudo-absence datasets are imported and duplicate entries are removed. The chosen variable sets from the feature selection (Step 3) are loaded.

```
> # Load the required packages for the variable selection.
```

```
>
> require (dismo)
> require (randomForest)
> options (warn=-1)
>
> # Choose the working directory.
>
> setwd ("YourWorkingDirectory")
>
> # Load the provided species data.
> # Note: The datasets do not include coordinates because the species is
> # endangered and the presence coordinates are thus not for public use.
>
> SQ_presences <- read.csv ("source/SQ_presences.csv", row.names=1)
> SQ_pseudoabs <- read.csv ("source/SQ_pseudoabs.csv", row.names=1)
>
> # Duplicate entries are removed before continuing:
>
> SQ_presences <- SQ_presences[!duplicated (SQ_presences),]
> SQ_pseudoabs <- SQ_pseudoabs[!duplicated (SQ_pseudoabs),]
>
> # Now load the variable sets chosen in the feature selection (Variable_Selection.R):
>
> load ("source/RFEresults_chosenVars.RData")
```

2. Species distribution modelling

We will use the aggregated set of all 10 feature selections for our models.

```
> chosenVars_allRFEs
[1] "dist_water"    "water"         "precipitation" "forest_edges"  "temperature"   "soil_BD"
[7] "pasture"
```

We need two sets of data for modelling: a training data set that is used to compile the model, and a testing dataset that is used for the threshold calculation for prediction.

Thus, we separate the presences into a training set (5/6th of the data) and a testing set (1/6th of the data):

```
> set.seed (42)
> groupPresence <- kfold(SQ_presences,6)
> pres_train <- SQ_presences [groupPresence!=1,chosenVars_allRFEs]
> pres_test <- SQ_presences [groupPresence==1,chosenVars_allRFEs]
```

Similarly, 360 pseudo-absences are chosen and separated so that the training set contains 300 and the testing set contains 60 observations.

```
> set.seed (42)
> abs_sample <- SQ_pseudoabs [sample(nrow(SQ_pseudoabs), 360),]
> groupAbsence <- kfold(nrow(abs_sample),6)
> abs_train <- abs_sample [groupAbsence!=1,chosenVars_allRFEs]
> abs_test <- abs_sample [groupAbsence==1,chosenVars_allRFEs]
```

Subsequently, the training data is combined and the random forest model is compiled:

```
> SQ_jointdata <- rbind (pres_train, abs_train)
> SQ_jointdata$occurrence <- c(rep (1,nrow(pres_train)),rep(0,nrow(abs_train)))
>
> # running the random forest function
>
> RFresult <- randomForest (SQ_jointdata [,1:7], SQ_jointdata [,8])
> RFresult

Call:
 randomForest(x = SQ_jointdata[, 1:7], y = SQ_jointdata[, 8])
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 0.07086081
                    % Var explained: 67.42

…

> sqrt(RFresult$mse[500])
[1] 0.2661969

…

RFresult$rsq[500]
[1] 0.6742061
```

Thus, by the 7 environmental variables obtained during the feature selection, we obtained a random forest model with an RMSE of 0.266 (of range 0 to 1) and a pseudo-R2 of 67.4 %.

3. Characteristics of the random forest model

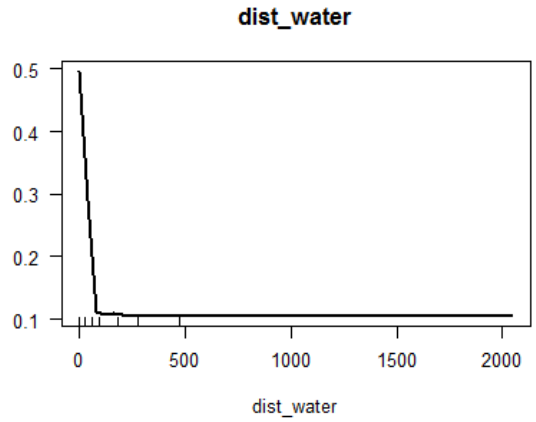
Random forest is a machine-learning method and thus does not give you a linear relationship between environmental variable and occurrence. However, the *importance ()* function gives you an information on the usefulness of all environmental variables in the random forest model in the form of the total decrease in node impurities (measured by the residual sum of squares for regression; a thorough explanation would be too detailed here, but if interested you will find information online).

