ModOperating Manual for BIOMOD

Wilfried Thuiller
Bruno Lafourcade, Miguel Araujo

May 7, 2009

Contents

0.1 Introduction

BIOMOD is an acronym for BIOdiversity MODelling. BIOMOD has been originally developed at the Centre d'Ecologie Fonctionnelle et Evolutive of the CNRS in Montpellier (France) and was partly funded by the FP5 ATEAM European Project. The package was developed for species distribution modelling but it can be used for modelling any kinds of distributions. The only restriction is that the dependent variable should be coded in a presence-absence binary format.
BIOMOD is a platform for ensemble forecasting of species distributions, enabling the explicit treatment of model uncertainties and the examination of species-environment relationships. It includes the ability to model species distributions with several techniques, test models with a wide range of approaches, project species distributions into the future using different climate scenarios and dispersal functions, assess species temporal turnover, plot species response curves, and test the strength of species interactions with predictor variables. Computationally, BIOMOD is a collection of functions running within the R (CRAN) software (programmed in R language) and allows the user to apply a range of statistical models to several dependent variables using a set of independent variables.

0.2 Installation

To run BIOMOD, please use the latest version of R. A large number of libraries are also required: rpart, MASS, gbm, gam, nnet, mda, randomForest, Design, Hmisc, reshape, plyr) before attempting to run BIOMOD.

Since march 2009, the BIOMOD functions are stored in a different format as it used to be. It is now an R package that is to be downloaded from this web page :

http://r-forge.r-project.org/R/?group$_id = 302$

It contains all the functions BIOMOD needs to work and the datasets necessary to run the examples. All the functions scripts are available by simply typing their names in the R console. A new user does not need to get into them, while more experienced users can eventually rewrite them and modify some internal parameters if they want to, but this is at their own risks as many functions have direct dependencies between them.

Once unzipped, you should put it in R's library directory. This is the example of a general root to get to that directory : "C:Files-2.8.0". It will obviously depend on where R is installed on your computer and on the R version you are using.

An extra file named "BIOMOD-R User Functions" aims to help the user to run BIOMOD in optimal conditions. This script presents pre-formatted calls to prepare the datasets, initialize BIOMOD, and run the different models. This is the script recommended to use all the time. You may for that reason modify it to your good will.

The recommended procedure is to first create a folder called BIOMOD. Then, create a new folder where to store the datasets, run the models and save the outputs and results. This is called the "workspace". In the examples below, the workspace is the directory "Biomod$_runs$".

It is from this folder that the files will be read and written. For example, you need to put a copy

of the User Function file in order to be able to open it once R's working directory is set to your workspace.
In the latter version of BIOMOD, the results will be stored outside R's workspace to counter the memory storage limitations of the software.While running BIOMOD, you will realise that additional folders will be created. First, the *Models()* function willcreated 2 folders named *models* and *pred*. As you might have guessed, they will respectively contain the models and the current predictions. Then, the *Projection()* function will create a folder to store the outputs for each projection scenario that is run (see the "Models' Projection" section).

0.3 Getting Started

0.3.1 Data preparation

In order to facilitate the learning of BIOMOD, a tutorial is provided here with artificial data. It is recommended that the user follows each step and run the models on these artificial datasets. Once the tutorial is completed the user should be able to run BIOMOD with his own data.

The first thing to do is to load the BIOMOD package like any other one. It will load all the functions required to run BIOMOD.

```
> library(BIOMOD)
```

If you obtain the following error message, or anything similar, then the package might not be located at the right place (see section above).

Error in library(BIOMOD) : no package named 'BIOMOD' has been found.

If everything seems right, then R knows the different BIOMOD functions. You can type any function name with a question mark in front to access to the help files. You will find a general explanation of what the function does, an explanation for the use of each parameter to be set and some examples. As a general case, the examples are more detailed and varied in the help files than in this present document.

```
> ?response.plot
```

All the information stored in the memory of the R software can be saved as a work session (or workspace). When beginning a new work session within R, you can load any previously saved work session, which will load all the functions, objects, results obtained in a previous session thus enabling you to continue exactly from where you left it.

```
> save.image()
```

It procuces an object in your working directory with a .RData extension. It is the actual name of the object produced if you run the example above, but you can (or should) add a name to it, for

example save.image("mywork.RData").

Now, BIOMOD is ready to run and the user can import the species and the environmental data. For practical reasons, we can store them in the same file and load the species/environment dataset provided with BIOMOD.

To load your own data from a text file, use the *read.table()* function:

```
> Sp.Env<-read.table("my_data.txt", h=T, sep="\t")
```

Type *?read.table* to get to the help file for more details and other possible extensions.

Here the separator was a tabulation but it could be a comma. Your file might also be in a csv format; in that case you should use the *read.csv* function to correctly load your data.

BIOMOD does not recognise geographical coordinates nor does it order the data according to these. The user should ensure that all datasets are kept in the same order.

NOTE: Missing values are not permitted in BIOMOD and will result in an error.

To load the example files :

```
> data(Sp.Env)
> data(CoorXY)
> Sp.Env[1:5,]
```

```
  Id   Var1 Var2  Var3  Var4  Var5  Var6  Var7 Species1 Species2
1  1 0.6683 4296 770.1 39.33 295.1 16.74 10.87        0        0
2  2 0.7596 4174 928.1 57.32 348.7 16.41 10.51        0        0
3  3 0.7424 4173 870.3 50.05 330.0 16.41 10.50        0        0
4  4 0.5543 4264 620.0 24.99 239.1 16.66 10.93        0        0
5  5 0.5489 4169 622.3 25.16 241.0 16.40 11.28        0        0
  Species3 Species4 Species5 Species6 Species7
1        0        0        1        0        1
2        0        0        0        0        1
3        0        0        0        0        0
4        0        0        1        0        0
5        0        0        1        0        0
```

- Id: An Id to keep track of the row numbers
- Var1 to Var7: Environmental variables (bioclimatic in that case)
- Species1 to Species7: Presence/absence of 7 species.

0.3.2 Initialisation

First, we need to set up the dataset in a correct format for BIOMOD by means of the *Initial.State* function.

Reminder of the syntax - rows are specified before the comma and columns after with a semi-colon separating the start and end column.

The syntax in the *Initial.State* function is the following:
Response: The response variables to model. In our example, the species occurrences are located from the column 9 to 15.

Explanatory: The explanatory or independent variables. In our example, the environmental variables, called Var1 to Var6 are located in the columns 2 to 8.

IndependentResponse: Truly independent response variables. This is used to evaluate the predictive accuracy of the models. If no truly independent data are available, type "NULL".

IndependentExplanatory: Truly independent explanatory variables. This should be used to evaluate the predictive accuracy of the models. If no truly independent data are available, type "NULL".

The calibration procedure

Ideally, one should always evaluate the predictive performance of a species distribution model using independent data. If this kind of data is available (informed in the *IndependentResponse* and *IndependentExplanatory* arguments), BIOMOD will calibrate the models on the calibration data and evaluate them using the independent data.

If no independent data for model evaluation exists, two alternatives are available to assess the predictive performance of the modles (see below the *Models()* function explanations of the *NbRunEval* argument)

In our example, we do not have truly independent data but I will give fake independant data (our original dataset in fact) for the purpose of examples showing. Just ignore it for the moment.

```
> Initial.State(Response = Sp.Env[,12:14], Explanatory = Sp.Env[,2:8],
 IndependentResponse = Sp.Env[,12:14], IndependentExplanatory = Sp.Env[,2:8])
> ls()
```

```
 [1] "BestModelByRoc"
 [2] "BestModelByTSS"
 [3] "Biomod.material"
 [4] "Biomod.PA.data"
 [5] "Biomod.PA.sample"
 [6] "consensus_Current_results"
 [7] "CoorXY"
 [8] "data.used"
 [9] "DataBIOMOD"
[10] "DataEvalBIOMOD"
[11] "Evaluation.results.Kappa"
[12] "Evaluation.results.Roc"
[13] "Evaluation.results.TSS"
[14] "Future1"
[15] "Models.information"
[16] "Pred_Species4"
[17] "Pred_Species5"
[18] "Pred_Species6_indpdt"
```

```
[19] "PredBestModelByKappa_Species4"
[20] "PredBestModelByRoc_Species4_Bin"
[21] "predind"
[22] "Proj_Future1_BestModelByTSS"
[23] "Proj_Future1_BestModelByTSS_Bin"
[24] "Proj_Future1_Species4"
[25] "Proj_Future1_Species4_BinRoc"
[26] "Sp.Env"
[27] "Species4_ANN_PA1"
[28] "Species4_CTA_PA1"
[29] "Species4_GLM_PA1"
[30] "Species4_MARS_PA1"
[31] "Species4_MDA_PA1"
[32] "Species4_RF_PA1"
[33] "Species5_GBM_PA1"
[34] "SpNoName.circles.2"
[35] "Total_consensus_Current"
[36] "VarImportance"
```

It creates one or several databases: DataBIOMOD and DataEvalBIOMOD, if you have independent data. The latter will be used during the testing of the models. Make sure to always keep these datasets unchanged and never delete them.

```
> DataBIOMOD[1:5,]
```

```
    Var1 Var2  Var3  Var4  Var5  Var6  Var7 Species4 Species5
1 0.6683 4296 770.1 39.33 295.1 16.74 10.87        0        1
2 0.7596 4174 928.1 57.32 348.7 16.41 10.51        0        0
3 0.7424 4173 870.3 50.05 330.0 16.41 10.50        0        0
4 0.5543 4264 620.0 24.99 239.1 16.66 10.93        0        1
5 0.5489 4169 622.3 25.16 241.0 16.40 11.28        0        1
  Species6
1        0
2        0
3        0
4        0
5        0
```

DataBIOMOD contains the environmental variables in the first columns, followed by the species occurrences. DataEvalBIOMOD has the same structure but it contains the data for testing of the models.

An object called Biomod.material is also produced which contains vital informtaion for most of the functions like the number of variables, the number of species, etc. Again, make sure to keep it unchanged.

0.4 Models

0.4.1 Short description

The function "Models" runs the different models implemented in BIOMOD, as well as their evaluation using three different techniques (kappa statistic, True Skill Statistics and ROC curve).

Nine different models are currently implemented:
- Generalised Linear Models (GLM)
- Generalised Additive Models (GAM)
- Classification Tree Analysis (CTA)
- Artificial Neural Networks (ANN)

- Surface Range Envelope (SRE)
- Generalised Boosting Model (GBM)
- Breiman and Cutler's random forest for classification and regression (randomForest)
- Mixture Discriminant Analysis (MDA)
- Multiple Adaptive Regression Splines (MARS)

The selection of each model is made by typing T (TRUE) or F (FALSE). There are also various parameters that needs setting up for some of the models. See below for the explanation.

All the selected models (= T) will run for each species on the calibration dataset. Below you can find a short explanation of each model and each parameter of the function. Note that they are not explained in the order they appear in the Models function. A more extensive description of the models can be found at the end of the Manual (see section "Models' description").

Models' selection and parameters

- *GLM = T, TypeGLM = "poly", Test = "BIC"*: Run a stepwise GLM (TRUE), using linear ("simple"), quadratic ("quad") or polynomial ("poly") terms. The stepwise procedure either uses the AIC or BIC criteria.

- *GBM = T, No.trees = 3000, CV.gbm = 5*: Run a generalised boosting model (GBM) (= boosted regression trees). The maximum number of trees can be user defined (default=3000). A cross-validation procedure to select the optimal number of trees is implemented. The defaul number of cross-validation is 5.

- *GAM = T, Spline = 4*: Run a generalised additive model (GAM) with a spline function with a degree of smoothing of 4 (similar to a polynomial of degree 3).

- *CTA = T, CV.tree = 50*: Run a classification tree analysis (CTA). The optimal length of the tree is estimated using cross-validation (default=50).

- *ANN = T, CV.ann = 2*: Run an artificial neural network (ANN). As different runs can provide different results, the best amount of weight decay and the number of units in the hidden layer is selected by using N-fold cross-validation (3 by default). The user can also select the number of cross-validations.

- *SRE = T, Perc025=T, Perc05=F*: Run an rectilinear surface range envelop (=BIOCLIM) using the percentile 0.025 or 0.05 as recommended by Nix or Busby.

- *MDA = T*: Run a mixture discriminant analysis using the MARS function for the regression part of the model.

- *MARS* = *T*: Run a multivariate adaptive regression spline.

- *RF* = *T*: Run a random forest model.

Evaluation of the models

- *ROC* = *T*: Evaluate the models using the Area Under the ROC (receiver operating characteristic curve) Curve (AUC)

- *Optimized.Threshold.ROC* = *T*: ROC is a threshold independent method. However, if the user wants to find the optimal threshold optimising the percentage of presence and absence correctly predicted, this threshold can be used to transform the probabilities of occurrence from models into presence and absence.

- *Kappa* = *T*: Evaluate the models using the Cohen's Kappa statistic. The treshold optimising the Kappa is kept.

- *TSS* = *T*: Evaluate the models using the True Skill Statistic (TSS). The treshold optimising the TSS is kept.

- *VarImport*: if True, this parameter enables a direct comparison of the explanatory variable importance across models. Once the models are trained (i.e. calibrated), a standard prediction is made. Then, one of the variables is randomized and a new prediction is made. The correlation score between that new prediction and the standard prediction is calculated and is considered to give an estimation of the variable importance in the model : if there is a good correlation score, i.e., the predictions do only slightly differ across the two methods, then the randomized variable does not influence the model in its prediction. This step is repeated n times for each variable independently.

NOTE : when the *VarImportance* function is run, the output is giving 1 minus the mean correlation for each variable. Therefore, a high score means a high importance. The results can also be given as a relative importance, i.e., the values are no longer related to the correlation scores but give a ranking of the variable importance (which sums up to one).

Using pseudo-absences

- *NbRepPA* = *0*: This will set the use of a pseudo-absences selection if higher than 0. Please refer to the Pseudo-absences section for detailed explanations. Various repetitions of this procedure can be done, multiplying the total number of runs that are to be done for each model.

- *strategy = 'random'*: the strategy to use for selecting pseudo absences. Can be either "circles", "squares", "per", "random" or "sre".

- *coor = CoorXY*: a two columned matrix giving the coordinates of your data points. It is needed for the "per", "circles" and "squares" strategies

- *distance = 3*: a value giving the distance to use for the "per", "circles" and "squares" strategy of the pseudo absences selection.

- *nb.absences = 2000*: the number of pseudo absences wanted to run the models with. They are randomly selected from the pool of pseudo absences available selected by the given strategy.

Determine the general evaluation procedure and the number of runs

The combination of the values given to the following arguments will determine in which way the models will be built and tested. Pay particular attention to these.

- *DataSplit*: the ratio used for splitting the original database in calibration and evaluation subsets during the evaluation procedure mentioned above. *Note that the function ensures that prevalence (ratio between the total number of presences and the total number of points) is conserved in the calibration and evaluation datasets*

- *NbRunEval*:
If no independent data for model evaluation exists, two alternatives are available. First, we can use a random data splitting procedure into, e.g., 70/30 % as commonly used (Araújo, et al. 2005b, Guisan and Thuiller 2005), where the models are calibrated on a random subset of 70 % of the data and evaluated on the remaining 30 %.

Secondly, a multiple cross-validation approach is available, where BIOMOD replicates the data splitting procedure N times, runs the models, record the predictive performance and then provide the mean of the cross-validation.

It gives a more robust estimate of the predictive performance of each selected model and it also provides an assessment of the sensitivity of the model to the initial conditions, i.e., to the species distribution data. Of course, it takes longer to run on basic personal computers.

a new way of calibrating / evaluating the models: To be reliable, predictions must be validated using independent data. As this information is often unavailable, an alternative is to partition randomly the data into a calibration (or training) and an evaluation (or testing) dataset. The calibration data is used during the fitting or learning process of the models, while the evaluation data ae used to estimate the prediction ability of the model. Within BIOMOD, this evaluation step can be done according to three different techniques (Kappa, ROC, TSS). However, this classic splitting method can add variability in the predictions when several runs are made: because the splitting of

the data is made at random, each run will have its own calibration and evaluation datasets resulting in inevitable differences during the calibration of models and the subsequent predictions. To address this problem, BIOMOD allows evaluation of model performance on different data split runs and then allows using 100 % of the data to make a final calibration of the models for prediction. In this case, the evaluation is more reliable and the predictions are not influenced by the random splitting of the data. It does, however, require more time to go through all the evaluation runs than when using a single run of the classic but necessarily biased splitting procedure. An extra evaluation of the final model can also be done if independent data are available.

- *Yweights*: Weights that the user can set for the response variables (a matrix with N columns for the N species). This is similar to an index of detectability for each site, which allows users to give stronger weights to more reliable presences or absences. It can be scaled up and put as a weight in the modeling process. For more information, see how *weights* is working in R.

- *KeepindependentPred*: If TRUE (and the truly independent data has been provided), theen the prediction on the independent data will be saved. If FALSE, only the predictive accuracy on the independent data are conserved (obtained by ROC, TSS and/or Kappa).

0.4.2 Running the models

We can now run the different models on our species. It takes only a few moments for each model to run. Here we will have 9(modles)*4(3 repetitions + final model)*2(PA repetitions) which makes 72 models per species, it will thus take several minutes.

It might be appropriate to fraction the modelling effort on basic personal computers (i.e. laptops), especially if your data has tens of thousands of rows (requiring longer calculation time). One can run one species at a time with all the models being put to true.
Note that in contrast with earlier versions of BIOMOD, it is unwise to run one model at a time as the results are now stored per species. Making several runs with different models will bring unwelcome trouble for analysing the outputs.

Please, be also aware that the *NbRunEval* and *NbRepPA* arguments can considerably enlarge your calculation time by multiplying the number of runs to be made for each species. Do not enter excessively high values for these two arguments **unless** you have sufficient patience and/or reasonable calculation power.

