Modelling Rare Species

Flavien Collart

2023-08-11

a. Modeling rare species

In this section, we will use the ecospat package to employ the method called "ensemble of small models (ESM)", which was developed by (Lomba et al. 2010; Breiner et al. 2015, 2018), which is particularly suitable for rare species. One of the major problems to model rare species is that the number of occurrences is usually scarce. Although some studies reported that species could be accurately modeled with very low sample size (e.g. 3 occurrences in van Proosdij et al. (2015)), sample size is problematic for the modeling procedure, with the risk of overfitting models when the number of occurrences is low compared to the number of predictors. In general, authors are limiting the number of predictors that are put in a model using the rule of thumb that not more than one predictor term should be used per 10 occurrences. To avoid this limitation, ESMs compute bivariate models and then combine all possible bivariate models into an ensemble. By averaging simple small models to an ensemble, ESMs avoid overfitting without losing explanatory power through reducing the number of predictor variables, and were shown to perform significantly better than standard SDMs with species having a low number of occurrences Breiner et al. (2015). For this section, we will be focusing on modeling the ecological niche of *Veronica alpina* in the Western Swiss Alps.

i. Pre-Modeling

The ESM functions of the ecospat package relies on biomod2. We thus need to first format our data by using the function 'BIOMOD_FormatingData', where species occurrences and associated coordinates, the environmental conditions and the name of the species of interest are given. In this example we use presence-absence data for 300 plots, where *Veronica alpina* is present in 12 locations. We also have 5 environmental predictors in our study area. These variables are the growing degree days (with a 0°C threshold), moisture index over the growing season (average values for June to August in mm day-1), the annual sum of radiation (in kJ m-2 year-1), Slope (in degrees), and the topographic position .

```
# Load the packages
library(ecospat)
library(biomod2)
library(terra)
library(viridis)
```

```
set.seed(123)
data("ecospat.testData")

# coordinates of the plots
xy <- ecospat.testData[,2:3]
# species presences and absences
sp_occ <- ecospat.testData$Veronica_alpina
sum(sp_occ) ## Number of occurrences
```

```
## [1] 12
```

```
# environmental data
env <- rast(system.file("extdata/ecospat.testEnv.tif", package="ecospat"))
```

```
# Formatting the data with the BIOMOD_FormatingData() function from the package biomod2
myBiomodData <- biomod2::BIOMOD_FormatingData(resp.var =
                                                as.numeric(sp_occ),
                                              expl.var = env,
                                              resp.xy = xy,
                                              resp.name =
                                                "Veronica.Alpina",
                                              filter.raster = TRUE)
```

ii. Core-Modeling

The function *ecospat.ESM.Modeling* is used to model the ecological niche of the species by generating bivariate models.

The argument *data* is for the formatted dataset object generated by BIOMOD_FormatingData.

The desired algorithms can be provided in the argument *models*. Model parameters can be adapted via the argument *models.options* by giving the object from the function BIOMOD_ModellingOptions() of the biomod2 package. As in the package biomod2, ESM can fit 12 different algorithms: Generalized Linear Model ('GLM'), Gradient Boosted Machine ('GBM'), eXtreme Gradient Boosting Training (XGBOOST), Generalized Additive Models ('GAM'), 'CTA', Artificial Neural Network ('ANN'), 'SRE', 'FDA', 'MARS', 'RF', Maximum entropy ('MAXENT', using the java software or 'MAXNET' from the maxnet package). Tuning to obtain the optimal parameters for the model can be realized with the argument *tune*. *Prevalence* can be set to build a "weighted response". If NULL, each observation (presence or absence) will have the same weight. You can also give a specific weight to observations via the argument *Yweights*.

To evaluate the models, the function performs a repeated split-sampling cross-validation using the arguments *DataSplit* and *NbRunEval*. *DataSplit* corresponds to the percentage of observations used to calibrate the models. *NbRunEval* indicates the number of times the split-sampling procedure is replicated. The function also allows user-defined cross-validations by giving a logical matrix in the argument *DataSplitTable*, where each row corrends to an observation and each column corresponds to a run. A value TRUE means that an observation will be used for model calibration while a FALSE is for model evaluation.

weighting.score corresponds to the evaluation metric that will be used to weight single bivariate models in the final ensemble model. The available evaluation metrics are: 'AUC', 'SomersD' (2xAUC-1), 'Kappa', 'TSS' or 'Boyce'.

which.biva allows to split the bivariate model procedure in several parts. For example, if which.biva is 1:3, only the three first variable combinations will be modeled. This allows to run different bivariate splits on different computers. However, it is better not to use this option if all models are run on a single computer. If you do so, make sure to give each of your modeling subset a unique *modeling.id.* and avoid space characters.