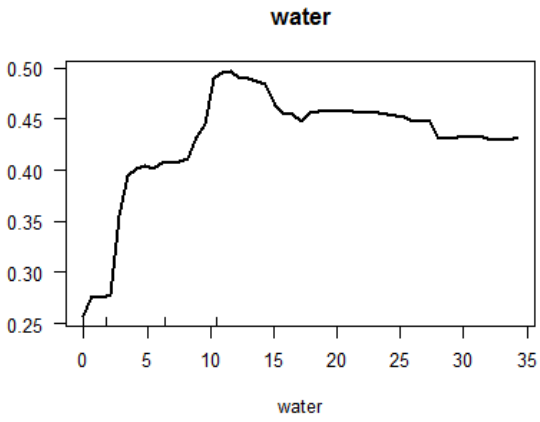
In the current example, we see that the most important environmental variables are all related to waterbody presence and supply. The distance to waterbodies is the most important variable in the present RF model, while the amount of forest edges is the least important one.

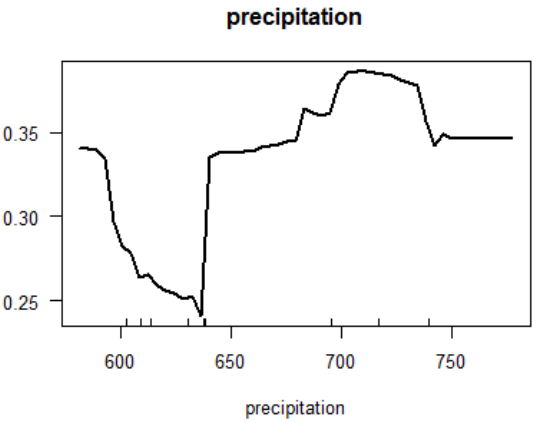
```
> importance(RFresult)
                IncNodePurity
dist_water          30.385066
water               16.756456
precipitation       11.162913
forest_edges         7.054125
temperature          8.538686
soil_BD              7.220469
pasture              8.430886
```

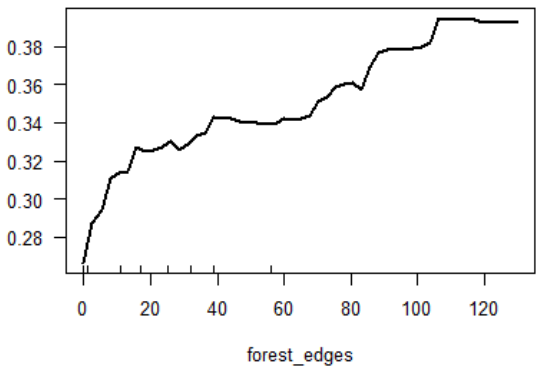
The importance itself does not tell you anything about the relationship between the environmental variable and the occurrence of the species. You can, however, look at the relationships by plotting partial dependence plots:

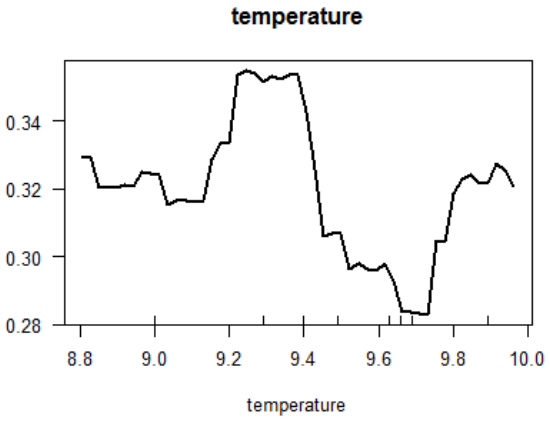
```
> par (mfrow=c(2,2))
> for (i in 1:7) { # i=1
+   VarName <- rownames (RFresult$importance) [i]
+   partialPlot(x=RFresult, pred.data=SQ_jointdata [,1:7], x.var=eval(VarName),
+             main=VarName, xlab=VarName, las=1, lwd=2)
+ }
```