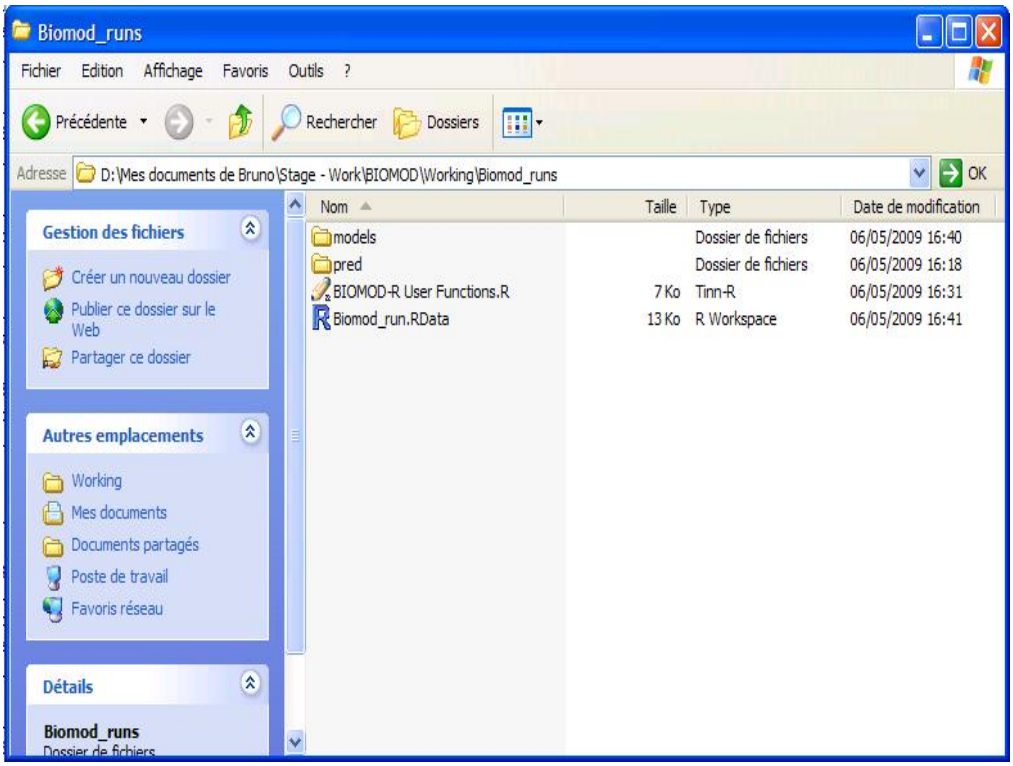
```
> #Models(GLM = T, TypeGLM = "poly", Test = "AIC", GBM = T, No.trees = 2000, GAM = T,
> #   Spline = 3, CTA = T, CV.tree = 50, ANN = T, CV.ann = 2, SRE = T, Perc025=T, Perc05=F, MDA = T,
> #   MARS = T, RF = T, NbRunEval = 3, DataSplit = 80, Yweights=NULL, Roc = T, Optimized.Threshold.Roc = T,
> #   Kappa = T, TSS=T, KeepPredIndependent = T, VarImport=5, NbRepPA=2, strategy="circles",
> #   coor=CoorXY, distance=2, nb.absences=1000)
>
> load("RUN.RData")
```

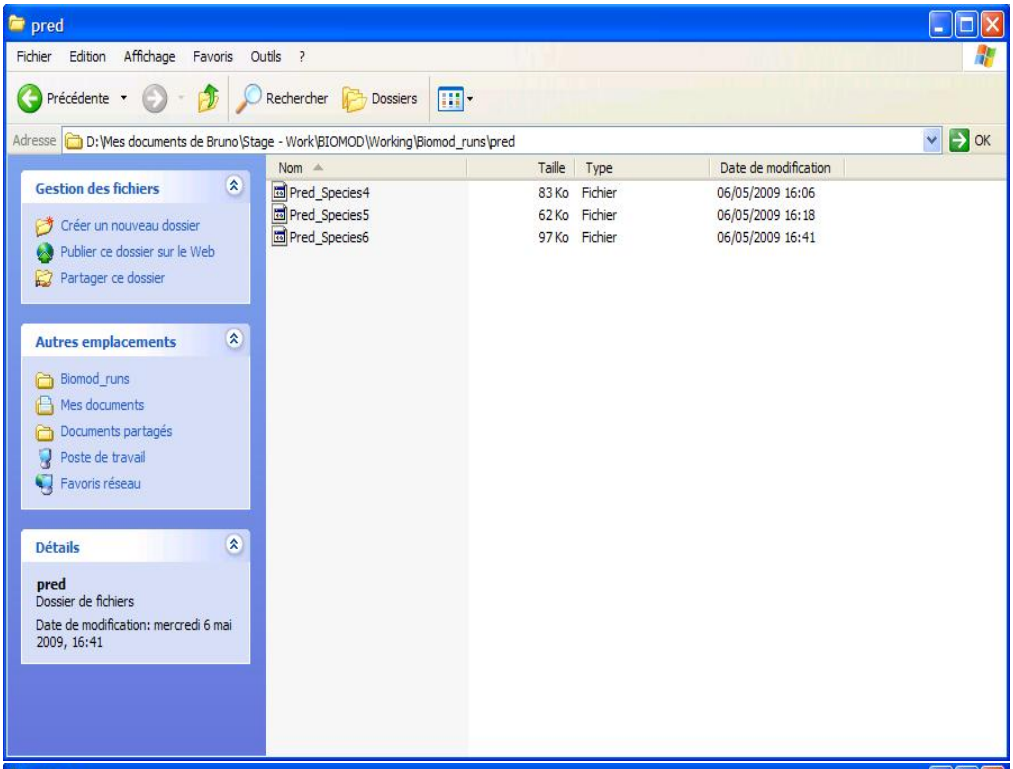
For the purpose of the example (even though the data does not ask for it) we used 2 pseudo-absences (PA) runs. Note that there has only been one PA run for Species5 because too little absences were available compared to the ones wanted. The nb.absences argument was set to 1000, but:

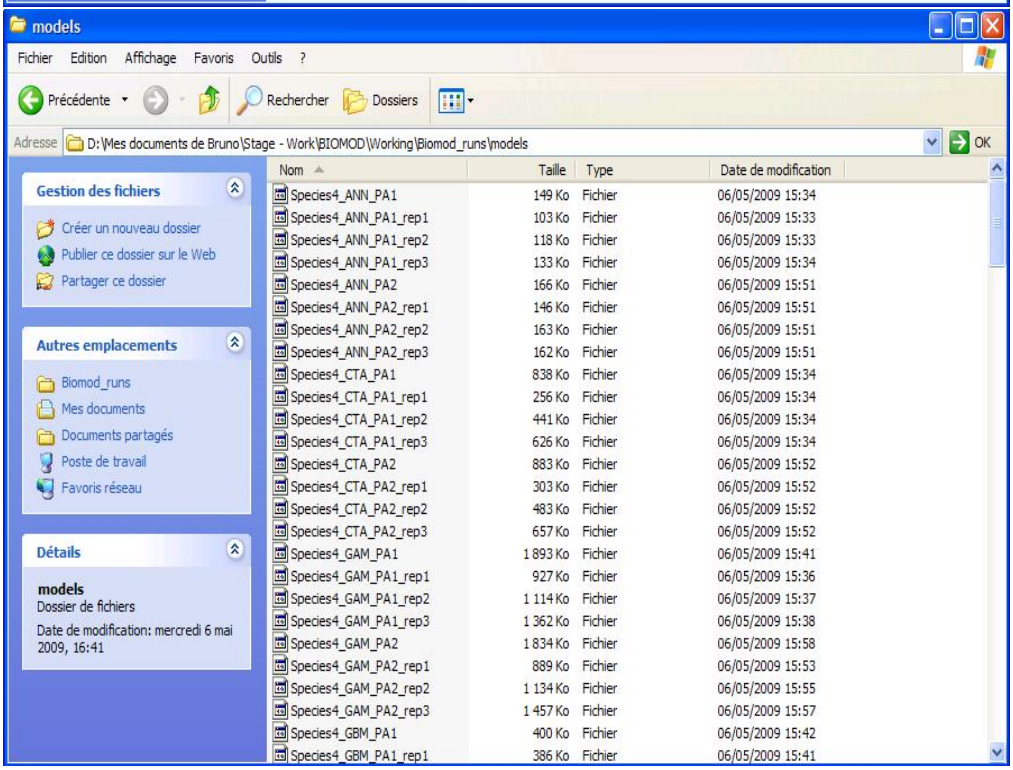
```
> #the number of data selected by the pseudo-absences procedure
> length(Biomod.PA.data$Species5)

[1] 1592

> #the number of presences for Species5
> sum(Sp.Env[,"Species5"])

[1] 678

> #Hence, the number of absences available for calibration
> length(Biomod.PA.data$Species5) - sum(Sp.Env[,"Species5"])

[1] 914
```

Too little absences are available. In this case, a single pseudo-absences run is made using all the absences available.

Your working folder should now look like this.




pred
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Dossiers
Adresse D:\Mes documents de Bruno\Stage - Work\BIOMOD\Working\Biomod_runs\pred
OK
Gestion des fichiers
Créer un nouveau dossier
Publier ce dossier sur le Web
Partager ce dossier
Autres emplacements
Biomod_runs
Mes documents
Documents partagés
Poste de travail
Favoris réseau
Détails
pred
Dossier de fichiers
Date de modification: mercredi 6 mai 2009, 16:41
Nom Taille Type Date de modification
Pred_Species4 83 Ko Fichier 06/05/2009 16:06
Pred_Species5 62 Ko Fichier 06/05/2009 16:18
Pred_Species6 97 Ko Fichier 06/05/2009 16:41


models
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Dossiers
Adresse D:\Mes documents de Bruno\Stage - Work\BIOMOD\Working\Biomod_runs\models
OK
Gestion des fichiers
Créer un nouveau dossier
Publier ce dossier sur le Web
Partager ce dossier
Autres emplacements
Biomod_runs
Mes documents
Documents partagés
Poste de travail
Favoris réseau
Détails
models
Dossier de fichiers
Date de modification: mercredi 6 mai 2009, 16:41
Nom Taille Type Date de modification
Species4_ANN_PA1 149 Ko Fichier 06/05/2009 15:34
Species4_ANN_PA1_rep1 103 Ko Fichier 06/05/2009 15:33
Species4_ANN_PA1_rep2 118 Ko Fichier 06/05/2009 15:33
Species4_ANN_PA1_rep3 133 Ko Fichier 06/05/2009 15:34
Species4_ANN_PA2 166 Ko Fichier 06/05/2009 15:51
Species4_ANN_PA2_rep1 146 Ko Fichier 06/05/2009 15:51
Species4_ANN_PA2_rep2 163 Ko Fichier 06/05/2009 15:51
Species4_ANN_PA2_rep3 162 Ko Fichier 06/05/2009 15:51
Species4_CTA_PA1 838 Ko Fichier 06/05/2009 15:34
Species4_CTA_PA1_rep1 256 Ko Fichier 06/05/2009 15:34
Species4_CTA_PA1_rep2 441 Ko Fichier 06/05/2009 15:34
Species4_CTA_PA1_rep3 626 Ko Fichier 06/05/2009 15:34
Species4_CTA_PA2 883 Ko Fichier 06/05/2009 15:52
Species4_CTA_PA2_rep1 303 Ko Fichier 06/05/2009 15:52
Species4_CTA_PA2_rep2 483 Ko Fichier 06/05/2009 15:52
Species4_CTA_PA2_rep3 657 Ko Fichier 06/05/2009 15:52
Species4_GAM_PA1 1 893 Ko Fichier 06/05/2009 15:41
Species4_GAM_PA1_rep1 927 Ko Fichier 06/05/2009 15:36
Species4_GAM_PA1_rep2 1 114 Ko Fichier 06/05/2009 15:37
Species4_GAM_PA1_rep3 1 362 Ko Fichier 06/05/2009 15:38
Species4_GAM_PA2 1 834 Ko Fichier 06/05/2009 15:58
Species4_GAM_PA2_rep1 889 Ko Fichier 06/05/2009 15:53
Species4_GAM_PA2_rep2 1 134 Ko Fichier 06/05/2009 15:55
Species4_GAM_PA2_rep3 1 457 Ko Fichier 06/05/2009 15:57
Species4_GBM_PA1 400 Ko Fichier 06/05/2009 15:42
Species4_GBM_PA1_rep1 386 Ko Fichier 06/05/2009 15:41

0.4.3 Analysing the outputs

There are now various objects stored in the workspace:

```
> ls()
```

```
 [1] "BestModelByRoc"
 [2] "BestModelByTSS"
 [3] "Biomod.material"
 [4] "Biomod.PA.data"
 [5] "Biomod.PA.sample"
 [6] "consensus_Current_results"
 [7] "CoorXY"
 [8] "data.used"
 [9] "DataBIOMOD"
[10] "DataEvalBIOMOD"
[11] "Evaluation.results.Kappa"
[12] "Evaluation.results.Roc"
[13] "Evaluation.results.TSS"
[14] "Future1"
[15] "Models.information"
[16] "Pred_Species4"
[17] "Pred_Species5"
[18] "Pred_Species6_indpdt"
[19] "PredBestModelByKappa_Species4"
[20] "PredBestModelByRoc_Species4_Bin"
[21] "predind"
[22] "Proj_Future1_BestModelByTSS"
[23] "Proj_Future1_BestModelByTSS_Bin"
[24] "Proj_Future1_Species4"
[25] "Proj_Future1_Species4_BinRoc"
[26] "Sp.Env"
[27] "Species4_ANN_PA1"
[28] "Species4_CTA_PA1"
[29] "Species4_GLM_PA1"
[30] "Species4_MARS_PA1"
[31] "Species4_MDA_PA1"
[32] "Species4_RF_PA1"
[33] "Species5_GBM_PA1"
[34] "SpNoName.circles.2"
[35] "Total_consensus_Current"
[36] "VarImportance"
```

Some of them are results from the run, like the Evaluation.results or the VarImportance objects, some others are usefull values to keep in memory (like Biomod.material, Biomod.PA.data, etc.) and some are dataframes (Sp.Env, DataBIOMOD).

Each model (excepted SRE) generates an object storing the different parameterisation, the importance of each variable (for GBM, GAM, randomForest), and the ANOVA for variable significance (GLM, GAM), and so on. This output is essential as it allows generating predictions, to know which variable has been selected and so on.

The models themselves are now stored out of the R workspace directly on the computers' hard disk. They are named after the model and the species' names, i.e. Species6_ MDA for example. There is also extensions of the names concerning the repetitions and the pseudo-absences runs, so that now one of our models will be Species6_ MDA_ PA1_ rep2.

Back loading the models and having them directly usable is very straightforward : simply use the *load* function to have the model restored in the R workspace, with the same name plus the directory root. This is also the case with the other outputs stored outside of R (predictions and projections).

Let's see an example with the GLM for the first species modelled (the syntax is not always handy but you will get used to it after a while):

```
> load("models/Species4_GLM_PA1")
> Species4_GLM_PA1


Call:  glm(formula = Species4 ~ poly(Var7, 3) + poly(Var3, 3) + poly(Var2,      3) + Var5 + poly(Var4, 2) + poly

Coefficients:
   (Intercept)  poly(Var7, 3)1  poly(Var7, 3)2  poly(Var7, 3)3
     -4.65e+00        4.71e+02       -4.83e+02        1.57e+02
poly(Var3, 3)1  poly(Var3, 3)2  poly(Var3, 3)3  poly(Var2, 3)1
      5.33e+02       -1.92e+02        9.35e+00        6.60e+02
poly(Var2, 3)2  poly(Var2, 3)3             Var5  poly(Var4, 2)1
     -8.62e+01       -3.61e+01       -1.51e-01       -2.88e+02
poly(Var4, 2)2  poly(Var1, 3)1  poly(Var1, 3)2  poly(Var1, 3)3
      1.39e+02        5.04e+03       -2.79e+03        8.13e+02
           Var6       I(Var5^3)
     -5.33e+00        1.33e-07

Degrees of Freedom: 1143 Total (i.e. Null);  1126 Residual
Null Deviance:            866
Residual Deviance: 194         AIC: 230
```

It shows the different variables retained in the final model.
The outputs also give the different coefficient values, the degrees of freedom, the residual deviance and the AIC of the final model. Of course, each model's outputs will not give the same information, as it depends on its specificity . A description of the outputs of each model is provided below (cf. OUPUTS and INTERPRETATION).

So, we have the outputs generated by Initial.State : - Sp.Env - CoorXY - DataBIOMOD - Biomod.material

We also have the objects produced by the *Models* function in the workspace in addition to the models and the predictions stored on the hard disk.

These are : - Evaluation.results.Roc - Evaluation.results.Kappa - Evaluation.results.TSS - VarImportance - Models.information.

The 3 first ones contain the scores of the evaluation procedure and the cutoffs for each model and for each species. *VarImportance* is rather explicit and contains the results of the variables' contribution analysis. *Models.information* is of little interest for the user, it contains essential informations to be used directly by the models to render projections.

We also get the following if NbRepPA is higher than 0 : - Biomod.PA.data - Biomod.PA.sample - SpNoName.circles.2 (or something close)

Biomod.PA.data contains the amount of data available after the inner run the pesudo-absence function. Biomod.PA.sample contains the rows to take from DataBIOMOD to get the data that

has been used for the calibration of each species and each PA run. The last object is a result of the pesudo-absence function inner run and is of no importance here (but see the "Pseudo-absences" section for explanations)

The predictions on the original dataset are stored independently for each species in an object following a 'Pred.Speciesname' logic and contains the probability of occurrence (habitat suitability index) for each run (if several runs) of the selected models.

NOTE: for calculation and memory storage purposes, this index is on a scale between 0 and 1000. To obtain a true probability of occurrence, rescaled between 0 and 1, simply divide each value by a thousand.

The same objects are produced for the independent data (if any) and the same logic is respected for the projections.

```
> load("pred/Pred_Species4")
```

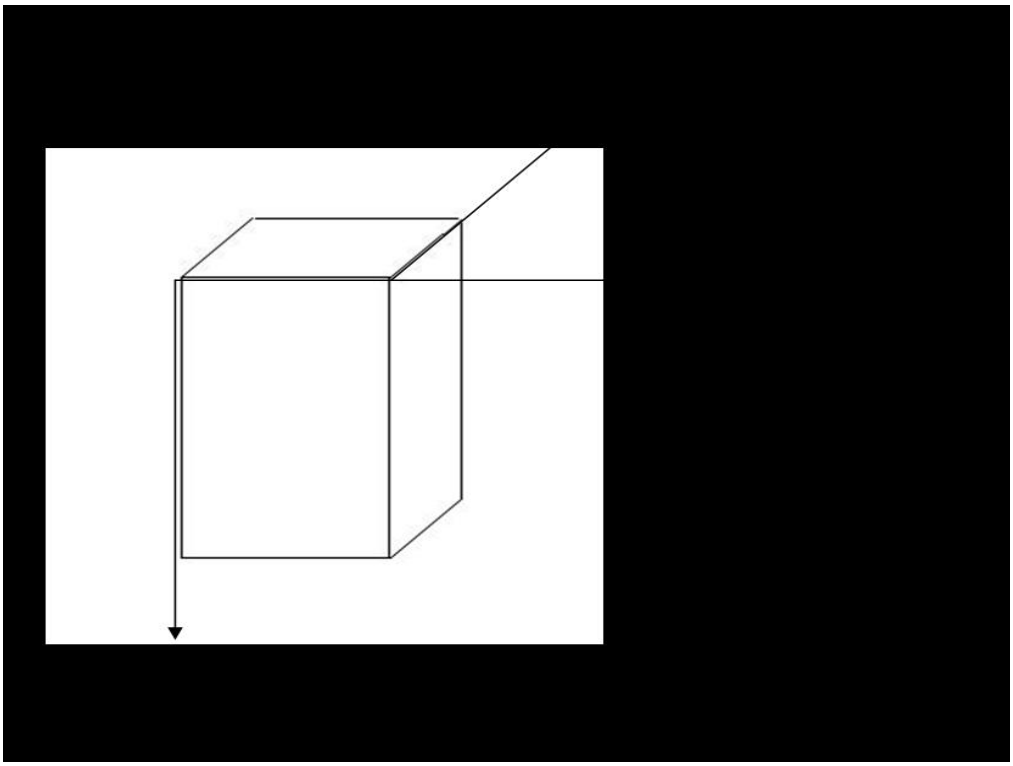
The trick is that these objects are no longer matrices but arrays (multiple dimensions) with 4 dimensions. The dimensions can be visualised as follows :

The first two build up a matrix where each column is the prediction of one of the models. The number of rows corresponds to the amount of data used for building those models.

```
> Pred_Species4[1:20,,1,1]
```

	ANN	CTA	GAM	GBM	GLM	MARS	MDA	RF	SRE
1	75	428	79	122	57	257	27	668	1000
2	1000	1000	809	790	869	668	982	904	1000
3	75	1000	241	700	209	436	460	798	1000
4	75	1000	373	642	460	444	362	750	1000
5	75	1000	490	947	495	537	877	941	1000
6	75	1000	689	869	757	750	999	949	1000
7	75	980	403	939	312	678	869	993	1000
8	75	980	445	957	470	616	501	986	1000
9	75	980	914	964	964	708	1000	918	1000
10	999	400	866	607	976	854	999	817	0
11	999	1000	935	846	994	867	1000	928	0
12	1000	980	997	950	999	653	1000	992	1000
13	75	980	859	976	941	824	999	982	1000
14	75	980	881	956	946	756	882	973	0
15	999	980	969	976	993	845	1000	986	1000
16	1000	333	998	861	999	712	999	841	1000
17	1000	1000	976	731	995	464	995	848	1000
18	1000	1000	977	925	996	795	1000	998	1000
19	1000	1000	996	934	999	547	1000	980	1000
20	75	1000	673	817	861	794	1000	964	0

Now, the third dimensions consists of a collection of 2-D matrices, one behind another, corresponding to the prediction produced by each repetition. The minimum for this dimension is 1. Considering that BIOMOD always produces a final model calibrated with 100% of the data given, the length of this third dimension is the value of the NbRunEval argument + 1. For example, with NbRunEval=10, you have 11 layers.