Parallel computing can be enabled with the argument *parallel*

The following step is to combine all the bivariate models into an ensemble. To so, we can use the function *ecospat.ESM.EnsembleModeling* which will need the object returned by *ecospat.ESM.Modeling*, the evaluation metric used to weight the bivariate models (*weighting.score*) and a *threshold* to remove poor performing models. The argument *models* allows to select one or several algorithms to realize the ensemble.

```
my.ESM.EF <- ecospat.ESM.EnsembleModeling(ESM.modeling.output = my.ESM,
                                          weighting.score = "SomersD",
                                          threshold = 0,
                                          models = NULL)
```

ESM performances resulted from the cross-validations can be observed in the object returned by *ecospat.ESM.EnsembleModeling*.

```
t(my.ESM.EF$ESM.evaluations)
```

Table 1: ESM performances based on a mean or standard deviations across bivariate model performances of a same run

model	RUN1_GLM	RUN2_GLM	RUN3_GLM
threshold	175	525	570
sensitivity	1	1	1
specificity	0.671	0.847	0.776
Kappa	0.155	0.332	0.238
AUC	0.771	0.926	0.859
sensitivity.sd	0	0	0
specificity.sd	0.051	0.039	0.045
Kappa.sd	0.071	0.127	0.100
AUC.sd	0.059	0.036	0.056
TSS	0.671	0.847	0.776
SomersD	0.541	0.853	0.718
MPA	0.234	0.616	0.612
Boyce	-0.138	0.474	-0.429
technique	GLM	GLM	GLM
RUN	RUN1	RUN2	RUN3

However, because a minimum sample size is needed to evaluate models (see (Jiménez-Valverde 2020)), it is recommended to evaluate ESMs using the pooling evaluation ((Collart and A. 2023)). The function *ecospat.ESM.EnsembleEvaluation* uses this approach, which consists of pooling the suitability values predicted with the hold-out data (evaluation dataset) across replicates. As the same observation(presence or absence or background point) is presumably sampled in several replicates, the suitability values for each data point are consequently averaged across replicates where they were sampled. This procedure generates a series of independent suitability values with a size approximately equal to that of the number of observations (the number of suitability values might be slightly lower than the number of original observations as some data points may not be sampled by chance in any of the n replicates). This function can compute several metrics, which can be selected with the argument *metrics*. If needed, *EachSmallModels* allows to evaluate each bivariate models via the pooling evaluation

```
my.ESM.EF.eval <- ecospat.ESM.EnsembleEvaluation(
                                ESM.modeling.output = my.ESM,
                                ESM.EnsembleModeling.output =
                                  my.ESM.EF,
                                metrics = c("SomersD","AUC","
                                            MaxTSS","MaxKappa",
                                            "Boyce"),
                                EachSmallModels = FALSE)
```

```
## Evaluation dataset obtained by the pooling evaluation
pred.test <- as.data.frame(my.ESM.EF.eval$ESM.fit)

## Evaluation scores of the ESM based on the pooling evaluation
my.ESM.EF.eval$ESM.evaluations
```

Table 2: ESM performances based on the pooling evaluation

	AUC	SomersD	Boyce	MaxKappa
GLM	0.847	0.694	0.73	0.25

Ecospat package also has numerous functions to compute model performances on your own. By providing model predictions and species observation.

