mopa (MOdelling with Pseudo Absences): A framework for species distribution modelling with improved pseudo-absence generation

by M. Iturbide et al.

02 June 2015

Contents

1 Introduction **1**

1.1 Installing the mopa package 1

2 Modelling steps **2**

2.1 Data pre-processing 2

2.2 Three–step pseudo–absences generation 4

2.2.1 STEP1: Environmental Profiling 4

2.2.2 STEP2: SDM performing with pseudo–absences generated into different extents of the unsuitable background. 6

2.2.2.1 At random (TS method) 6

2.2.2.2 With k-means clustering (TSKM method) 7

2.2.3 STEP3: selection of the optimum background extent and corresponding fitted model . 8

3 References **12**

1 Introduction

This is a tutorial containing a full worked example of species distribution modelling using the RSEP method and the three-step methods (TS and TSKM methods) for pseudo-absence generation presented in Iturbide et al., 2015, combining environmental profiling and spatial extent restriction of the background (see Fig. 2 in the cited article). We illustrate the steps followed to produce some of the analyses presented in the article using the R package `mopa` (MOdelling Pseudo Absences).

NOTE: Most functions presented are implemented in `mopa`. In the few cases in which functions from other packages are used, the package name is always explicitly indicated.

1.1 Installing the mopa package

The `mopa` package is available on GitHub. The stable release fo the package is on the 'master' branch.

We recomend the `devtools` package to download and install `mopa` directly from the stable branch of the repository:

```
if (!require(devtools)) install.packages("devtools")
devtools::install_git("https://github.com/SantanderMetGroup/mopa.git")
```

Once installed, we load the package:

```
library(mopa)
```

```
## Loading required package: raster
## Loading required package: sp
```

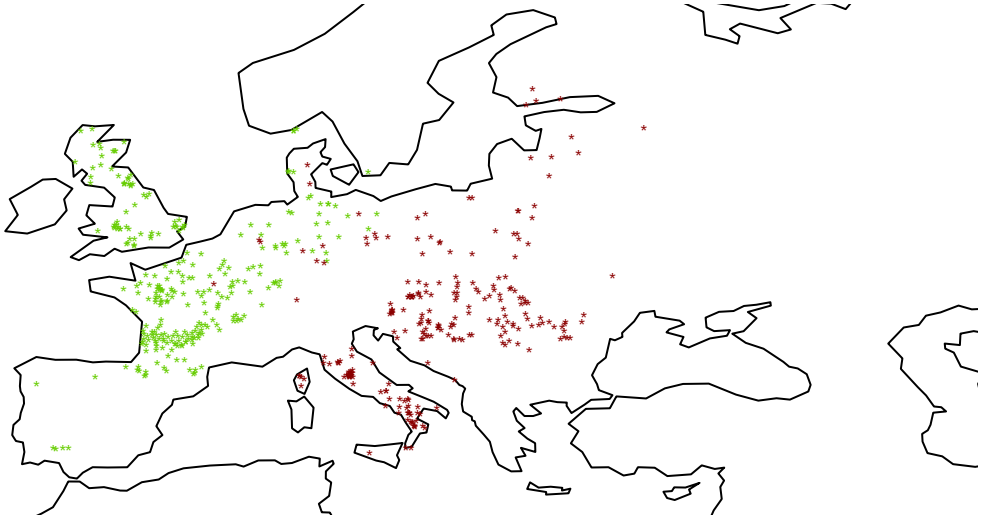
2 Modelling steps

2.1 Data pre-processing

In this article we indicate the adequacy of using different ecotypes for modelling species distributions. Thus, functions in the `mopa` package are intended to deal with more than one group of presences simultaneously. In this example, we use a `list` of 2 different Oak haplotypes (H11 and H5, see Table 1 in the manuscript), but the same steps may be followed in a joint analysis of multiple species. Of course, the most typical case (i.e., dealing with with a single group of species) can be also easily accomplished by providing a `data.frame`.