Note that the firts layer is always the final model, then come the repetitions.

```
> #the final model
> Pred_Species4[1:20,,1,1]
```

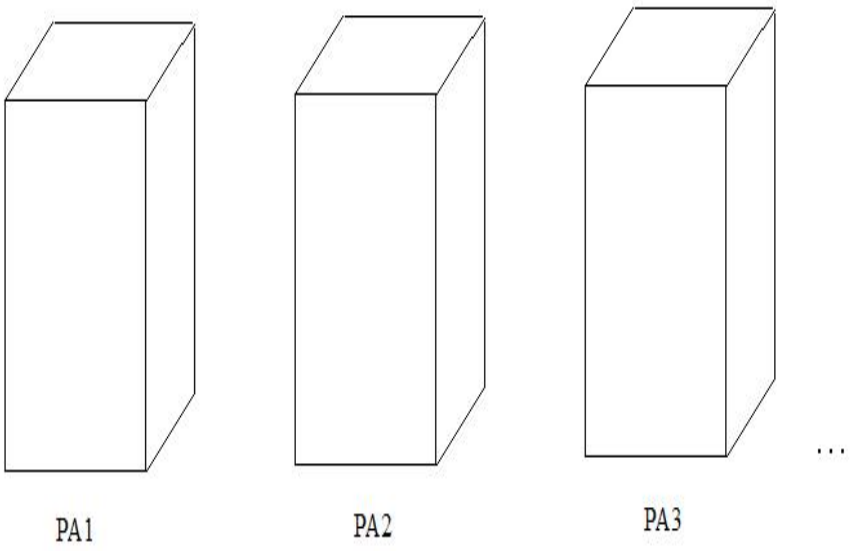
	ANN	CTA	GAM	GBM	GLM	MARS	MDA	RF	SRE
1	75	428	79	122	57	257	27	668	1000
2	1000	1000	809	790	869	668	982	904	1000
3	75	1000	241	700	209	436	460	798	1000
4	75	1000	373	642	460	444	362	750	1000
5	75	1000	490	947	495	537	877	941	1000
6	75	1000	689	869	757	750	999	949	1000
7	75	980	403	939	312	678	869	993	1000
8	75	980	445	957	470	616	501	986	1000
9	75	980	914	964	964	708	1000	918	1000
10	999	400	866	607	976	854	999	817	0
11	999	1000	935	846	994	867	1000	928	0
12	1000	980	997	950	999	653	1000	992	1000
13	75	980	859	976	941	824	999	982	1000
14	75	980	881	956	946	756	882	973	0
15	999	980	969	976	993	845	1000	986	1000
16	1000	333	998	861	999	712	999	841	1000
17	1000	1000	976	731	995	464	995	848	1000
18	1000	1000	977	925	996	795	1000	998	1000
19	1000	1000	996	934	999	547	1000	980	1000
20	75	1000	673	817	861	794	1000	964	0

```
> #the second repetition model
> Pred_Species4[1:20,,3,1]
```

	ANN	CTA	GAM	GBM	GLM	MARS	MDA	RF	SRE
1	110	11	61	62	38	270	15	613	1000

```
2    923 1000 907 798 969  765  411 845 1000
3    110  500 260 712 185  468  986 829 1000
4    110  500 446 487 505  472  981 685 1000
5    110 1000 537 932 510  559  998 932 1000
6    110 1000 785 822 886  859  367 825 1000
7    110 1000 483 966 383  713  992 990 1000
8    110 1000 454 965 461  702  997 986 1000
9   1000 1000 929 964 976  866 1000 993 1000
10 1000  400 886 638 989  895  999 860    0
11 1000 1000 927 818 995  978 1000 905    0
12 1000 1000 998 959 999  700 1000 998 1000
13  993 1000 857 943 932  834  999 968 1000
14  999 1000 895 893 951  800  999 964    0
15 1000 1000 968 938 992  887  999 961 1000
16 1000  400 999 836 999  686  999 820 1000
17 1000  500 980 513 995  508  999 613 1000
18 1000 1000 980 915 997  896 1000 997 1000
19 1000 1000 997 912 999  584 1000 961 1000
20  924 1000 748 833 895  821  999 970    0
```

The fourth dimension represents the number of pseudo-absences repetitions that have been made. In the case where NbRepPA=0, the dimension is simply 1 (not 0).



You will never visualise it this way with R though. It is just an abstract view of how it is sorted. Some usefull functions not to get lost are *dim()* and *dimnames()*. The first one gives you the number of layers for each dimension, the second will give you their names respectively.

```
> dim(Pred_Species4)

[1] 1144    9    4    2
```

```
> #the [-1] is to avoid the names of the rows, generally not usefull
> dimnames(Pred_Species4)[-1]


[[1]]
[1] "ANN"  "CTA"  "GAM"  "GBM"  "GLM"  "MARS" "MDA"  "RF"   "SRE"

[[2]]
[1] "total.data" "rep1"       "rep2"       "rep3"

[[3]]
[1] "PA1" "PA2"
```

For instance, we examine the probability of occurrence of the first species, modelled with GBM. Here we just display 20 rows in the middle (or sites).

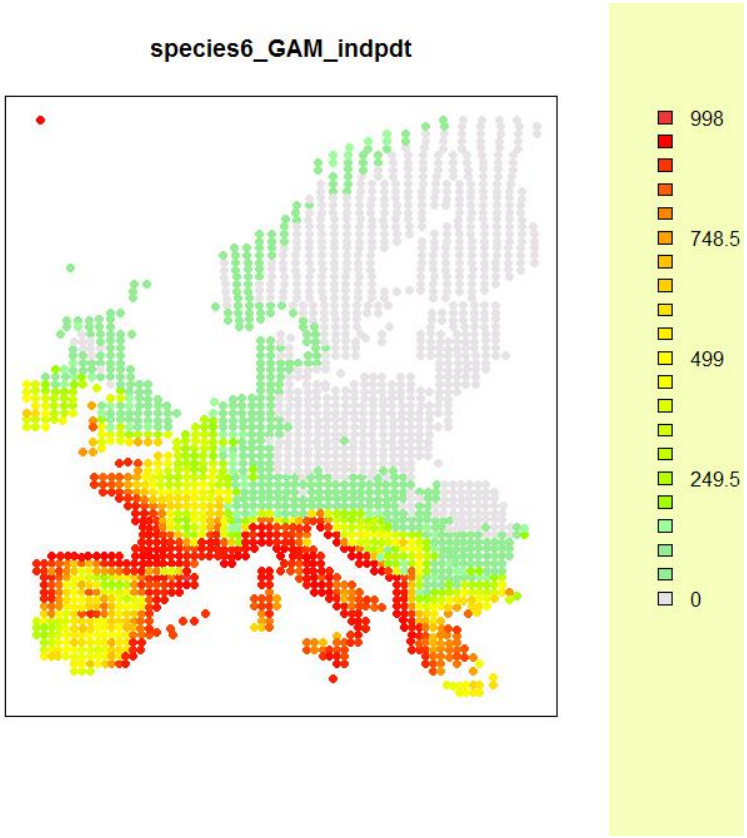
```
> #if you don't inform the 3rd and 4th dimension (you still need commas), will have all of them at once
> load("pred/Pred_Species5")
> Pred_Species5[481:500,"GBM",,]


    total.data rep1 rep2 rep3
481        757  779  791  798
482        838  850  830  833
483        520  460  484  576
484        720  682  740  711
485        457  448  352  498
486        927  944  924  962
487        910  920  922  933
488        753  554  678  585
489        984  981  979  989
490        986  982  981  989
491        989  985  985  991
492        910  852  896  851
493        743  721  794  648
494        986  977  985  987
495        987  982  984  989
496        991  987  988  992
497        991  987  988  993
498        985  982  981  986
499        991  987  988  992
500        987  982  985  989
```

Note that we only have one layer of the 4th dimension whan NbRepPA=2. That is because, as explained above, there has only been one PA run for that species.

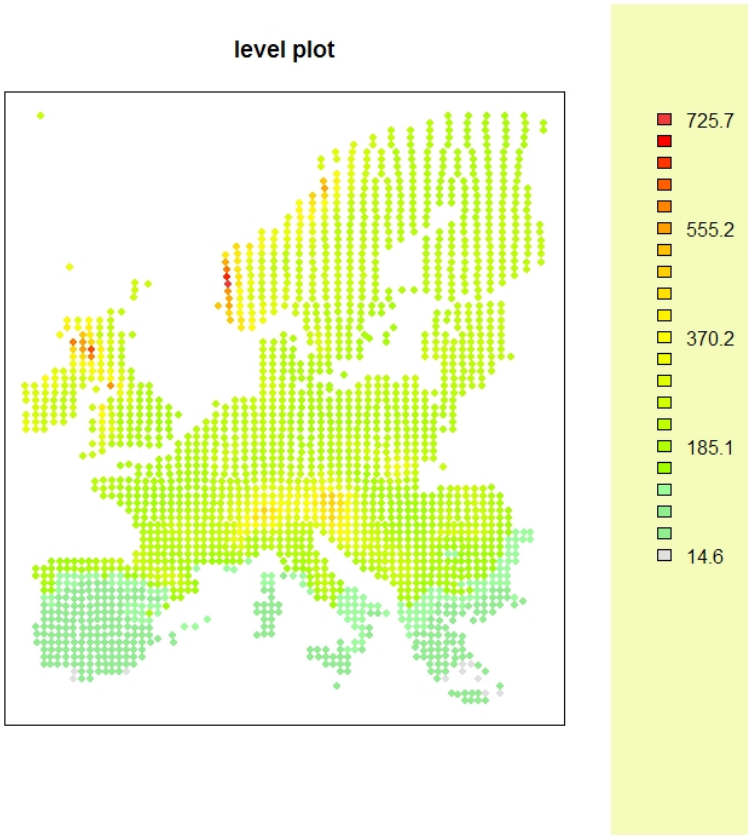
Note that because GBM contains a stochastic component, you might end up with slightly different values on these example runs.

To plot the predictions, use the *plot.level* function. It requires two inputs : the vector of values that you want to plot and the coordinates of your data points. This function works with any type of data. Because we have chosen to run the models with pseudo-absence data, plotting the partial predictions is not very convinient. We will plot instead the values of the fake independant data (which is just the full original dataset) for the GAM, and the values of one the variables used to calibrate the models.

```
> load("pred/Pred_Species6_indpdt")
> level.plot(Pred_Species6_indpdt[,"GAM",1,1], CoorXY, title='species6_GAM_indpdt')
```




```
> level.plot(Sp.Env[,5], CoorXY)
```



Note that the independent predictions are only made on the final 100% model and not on the repetitions. To check it :

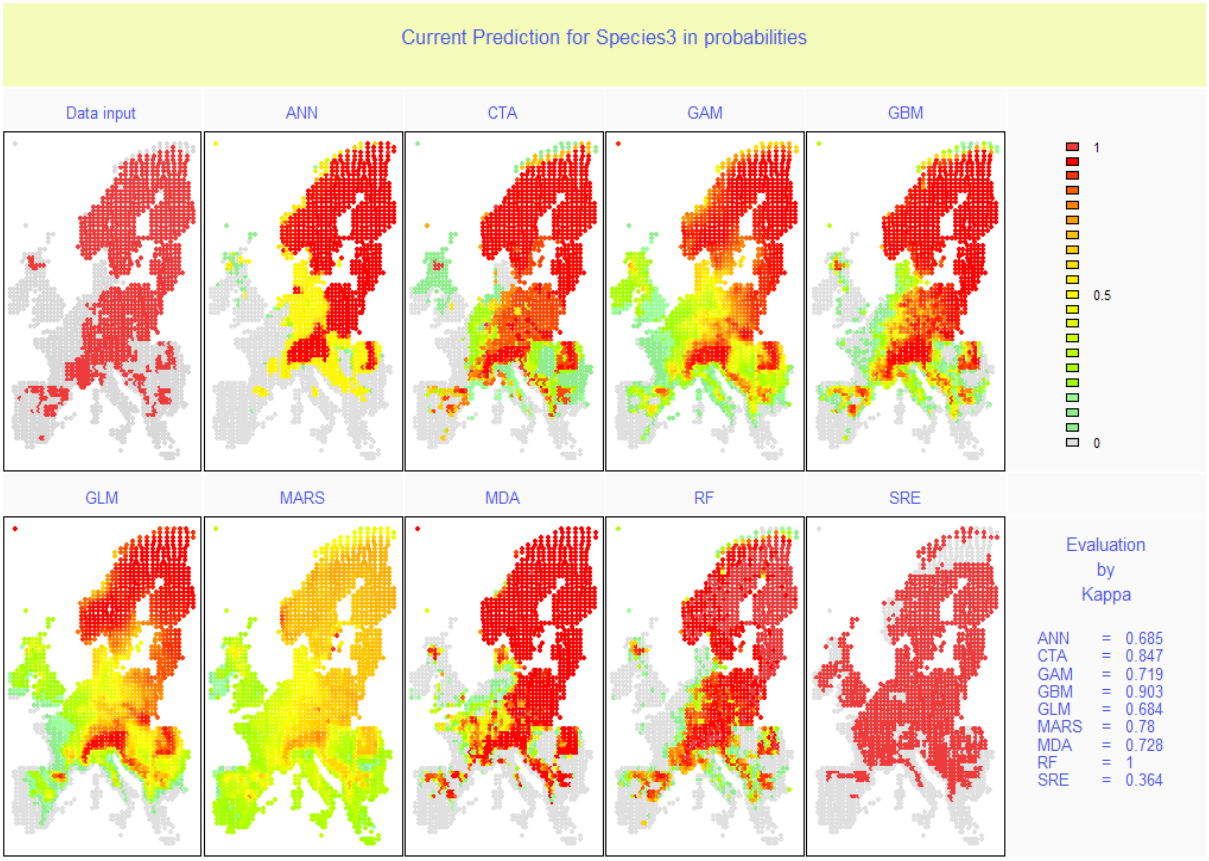
```
> Pred_Species6_indpdt[1:10,,,]


, , total.data, PA1

    ANN CTA GAM GBM GLM MARS MDA  RF  SRE
1  1000  34 204  31  47   92   0  25    0
2  1000 200 268 304 232   74   0 256    0
3  1000  34 192 113 104   66   0 112    0
4  1000  34 345  64  19  163   0  32    0
5  1000  34 509  72  21  204   0  41    0
6  1000  34 396  59  37  165   0  16    0
7  1000  34 595  69  63  212   0  26    0
8  1000  34 182  57  31   99   0  29    0
9  1000  34 252  72  24  132   0  41    0
10 1000  34 262  52  57  128   0  12 1000

, , rep1, PA1

   ANN CTA GAM GBM GLM MARS MDA RF SRE
1   NA  NA  NA  NA  NA   NA  NA NA  NA
2   NA  NA  NA  NA  NA   NA  NA NA  NA
3   NA  NA  NA  NA  NA   NA  NA NA  NA
4   NA  NA  NA  NA  NA   NA  NA NA  NA
5   NA  NA  NA  NA  NA   NA  NA NA  NA
6   NA  NA  NA  NA  NA   NA  NA NA  NA
7   NA  NA  NA  NA  NA   NA  NA NA  NA
8   NA  NA  NA  NA  NA   NA  NA NA  NA
9   NA  NA  NA  NA  NA   NA  NA NA  NA
```

```
10  NA  NA  NA  NA  NA   NA  NA NA  NA

, , rep2, PA1

   ANN CTA GAM GBM GLM MARS MDA RF SRE
1   NA  NA  NA  NA  NA   NA  NA NA  NA
2   NA  NA  NA  NA  NA   NA  NA NA  NA
3   NA  NA  NA  NA  NA   NA  NA NA  NA
4   NA  NA  NA  NA  NA   NA  NA NA  NA
5   NA  NA  NA  NA  NA   NA  NA NA  NA
6   NA  NA  NA  NA  NA   NA  NA NA  NA
7   NA  NA  NA  NA  NA   NA  NA NA  NA
8   NA  NA  NA  NA  NA   NA  NA NA  NA
9   NA  NA  NA  NA  NA   NA  NA NA  NA
10  NA  NA  NA  NA  NA   NA  NA NA  NA

, , rep3, PA1

   ANN CTA GAM GBM GLM MARS MDA RF SRE
1   NA  NA  NA  NA  NA   NA  NA NA  NA
2   NA  NA  NA  NA  NA   NA  NA NA  NA
3   NA  NA  NA  NA  NA   NA  NA NA  NA
4   NA  NA  NA  NA  NA   NA  NA NA  NA
5   NA  NA  NA  NA  NA   NA  NA NA  NA
6   NA  NA  NA  NA  NA   NA  NA NA  NA
7   NA  NA  NA  NA  NA   NA  NA NA  NA
8   NA  NA  NA  NA  NA   NA  NA NA  NA
9   NA  NA  NA  NA  NA   NA  NA NA  NA
10  NA  NA  NA  NA  NA   NA  NA NA  NA

, , total.data, PA2

   ANN CTA GAM GBM GLM MARS MDA  RF  SRE
1  556 500 238  30  25   93   0  13    0
2  556   0 304 234 249   41   0 101    0
3  556   0 215 100 101    0   0 182    0
4  556   0 311  57   8  227   0   2    0
5  556   0 469  63   8  292   0   8    0
6  556   0 353  59  24  241   0  10    0
7  556   0 551  55  39  320   0  16    0
8  556   0 191  47  14  113   0  66    0
9  556   0 249  76  11  167   0  22    0
10 556   0 293  61  35  147   0  25 1000

, , rep1, PA2

   ANN CTA GAM GBM GLM MARS MDA RF SRE
1   NA  NA  NA  NA  NA   NA  NA NA  NA
2   NA  NA  NA  NA  NA   NA  NA NA  NA
3   NA  NA  NA  NA  NA   NA  NA NA  NA
4   NA  NA  NA  NA  NA   NA  NA NA  NA
5   NA  NA  NA  NA  NA   NA  NA NA  NA
6   NA  NA  NA  NA  NA   NA  NA NA  NA
7   NA  NA  NA  NA  NA   NA  NA NA  NA
8   NA  NA  NA  NA  NA   NA  NA NA  NA
9   NA  NA  NA  NA  NA   NA  NA NA  NA
10  NA  NA  NA  NA  NA   NA  NA NA  NA

, , rep2, PA2

   ANN CTA GAM GBM GLM MARS MDA RF SRE
1   NA  NA  NA  NA  NA   NA  NA NA  NA
2   NA  NA  NA  NA  NA   NA  NA NA  NA
3   NA  NA  NA  NA  NA   NA  NA NA  NA
4   NA  NA  NA  NA  NA   NA  NA NA  NA
5   NA  NA  NA  NA  NA   NA  NA NA  NA
6   NA  NA  NA  NA  NA   NA  NA NA  NA
7   NA  NA  NA  NA  NA   NA  NA NA  NA
8   NA  NA  NA  NA  NA   NA  NA NA  NA
9   NA  NA  NA  NA  NA   NA  NA NA  NA
10  NA  NA  NA  NA  NA   NA  NA NA  NA
```

```
, , rep3, PA2

   ANN CTA GAM GBM GLM MARS MDA RF SRE
1   NA  NA  NA  NA  NA   NA  NA NA  NA
2   NA  NA  NA  NA  NA   NA  NA NA  NA
3   NA  NA  NA  NA  NA   NA  NA NA  NA
4   NA  NA  NA  NA  NA   NA  NA NA  NA
5   NA  NA  NA  NA  NA   NA  NA NA  NA
6   NA  NA  NA  NA  NA   NA  NA NA  NA
7   NA  NA  NA  NA  NA   NA  NA NA  NA
8   NA  NA  NA  NA  NA   NA  NA NA  NA
9   NA  NA  NA  NA  NA   NA  NA NA  NA
10  NA  NA  NA  NA  NA   NA  NA NA  NA
```

A more fancy version of the *plot.level* function is the *plot.map* function. It works exclusively within BIOMOD and enables a facilitated visualisation and comparison of the different outputs of the model. It requires the model and the species for which you want the plot, but also the format you want your predictions in, and with which method (if a method is needed, for instance having a binary or filtered format which requires a threshold). For example, here we plot the predictions on original data in probabilities for all the models and for the first species.

```
> plot.map(Sp=1, model='all', method='Kappa', format.type='probs', wanted='prediction')
```



You can modify the color gradient by setting the *color.gradient* argument to either *red* (the default), *blue* or *grey*.