For example, the Boyce index which only requires presences can be calculated with the function *ecospat.boyce*. The argument *obs* should contain the model prediction for the presences while *fit* should contain the predictions of background points. Correlation measurement can be changed via the argument *method*

```
boyce.index <- ecospat.boyce(fit = pred.test$Fit_GLM,
                             obs = pred.test$Fit_GLM[pred.test$resp==1],
                             PEplot = FALSE,
                             method = "spearman")
boyce.index$cor
```

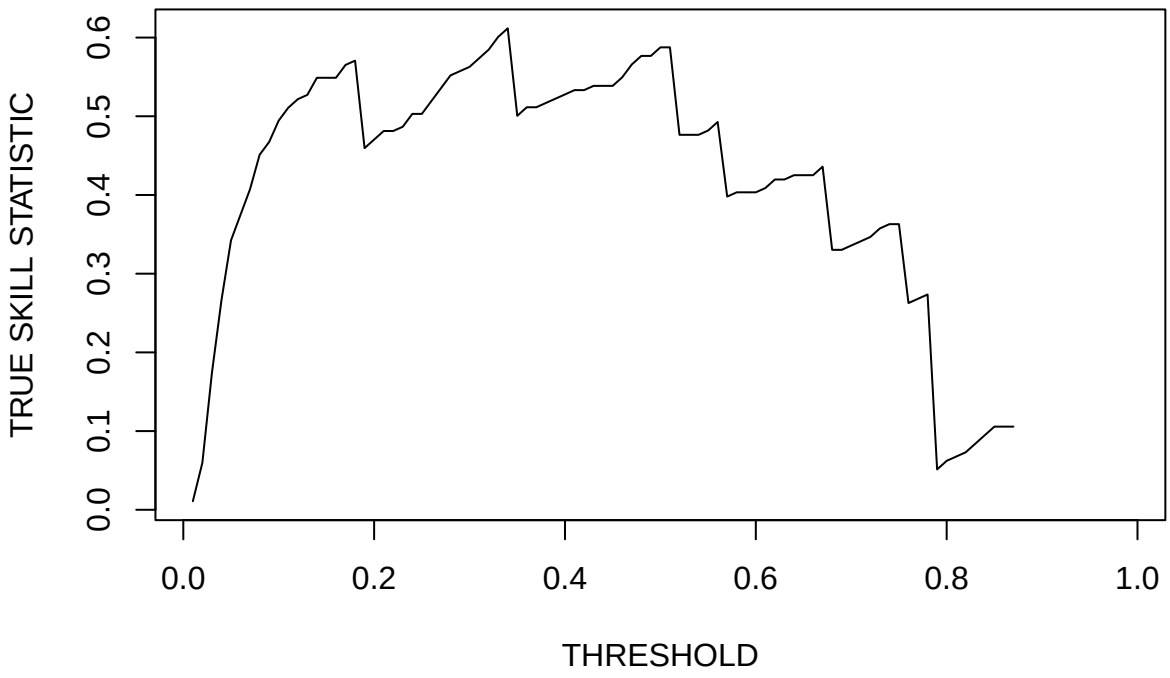
```
## [1] 0.73
```

MaxTSS and MaxKappa can be estimated via the functions *ecospat.max.tss* and *ecospat.max.kappa* and the variations of TSS and Kappa metric on a threshold can be done with *ecospat.plot.tss* and *ecospat.plot.kappa*

```
MaxTSS <- ecospat.max.tss(Pred = pred.test$Fit_GLM,
                          Sp.occ = pred.test$resp)
MaxTSS$max.TSS
```

```
## [1] 0.611715
```