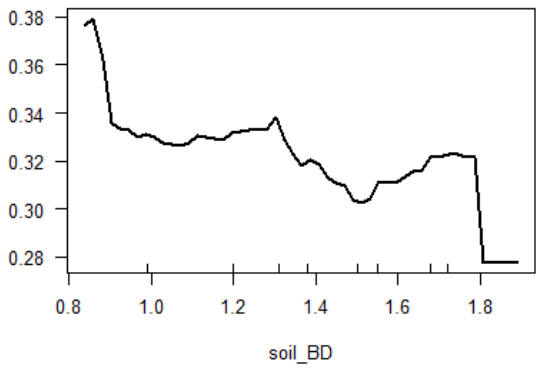

dist_water
0.5
0.4
0.3
0.2
0.1
0
500
1000
1500
2000
dist_water

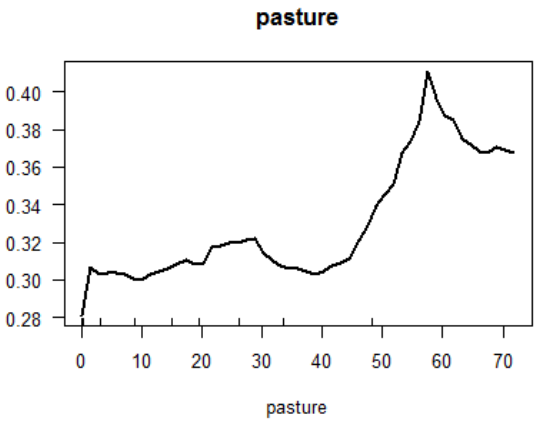

water
0.50
0.45
0.40
0.35
0.30
0.25
0
5
10
15
20
25
30
35
water


precipitation
0.35
0.30
0.25
600
650
700
750
precipitation


forest_edges
0.38
0.36
0.34
0.32
0.30
0.28
0
20
40
60
80
100
120
forest_edges


temperature
0.34
0.32
0.30
0.28
8.8
9.0
9.2
9.4
9.6
9.8
10.0
temperature


soil_BD
0.38
0.36
0.34
0.32
0.30
0.28
0.8
1.0
1.2
1.4
1.6
1.8
soil_BD


pasture
0.40
0.38
0.36
0.34
0.32
0.30
0.28
0
10
20
30
40
50
60
70
pasture

Due to random forest being a machine learning method, the relationships are not smooth but reflect a decision result. However, the partial dependence plots show a clear trend for all environmental variables. Here are some examples:

- The probability of the species being present is highest directly next to waterbodies and lowers fast with increasing distance.

- For the species, it is optimal if about 12 % of the surrounding area is covered by water

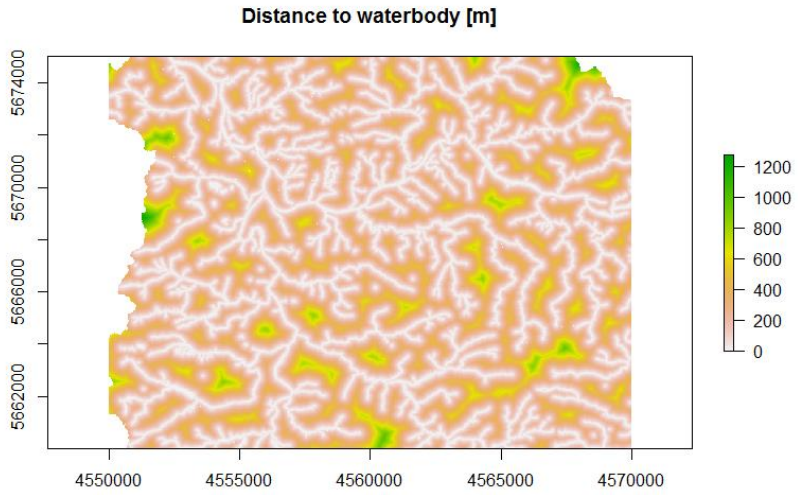
- the more forest edges and the more pasture, the better

For each species, information of the usual (breeding) habitat can also be found in numerous publications that should be used to cross-check if the present model works realistically.