0.5 Output and interpretation

0.5.1 Interpretation and use of GLM

Depending on what options have been selected during the model setup, the GLM model can contain the various components: the GLM object; This is a reference to the memory of the calibration process. It provides an explanation of the dependent variables selected by the stepwise procedure as well as the residual and null deviances of the model.

```
> load("models/Species4_GLM_PA1")
> Species4_GLM_PA1


Call:  glm(formula = Species4 ~ poly(Var7, 3) + poly(Var3, 3) + poly(Var2,      3) + Var5 + poly(Var4, 2) + poly

Coefficients:
   (Intercept)  poly(Var7, 3)1  poly(Var7, 3)2  poly(Var7, 3)3
     -4.65e+00        4.71e+02       -4.83e+02        1.57e+02
poly(Var3, 3)1  poly(Var3, 3)2  poly(Var3, 3)3  poly(Var2, 3)1
      5.33e+02       -1.92e+02        9.35e+00        6.60e+02
poly(Var2, 3)2  poly(Var2, 3)3            Var5  poly(Var4, 2)1
     -8.62e+01       -3.61e+01       -1.51e-01       -2.88e+02
poly(Var4, 2)2  poly(Var1, 3)1  poly(Var1, 3)2  poly(Var1, 3)3
      1.39e+02        5.04e+03       -2.79e+03        8.13e+02
           Var6       I(Var5^3)
     -5.33e+00        1.33e-07

Degrees of Freedom: 1143 Total (i.e. Null);  1126 Residual
Null Deviance:            866
Residual Deviance: 194         AIC: 230


> summary(Species4_GLM_PA1)


Call:
glm(formula = Species4 ~ poly(Var7, 3) + poly(Var3, 3) + poly(Var2,
    3) + Var5 + poly(Var4, 2) + poly(Var1, 3) + Var6 + I(Var5^3),
    family = binomial, data = DataBIOMOD[calib.lines, ], weights = Yweights[calib.lines,
        i])

Deviance Residuals:
      Min         1Q     Median         3Q        Max
-2.58e+00  -7.55e-02  -6.21e-04  -2.11e-08   3.00e+00

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -4.65e+00   2.64e+01   -0.18   0.8605
poly(Var7, 3)1  4.71e+02   1.80e+02    2.62   0.0087 **
poly(Var7, 3)2 -4.83e+02   7.47e+01   -6.46  1.0e-10 ***
poly(Var7, 3)3  1.57e+02   7.83e+01    2.01   0.0447 *
poly(Var3, 3)1  5.33e+02   9.36e+01    5.70  1.2e-08 ***
poly(Var3, 3)2 -1.92e+02   4.24e+01   -4.52  6.3e-06 ***
poly(Var3, 3)3  9.35e+00   1.48e+01    0.63   0.5272
poly(Var2, 3)1  6.60e+02   2.33e+02    2.83   0.0046 **
poly(Var2, 3)2 -8.62e+01   1.02e+02   -0.85   0.3967
poly(Var2, 3)3 -3.61e+01   4.53e+01   -0.80   0.4258
Var5           -1.51e-01   2.67e-02   -5.65  1.6e-08 ***
poly(Var4, 2)1 -2.88e+02   5.49e+01   -5.25  1.5e-07 ***
poly(Var4, 2)2  1.39e+02   4.43e+01    3.14   0.0017 **
poly(Var1, 3)1  5.04e+03   2.07e+03    2.44   0.0148 *
poly(Var1, 3)2 -2.79e+03   1.16e+03   -2.41   0.0159 *
poly(Var1, 3)3  8.13e+02   3.33e+02    2.44   0.0145 *
Var6           -5.33e+00   1.65e+00   -3.23   0.0012 **
I(Var5^3)       1.33e-07   5.46e-08    2.44   0.0148 *
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 865.93  on 1143  degrees of freedom
Residual deviance: 194.00  on 1126  degrees of freedom
AIC: 230

Number of Fisher Scoring iterations: 17
```

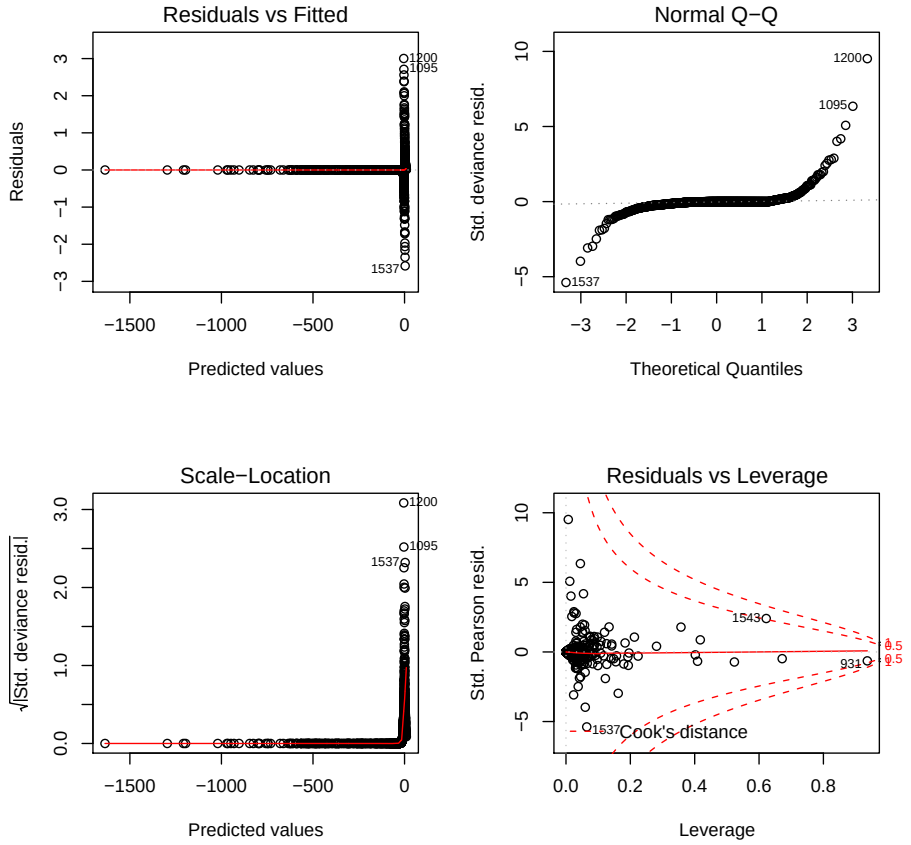
The next call obtains the anova results and the details of the stepwise procedure type. Note that the independent variables are ranked by their AIC importance.

```
> Species4_GLM_PA1$anova
```

```
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
Species4 ~ 1

Final Model:
Species4 ~ poly(Var7, 3) + poly(Var3, 3) + poly(Var2, 3) + Var5 +
    poly(Var4, 2) + poly(Var1, 3) + Var6 + I(Var5^3)


             Step Df Deviance Resid. Df Resid. Dev   AIC
1                                  1143      865.9 867.9
2 + poly(Var7, 3)  3  317.503      1140      548.4 556.4
3 + poly(Var3, 3)  3  143.076      1137      405.4 419.4
4 + poly(Var2, 3)  3  109.999      1134      295.4 315.4
5          + Var5  1   36.144      1133      259.2 281.2
6 + poly(Var4, 2)  2   41.559      1131      217.7 243.7
7 + poly(Var1, 3)  3   12.082      1128      205.6 237.6
8          + Var6  1    7.195      1127      198.4 232.4
9     + I(Var5^3)  1    4.372      1126      194.0 230.0
```

The function *plot* of R will give the basic and usual outputs for GLM. They are useful but not entirely relevant in the case of the logistic regression.

```
> par(mfrow=c(2,2))
> plot(Species4_GLM_PA1)
```



0.5.2 Interpretation and use of GBM

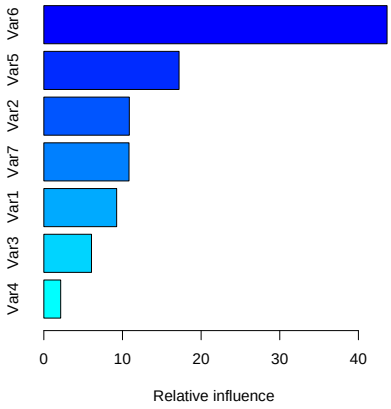
The same kind of outputs obtained for GLM can be extracted for GBM and for the others models. Here we willjust present the outputs of GBM.

The function summary computes the relative influence of each variable in the gbm object. This returns the reduction attributable to each variable in sum of squared error in predicting the gradient on each iteration. It describes the relative influence of each variable in reducing the loss function. It returns a data frame where the first component is the variable name and the second is the computed relative influence, normalized to sum up to 100.
Make sure the GBM library is uploaded.

```
> load("models/Species5_GBM_PA1")
> summary(Species5_GBM_PA1)
```

```
   var rel.inf
1 Var6  43.643
2 Var5  17.195
3 Var2  10.869
4 Var7  10.830
5 Var1   9.259
```

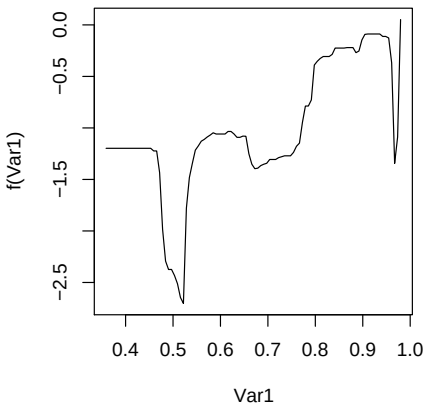
```
6 Var3   6.061
7 Var4   2.143
```



Response curves.

The GBM library allows to plot the response curves of the species against the environmental variables selected by the models.
i.var: a vector of indices or the names of the variables to plot. If using indices, the variables are indexed in the same order as they appear in the initial 'gbm' formula. For instance, here BIOMOD will plot the first variable in the model.

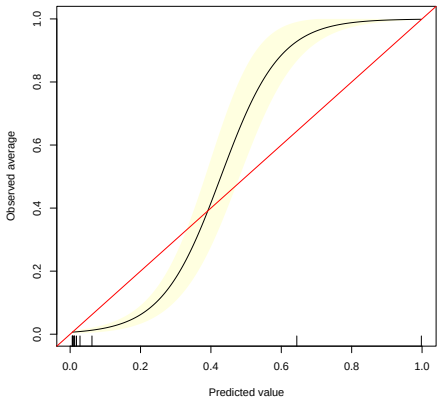
```
> plot(Species5_GBM_PA1, i.var=1)
```



The user can also use the custom *response.plot* function presented for GLM.

The *gbm* library also provides an experimental diagnostic tool that plots the fitted values versus the actual average values. Uses gam to estimate $E(y|p)$. Well-calibrated predictions imply that $E(y|p) = p$. The plot also includes a pointwise 95 band.

```
> library(gbm)
> #let's store the data that was used for calibration of the first PA run
> #for Species4 to simplify the code
> data.used <- DataBIOMOD[Biomod.PA.sample$Species4$PA1,8]
> calibrate.plot(data.used, Pred_Species4[,"GBM",1,1]/1000)
```



The function requires the observed presence-absence of the selected species and the predictions. Note that this function can also be used with any models in R-BIOMOD.

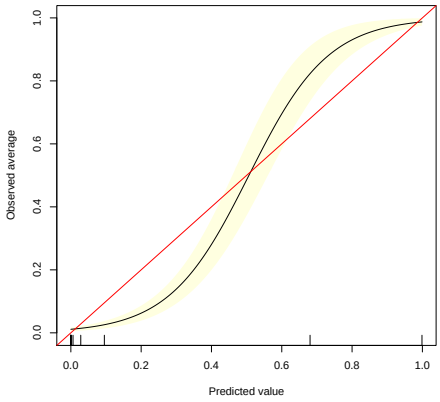
0.5.3 Interpretation and use of GAM

The outputs are very similar than for GLM.

Response curves can be plotted easily with an internal function from the *gam*.

All the tools provided by R to examine GAM results are available (make sure the GAM library is uploaded). As shown for GBM outputs, the user can use the "calibrate.plot" function of the library gbm to plot the accuracy of the model:

```
> calibrate.plot(data.used, Pred_Species4[,"GAM",1,1]/1000)
```



0.5.4 Interpretation and use of CTA

There are several useful outputs in CTA models. A critical one is *frame* which gives the details of the node, the explained deviance by each node (dev) and the probability of occurrences (yval).

```
> load("models/Species4_CTA_PA1")
> names(Species4_CTA_PA1)
```

```
 [1] "frame"     "where"     "call"      "terms"     "cptable"
 [6] "splits"    "method"    "parms"     "control"   "functions"
[11] "y"         "ordered"
```

```
> Species4_CTA_PA1$frame
```

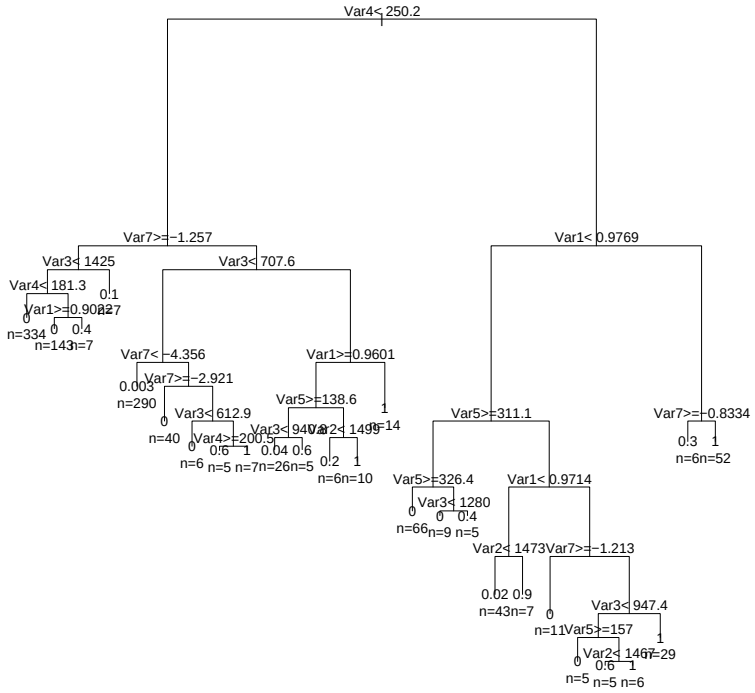
```
       var    n   wt      dev     yval complexity ncompete nsurrogate
1     Var4 1144 1144 125.8741 0.125874  0.1986814        4          4
2     Var7  900  900  41.8489 0.048889  0.0477305        4          5
4     Var3  491  491   3.9674 0.008147  0.0036968        4          1
8     Var4  484  484   2.9814 0.006198  0.0036968        4          5
16  <leaf>  334  334   0.0000 0.000000  0.0010000        0          0
17    Var1  150  150   2.9400 0.020000  0.0036968        4          0
34  <leaf>  143  143   0.0000 0.000000  0.0010000        0          0
35  <leaf>    7    7   1.7143 0.428571  0.0010000        0          0
9   <leaf>    7    7   0.8571 0.142857  0.0010000        0          0
5     Var3  409  409  36.0880 0.097800  0.0477305        4          4
10    Var7  348  348  10.6523 0.031609  0.0207005        4          5
20  <leaf>  290  290   0.9966 0.003448  0.0009657        0          0
21    Var7   58   58   8.2759 0.172414  0.0207005        4          5
42  <leaf>   40   40   0.0000 0.000000  0.0010000        0          0
43    Var3   18   18   4.4444 0.555556  0.0207005        4          4
86  <leaf>    6    6   0.0000 0.000000  0.0010000        0          0
87    Var4   12   12   1.6667 0.833333  0.0037074        4          5
174 <leaf>    5    5   1.2000 0.600000  0.0010000        0          0
175 <leaf>    7    7   0.0000 1.000000  0.0010000        0          0
11    Var1   61   61  15.2131 0.475410  0.0397250        4          5
22    Var5   47   47  10.2128 0.319149  0.0261483        4          4
44    Var3   31   31   3.4839 0.129032  0.0105052        4          2
88  <leaf>   26   26   0.9615 0.038462  0.0012833        0          0
89  <leaf>    5    5   1.2000 0.600000  0.0010000        0          0
45    Var2   16   16   3.4375 0.687500  0.0206887        4          5
90  <leaf>    6    6   0.8333 0.166667  0.0010000        0          0
91  <leaf>   10   10   0.0000 1.000000  0.0010000        0          0
23  <leaf>   14   14   0.0000 1.000000  0.0010000        0          0
3     Var1  244  244  59.0164 0.409836  0.1535166        4          0
```

```
6     Var5  186  186  35.1237 0.252688  0.0593425        4          5
12    Var5   80   80   1.9500 0.025000  0.0029792        4          1
24  <leaf>   66   66   0.0000 0.000000  0.0010000        0          0
25    Var3   14   14   1.7143 0.142857  0.0029792        4          3
50  <leaf>    9    9   0.0000 0.000000  0.0010000        0          0
51  <leaf>    5    5   1.2000 0.400000  0.0010000        0          0
13    Var1  106  106  25.8962 0.424528  0.0593425        4          5
26    Var2   50   50   6.0200 0.140000  0.0332563        4          3
52  <leaf>   43   43   0.9767 0.023256  0.0009502        0          0
53  <leaf>    7    7   0.8571 0.857143  0.0010000        0          0
27    Var7   56   56  12.2143 0.678571  0.0500752        4          2
54  <leaf>   11   11   0.0000 0.000000  0.0010000        0          0
55    Var3   45   45   5.9111 0.844444  0.0169802        4          5
110   Var5   16   16   3.9375 0.562500  0.0169802        4          5
220 <leaf>    5    5   0.0000 0.000000  0.0010000        0          0
221   Var2   11   11   1.6364 0.818182  0.0034667        4          5
442 <leaf>    5    5   1.2000 0.600000  0.0010000        0          0
443 <leaf>    6    6   0.0000 1.000000  0.0010000        0          0
111 <leaf>   29   29   0.0000 1.000000  0.0010000        0          0
7     Var7   58   58   4.5690 0.913793  0.0179136        4          4
14  <leaf>    6    6   1.3333 0.333333  0.0010000        0          0
15  <leaf>   52   52   0.9808 0.980769  0.0014361        0          0
```

This table is easier to read by plotting the tree in the same time.
Make sure the *rpart* library is loaded.

Note that the *plot* function does not display the label and text by default. The user must use the *text* function to add the text

```
> plot(Species4_CTA_PA1, margin=0.05)
> text(Species4_CTA_PA1, use.n=T)
```



Even with CTA, the *response.plot* function allows to plot the response curves. This shows the categorical rule-based approach of CTA which makes sharp relationships.