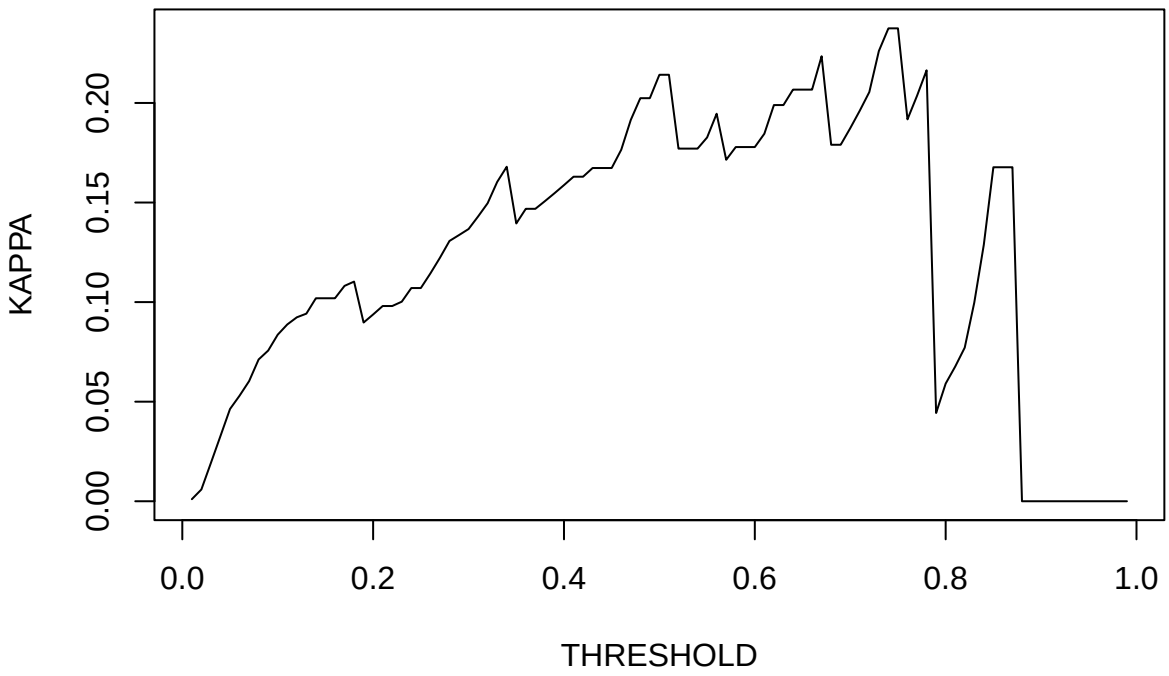
```
ecospat.plot.tss(Pred = pred.test$Fit_GLM,
                 Sp.occ = pred.test$resp)
```




```
MaxKappa <- ecospat.max.kappa(Pred = pred.test$Fit_GLM,
                              Sp.occ = pred.test$resp)
MaxKappa$max.Kappa
```

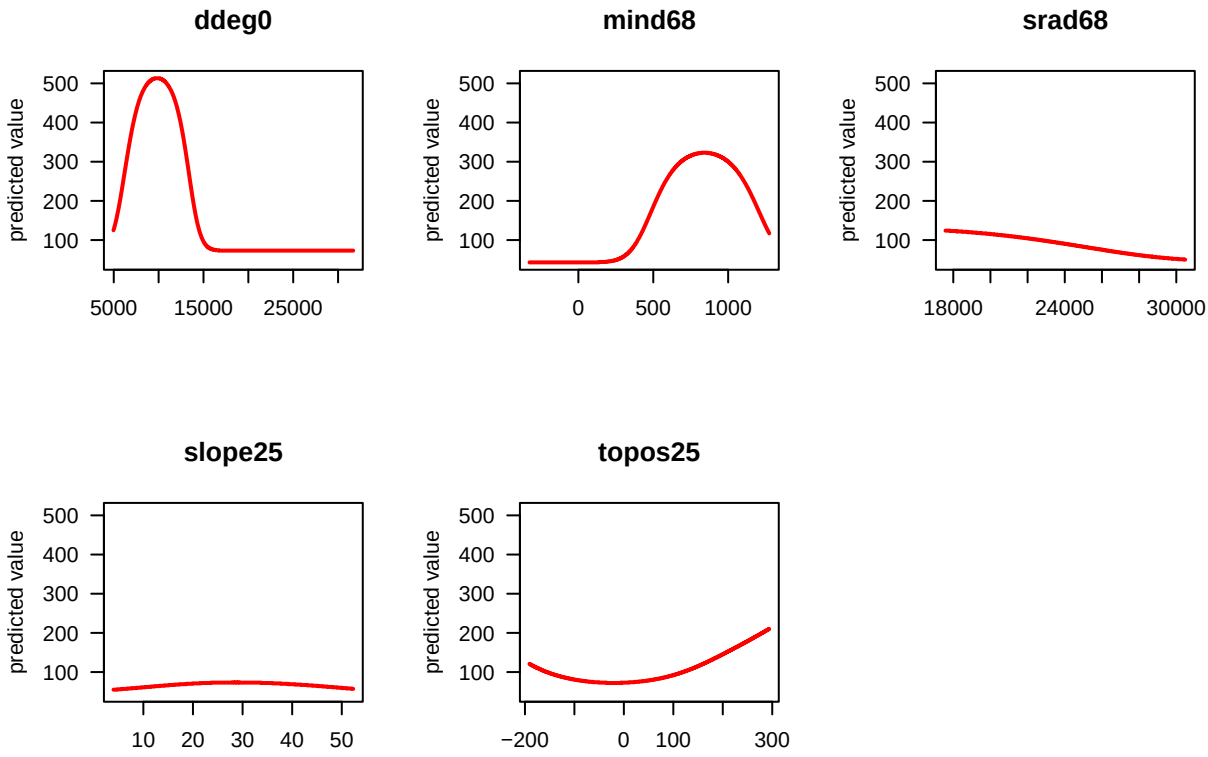
```
## [1] 0.2496419
```

```
ecospat.plot.kappa(Pred = pred.test$Fit_GLM,
                   Sp.occ = pred.test$resp)
```