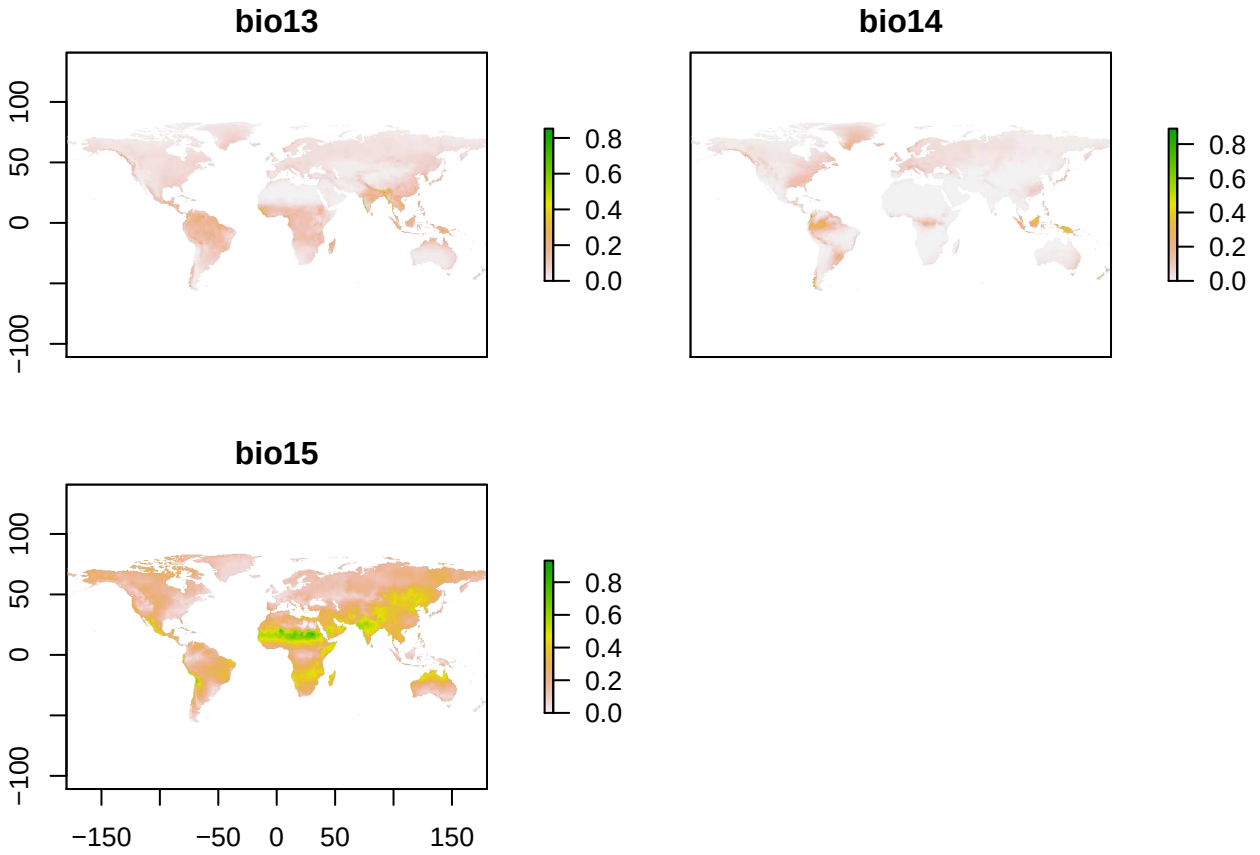
The data set of species in this example is `Oak_phylo2` and is provided with the `mopa` package. This is a modified subset of the *Quercus sp Europe Petit 2002* database (Petit et al., 2002b), which is available in the *Georeferenced Database of Genetic Diversity* or $(GD)^2$. To aid in map representation, a dataset called `wrld` containing a World map is also included in the package.

```
# Load map and Oak data
data(wrld)
data(Oak_phylo2)
# Map
plot(wrld, asp = 1, xlim= c(-10,50), ylim=c(40,60))
for (i in 1:length(Oak_phylo2)) {
  points(Oak_phylo2[[i]], pch = "*", cex = 0.5, col = colors()[i*50])
}
```



Predictor variables are typically stored in raster files. The different raster layers can be efficiently handled in R using the utilities of the `raster` package. `mopa` uses as input this type of raster objects. In particular, multiple layers can be arranged in a collection of `RasterLayers` objects called a `RasterStack` (see `?raster::raster` for more information on `raster` objects).

```
# RatserStack of environmental variables
data(biostack)
plot(biostack)
```



The regularly distributed grid points covering the continental area can be created with functions from the `raster` and `sp` packages as follows:

```
# Extract raster values at grid coordinates
ac <- xyFromCell(biostack[[1]], 1:ncell(biostack[[1]]))
ex <- extract(biostack[[1]], ac)
# Convert to a Spatial object and define projection
sp_grid <- SpatialPoints(ac[-which(is.na(ex)), ])
projection(sp_grid) <- CRS("+proj=longlat +init=epsg:4326")
```

For ease of use, the previous steps can be skipped, as the `sp_grid` object is already included in the `mopa` package, covering the whole world at 10 km resolution.

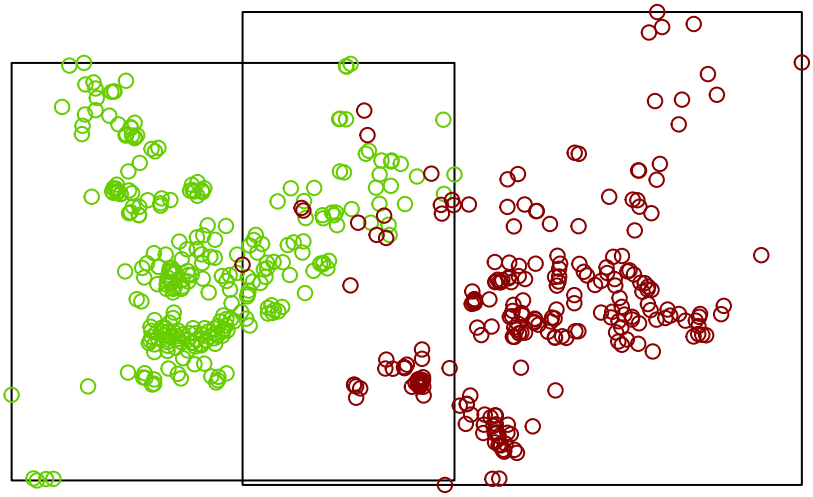
```
data(sp_grid)
```

Function `boundingCoords` creates a matrix of bounding coordinates around the known presence localities of the target species. In this case, since the `Oak_phylo2` list contains two groups of points (one for each haplotype considered), a list of two matrices is created.

```
oak.extension <- boundingCoords(xy = Oak_phylo2)
```