0.5.5 Interpretation and use of ANN

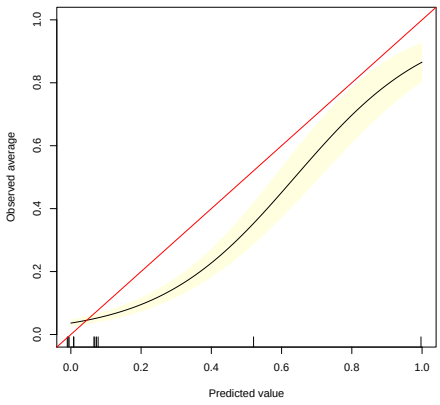
Similarly to GLM, GAM, or CTA, we can plot the response curves of the species to the selected environmental variables using the *plot.response* function.

The user can also plot the relative goodness of fit of the model, using the *calibrate.plot* function from the library(gbm):

```
> load("models/Species4_ANN_PA1")
> names(Species4_ANN_PA1)
```

```
 [1] "n"             "nunits"        "nconn"         "conn"
 [5] "nsunits"       "decay"         "entropy"       "softmax"
 [9] "censored"      "value"         "wts"           "convergence"
[13] "fitted.values" "residuals"     "lev"           "call"
[17] "terms"         "coefnames"     "xlevels"
```

```
> calibrate.plot(data.used, Pred_Species4[,"ANN",1,1]/1000)
```



0.5.6 Interpretation and use of SRE

There are no models here as this is simply a rectilinear envelop, but Like all the other models the predictions are stored.

Note also that there is no ROC evaluation available, since SRE does not provide probability values but only the presence-absence of the species. For this reason also the *calibrat.plot* function cannot work. Only TSS and Kappa are available.

0.5.7 Interpretation and use of MDA

Depending on what options have been selected during the model setup, the model can contain the following components: For instance for the first species:

```
> load("models/Species4_MDA_PA1")
> summary(Species4_MDA_PA1)
```

```
                 Length Class  Mode
percent.explained  5     -none- numeric
values             5     -none- numeric
means             30     -none- numeric
theta.mod         25     -none- numeric
dimension          1     -none- numeric
sub.prior          2     -none- list
fit               14     mars   list
call               4     -none- call
weights            2     -none- list
prior              2     table  numeric
assign.theta       2     -none- list
deviance           1     -none- numeric
confusion          4     -none- numeric
terms              3     terms  call
```

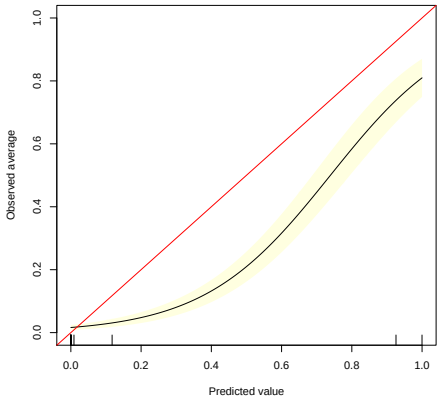
```
> Species4_MDA_PA1
```

```
Call:
mda(formula = eval(parse(text = paste(SpNames[i], paste(scopeExpSyst(DataBIOMOD[1:10,
    1:NbVar], "MDA"), collapse = "")))), data = DataBIOMOD[calib.lines,
    ], method = mars)

Dimension: 5

Percent Between-Group Variance Explained:
    v1     v2     v3     v4     v5
 66.31  83.13  96.32  98.94 100.00

Training Misclassification Error: 0.06731 ( N = 1144 )

Deviance: 479.9
```

Similarly to the previous models, we can also plot the response curves of the species against the selected environmental variables using the *response.plot* function.

To plot the relative goodness of fit using the *calibrate.plot* function of the library(gbm).

```
> calibrate.plot(data.used, Pred_Species4[,"MDA",1,1]/1000)
```



0.5.8 Interpretation and use of MARS

Depending on what options have been selected during the model setup, MDA.list can contain the following components: For instance for the first species

```
> load("models/Species4_MARS_PA1")
> summary(Species4_MARS_PA1)
```

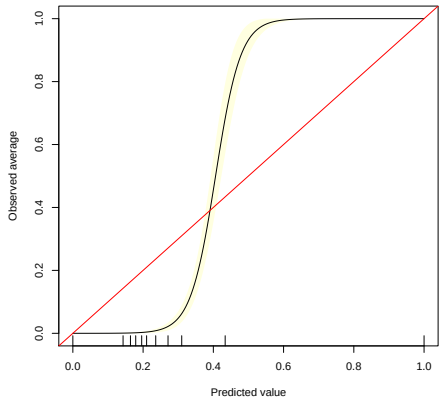
```
               Length Class  Mode
call              4   -none- call
all.terms        21   -none- numeric
selected.terms   16   -none- numeric
penalty           1   -none- numeric
degree            1   -none- numeric
nk                1   -none- numeric
thresh            1   -none- numeric
gcv               1   -none- numeric
factor          147   -none- numeric
cuts            147   -none- numeric
```

```
residuals       1144  -none- numeric
fitted.values   1144  -none- numeric
lenb               1  -none- numeric
coefficients      16  -none- numeric
x              18304  -none- numeric
```

Like for the previous models, we can also plot the response curves of the species against the selected environmental variables using the *plot.response* function.

To plot the relative goodness of fit using the *calibrate.plot* function of the library(gbm):

```
> calibrate.plot(data.used, (Pred_Species4[,"MARS",1,1]/1000))
```



0.5.9 Interpretation and use of RF

```
> load("models/Species4_RF_PA1")
> summary(Species4_RF_PA1)
```

```
                Length Class  Mode
call               6   -none- call
type               1   -none- character
predicted       1144   factor numeric
err.rate        2250   -none- numeric
confusion          6   -none- numeric
votes           2288   -none- numeric
oob.times       1144   -none- numeric
classes            2   -none- character
importance        28   -none- numeric
importanceSD      21   -none- numeric
localImportance    0   -none- NULL
proximity          0   -none- NULL
ntree              1   -none- numeric
mtry               1   -none- numeric
forest            14   -none- list
y               1144   factor numeric
test               0   -none- NULL
inbag              0   -none- NULL
```

The importance of each variable, as produced by random Forest, can be extracted.

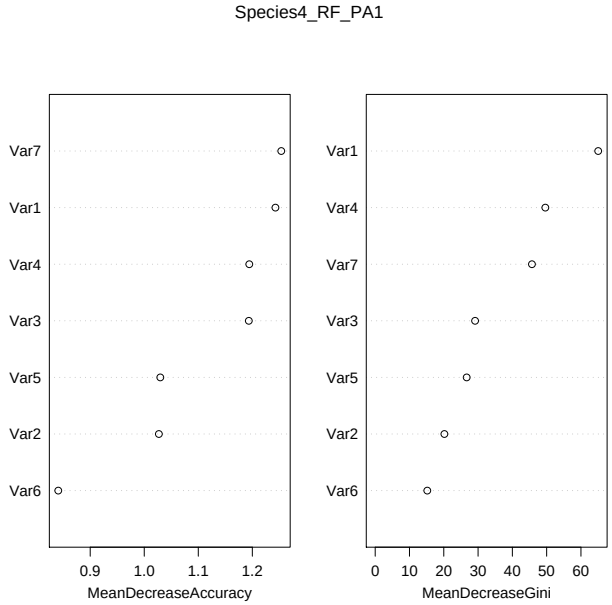
```
> Species4_RF_PA1$importance
```

```
           0       1 MeanDecreaseAccuracy MeanDecreaseGini
Var1 0.04915 0.18683                0.06647            65.03
Var2 0.03896 0.01943                0.03643            20.18
Var3 0.08045 0.12820                0.08657            29.13
Var4 0.10259 0.17605                0.11196            49.61
Var5 0.05020 0.01208                0.04536            26.68
Var6 0.04454 0.01771                0.04110            15.19
Var7 0.13554 0.22869                0.14709            45.71
```

Here are the definitions of the variables' importance measures.
- Mean Decrease Accuracy: For each tree, the prediction accuracy on the out-of-bag portion of the data is recorded. Then the same is done after permuting each predictor variable. The difference between the two accuracies are then averaged across all trees, and normalized by the standard error.
- Mean Decrease Gini: The second measure is the total decrease in node impurities from splitting on the variable, averaged over all trees. For classification, the node impurity is measured by the Gini index.

Similarly, a dotchart of variable importance as measured by a Random Forest can be plotted.

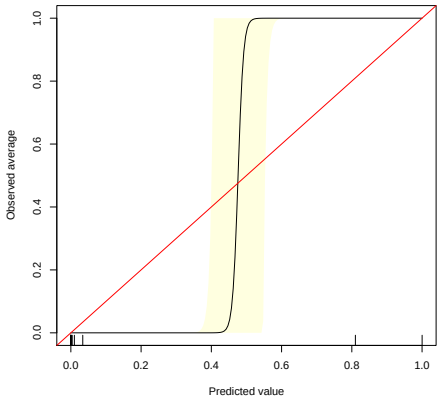
```
> varImpPlot(Species4_RF_PA1)
```



An assesment of the importance of each variable using the permutation process (similar for all the models) is also available if this option was selected. Similarly, the response curves using the *plot.response* can be plotted.

To plot the relative goodness-of-fit using the *calibrate.plot* function of the library(gbm).

```
> calibrate.plot(data.used, (Pred_Species4[,"RF",1,1]/1000))
```



0.5.10 Evaluation of the predictive performance

There are three available techniques for making an assessment of a model's performance (c.f. 0.14 Predictive Performance description). If ROC, Kappa and/or TSS is selected, the correspondant technique will be run on the cross-validation step models (if any cross- validation are wanted) and on the final model calibrated on 100% of the data. Performance measures are stored indidividually for each species and model, and for each run.

A summary table of the type "Evaluation.results.method" are produced by the *Models* function containing the predictive performance of each model which is convenient for making comparisons across methods and taxa.

```
> Evaluation.results.Kappa
```

```
$Species4_PA1
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.756        0.524      0.6633   80.0       71.53
CTA             0.676        0.615      0.9492   40.0       97.92
GAM             0.838        0.498      0.8837  229.8       91.67
GBM             0.775        0.645      0.9372  154.8       97.22
GLM             0.794         0.45      0.8985  199.8       93.75
MARS            0.717        0.579      0.8737  350.0       91.67
MDA             0.734        0.571      0.8524  120.0       94.44
RF              0.822        0.799      1.0000  340.0      100.00
SRE             0.426        0.319      0.6359   10.0       76.39
     Specificity
ANN         94.8
CTA         97.0
GAM         96.7
GBM         96.5
GLM         96.1
MARS        95.7
MDA         90.8
RF         100.0
SRE         87.2

$Species4_PA1_rep1
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.802         none      0.7717  570.0       70.83
CTA             0.757         none      0.8608  410.0       84.72
GAM             0.832         none      0.8589  449.6       86.11
GBM             0.761         none      0.8838  283.8       88.89
```

```
GLM             0.779         none      0.8421  379.6       86.81
MARS            0.659         none      0.8021  480.0       77.08
MDA             0.713         none      0.8283  700.0       86.81
RF              0.864         none      0.9724  330.0       98.61
SRE             0.369         none      0.4932   10.0       76.39
     Specificity
ANN         99.0
CTA         98.8
GAM         98.5
GBM         98.7
GLM         97.9
MARS        98.6
MDA         97.5
RF          99.5
SRE         87.0

$Species4_PA1_rep2
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.596         none      0.7696  560.0       71.53
CTA             0.629         none      0.8454  410.0       84.03
GAM             0.790         none      0.8597  439.6       86.81
GBM             0.721         none      0.8794  333.4       84.72
GLM             0.757         none      0.8591  619.4       83.33
MARS            0.715         none      0.8017  370.0       86.81
MDA             0.723         none      0.8213  200.0       88.89
RF              0.740         none      0.9512  340.0       93.06
SRE             0.446         none      0.5014   10.0       73.61
     Specificity
ANN         98.8
CTA         98.5
GAM         98.4
GBM         99.3
GLM         99.0
MARS        96.7
MDA         96.9
RF          99.8
SRE         88.3

$Species4_PA1_rep3
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.870         none      0.7711  550.0       72.22
CTA             0.641         none      0.8698  410.0       86.11
GAM             0.893         none      0.8617  509.5       85.42
GBM             0.842         none      0.8949  263.9       93.06
GLM             0.846         none      0.8482  339.7       88.89
MARS            0.776         none      0.7875  470.0       75.00
MDA             0.765         none      0.7259  520.0       86.81
RF              0.864         none      0.9686  320.0       98.61
SRE             0.465         none      0.4930   10.0       79.17
     Specificity
ANN         98.7
CTA         98.8
GAM         98.7
GBM         98.3
GLM         97.7
MARS        98.6
MDA         94.2
RF          99.4
SRE         86.1

$Species4_PA2
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.779        0.503      0.8003 230.00       83.33
CTA             0.777        0.611      0.9601  30.00       99.31
GAM             0.869        0.519      0.8877 309.69       90.97
GBM             0.864        0.658      0.9492 202.80       97.22
GLM             0.853        0.512      0.9022  89.91       97.22
MARS            0.838        0.592      0.8775 360.00       93.75
MDA             0.814        0.602      0.8925 200.00       93.75
RF              0.895        0.773      1.0000 310.00      100.00
SRE             0.466        0.319      0.6519  10.00       76.39
     Specificity
```

```
ANN         96.7
CTA         96.7
GAM         97.8
GBM         97.7
GLM         93.0
MARS        94.0
MDA         95.5
RF         100.0
SRE         88.8

$Species4_PA2_rep1
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.765         none      0.7584  480.0       79.17
CTA             0.821         none      0.8894  410.0       90.97
GAM             0.847         none      0.8808  429.6       89.58
GBM             0.870         none      0.9225  432.6       90.28
GLM             0.839         none      0.8580  519.5       85.42
MARS            0.820         none      0.7970  560.0       74.31
MDA             0.830         none      0.8508  400.0       85.42
RF              0.919         none      0.9841  390.0       98.61
SRE             0.533         none      0.5320   10.0       79.17
     Specificity
ANN         96.9
CTA         98.5
GAM         98.5
GBM         99.5
GLM         98.6
MARS        99.0
MDA         98.4
RF          99.8
SRE         88.1

$Species4_PA2_rep2
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.808         none      0.7893  430.0       77.78
CTA             0.684         none      0.8552  210.0       88.89
GAM             0.820         none      0.8779  469.5       86.81
GBM             0.802         none      0.9153  353.2       90.97
GLM             0.802         none      0.8645  609.4       84.72
MARS            0.795         none      0.8535  420.0       84.72
MDA             0.693         none      0.7800  390.0       88.89
RF              0.826         none      0.9678  500.0       95.83
SRE             0.322         none      0.5137   10.0       73.61
     Specificity
ANN         98.1
CTA         97.9
GAM         98.9
GBM         99.2
GLM         98.9
MARS        98.6
MDA         95.6
RF          99.8
SRE         88.9

$Species4_PA2_rep3
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.763         none      0.6518  310.0       65.97
CTA             0.826         none      0.8779  410.0       90.28
GAM             0.940         none      0.8831  439.6       88.19
GBM             0.921         none      0.9201  353.2       92.36
GLM             0.919         none      0.8589  479.5       86.11
MARS            0.900         none      0.8573  470.0       81.94
MDA             0.919         none      0.8614  370.0       88.19
RF              0.940         none      0.9881  410.0       99.31
SRE             0.542         none      0.5111   10.0       77.08
     Specificity
ANN         96.5
CTA         98.3
GAM         98.8
GBM         99.1
GLM         98.5
MARS        99.2
```

```
MDA         98.2
RF          99.8
SRE         87.7

$Species5_PA1
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.865        0.479      0.8614  520.0       93.36
CTA             0.933        0.594      0.9530  410.0       97.49
GAM             0.918        0.567      0.9078  379.6       96.90
GBM             0.940         0.61      0.9696  381.3       98.82
GLM             0.925        0.582      0.9141  509.5       95.13
MARS            0.912        0.561      0.9256  480.0       96.61
MDA             0.878        0.533      0.8870  170.0       93.95
RF              0.959        0.686      1.0000  390.0      100.00
SRE             0.665        0.436      0.6772   10.0       82.60
     Specificity
ANN        92.78
CTA        97.81
GAM        93.87
GBM        98.14
GLM        96.28
MARS       95.95
MDA        94.75
RF        100.00
SRE        85.12

$Species5_PA1_rep1
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.904         none      0.8976  340.0       95.28
CTA             0.949         none      0.9486  410.0       96.61
GAM             0.935         none      0.9078  439.6       95.87
GBM             0.949         none      0.9564  342.3       98.23
GLM             0.948         none      0.9135  599.4       93.07
MARS            0.936         none      0.9115  550.0       95.58
MDA             0.910         none      0.8845  330.0       93.66
RF              0.968         none      0.9936  460.0       99.85
SRE             0.633         none      0.6606   10.0       81.56
     Specificity
ANN        94.75
CTA        98.14
GAM        95.19
GBM        97.59
GLM        97.81
MARS       95.73
MDA        94.86
RF         99.56
SRE        84.68

$Species5_PA1_rep2
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.853         none      0.8492  440.0       93.36
CTA             0.968         none      0.9512  410.0       96.90
GAM             0.916         none      0.9103  439.6       95.87
GBM             0.948         none      0.9526  401.4       97.94
GLM             0.910         none      0.9097  609.4       93.36
MARS            0.903         none      0.9051  510.0       95.13
MDA             0.827         none      0.8144  180.0       93.22
RF              0.974         none      0.9949  370.0       99.56
SRE             0.712         none      0.6696   10.0       83.63
     Specificity
ANN        92.01
CTA        98.14
GAM        95.40
GBM        97.48
GLM        97.26
MARS       95.51
MDA        89.06
RF         99.89
SRE        83.81

$Species5_PA1_rep3
     Cross.validation indepdt.data total.score Cutoff Sensitivity
```

```
ANN             0.840         none      0.8511  480.0       91.59
CTA             0.884         none      0.9433  340.0       95.72
GAM             0.902         none      0.9078  429.6       95.72
GBM             0.923         none      0.9577  350.9       98.23
GLM             0.916         none      0.9190  519.5       94.99
MARS            0.897         none      0.9128  480.0       95.43
MDA             0.897         none      0.8881  330.0       93.07
RF              0.936         none      0.9872  520.0       99.12
SRE             0.650         none      0.6725   10.0       81.27
     Specificity
ANN        93.54
CTA        98.36
GAM        95.30
GBM        97.70
GLM        96.83
MARS       95.95
MDA        95.62
RF         99.56
SRE        86.00

$Species5_PA2
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN               NA            NA          NA     NA          NA
CTA               NA            NA          NA     NA          NA
GAM               NA            NA          NA     NA          NA
GBM               NA            NA          NA     NA          NA
GLM               NA            NA          NA     NA          NA
MARS              NA            NA          NA     NA          NA
MDA               NA            NA          NA     NA          NA
RF                NA            NA          NA     NA          NA
SRE               NA            NA          NA     NA          NA
     Specificity
ANN           NA
CTA           NA
GAM           NA
GBM           NA
GLM           NA
MARS          NA
MDA           NA
RF            NA
SRE           NA

$Species5_PA2_rep1
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN               NA            NA          NA     NA          NA
CTA               NA            NA          NA     NA          NA
GAM               NA            NA          NA     NA          NA
GBM               NA            NA          NA     NA          NA
GLM               NA            NA          NA     NA          NA
MARS              NA            NA          NA     NA          NA
MDA               NA            NA          NA     NA          NA
RF                NA            NA          NA     NA          NA
SRE               NA            NA          NA     NA          NA
     Specificity
ANN           NA
CTA           NA
GAM           NA
GBM           NA
GLM           NA
MARS          NA
MDA           NA
RF            NA
SRE           NA

$Species5_PA2_rep2
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN               NA            NA          NA     NA          NA
CTA               NA            NA          NA     NA          NA
GAM               NA            NA          NA     NA          NA
GBM               NA            NA          NA     NA          NA
GLM               NA            NA          NA     NA          NA
MARS              NA            NA          NA     NA          NA
```

```
MDA                NA           NA           NA     NA           NA
RF                 NA           NA           NA     NA           NA
SRE                NA           NA           NA     NA           NA
     Specificity
ANN           NA
CTA           NA
GAM           NA
GBM           NA
GLM           NA
MARS          NA
MDA           NA
RF            NA
SRE           NA

$Species5_PA2_rep3
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN                NA           NA           NA     NA           NA
CTA                NA           NA           NA     NA           NA
GAM                NA           NA           NA     NA           NA
GBM                NA           NA           NA     NA           NA
GLM                NA           NA           NA     NA           NA
MARS               NA           NA           NA     NA           NA
MDA                NA           NA           NA     NA           NA
RF                 NA           NA           NA     NA           NA
SRE                NA           NA           NA     NA           NA
     Specificity
ANN           NA
CTA           NA
GAM           NA
GBM           NA
GLM           NA
MARS          NA
MDA           NA
RF            NA
SRE           NA

$Species6_PA1
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.844        0.539      0.8681   40.0       98.21
CTA             0.877        0.662      0.9572  210.0       98.72
GAM             0.881        0.695      0.9006  399.2       95.66
GBM             0.908        0.705      0.9657  341.3       98.47
GLM             0.924        0.702      0.9339  339.7       97.19
MARS            0.907        0.686      0.9344  360.0       96.94
MDA             0.875        0.552      0.9051   20.0       95.41
RF              0.927        0.767      1.0000  340.0      100.00
SRE             0.695        0.407      0.7182   10.0       83.42
     Specificity
ANN         88.6
CTA         97.0
GAM         94.4
GBM         98.1
GLM         96.2
MARS        96.5
MDA         95.1
RF         100.0
SRE         88.4

$Species6_PA1_rep1
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.849         none      0.8466  580.0       95.41
CTA             0.885         none      0.9346  430.0       95.92
GAM             0.852         none      0.8775  399.2       95.66
GBM             0.909         none      0.9342  470.8       94.90
GLM             0.919         none      0.9181  469.5       93.62
MARS            0.921         none      0.9253  440.0       94.39
MDA             0.883         none      0.8916  740.0       92.09
RF              0.946         none      0.9893  510.0       98.72
SRE             0.732         none      0.6971   10.0       84.44
     Specificity
ANN         92.8
CTA         97.9
```

```
GAM         94.6
GBM         98.3
GLM         97.9
MARS        98.0
MDA         97.0
RF          99.9
SRE         88.3

$Species6_PA1_rep2
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.788         none      0.8233  740.0       92.60
CTA             0.859         none      0.9241  510.0       95.41
GAM             0.854         none      0.8785  558.9       88.01
GBM             0.887         none      0.9311  589.1       91.84
GLM             0.905         none      0.9222  419.6       95.15
MARS            0.881         none      0.9243  430.0       95.92
MDA             0.888         none      0.8906   80.0       95.66
RF              0.905         none      0.9788  360.0       98.98
SRE             0.684         none      0.7034   10.0       82.91
     Specificity
ANN         92.6
CTA         97.5
GAM         98.0
GBM         99.4
GLM         97.5
MARS        97.3
MDA         95.4
RF          99.2
SRE         89.5

$Species6_PA1_rep3
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.894         none      0.8569  460.0       96.17
CTA             0.886         none      0.9335  510.0       93.37
GAM             0.938         none      0.8926  468.6       93.62
GBM             0.929         none      0.9419  410.7       97.19
GLM             0.947         none      0.9210  429.6       96.17
MARS            0.920         none      0.9207  420.0       95.41
MDA             0.855         none      0.8663  390.0       93.88
RF              0.930         none      0.9859  440.0       99.74
SRE             0.669         none      0.6893   10.0       84.18
     Specificity
ANN         93.1
CTA         98.9
GAM         96.4
GBM         97.8
GLM         97.0
MARS        97.3
MDA         94.7
RF          99.3
SRE         87.9

$Species6_PA2
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.904         0.55      0.9047  470.0       96.17
CTA             0.876        0.657      0.9527  170.0       98.47
GAM             0.880        0.686      0.9112  389.2       95.92
GBM             0.884        0.688      0.9626  381.7       97.96
GLM             0.899        0.693      0.9343  389.6       96.43
MARS            0.896        0.649      0.9298  320.0       96.68
MDA             0.882        0.584      0.8928  430.0       93.88
RF              0.938        0.759      1.0000  340.0      100.00
SRE             0.690        0.407      0.7092   10.0       83.42
     Specificity
ANN         94.3
CTA         96.8
GAM         95.2
GBM         98.3
GLM         97.0
MARS        96.3
MDA         95.4
RF         100.0
```

```
SRE         87.5

$Species6_PA2_rep1
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.903         none      0.8970  500.0       89.80
CTA             0.893         none      0.9462  430.0       94.64
GAM             0.860         none      0.8904  558.9       87.76
GBM             0.883         none      0.9327  392.7       95.66
GLM             0.910         none      0.9270  499.5       94.39
MARS            0.889         none      0.9260  440.0       92.09
MDA             0.887         none      0.8872  840.0       90.56
RF              0.929         none      0.9822  350.0       98.72
SRE             0.699         none      0.7167   10.0       83.67
     Specificity
ANN         98.3
CTA         99.1
GAM         98.8
GBM         97.9
GLM         98.1
MARS        99.0
MDA         97.4
RF          99.5
SRE         90.0

$Species6_PA2_rep2
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.880         none      0.8856  400.0       93.11
CTA             0.826         none      0.9246  410.0       92.86
GAM             0.893         none      0.8954  379.2       95.41
GBM             0.873         none      0.9323  402.0       94.64
GLM             0.884         none      0.9273  409.6       94.90
MARS            0.882         none      0.9328  470.0       91.58
MDA             0.856         none      0.8949  100.0       94.64
RF              0.938         none      0.9876  300.0       99.23
SRE             0.689         none      0.6870   10.0       82.91
     Specificity
ANN         96.2
CTA         98.6
GAM         95.8
GBM         98.3
GLM         97.9
MARS        99.6
MDA         96.1
RF          99.6
SRE         88.4

$Species6_PA2_rep3
     Cross.validation indepdt.data total.score Cutoff Sensitivity
ANN             0.929         none      0.8949  500.0       89.29
CTA             0.908         none      0.9394  340.0       95.15
GAM             0.887         none      0.8904  489.0       90.31
GBM             0.896         none      0.9320  470.3       93.88
GLM             0.903         none      0.9192  399.6       95.92
MARS            0.918         none      0.9226  390.0       92.35
MDA             0.903         none      0.8780  250.0       91.84
RF              0.946         none      0.9893  500.0       98.72
SRE             0.682         none      0.6875   10.0       82.40
     Specificity
ANN         98.4
CTA         98.5
GAM         97.7
GBM         98.6
GLM         97.0
MARS        98.7
MDA         96.3
RF          99.9
SRE         88.7
```