Model performances can also be checked by observing species response curves to each environmental predictors. To do so, The function *ecospat.ESM.responsePlot* can be used. This function is an adaptation of the Evaluation Strip method proposed by Elith et al. (2005) and needs the objects returned by *ecospat.ESM.Modeling* and *ecospat.ESM.EnsembleModeling*. The statistic used to keep constant the other predictor while generated the response curve for a predictor can be changed via the argument *fixed.var.metric*

```
response.plots <- ecospat.ESM.responsePlot(ESM.EnsembleModeling.output =
                                             my.ESM.EF,
                                           ESM.modeling.output = my.ESM,
                                           fixed.var.metric = 'median')
```



To check the contribution of each variable, you can use the function *ecospat.ESM.VarContrib*. This function computes the ratio between sum of weights of bivariate models where a focal variable was used and sum of weights of bivariate models where the focal variable was not used. The ratio is corrected for the number of models with or without the focal variable. This ratio gives an indication on the proportional contribution of the variable in the final ensemble model. A value of higher than 1 indicates that the focal variable has a higher contribution than average. For the ensemble model, a weighted mean is applied among model algorithms.

```
var.contrib <- ecospat.ESM.VarContrib(ESM.modeling.output = my.ESM,
                                      ESM_EF.output = my.ESM.EF)
var.contrib
```

Table 3: Variable contributions to ESMs

	GLM
ddeg0	1.596
mind68	1.326
srad68	0.873
slope25	0.563
topos25	0.898

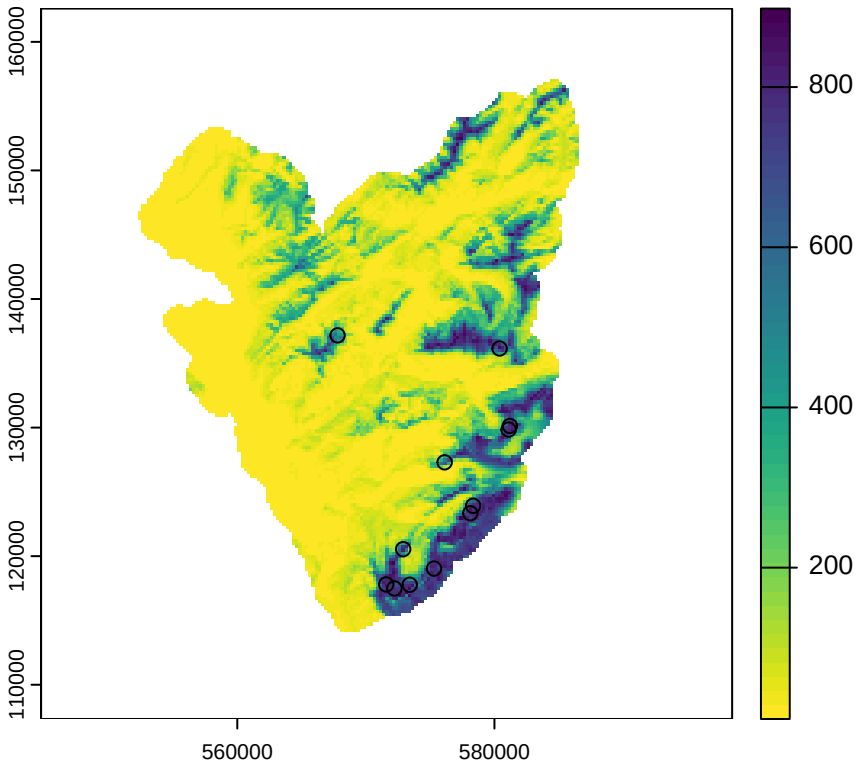
After checking the model performances and the response curves, models can be projected using two functions: *ecospat.ESM.Projection* which projects each bivariate model and *ecospat.ESM.EnsembleProjection* which generates the ensemble of these bivariate models. *new.env* argument allows to project models onto a new