Function `delimit` creates a rectangular polygon shape from the bounding coordinates and does the intersection of the background points. A `list` with two objects is obtained: (1) `bbs`: polygon shape of the bounding boxes and (2) `bbs.grid`: a list of data frames of the background point grid limited by the bounding coordinates.

```
box.grid <- delimit(bounding.coords = oak.extension, grid = sp_grid,
                    names = names(Oak_phylo2))
# Plot presences and bounding boxes
plot(box.grid$bbs, asp = 1)
for (i in 1:length(Oak_phylo2)){
  points(Oak_phylo2[[i]], col = colors()[i*50])
}
```



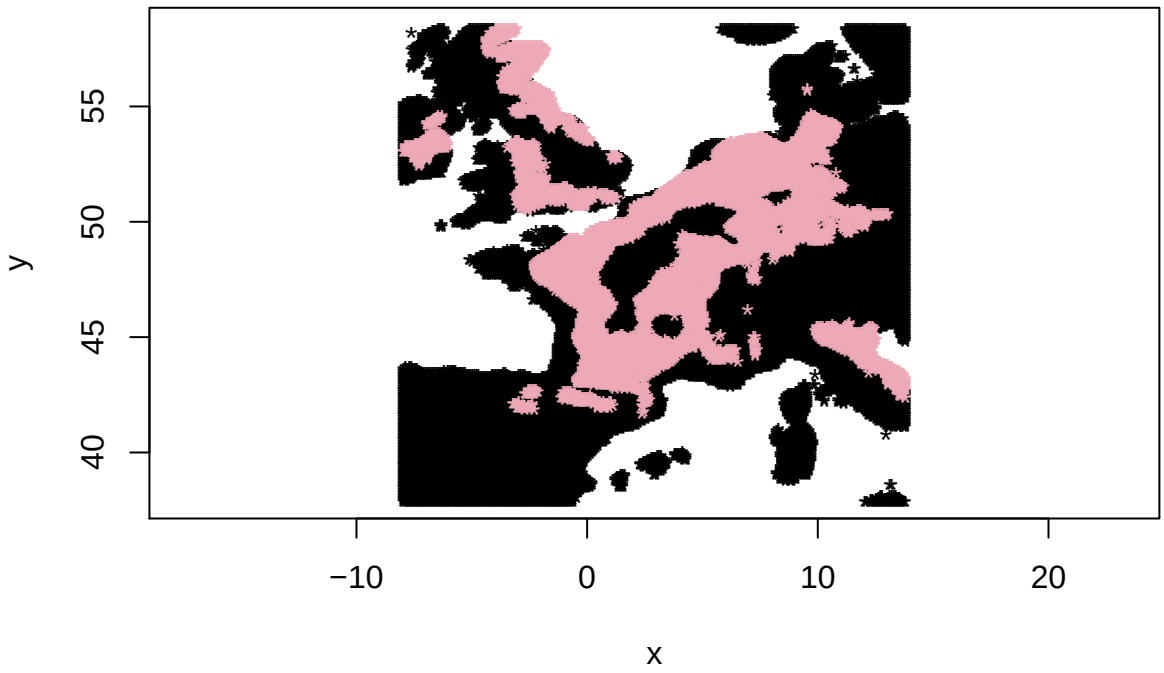
2.2 Three–step pseudo–absences generation

In this section we illustrate the steps followed to generate pseudo-absences following the RSEP, TS and TSKM procedures described in the manuscript.

2.2.1 STEP1: Environmental Profiling

The first step is the selection of the environmentally unsuitable areas using a presence-only algorithm. In `mopa` this is done using a support vector machine-based algorithm that performs a preliminary binary classification of the study region (suitable/unsuitable) using as input the environmental conditions of the presence localities. This is done by function `OCSVMprofiling` which runs the one-class support vector machine algorithm (OCSVM) for each Oak group of the example:

```
unsuitable.bg <-OCSVMprofiling(xy = Oak_phylo2, varstack = biostack,
                               bbs.grid = box.grid$bbs.grid)
# Plot areas predicted as suitable (presence) and unsuitable (absence) for group H11
plot(unsuitable.bg$absence$H11, pch = "*", asp = 1)
points(unsuitable.bg$presence$H11, pch = "*", col = "pink2")
```



If the RSEP method is selected for pseudo-absence data generation, only the first step is needed, thus, at this point we can create random pseudo-absences in the unsuitable background and perform SDM. The same functions as in the TS and TSKM methods might be used for this purpose, these are `PseudoAbsences` and `allModeling`, which are detailed in the section below (STEP2).

```
# Pseudo--absences are generated at random, in equal number
# to presences (prevalence) and keeping a 10 km distance to
# presences (exclusion buffer)
rsep_random <-PseudoAbsences(xy = Oak_phylo2,
                             bg.grids = unsuitable.bg$absence,
                             exclusion.buffer = 0.083,
                             prevalence = 0.5, kmeans = FALSE)

# Bind presences and absences before modelling
presausRSEP <- list()
for (i in 1:length(Oak_phylo2)){
  xyp <- cbind(Oak_phylo2[[i]], rep(1, nrow(Oak_phylo2[[i]])))
  xya <- cbind(rsep_random[[i]], rep(0, nrow(rsep_random[[i]])))
  colnames(xyp) <-  c("x", "y", "p")
  colnames(xya) <-  colnames(xyp)

  presausRSEP[[i]]<- rbind(xyp, xya)
}
names(presausRSEP) <- names(Oak_phylo2)

# Modelling
modirsRSEP <- allModeling(data = presausRSEP, varstack = biostack, k = 10,
                      algorithm = "mars", destdir = getwd(),
                      projection = CRS("+proj=longlat +init=epsg:4326"))
```