Taking the example of the first PA run for Species4 : there are 4 different matrices, one for each run (3 repetitions with a 80-20% partitioning and the final 100% model). For the first repetition, (Evaluation.results.Kappa$Species4$_PA1_rep1), *the first column is the score on the remaining 20% of the data after calibration*

For the final model (Evaluation.results.Kappa$Species4_PA1), *the first column is the average of the cross-validation of all the repetitions. The second one is the score when the model is evaluated on independent data if any is available, a*

To display the predictive accuracy by Roc of the GLM for the first species modelled

```
> Evaluation.results.Roc$Species4_PA1["GLM",]
```

```
    Cross.validation indepdt.data total.score  Cutoff Sensitivity
GLM            0.973        0.937       0.989 121.878      94.444
    Specificity
GLM        94.4
```

As you can see the GLM has a high predictive accuracy on this particular species. The fairly small decrease of accuracy from the Calibration to the Evaluation is an indication that the model does not tend to overfit the data.
If the Optimized Threshold by ROC has been selected, a cutoff is available (fourth column) estimated using the model built with all data for calibration. It represents the best probability threshold maximising the percentage of presence and absence correctly predicted for the evaluation data. The sensitivity and specifity associated with that threshold are given in the last two columns. This threshold value will be used later to transform probabilities into presence-absence (binary format) or filtered values.

As for the probabilities, the thresholds are scalled from 0 to 1000.

The same structure is kept for Roc, Kappa and TSS methods.

0.5.11 Importance of each variable

It is always difficult to compare predictions from different models as they do not rely on the same algorithms, techniques and assumptions about the expected relationship between species distributions and the environment. With a permutation procedure, BIOMOD can extract a measure of relative importance of each variable that is independent of the model. As for the predictive accuracy, the results are stored individually per species and per model. It might be more convenient to extract the results in a summary table.
Running the *Models* function will produce an object called "VarImportance" (only if VarImp was put higher than 0 in the function call). Let's have a look at it.

```
> VarImportance
```

```
$Species4
      Var1  Var2  Var3  Var4  Var5  Var6  Var7
ANN  0.005 0.328 1.117 0.608 0.799 0.508 0.497
CTA  0.439 0.140 0.307 0.578 0.047 0.000 0.662
GAM  0.118 0.178 1.067 0.463 0.668 0.000 0.461
GBM  0.082 0.089 0.296 0.125 0.093 0.007 0.578
GLM  0.173 0.724 0.939 0.435 0.646 0.830 0.710
MARS 0.446 0.729 0.603 0.000 0.000 0.726 0.985
MDA  0.071 0.714 0.906 0.379 0.552 0.847 0.603
RF   0.176 0.065 0.254 0.316 0.060 0.024 0.484
SRE  0.076 0.005 0.093 0.064 0.033 0.006 0.219
```

```
$Species5
      Var1  Var2  Var3  Var4  Var5  Var6  Var7
ANN  0.058 1.260 0.995 0.755 0.464 0.397 1.180
CTA  0.309 0.039 0.074 0.011 0.299 0.410 0.037
GAM  0.100 1.047 0.883 0.683 0.346 0.263 0.340
GBM  0.033 0.197 0.028 0.003 0.155 0.037 0.028
GLM  0.125 0.391 0.732 0.568 0.332 0.619 0.490
MARS 0.680 0.825 0.000 0.000 0.291 0.920 0.564
MDA  0.066 0.569 0.473 0.406 0.275 0.607 0.210
RF   0.089 0.122 0.065 0.011 0.105 0.133 0.036
SRE  0.022 0.025 0.023 0.027 0.097 0.020 0.069

$Species6
      Var1  Var2  Var3  Var4  Var5  Var6  Var7
ANN  0.000 0.086 0.695 0.744 0.370 0.039 0.517
CTA  0.373 0.101 0.272 0.133 0.018 0.012 0.625
GAM  0.458 1.186 0.623 0.173 0.262 0.444 1.195
GBM  0.292 0.015 0.060 0.094 0.021 0.010 0.328
GLM  0.233 1.238 0.000 0.139 0.095 0.546 1.258
MARS 0.942 0.177 0.292 0.517 0.000 0.000 0.661
MDA  0.261 0.786 0.447 0.359 0.272 0.348 0.287
RF   0.165 0.059 0.094 0.080 0.030 0.048 0.380
SRE  0.085 0.054 0.005 0.030 0.062 0.021 0.097
```

Note that the importance of the variables is only calculated for the final model.

Remember that the importance of each variable is one minus the correlation score between the original prediction and the prediction made with a permuted variable. The range of value is therefore scaled between 0 and 1. A value of 1 means high importance whereas a value is 0 means no importance.

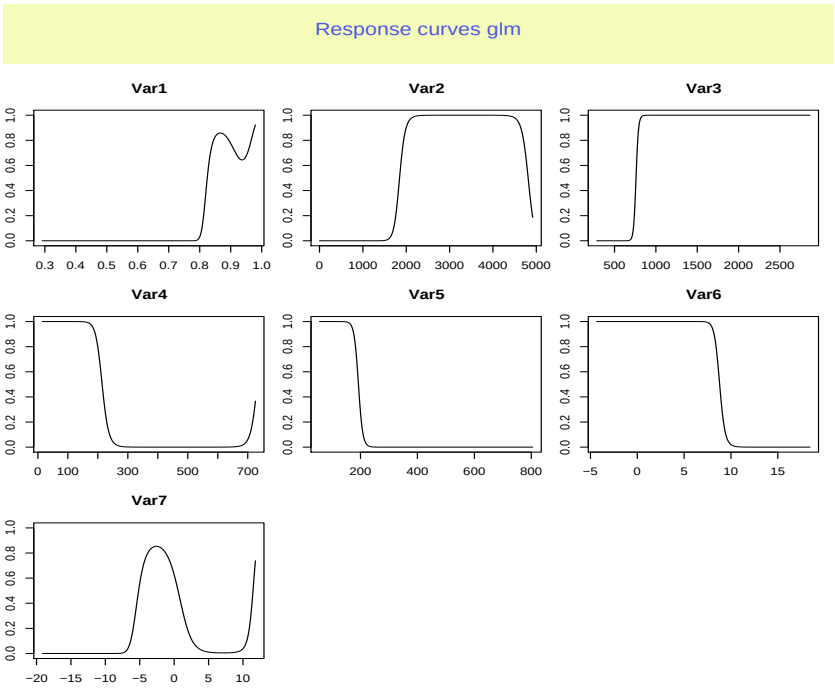
NOTE : The original correlation can be negative. We consider these cases to represent an even bigger influence of the permutated variable on the prediction than with a correlation of 0. The variable importance estimation will therefore still be given as 1 minus the correlation score and, as a consequence, turn into values higher than 1. These cases are not so rare.

0.5.12 Response curves

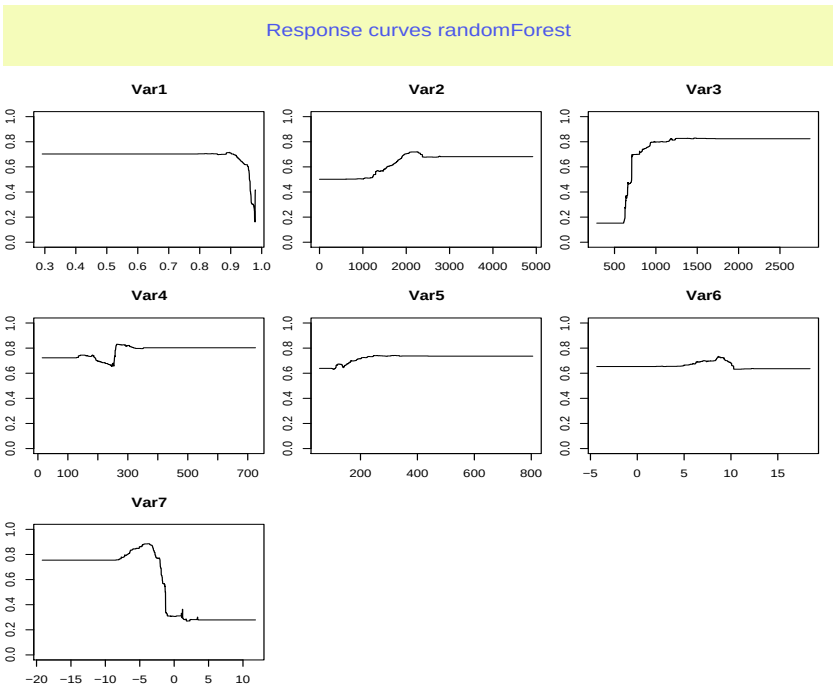
BIOMOD allows plotting the response curves of every model in the good scale. The *response.plot* function must be used to this matter. This function requires a selected model and a selected species to plot the response curves.

Here are two examples of the GLM and RF for the first species modelled. You need to load the model (those ones have already been loaded in prior calls), type its name in the first argument, then give the variables for which you want to see the curves.

```
> response.plot(Species4_GLM_PA1, Sp.Env[2:8])
```




```
> response.plot(Species4_RF_PA1, Sp.Env[2:8])
```



For this, N-1 variables are held constant at their mean value whilst the variable of interest contains 100 points varying across the maximum and the minimum of the variable's range. Variation in predictions, made to these 100 cells, only reflects the effects of variation of the one selected variable.

Thus, a plot of these predictions allows visualisation of the modelled response to the variable of interest, contingent on the other variables being held constant.
This is done subsequently for all the selected variables.

0.5.13 Predictions on the original dataset

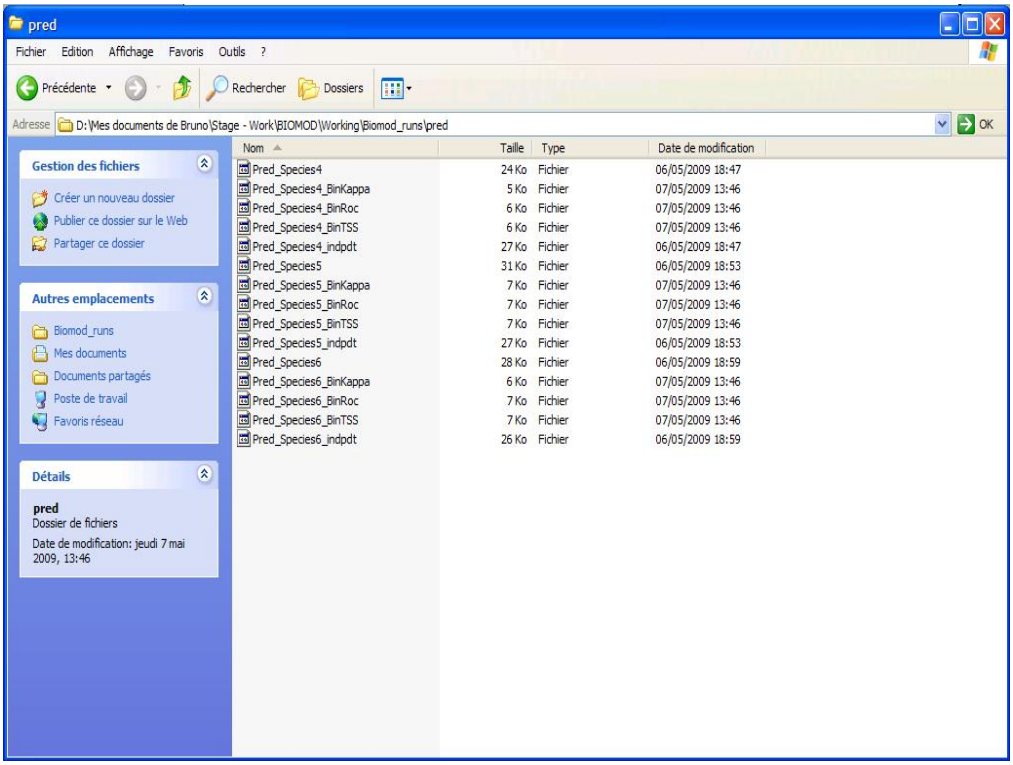
The predictions made by each model for each species are stored inside the *pred* folder. We considered it to be more convenient to have a matrix with the predictions by species for all the models.

```
> CurrentPred(GLM=T, GBM=T, GAM=T, CTA=T, ANN=T, SRE=T, MDA=T, MARS=F, RF=T,
 BinRoc=T, BinKappa=T, BinTSS=T, FiltKappa=T)
```

For each selected model (the models that were not run will be automatically switched off) an new object will be created for each species of the type Pred_Speciesname, Pred_Species4 for example (do keep in mind that they are not stored directly in the workspace and that a loading of the file is necessary). They contain the predictions made by all the models expressed as a probability of occurrence (remember that the scale is between 0 and 1000).