data.frame, *SpatRaster* of ecological values while *name.env* allows to give a name to the projection. Parallel computing can be enabled with the argument *parallel*

```
### Projection of simple bivariate models into new space
my.ESM.proj.current <- ecospat.ESM.Projection(ESM.modeling.output = my.ESM,
                                              new.env = env,
                                              name.env = "currentTime",
                                              parallel = FALSE,
                                              cleanup = FALSE)

### Projection of calibrated ESMs into new space
my.ESM.EF.proj.current <- ecospat.ESM.EnsembleProjection(
                                                  ESM.prediction.output =
                                                    my.ESM.proj.current,
                                                  ESM.EnsembleModeling.output =
                                                    my.ESM.EF,
                                                  chosen.models = "all")
```

```
## projected ESM of Veronica alpina at present time
plot(my.ESM.EF.proj.current,
     col = rev(viridis::viridis(50)))
points(xy[sp_occ==1,],
       col = "black")
```



iii. Post-modeling

ESM projections can be afterwards binarized. To binarize these maps, diverse thresholds can be computed via the function *ecospat.ESM.threshold*. This function also provides evaluation scores for the full model (thus, evaluating the fit of the model but not the transferability).

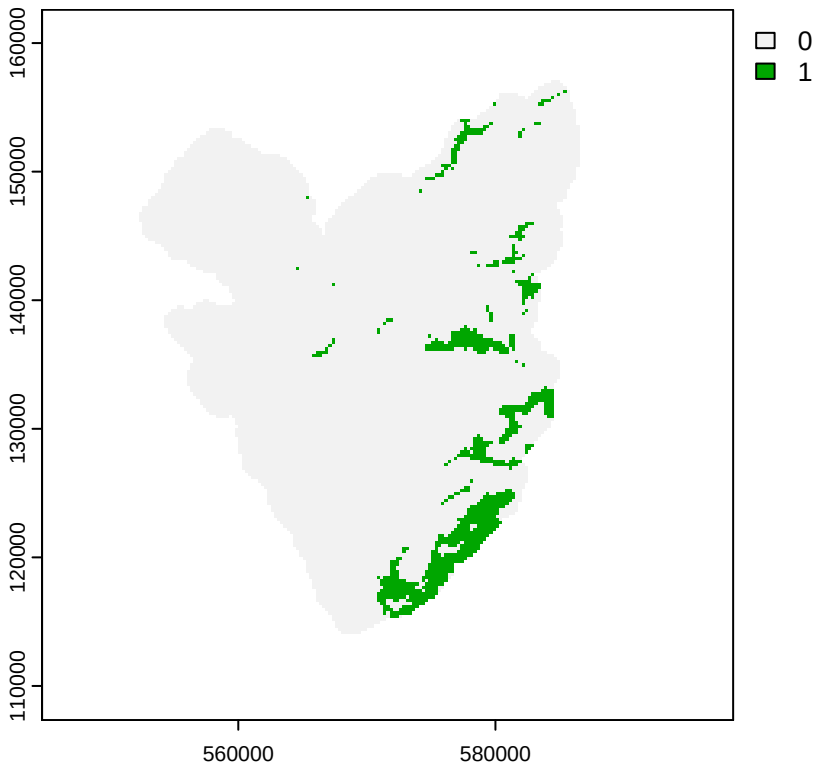
```
Thr <- ecospat.ESM.threshold(ESM.EnsembleModeling.output = my.ESM.EF,
                             PEplot = FALSE)
t(Thr)
```