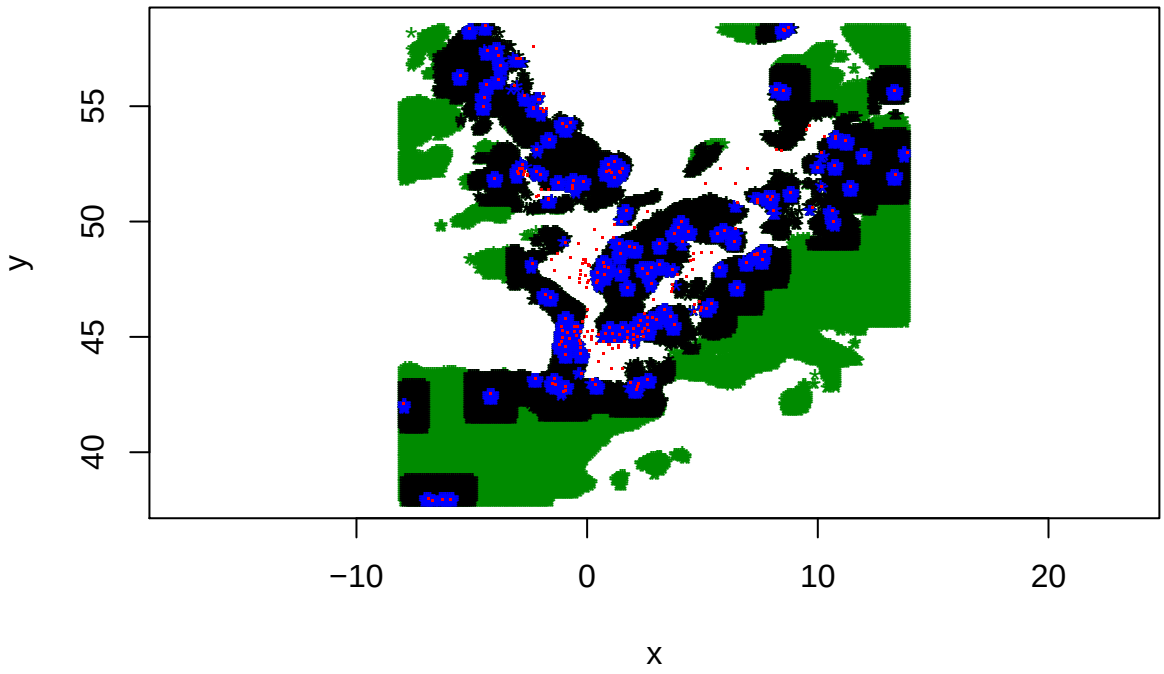
2.2.2 STEP2: SDM performing with pseudo–absences generated into different extents of the unsuitable background.

In the second step, SDMs are performed with pseudo–absences generated into different extents of the unsuitable background. Several functions are involved in this step. Function `bgRadio` performs the partition of the background space considering multiple distance thresholds. In other words, it creates backgrounds of different spatial extent for each species/population. In the example below, extents are created for a sequence of 10 km between distances, from 20 km to half the length of the diagonal of the bounding box, as described in Sec. 2.4 of the manuscript. A list of matrices containing xy coordinates is returned, each matrix corresponding to a different background extent tested.

```
ext <- bgRadio(xy = Oak_phylo2, bounding.coords = oak.extension,
               bg.absence = unsuitable.bg$absence, start = 0.166,
               by = 0.083, unit = "decimal degrees")
```

```
## [1] "creating background point-grids for species 1 out of 2"
## [1] "creating background point-grids for species 2 out of 2"
```

```
# Plot presences for group H11 and background extents of 20, 120 and 520 km
plot(ext$H11$km520, col = "green4", pch = "*", asp = 1)
points(ext$H11$km120, pch = "*")
points(ext$H11$km20, pch = "*", col = "blue")
points(Oak_phylo2$H11, col = "red", pch = ".", cex = 1.5)
```