It might be useful to extract the presence/absence predictions. To do so, switch *BinRoc, BinKappa* and/or *BinTSS* to TRUE and each probability of occurence will be transformed into presence and absence using the cutoff maximising the models accuracy according to Roc, Kappa or TSS.

Additional datasets will be created: Pred_Species4_BinRoc, Pred_Species4_BinKappa, Pred_Species4_BinTSS, Pred_Species4_FiltKappa, and so on. The files with no extensions are to be read with R only while the text files can easily be used for other purposes (e.g. load it into another software).


pred
Fichier Edition Affichage Favoris Outils ?
Précédente
Rechercher
Dossiers
Adresse
D:\Mes documents de Bruno\Stage - Work\BIOMOD\Working\Biomod_runs\pred
OK
Gestion des fichiers
Créer un nouveau dossier
Publier ce dossier sur le Web
Partager ce dossier
Autres emplacements
Biomod_runs
Mes documents
Documents partagés
Poste de travail
Favoris réseau
Détails
pred
Dossier de fichiers
Date de modification: jeudi 7 mai 2009, 13:46
Nom
Taille
Type
Date de modification
Pred_Species4 24 Ko Fichier 06/05/2009 18:47
Pred_Species4_BinKappa 5 Ko Fichier 07/05/2009 13:46
Pred_Species4_BinRoc 6 Ko Fichier 07/05/2009 13:46
Pred_Species4_BinTSS 6 Ko Fichier 07/05/2009 13:46
Pred_Species4_indpdt 27 Ko Fichier 06/05/2009 18:47
Pred_Species5 31 Ko Fichier 06/05/2009 18:53
Pred_Species5_BinKappa 7 Ko Fichier 07/05/2009 13:46
Pred_Species5_BinRoc 7 Ko Fichier 07/05/2009 13:46
Pred_Species5_BinTSS 7 Ko Fichier 07/05/2009 13:46
Pred_Species5_indpdt 27 Ko Fichier 06/05/2009 18:53
Pred_Species6 28 Ko Fichier 06/05/2009 18:59
Pred_Species6_BinKappa 6 Ko Fichier 07/05/2009 13:46
Pred_Species6_BinRoc 7 Ko Fichier 07/05/2009 13:46
Pred_Species6_BinTSS 7 Ko Fichier 07/05/2009 13:46
Pred_Species6_indpdt 26 Ko Fichier 06/05/2009 18:59

0.6 Models' projection

For all the models currently implemented, BIOMOD is able to project potential distributions of species or land-use classes for other areas, other resolutions or other times. BIOMOD does not utilise the geographical coordinates nor does it perform a re-ordering of the data for making projections. The user should ensure that all datasets are kept in the same order in order to allow unmistaken comparisons between observed and predicted maps.

To make the projections, use the function *Projection*.

Two hypothetical future climate databases are provided with BIOMOD. They are simply called *Future1* and *Future2*. Note that the labels of the columns should be exactly the same as those of the explanatory variables (or independent variables) of the calibration datasets.

The syntax is very similar to previous functions. First add the new data (e.g. climate change scenario), then the prefix name of the output (Proj.name), and then the models for which the projections have to be made. Then, the user can select if the data needs to be transformed into a binary presence/absence format, or be filtered by a threshold (Kappa, ROC or TSS).

The Proj.name argument is very important as it will be used to store the results and also used by other functions to reload this data. The *Projection* function will create a directory using that name. In our case, it will produce "proj.Future1" next to "pred" and "models" in the working directory. A directory is created for each run of the function with a different scenario.

```
> #like for calibrating the models, you can load your own data
> #Here we use the example file
> data(Future1)
> Projection(Proj = Future1[,2:8], Proj.name='Future1', GLM = T, GBM = T, GAM = T,
    CTA = T, ANN = T, SRE = T, Perc025=T, Perc05=F, MDA =T, MARS = T, RF = T,
    BinRoc = T, BinKappa = T, BinTSS = T, FiltRoc = T, FiltKappa = T, FiltTSS = T,
    repetition.models=T)


Species4
Species5
Species6
```

Let's check the future projections made by GLM:

```
> load("proj.Future1/Proj_Future1_Species4")
> load("proj.Future1/Proj_Future1_Species4_BinRoc")
> Proj_Future1_Species4[740:760,,1,1]


     ANN  CTA GAM GBM GLM MARS  MDA  RF  SRE
740 1000  333 316 311   0  603   30 548    0
741    0    0   6  20   0  547    0  18    0
742   99    0  47  59   0  512    0 401    0
743    0    0   4  13   0  530    0   5    0
744   75 1000 190 719   0  617 1000 829 1000
745   76    0 226 232   0  595  999 410 1000
746   75 1000  93 855   0  648    0 929    0
```

```
747   75 1000  67 361   0  490    0 561 1000
748   75  333   7 302   0  419    0 518    0
749   75  333  66 423   0  671   36 661 1000
750   75  333  18 284   0  513    0 608    0
751   75 1000  63 811   0  631    0 830 1000
752    0    0   0  10   0  396    0   0    0
753    0    0   0   6   0  396    0   0    0
754    0    0   0   8   0  391    0   0    0
755    0    0   0   6   0  390    0   0    0
756    0    0   0  12   0  418    0   1    0
757    0    0   0  12   0  404    0   0    0
758    0    0   0  12   0  434    0   2    0
759    0    0   0  12   0  416    0   2    0
760    0    0   6  12   0  495    0   2    0
```

```
> Proj_Future1_Species4_BinRoc[740:760,,1,1]
```

```
    ANN CTA GAM GBM GLM MARS MDA RF SRE
740   1   1   1   1   0    1   0  0   0
741   0   0   0   0   0    1   0  0   0
742   1   0   0   0   0    1   0  0   0
743   0   0   0   0   0    1   0  0   0
744   0   1   1   1   0    1   1  1   1
745   1   0   1   1   0    1   1  0   1
746   0   1   0   1   0    1   0  1   0
747   0   1   0   1   0    1   0  0   1
748   0   1   0   1   0    1   0  0   0
749   0   1   0   1   0    1   0  1   1
750   0   1   0   1   0    1   0  0   0
751   0   1   0   1   0    1   0  1   1
752   0   0   0   0   0    1   0  0   0
753   0   0   0   0   0    1   0  0   0
754   0   0   0   0   0    1   0  0   0
755   0   0   0   0   0    1   0  0   0
756   0   0   0   0   0    1   0  0   0
757   0   0   0   0   0    1   0  0   0
758   0   0   0   0   0    1   0  0   0
759   0   0   0   0   0    1   0  0   0
760   0   0   0   0   0    1   0  0   0
```

0.7 Models' optimisation

0.7.1 Predictions on the original datasets

BIOMOD has been programmed to allow direct comparisons between models during the process. This provides a flexible way to derive optimised predictions.

The function *PredictionBestModel* will check, iteratively for each run, which model has the highest predictive accuracy according to the selected method (Roc, Kappa or TSS). Type T (TRUE) or F (FALSE) for each model you want for the optimisation. Note that if you have run the *Models* function using all models, it is not necessary to run the optimisation on all the models, but only the one which might be of interest.

The function will create new datasets prefixed *PredBestModelByXXX* (with XXX being replaced by the evaluation method used, Kappa, ROC or TSS) where the predictions on the original dataset will be stored according to the model selected. For instance, the first species could be predicted using GLM, while the second one by GAM. The selected model, the predictive accuracy, the associated threshold as well as the sensitivity and specificity of the selected models are stored in the new dataset: *BestModelByRoc*. One could choose only the optimisation run on only one evaluation method (e.g. *method='Kappa'*), or all (e.g. *method='all'*). Two additional options can also be

selected : as the previous option generates probability values, users who want binary transformation can type: *Bin.trans = T*. In this case, new datasets will be created depending on the evalution method used, e.g. *PredBestModelByRoc.BinRoc*.

If users want probability values above the threshold used to predict presences to be kept (i.e., only probabilities below the threshold are set to zero, the others are left as they were), then type: *Filt.trans = T*.

In our example, we could compare all the models we run for the different species using the three different evaluation methods available. We also transform the probabilities into the presence/absence and filtered probabilities.

```
> PredictionBestModel(GLM=T,GBM=T, GAM=T, CTA=T, ANN=T, MDA=T, MARS=F, RF=T, SRE=T,
  method='all', Bin.trans = T, Filt.trans = T)
```

Multimodel comparison according to the TSS statistic:

```
> load("pred/BestModelByTSS")
> BestModelByTSS
```

```
$Species4
         Best.Model Cross.validation indepdt.data total.score Cutoff
PA1              RF            0.901         0.96      1.0000    340
PA1_rep1         RF            0.924         none      0.9841    280
PA1_rep2         RF            0.854         none      0.9422    130
PA1_rep3         RF            0.925         none      0.9811    250
PA2              RF            0.929        0.954      1.0000    310
PA2_rep1         RF            0.945         none      0.9891    370
PA2_rep2         RF            0.863         none      0.9732    330
PA2_rep3         RF            0.980         none      0.9960    330
         Sensitivity Specificity
PA1           100.00       100.0
PA1_rep1       99.31        99.1
PA1_rep2       97.22        97.0
PA1_rep3       99.31        98.8
PA2           100.00       100.0
PA2_rep1       99.31        99.6
PA2_rep2       97.92        99.4
PA2_rep3      100.00        99.6

$Species5
         Best.Model Cross.validation indepdt.data total.score Cutoff
PA1              RF            0.960        0.741      1.0000    390
PA1_rep1         RF            0.973         none      0.9941    460
PA1_rep2         RF            0.974         none      0.9945    370
PA1_rep3         RF            0.934         none      0.9868    520
         Sensitivity Specificity
PA1           100.00      100.00
PA1_rep1       99.85       99.56
PA1_rep2       99.56       99.89
PA1_rep3       99.12       99.56

$Species6
         Best.Model Cross.validation indepdt.data total.score Cutoff
PA1              RF            0.950        0.875      1.0000    340
PA1_rep1         RF            0.960         none      0.9904    360
PA1_rep2         RF            0.937         none      0.9829    320
PA1_rep3         RF            0.952         none      0.9904    440
PA2              RF            0.937         0.87      1.0000    340
PA2_rep1         RF            0.937         none      0.9843    300
PA2_rep2         RF            0.942         none      0.9883    300
PA2_rep3         RF            0.934         none      0.9863    360
         Sensitivity Specificity
PA1           100.00       100.0
PA1_rep1       99.74        99.3
PA1_rep2       99.49        98.8
PA1_rep3       99.74        99.3
```

```
PA2           100.00       100.0
PA2_rep1       99.23        99.2
PA2_rep2       99.23        99.6
PA2_rep3       99.23        99.4
```

Multimodel comparison according to the ROC:

```
> load("pred/BestModelByRoc")
> BestModelByRoc
```

```
$Species4
         Best.Model Cross.validation indepdt.data total.score Cutoff
PA1              RF             0.98         0.99           1    620
PA1_rep1         RF            0.982         none       0.998    313
PA1_rep2         RF            0.971         none       0.997    133
PA1_rep3         RF            0.987         none       0.999    280
PA2              RF            0.992        0.989           1    592
PA2_rep1         RF            0.997         none           1    309
PA2_rep2         RF             0.98         none       0.998    176
PA2_rep3         RF            0.998         none           1    369
         Sensitivity Specificity
PA1              100         100
PA1_rep1      99.306        99.1
PA1_rep2      97.222        97.1
PA1_rep3      98.611        98.9
PA2              100         100
PA2_rep1      99.306        99.3
PA2_rep2      97.917        97.9
PA2_rep3      99.306        99.6

$Species5
         Best.Model Cross.validation indepdt.data total.score Cutoff
PA1              RF            0.996        0.934           1    649
PA1_rep1         RF            0.999         none           1    534
PA1_rep2         RF            0.999         none           1    364
PA1_rep3         RF             0.99         none       0.999    437
         Sensitivity Specificity
PA1              100         100
PA1_rep1      99.558      99.562
PA1_rep2      99.558      99.562
PA1_rep3      99.263      99.234

$Species6
         Best.Model Cross.validation indepdt.data total.score Cutoff
PA1              RF            0.996        0.974           1    654
PA1_rep1         RF            0.998         none           1    392
PA1_rep2         RF            0.994         none           1    352
PA1_rep3         RF            0.996         none           1    454
PA2              RF            0.996        0.971           1    654
PA2_rep1         RF            0.996         none           1    306
PA2_rep2         RF            0.996         none           1    289
PA2_rep3         RF            0.996         none       0.999    316
         Sensitivity Specificity
PA1              100         100
PA1_rep1      99.235        99.3
PA1_rep2       98.98          99
PA1_rep3       99.49        99.4
PA2              100         100
PA2_rep1      99.235        99.2
PA2_rep2      99.235        99.2
PA2_rep3      99.235        99.2
```

Multimodel predictions according to the Kappa statistic

```
> load("pred/PredBestModelByKappa_Species4")
> PredBestModelByKappa_Species4[740:760,]
```

```
    PA1 PA1_rep1 PA1_rep2 PA1_rep3 PA2 PA2_rep1 PA2_rep2 PA2_rep3
740   0        0        0        0   4        1        0        1
```

```
741   1        0        1        5   0        6        1        2
742   0        0        0        0   0        0        0        0
743  77      145      224       81   0        6        1        1
744   2        6       20        5   0        0        5        2
745   9        6        5       10   0        0        0        0
746   6        8        0        1   6       12        4        4
747   0        0        0        0   0        2        2        0
748  13       20        6       13   0        0        0        0
749   0        0        0        0   0        0        0        0
750   2        0        4        0   2       10       21       13
751   1       22        1        4   9       24       30       13
752   0        2        0        0   0       16        0        0
753   5        4       14        4   0        0        0        0
754   0        0        0        0   1        1        2        4
755   1        6        4        2   4        9        4        1
756  64       70       84       52   5        6        0        5
757  69       44      166      280  17       28       21       54
758  17       14       38       30   1        0        1        2
759 116      106       97      157   0       17        2        4
760  13       20        9        6  29       24       58       41
```

Multimodel predictions according to the ROC, transformed in binary presence/absence

```
> load("pred/PredBestModelByRoc_Species4_Bin")
> PredBestModelByRoc_Species4_Bin[740:760,]
```

```
    PA1 PA1_rep1 PA1_rep2 PA1_rep3 PA2 PA2_rep1 PA2_rep2 PA2_rep3
740   0        0        0        0   0        0        0        0
741   0        0        0        1   0        1        0        0
742   0        0        0        0   0        0        0        0
743   1        0        1        1   0        1        0        0
744   0        1        1        1   0        0        1        0
745   1        1        1        0   0        0        0        0
746   0        1        0        0   1        0        1        1
747   0        0        0        0   0        0        1        0
748   0        0        1        0   0        0        0        0
749   0        0        0        0   0        0        0        0
750   0        0        1        0   0        0        1        0
751   0        0        0        1   1        0        1        0
752   0        0        0        0   0        0        0        0
753   0        1        1        1   0        0        0        0
754   0        0        0        0   0        0        1        1
755   0        1        1        0   0        1        1        0
756   1        1        1        1   0        1        0        1
757   1        1        1        1   0        0        1        1
758   0        0        1        1   0        0        0        0
759   0        0        1        0   0        0        1        1
760   0        0        1        1   0        0        1        1
```

0.7.2 Projections onto the future or other areas

Depending on the model that has been selected as the *best model* into the PredictionBestModel function, optimisation for the future can also be performed using the functions *ProjectionBestModel* according to the selected evaluation method (ROC, Kappa or TSS).

The syntax is the same than in the *Projections* function. The user only needs to give the name of the climatic that will be used when running the *projection* function. Similarly to the *PredictionBestModel* function, the user can also specify if he wants the optimised-projections transformed into presence-absence or filtered, respectively typing: Bin.trans=T and Filt.trans=T.

```
> ProjectionBestModel(Proj.name='Future1', Bin.trans=T, Filt.trans=T, method='all')
> load("proj.Future1/Proj_Future1_BestModelByTSS")
> dim(Proj_Future1_BestModelByTSS)
```

```
[1] 2264    8    3
```

```
> dimnames(Proj_Future1_BestModelByTSS)[-1]
```

```
[[1]]
[1] "PA1"      "PA1_rep1" "PA1_rep2" "PA1_rep3" "PA2"      "PA2_rep1"
[7] "PA2_rep2" "PA2_rep3"

[[2]]
[1] "Species4" "Species5" "Species6"
```

For projections, the best models results are stored in 3-D arrays where the second dimension is the repetition runs and the third dimension is the species.

```
> Proj_Future1_BestModelByTSS[740:760,,"Species4"]
```

```
    PA1 PA1_rep1 PA1_rep2 PA1_rep3 PA2 PA2_rep1 PA2_rep2 PA2_rep3
740 548      534      564      625 428      482      496      580
741  18       28       20       26  20       37       28       30
742 401      349      425      452 329      288      408      316
743   5       13        6        6   6       14        8       18
744 829      925      876      897 881      800      921      908
745 410      317      473      526 393      348      593      585
746 929      962      946      898 929      913      929      901
747 561      501      349      652 550      565      697      709
748 518      434      540      624 345      445      405      510
749 661      585      668      740 641      721      706      690
750 608      521      642      684 525      641      568      597
751 830      928      905      905 893      845      925      948
752   0        0        0        0   0        0        1        1
753   0        0        0        0   0        0        0        0
754   0        1        0        0   2        1        0        0
755   0        0        0        0   0        0        0        0
756   1        4        1        0   0        1        2        2
757   0        1        0        0   0        0        0        0
758   2        4        1        1   0        1        8        1
759   2        0        0        0   0        1        0        1
760   2        4        0       10   4        5        4        5
```

```
> load("proj.Future1/Proj_Future1_BestModelByTSS_Bin")
> Proj_Future1_BestModelByTSS_Bin[740:760,,"Species4"]
```

```
    PA1 PA1_rep1 PA1_rep2 PA1_rep3 PA2 PA2_rep1 PA2_rep2 PA2_rep3
740   1        1        1        1   1        1        1        1
741   0        0        0        0   0        0        0        0
742   1        1        1        1   1        0        1        0
743   0        0        0        0   0        0        0        0
744   1        1        1        1   1        1        1        1
745   1        1        1        1   1        0        1        1
746   1        1        1        1   1        1        1        1
747   1        1        1        1   1        1        1        1
748   1        1        1        1   1        1        1        1
749   1        1        1        1   1        1        1        1
750   1        1        1        1   1        1        1        1
751   1        1        1        1   1        1        1        1
752   0        0        0        0   0        0        0        0
753   0        0        0        0   0        0        0        0
754   0        0        0        0   0        0        0        0
755   0        0        0        0   0        0        0        0
756   0        0        0        0   0        0        0        0
757   0        0        0        0   0        0        0        0
758   0        0        0        0   0        0        0        0
759   0        0        0        0   0        0        0        0
760   0        0        0        0   0        0        0        0
```

Note that it is necessary to have run the PredictionBestModel before running the ProjectionBestModel.

You can check all the new objects that have been created by going through the "pred" and "proj.Future1" directories.

0.8 Ensemble Forecasting

One difficulty with the use of species distribution models is that the number of techniques available is large and is increasing steadily, making it difficult for 'non-aficionados' to select the most appropriate methodology for their needs ((Elith, J. et al. 2006, Heikkinen, R. et al. 2006)). Recent analyses have also demonstrated that discrepancies between different techniques can be very large, making the choice of the appropriate model even more difficult. This is particularly true when models are used to project distributions of species into independent situations, which is the case of projections of species distributions under future climate change scenarios ((Pearson, R. G. et al. 2006, Thuiller, W. 2004)). A solution for this inter-model variability is to fit ensembles of forecasts by simulating across more than one set of initial conditions, model classes, model parameters, and boundary conditions (for a review see Araújo & New 2007) and analyse the resulting range of uncertainties with bounding box, consensus and probabilistic methodologies rather than lining up with a single modelling outcome ((Araújo, M. B. and New, M. 2007, Thuiller, W. 2007)). BIOMOD offers such a platform for ensemble forecasting.