Table 4: Various threshold and fit performances of ESM

	Full_GLM_ESM
sensitivity	0.917
specificity	0.845
Kappa	0.280
AUC	0.922
sensitivity.sd	0.083
specificity.sd	0.022
Kappa.sd	0.070
AUC.sd	0.023
SomersD	0.843
Boyce	NA
TSS	0.761
TSS.th	0.645
MPA1.0	0.386
MPA0.95	0.737
MPA0.90	0.776
Boyce.th.min	0.340
Boyce.th.max	0.722

Model projections can be afterwards binarized with the function *ecospat.binary.model* which need in the arguments *Pred*, a spatial grid and *Threshold*, the value of the threshold.

```
### Binarization of the projected ESM based on the threshold maximizing the TSS
my.ESM.EF.proj.current.bin <- ecospat.binary.model(Pred = my.ESM.EF.proj.current,
                                                   Threshold = (Thr$TSS.th)*1000)
plot(my.ESM.EF.proj.current.bin)
```

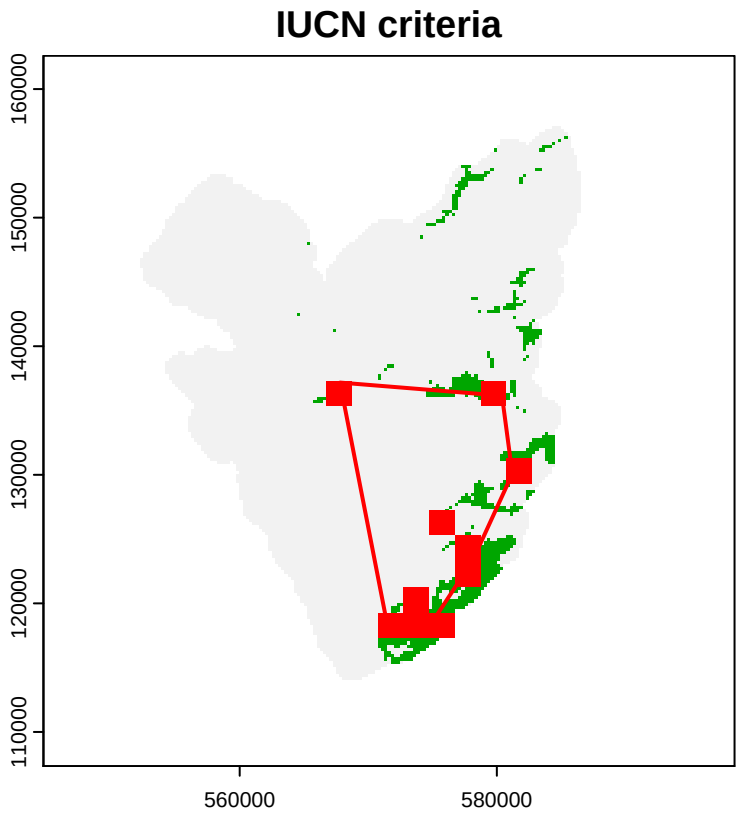


After binarizing maps, one could quantify the species range size or the occupied patches from ESM maps and IUCN criteria. In the ecospat package, the function *ecospat.rangesize* and *ecospat.occupied.patch* are made for these purposes.

More precisely, *ecospat.rangesize* allows quantifying Area of Occupancy AOO and the Extent of Occurrence EOO. Numerous parameters are available and are describe when running in R ?ecospat.rangesize