4. Prediction with the random forest model

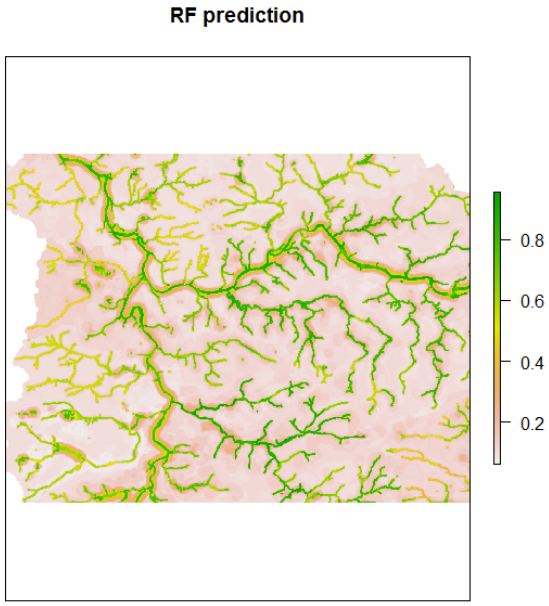
Example environmental data is provided as a raster stack in the folder *"source"*.

```
> load ("source/SQ_VariableMaps.RData")
> # let's plot the distance to waterbodies as an example:
> par (mfrow=c(1,1))
> plot (SQ_variables[["dist_water"]], main="Distance to waterbody [m]")
```



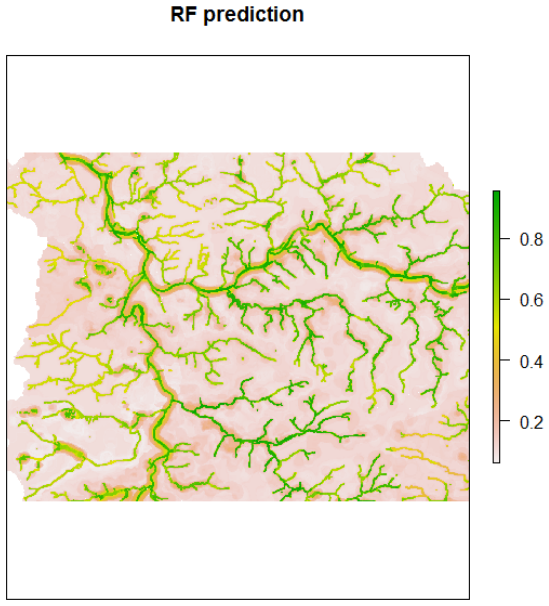
The rasterstack contains maps of all environmental variables in an example area. We predict the occurrence of our species and plot the result:

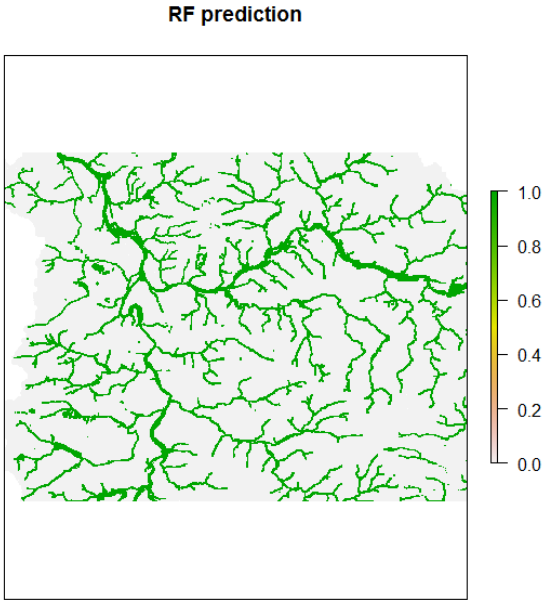
```
> RFprediction <- predict (SQ_variables,RFresult)
> par (mfrow=c(1,2))
> plot (RFprediction, axes=FALSE, main="RF prediction")
```



We see that the prediction yields values between 0 and 1, with high values centered on the river system and low values in all other areas. To determine whether occurrence is likely or not, we have to calculate a threshold based on the testing data we held back before modelling. This threshold is then used to decide whether occurrence is likely or not.

```
> RFevaluation <- evaluate (pres_test,a=abs_test,RFresult)
> RFtreshold <- threshold(RFevaluation,'spec_sens')
> RFpred_th <- RFprediction
> RFpred_th [RFpred_th > RFtreshold] <- 1; RFpred_th [RFpred_th < 1] <- 0
> plot (RFpred_th, axes=FALSE, main="RF prediction")
```





5. Concluding remarks

Above we used a dataset of 141 presence observations and 300 absences to create a random forest model for species distribution modelling. However, similar to the feature selection, it is recommended to repeat modelling at least 10 times with different pseudo-absence datasets to get reliable results (Barbet-Massin et al., 2012).