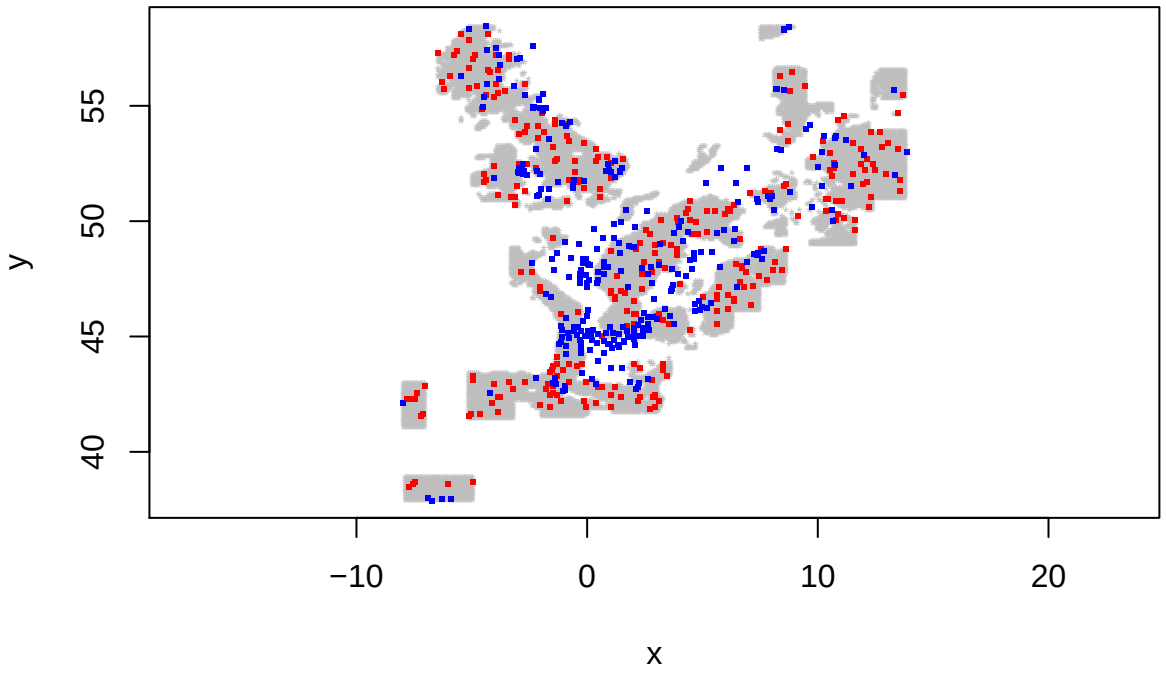
Function `PseudoAbsences` creates pseudo–absences either at random (TS method) or using the k-means clustering approach (TSKM method) for all the background extents considered. Prevalence (proportion of presences against pseudo–absences) and the exclusion buffer (minimum distance to be kept to presences without pseudo-absences) can also be set in this function using the arguments `prevalence` and `exclusion.buffer`.

2.2.2.1 At random (TS method) In the example below, pseudo–absences are generated at random, in equal number to presences (prevalence) and keeping a 10 km distance to presences (exclusion buffer).

```
pa_random <-PseudoAbsences(xy = Oak_phylo2, bg.grids = ext, exclusion.buffer = 0.083,
                           prevalence = 0.5, kmeans = FALSE)
```

```
## [1] "generating pseudo-absences for species 1 out of 2"
## [1] "generating pseudo-absences for species 2 out of 2"
```

```
# Plot presences/pseudo-absences for group H11 considering the background extent of 120 km
plot(ext$H11$km120, pch="*", col= "grey", cex=.5, asp=1)
points(pa_random$H11$km120, col= "red", pch=".", cex=3)
points(Oak_phylo2$H11, col= "blue", pch=".", cex=3)
```