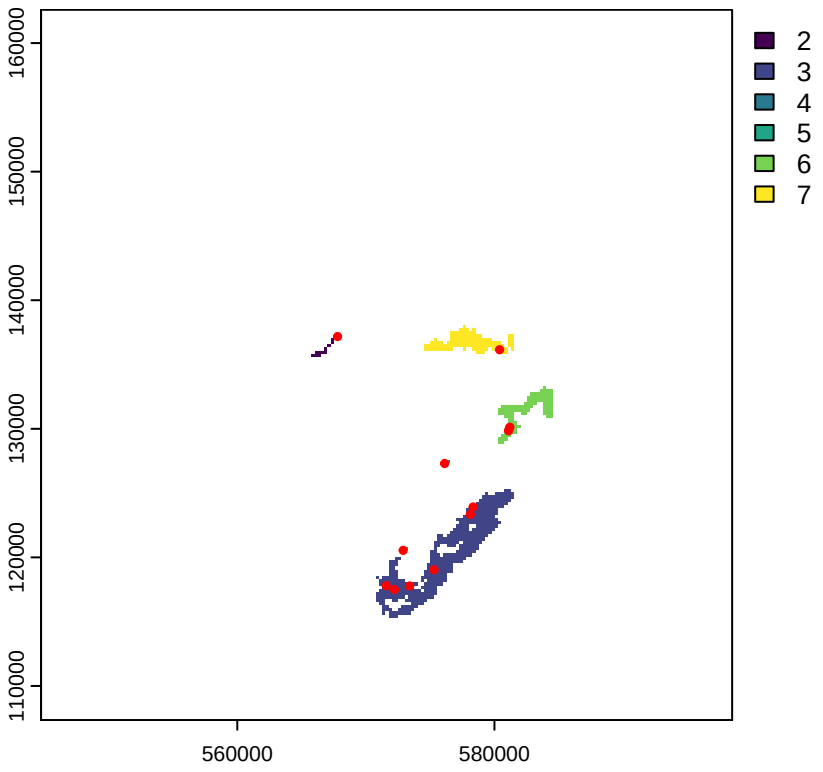
```
rangesize <- ecospat.rangesize(my.ESM.EF.proj.current.bin,
                               xy = xy[sp_occ==1,],
                               AOO.circles = TRUE,
                               lonlat = FALSE)

plot(my.ESM.EF.proj.current.bin,legend = FALSE,
     main = "IUCN criteria")
plot(rangesize$RangeObjects$AOO,
     add = TRUE,
     col = "red",
     legend = FALSE)
plot(rangesize$RangeObjects$EOO@polygons,
     add = TRUE,
     border = "red",
     lwd = 2)
```



ecospat.occupied.patch quantified the number of patches where species occupied based on species distribution predictions, species occurrences and a buffer value (in meter) around species occurrences.

```
ocp  <- ecospat.occupied.patch(my.ESM.EF.proj.current.bin,
                               xy[sp_occ==1,],
                               buffer = 500)
plot(ocp,
     col = viridis::viridis(50)) ## occupied patches: colored areas
points(xy[sp_occ==1,],col = "red",
       cex = 0.5,
       pch = 19)
```



Breiner, F. T., Guisan A., Bergamini A., and Nobis M. P. 2015. "Overcoming Limitations of Modelling Rare Species by Using Ensembles of Small Models." *Methods in Ecology and Evolution* 6 (10): 1210–8.

Breiner, F. T., Nobis M. P., Bergamini A., and Guisan A. 2018. "Optimizing Ensembles of Small Models for Predicting the Distribution of Species with Few Occurrences." Edited by Nick Isaac. *Methods in Ecology and Evolution* 9 (4): 802–8. https://doi.org/10.1111/2041-210x.12957.

Collart, F., and Guisan A. 2023. "Small to Train, Small to Test: Dealing with Low Sample Size in Model Evaluation." *Ecological Informatics* 75 (July): 102106. https://doi.org/10.1016/j.ecoinf.2023.102106.

Elith, J., Ferrier S., Huettmann F., and Leathwick J. 2005. "The Evaluation Strip: New and Robust Method for Plotting Predicted Responses from Species Distribution Models." *Ecological Modelling* 186 (3): 280–89. https://doi.org/10.1016/j.ecolmodel.2004.12.007.

Jiménez-Valverde, A. 2020. "Sample Size for the Evaluation of Presence-Absence Models." *Ecological Indicators* 114 (July): 106289. https://doi.org/10.1016/j.ecolind.2020.106289.

Lomba, A., Pellissier L., Randin C., Vicente J., Moreira F., Honrado J., and Guisan A. 2010. "Overcoming the Rare Species Modelling Paradox: A Novel Hierarchical Framework Applied to an Iberian Endemic Plant." *Biological Conservation* 143 (11): 2647–57. https://doi.org/10.1016/j.biocon.2010.07.007.

van Proosdij, A. S. J., Sosef M. S. M., Wieringa J. J., and Raes N. 2015. "Minimum Required Number of Specimen Records to Develop Accurate Species Distribution Models." *Ecography* 39 (6): 542–52. https://doi.org/10.1111/ecog.01509.