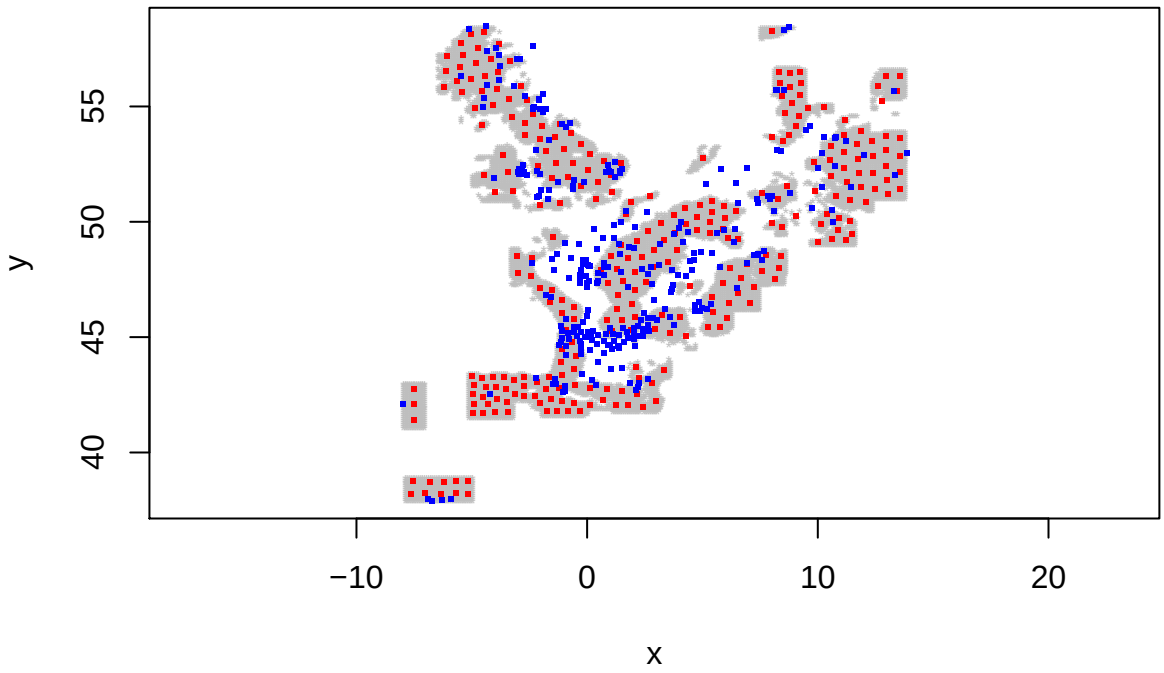


2.2.2.2 With k-means clustering (TSKM method) In the example below, pseudo-absences are generated with k-means clustering, in equal number to presences (prevalence) and keeping a 10 km distance to presences (exclusion buffer).

```
pa_kmeans <-PseudoAbsences(xy = Oak_phylo2, bg.grids = ext, exclusion.buffer = 0.083,
                           prevalence = 0.5, kmeans = TRUE, varstack = biostack)
```

```
## [1] "generating pseudo-absences for species 1 out of 2"
## [1] "generating pseudo-absences for species 2 out of 2"
```

```
# Plot presences/pseudo-absences for group H11 considering the background extent of 120 km
plot(ext$H11$km120, pch = "*", col = "grey", cex = .5, asp = 1)
points(pa_kmeans$H11$km120, col = "red", pch = ".", cex = 3)
points(Oak_phylo2$H11, col = "blue", pch = ".", cex = 3)
```



Function `bindPresAbs` binds presence and pseudo-absence data for each background extension.

```
presausTS <- bindPresAbs(presences = Oak_phylo2, absences = pa_random)
```

The `allModeling` function performs the species distribution modelling and the k-fold cross-validation for a set of presence/absence data per species, corresponding to different background extents. Algorithms supported are "glm", "svm", "maxent", "mars", "randomForest", "cart.rpart" and "cart.tree". In the example below, we perform a 10–fold cross validation using the "mars" modelling algorithm.

```
modirsTS <- allModeling(data = presausTS, varstack = biostack, k = 10,
                        algorithm = "mars", destdir = getwd(),
                        projection = CRS("+proj=longlat +init=epsg:4326"))
```

Named *.Rdata* objects are stored in the specified directory in `destdir`. Each file is automatically named, indicating the algorithm, background extent, and species/population in this order (if a single species is provided, no name is given for the species). For instance, the file *mars_bgkm20_hgH11.Rdata* stores the R object (a `list`) containing the SDM results for the MARS algorithm, considering a 20 km background extent and the Oak group H11. In particular, the output list contains the following components:

```
(1)`allmod`: fitted model with all data for training.
(2)`auc`: AUC statistic in the cross validation.
(3)`kappa`: kappa statistic in the cross validation.
(4)`tss`: true skill statistic in the cross validation.
(5)`mod`: fitted model with partitioned data.
(6)`p`: cross-validated model predictions.
```

2.2.3 STEP3: selection of the optimum background extent and corresponding fitted model

In the third step, AUCs obtained and corresponding extents are fitted to a Michaelis-Menten model to extract the *Vm* coefficient (equation 1 in the manuscript). Then, the minimum extent at which the AUC surpasses

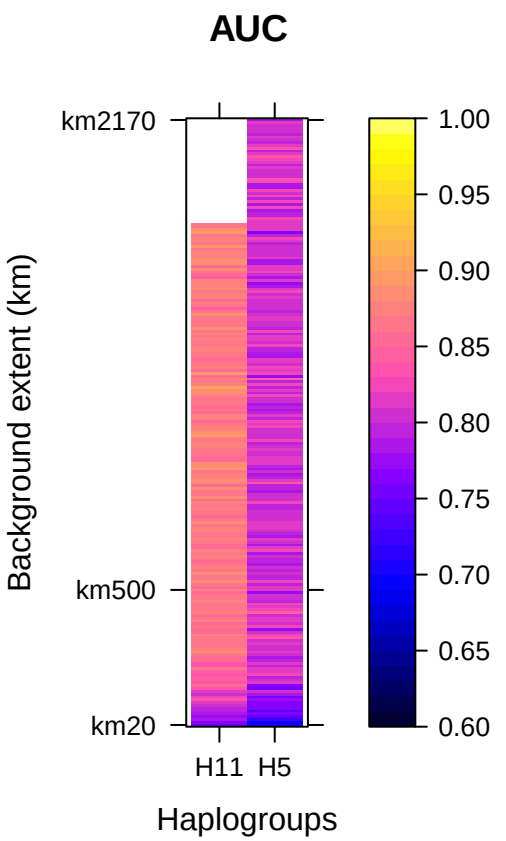
the *Vm* value is selected as the threshold extent (see Figure 3 in the manuscript), being the corresponding fitted SDM the definitive to predict suitability probabilities in the study area.

We next indicate how to plot the results of the optimal spatial extent selection using the `lattice` package. First we load the data generated with the `allModeling` function and extract the corresponding AUC values. Function `loadTestValues` loads and stores AUC data in a matrix:

```
auc_mars <-loadTestValues(data = presausTS, test = "auc", algorithm = "mars")
```

```
## [1] "loading values for species 1"
## [1] "loading values for species 2"
```

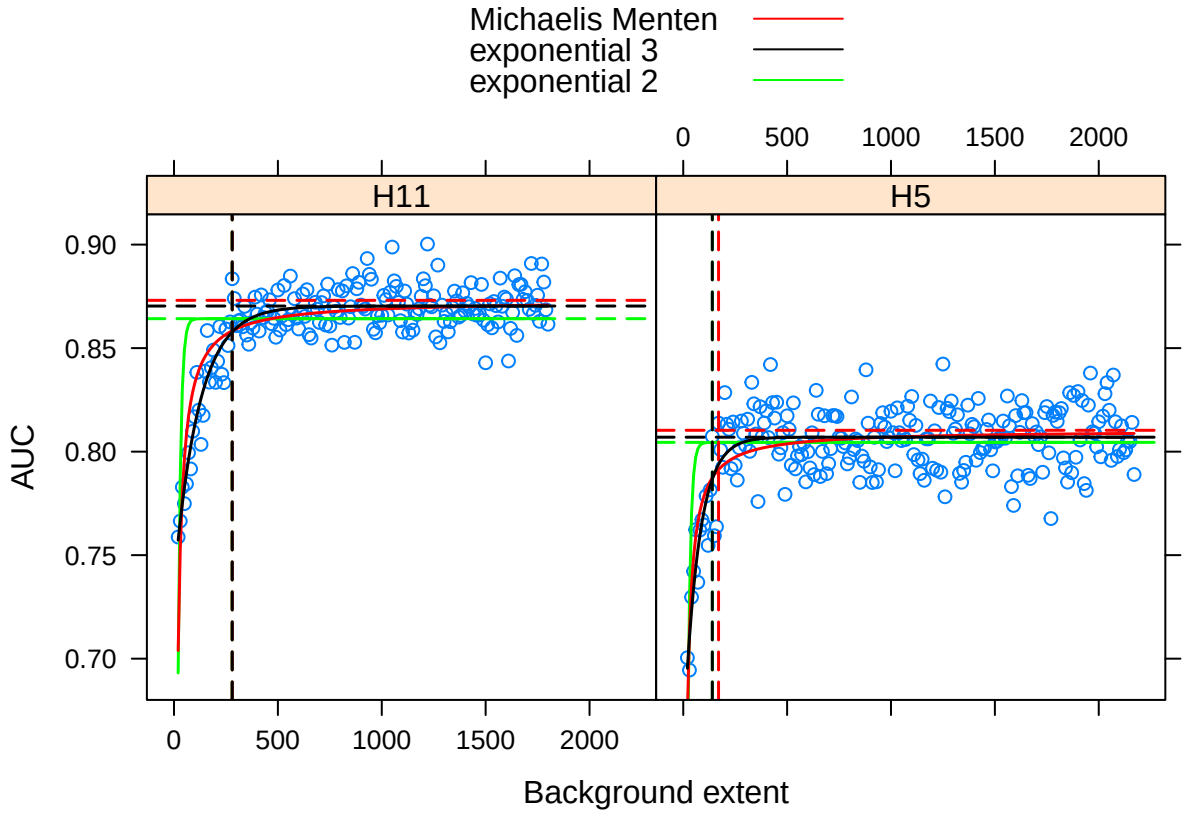
```
library(lattice)
levelplot(auc_mars, aspect = 5,
          scales = list(y = list(cex = 0.8,
              at = c(1, 49, ncol(auc_mars)),
              labels = c(colnames(auc_mars)[1],
                         colnames(auc_mars)[49],
                         colnames(auc_mars)[ncol(auc_mars)]))),
          at = seq(0.6, 1, 0.01), col.regions = bpy.colors,
          xlab = "Haplogroups", ylab = "Background extent (km)", main = "AUC")
```



Model fitting is done by function `indextent`, that internally uses the `nls` function of R package `stats`. An index of the threshold extents is obtained. A fitted model plot (as in Fig. 3 of the manuscript) is also returned if argument `diagrams` is set to `TRUE`.

```
ind <- indextent(testmat = auc_mars, diagrams = TRUE)
```

```
## ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
## residual sum of squares for species  H11
## exponential3 = 0.0179834953861648
## Michaelis Menten = 0.0259861414251489
## exponential2 = 0.0653498378188769
## best function =  exponential3
## ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
## residual sum of squares for species  H5
## exponential3 = 0.0441385733965479
## Michaelis Menten = 0.047446914127977
## exponential2 = 0.0668493212051576
## best function =  exponential3
## ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```




```
ind
```

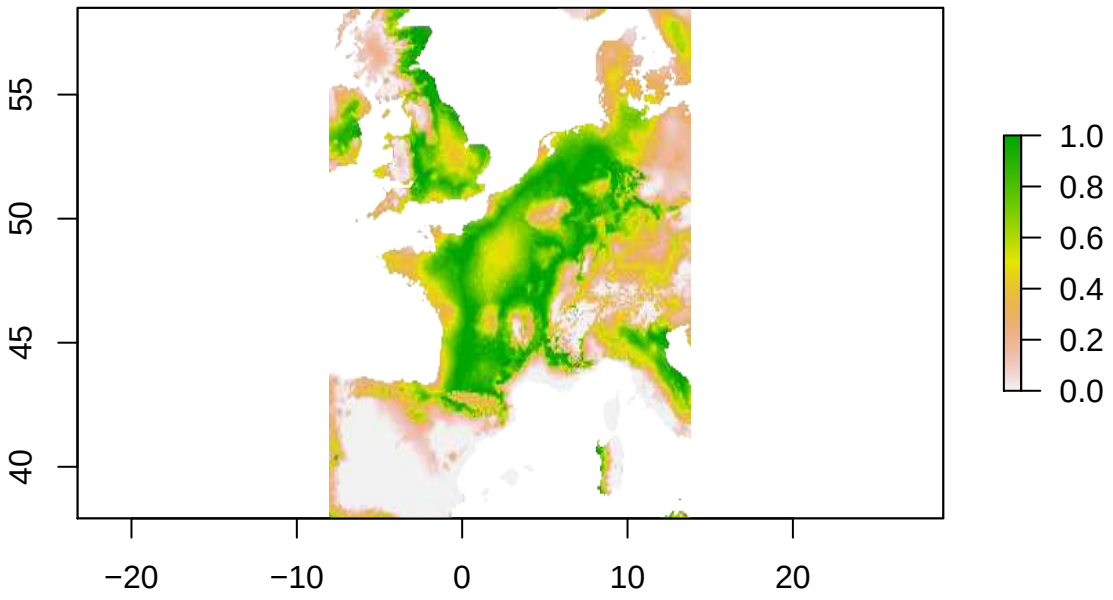
```
## km280 km140
##    27    13
```

The `ind` object in the example above gives the index to extract the best model components and associated data, by means of function `loadDefinitiveModel`.

```
def <-loadDefinitiveModel(data = presausTS, extents = ind, slot = "allmod", algorithm = "mars")
```

Once the optimal SDMs are chosen, we can generate the resulting suitability maps. In the example below we use function `biomat` for preparing a matrix with the variables for prediction in the study area. Then, the predictions are converted to a raster format with functions `SpatialPixelsDataFrame` (from the `sp` package) and `raster` (from the `raster` package).

```
# Suitability map for the Oak group H11
# Function 'biomat' prepares matrix with variables for projection
projectionland <- biomat(cbind(box.grid$bbs.grid$H11,
                               rep(1, nrow(box.grid$bbs.grid$H11))), biostack)
# Prediction
p <- predict(def$H11, projectionland[ ,-1])
p[which(p < 0)] <- 0
p[which(p > 1)] <- 1
# Convert prediction to a raster object
spp <- SpatialPixelsDataFrame(box.grid$bbs.grid$H11, as.data.frame(p))
ras <- raster(spp)
plot(ras)
```



We can combine functions in `mopa` to apply alternative methods of pseudo-absence data generation. Functions performing each step in RSEP, TS and TSKM are indicated in the conceptual diagram of the manuscript (Fig. 2). Functions involved in the TS and TSKM methods are:

`boundingCoords + delimit + OCSVMprofiling + bgRadio + pseudoAbsences + bindPresAbs + allModeling + loadTestValues + indextent + loadDefinitiveModel`,

while the RSEP method only applies the first step of the Three-step methods, being the involved functions:

`boundingCoords + delimit + OCSVMprofiling + pseudoAbsences + allModeling`.

If we want to establish a threshold distance of the background but are not interested in doing an environmental profiling of the background in the previous step, we can combine functions this way:

`boundingCoords + delimit + bgRadio + pseudoAbsences + bindPresAbs + allModeling + loadTestValues + indextent + loadDefinitiveModel`.

```
print(sessionInfo())
```

```
## R version 3.2.0 (2015-04-16)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.2 LTS
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=es_ES.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=es_ES.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=es_ES.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] lattice_0.20-31 mopa_0.1.0      raster_2.3-40   sp_1.1-0
##
## loaded via a namespace (and not attached):
##  [1] formatR_1.2           TeachingDemos_2.9     class_7.3-12
##  [4] tools_3.2.0           rpart_4.1-9           digest_0.6.8
##  [7] goftest_1.0-2         evaluate_0.7          nlme_3.1-120
## [10] PresenceAbsence_1.1.9 mgcv_1.8-6            Matrix_1.2-0
## [13] rgdal_0.9-3           yaml_2.1.13           spam_1.0-1
## [16] dismo_1.0-12          e1071_1.6-4           stringr_1.0.0
## [19] knitr_1.10.5          grid_3.2.0            tree_1.0-35
## [22] sampling_2.6          plotrix_3.5-12        rmarkdown_0.6.1
## [25] plotmo_3.1.0          polyclip_1.3-0        deldir_0.1-9
## [28] magrittr_1.5          tensor_1.5            splancs_2.01-37
## [31] htmltools_0.2.6       MASS_7.3-39           randomForest_4.6-10
## [34] abind_1.4-3           spatstat_1.41-1       lpSolve_5.6.11
## [37] earth_4.3.0           stringi_0.4-1
```

3 References

Iturbide, M., Bedia, J., Herrera, S., del Hierro, O., Pinto, M., Gutiérrez, J.M., 2015. A framework for species distribution modelling with improved pseudo-absence generation. Ecological Modelling DOI: 10.1016/j.ecolmodel.2015.05.018.

Petit, R. J., Csaikl, U. M., Bordács, S., Burg, K., Coart, E., Cottrell,J., van Dam, B., Deans, J. D., Dumolin-Lapégue, S., Fineschi, S., Finkeldey, R., Gillies, A., Glaz, I., Goicoechea, P. G., Jensen,J. S., König, A. O., Lowe, A. J., Madsen, S. F., Mátyás, G., Munro, R. C., Olalde, M., Pemonge, M.-H., Popescu, F., Slade, D., Tabbener, H., Taurchini, D., de Vries, S. G. M., Ziegenhagen, B., Kremer, A., 2002b. Chloroplast DNA variation in european white oaks: Phylogeography and patterns of diversity based on data from over 2600 populations. Forest Ecology and Management 156 (1–3), 5–26.