Several approaches are available for combining ensembles of models in BIOMOD. Here is an example of the use of the *Ensemble.Forecasting* function as well as some details of the different strategies:

Four straightforward means of 'committee averaging' (giving the same weight to all the elements) are done across all the models for each run:

- on the probabilities
- on the binary projection according to the Roc method,
- on the binary projection according to the Kappa method,
- on the binary projection according to the TSS method.

A weighted approach is also available that ranks the models using their evaluation score.

Making a mean on the 0-1 projections gives some sort of probability of presence. For example, for a given site and with the TSS method, 6 projections give a "1" and 2 give a "0". The mean will be 0.75. It is extracted from binary projection and it is therefore not possible to determine a prior threshold. Conversion into binary is nevertheless possible (see *binary* below).

The median value is also calculated on the probabilities given by the models. It is considered to be more reliable because it is less influenced by extreme values. A weighting is not possible, nor the determination of a threshold from the already existing ones.

```
> Ensemble.Forecasting(Proj.name= "Future1", weight.method='Roc', PCA.median=T,
 binary=T, bin.method='Roc', Test=F, decay=1.6, repetition.models=T)
```

The function returns a list that is also stored in R's memory. In our case, it will be called "consensus$_{F}uture1_{r}esults$".$Itcontainsallthecomputationalinformationthathasbeenusedtorendertheensembleforecasts$ $True$), $theweightsawardedtothemodelsintheweightingprocess, themodelselectedbythePCA.medianmethod(ifsettoTrue$

Options:

repetition.models: You can choose to switch on or off the repetition models. If selected, the function will calculate the ensemble forecasts for each run and generate a final one which produces a general ensemble forecast across all the runs for each method.This total consensus is done inconsistently of this argument being set to TRUE or FALSE.

weight.method: the method for ranking the models according to their predictive performance. The *decay* gives the relative importance of the weights. The default weight decay is 1.6; See the example below.

models	GAM	GBM	GLM	ANN	RF	MARS	CTA	MDA
score with Roc	0.96	0.92	0.90	0.88	0.87	0.75	0.72	0.68
decay of 1	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
decay of 1.2	0.217	0.181	0.151	0.126	0.105	0.087	0.073	0.061
decay of 1.6	0.384	0.240	0.150	0.094	0.059	0.037	0.023	0.014
decay of 2	0.502	0.251	0.125	0.063	0.031	0.016	0.008	0.004

You can type in any value (it has however to be higher than 1) depending on the strength of discrimination that you want. A decay of 1 is equivalent to a committee averaging (i.e. same weights given to all elements).

PCA.median: this is an alternative approach for obtaining a hierarchie of models in an ensemble that does not depend on the performance of each modelling technique.
A PCA is run with projected probabilities of all of the models selected. In the current version of BIOMOD, the consensus model is the model whose projection is the most correlated with the first axis of the PCA. However, the PCA approach can be used in several ways. It can be used to select one single consensus model (as currently implemented in BIOMOD), but it can also be used to allow committee averaging across consensus models (models with high loads in the first axis of PCA), or be used to allow committee averaging across models ranking high in different axes of the PCA. Implementations of these methods can be found in Thuiller (2004), Araújo et al. (2005), and Araújo et al. (2006).
In the current version of BIOMOD no output is produced for this option, the name of the such selected model is kept in the function's information output.

binary: by setting this argument to True, the ensemble forecasting function will also render the consensus projections in a binary format. The thresholds used differ from one method to the other:
- mean on probabilities: converted in binary format by a mean threshold (thus giving 3 possibilities - Roc, Kappa or TSS; you need to set it in the *bin.method* argument),
- weighted mean on probabilities: converted in binary by a weighted mean threshold (using the same method than for ranking, i.e. the *weight.method* argument),

- Roc-Kappa-TSS means: an arbitrary value of 500 (corresponding to a probability of 0.5) is used, meaning that a site is considered suitable if at least half of the projections have projected a presence.

Test: This option will test the efficiency of the consensus method on the data given for calibration. A Roc evaluation is run and the score will be given in the output of the function as the "test.results".

OUTPUTS

This function will be run for all the species at once. It will produce an object per species. These objects are arrays of three dimensions :

```
> load("proj.Future1/consensus_Species6_Future1")
> dim(consensus_Species6_Future1)
> dimnames(consensus_Species6_Future1)[-1]
```

The second dimension is the repetition runs and the third dimension is the consensus methods. There is also an object called "Total$_{c}onsensus_{F}uture1$"$thatmakesasingleoutputoutofalltherepetitions$.

```
> load("proj.Future1/Total_consensus_Future1")
> dim(Total_consensus_Future1)
> dimnames(Total_consensus_Future1)[-1]
```

Now the second dimension is the species. Let's see and plot some of these :

```
> Total_consensus_Future1[1:20,,1]
> data <- Total_consensus_Future1
> par(mfrow=c(2,6))
> par(mar=c(0.6,0.6,2,0.6))
> level.plot(data[,1,1], CoorXY, show.scale=F, title='sp4_mean', cex=0.5)
> level.plot(data[,1,2], CoorXY, show.scale=F, title='sp4_weighted_mean', cex=0.5)
> level.plot(data[,1,3], CoorXY, show.scale=F, title='sp4_median', cex=0.5)
> level.plot(data[,1,4], CoorXY, show.scale=F, title='sp4_Roc_mean', cex=0.5)
> level.plot(data[,1,5], CoorXY, show.scale=F, title='sp4_Kappa_mean', cex=0.5)
> level.plot(data[,1,6], CoorXY, show.scale=F, title='sp4_Roc_mean', cex=0.5)
> level.plot(data[,2,1], CoorXY, show.scale=F, title='sp5_mean', cex=0.5)
> level.plot(data[,2,2], CoorXY, show.scale=F, title='sp5_weighted_mean', cex=0.5)
> level.plot(data[,2,3], CoorXY, show.scale=F, title='sp5_median', cex=0.5)
> level.plot(data[,2,4], CoorXY, show.scale=F, title='sp5_Roc_mean', cex=0.5)
> level.plot(data[,2,5], CoorXY, show.scale=F, title='sp5_Kappa_mean', cex=0.5)
> level.plot(data[,2,6], CoorXY, show.scale=F, title='sp5_TSS_mean', cex=0.5)
```

if binary is set to True, the same names are used with a terminal $_{B}incontainingtheconsensusresultsinbinaryformat$.

0.9 Migration

This function allows the inclusion of a very simple migration process when projecting species distributions into the future. The function constraints the projection to occur in a delimited perimeter around the current distribution. The delimited perimeter has to be decided by the user.
The function uses two datasets: the current species distributions and the the future (assuming by default unlimited migration).
The latitude and longitude of the datasets need to be specified in order to calculate the distances allowed for migration.

Note *that to be able to use this function, both current and future datasets must be ordered in the same way and have the same coordinates and the same resolution.*

Then the migration rate has to be specified. Two options are available. Either the user can specify the same rate for all the species modelled (a number must be given) or specify a different rate for every species modelled (a vector must be given).
For the generic migration rate, type the maximum distance the species could migrate according to the time slice modelled.
For the species-specific migration rate, create a vector (number of rows = number of species) containing for each species the maximum distance the species could migrate.

Finally, give the name where the projections using limited migration will be stored.
For instance Future1.Migration.1km.per.year.

Note that the rate of migration should be given in degrees. For instance for a species with a maximum of 1 minute (1.6km) by 10 years. If we project its distribution in 50 years: Rate = $1x0.16667x5$ *(where 0.01667 is the conversion from minute to degree).*

For projection in 2080: Rate = $1x0.16667x8$.

For a maximum rate of 3 minutes per 10 years (4.8km) in 2080: Rate = $3x0.16667x8$
we will run a projection with the original dataset to have it in the same format as the Future1 projection. We will use the overall mean consensus in binary
Projection(Proj = Sp.Env[,2:8], Proj.name='Current', GLM = T, GBM = T, GAM = T, CTA = T, ANN = T, SRE = T, Perc025=T, Perc05=F, MDA =T, MARS = T, RF = T, BinRoc=T, BinKappa=T, BinTSS=T, FiltRoc=T, FiltKappa=T, FiltTSS=T, repetition.models=T)
Ensemble.Forecasting(Proj.name= "Current", weight.method='Roc', PCA.median=T, binary=T, bin.method='Roc', Test=F, decay=1.6, repetition.models=T)
$load("proj.Future1/Total_consensus_Future1")load("proj.Current/Total_consensus_Current")$
$Migration(CurrentPred = Total_consensus_Current[,,1], FutureProj = Total_consensus_Future1[,,1], X = CoorXY[,1], Y = CoorXY[,2], MaxMigr = 5 * 0.16667 * 8, Pred.Save = "Future1.Migration")$
Future1.Migration[740:760,]

0.10 Species Turnover

This function allows to estimate species loss, gained, and turnover by pixel for the time slice considered. The function uses two datasets: the current species distributions and the future one (for instance after accounting for migration). Note that predictions for current and future must be in a binary (presence and absence) format. Finally, give the name where the turnover summaries will be stored.

In the stored database, 10 columns are created.

The first four columns are relative numbers: Disa represents the number of species predicted to disappear from the given pixel. Stable0 is the number of species which are currently not in the given pixel and not predicted to migrate. Stable1 represents the number of species currently occurring in

the given pixel, and predicted to remains into the future. Gain represent the number of species which are currently absent but predicted to migrate in the given pixel.

PercLoss, PercGain and Turnover are the related percentage estimated as the following:
- PercLoss = 100 x L/(SR)
- PercGain = 100 x G/(SR)
- Turnover = 100 x (L+G)/(SR+G)
Where SR is the current species richness.

CurrentSR represent the current modelled species richness in the given pixel.
FutureSR0Disp represents the future modelled species richness assuming no migration of species
FutureSR1Disp represents the future modelled species richness assuming migration (depending on the datasets given in input, if Migration has been used or not).

Biomod.Turnover(CurrentPred = PredBestModelByRoc.Bin, FutureProj = Future1.Migration, Turnover.Save= "Turnover.2050")
Turnover.2050[740:760,]

0.11 Species Range Change

This function allows to estimate the proportion and relative number of pixels (or habitat) lost, gained and stable for the time slice considered.
The function uses two datasets. The current species distributions and the future one. Note that predictions for current and future must be in a binary (presence and absence) format. Finally, give the name where the species range change summaries will be stored.

$load("proj.Current/Total_consensus_Current_Bin")load("proj.Future1/Total_consensus_Future1_Bin")$
$Biomod.RangeSize(CurrentPred = Total_consensus_Current_Bin[,,1], FutureProj = Future1.Migration, SpChange.S$
$"SpChange.2050")$
A list of two datasets is created: Compt.By.Species and Diff.By.Pixel

Diff.By.Pixel stores useful information for each species. The species are in columns and the pixel in rows. For each species, a pixel could have four different values:

-2 if the given pixel is predicted to be lost by the species.
-1 if the given pixel is predicted to be stable for the species.
0 is the given pixel was not occupied, and will not be into the future.
1 if the given pixel was not occupied, and is predicted to be into the future.

$SpChange.2050Diff.By.Pixel[740:760,]$
This table could be easily plotted into GIS software in order to represent the pattern of change for the selected species.

Compt.By.Species stores the summary of range change for each species (by rows).
The first four columns are relative numbers: Disa represents the number of pixels predicted to be lost by the given species. Stable0 is the number of pixels which are not currently occupied by the given species and not predicted to be. Stable1 represents the number of pixels currently occupied by the given species, and predicted to remain occupied into the future. Gain represent the number of pixels which are currently not occupied by the given species but predicted to be into the future.
PercLoss, PercGain and SpeciesRangeChange are the related percentage estimating as the following:
- CurrentRangeSize represent the modelled current range size (number of pixels occupied) of the given species.
- FutureRangeSize0Disp represents the future modelled range size assuming no migration of the given species.
- FutureRangeSize1Disp represents the future modelled range size assuming migration of the given species (depending on the datasets given in input, if Migration has been used or not).

SpChange.2050Compt.By.Species

0.12 Other Functionalities

This section presents a series of functionalities that are not directly related to the functioning of BIOMOD. These are to be used on any datasets, considering that they follow the specific requirements. Thus, you do not need to run BIOMOD to use them.

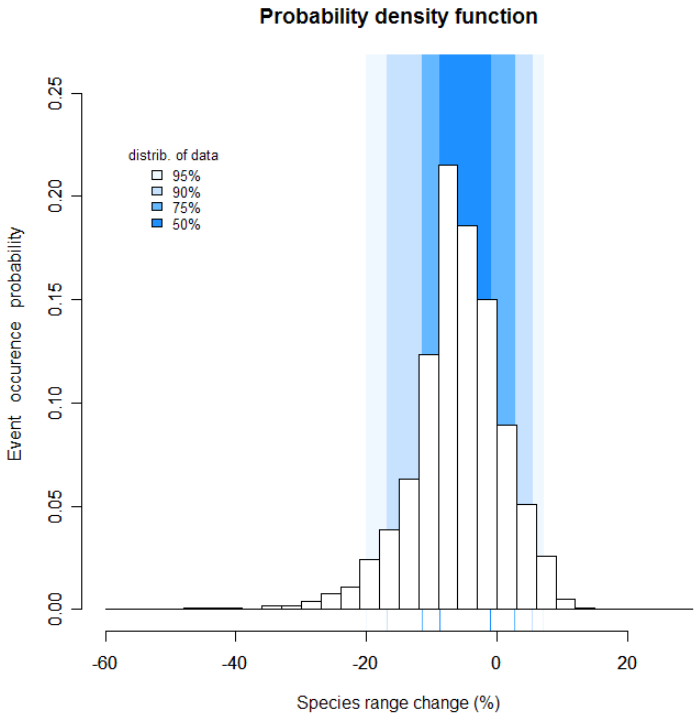
However, do NOT copy the lines presented here and try to run them in an R console. This will inevitably end up in an error message. The code here is used as an example.

0.12.1 Probability Density Function

Using a variety of parameters in modelling will inevitably bring variability in predictions, especially when it comes to making future predictions. This function enables an overall viewing of the future predictions range per species and gives the likelihood of range shift estimations.

The future range changes are calculated as a percentage of the species' present state. For example, if a species currently occupies 100 cells and is estimated by a model to cover 120 cells in the future, the range change will be + 20%.

```
> ProbDensFunc (initial=Sp.Env[,9], projections=Proj[,1:120], distrib=T, cusn=T, groups=gp, resolution=5)
```



initial: a vector in a binary format (ones and zeros) representing the current distribution of a species which will be used as a reference for the range change calculations.

projection: a matrix grouping all the predictions where each column is a single prediction. MAke sure you keep projections in the same order as the initial vector (line1=site1, line2=site2, etc.).

distrib: if true, the optimal way for condensing 50, 75, 90 and 95% of the data will be calculated and shown on the graph.

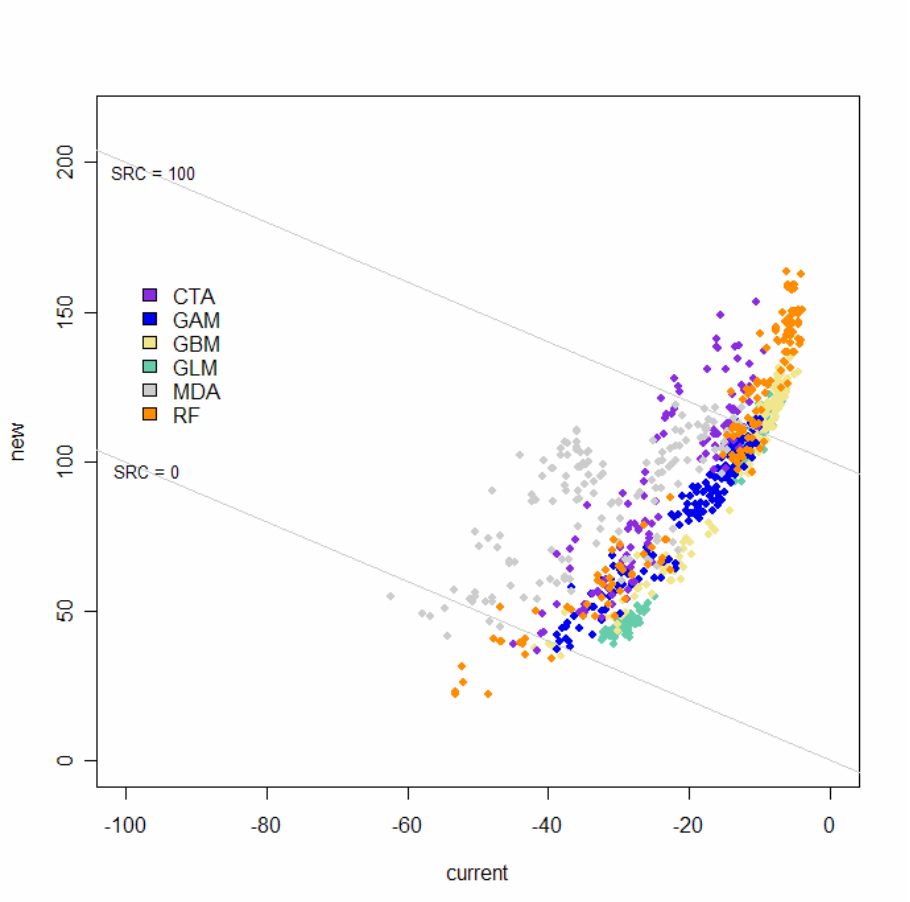
Resolution: the step used for classes of prediction in graphics. The default value is 5.

***NOTE**: modifying the resolution will directly influence the probability scale. Bigger classes will cumulate a greater number of predictions and therefore represent a greater fraction of the total predictions. The probability is in fact that of the class and not of isolated events.*

cvsn: stands for current vs new. If true, the range change calculations will be of two types: the percentage of cells currently occupied by the species to be lost, and the relative percentage of cells currently unoccupied but projected to be, namely 'new' cells, compared to current surface range.

With the example above where the species will have 120 suitable sites in the future whilst only 100 at present, this might be the result of different events. A case could be that the 100 present cells are kept and an additional 20 new sites makes the 120 cells. Another possibility is that the 100 current cells are predicted to be lost with 120 new cells, also giving 120 total cells in future.

These two cases bring the same SRC calculations results, but whilst the first case does not imply much as in survival strategies (the current populations will still be in good conditions in future, plus even having new potential territories to explore and colonise), the second case, however, implies a strong migrating effort for the populations to stay in suitable environments. Those two cases and all in-between possibilities are distinguishable with this method.



Here, each dot is a projection. For example, the one furthest on the left gives the following information: approximately -60% of the current sites will be lost and 50% of new sites will be gained. The SRC is very simply the addition of these two values : -10%. See how this single value does not reflect every thing that is going on: it does not tell that more than half of current habitats are projected to be lost, which would surely lead to different management decisions.

The two lines represent where the SRC value is 0 (no absolute change in the number of suitable sites) and +100% (the species will double its current potential distribution size). Along those line, you have all the possibilities for giving that one value (-10+10=0 ; -40+40=0 ; ...).

An extra feature on this graph is the colours. They enable to differenciate groups of projections with the present example of the models. It enables to view where the variability in projection comes from (see the description of groups below). You will have as many as these graphs as lines that you have in the groups matrix.

groups: an option for ungrouping the projections enabling a separated visualisation of the prediction range per given group. A matrix is expected where each column is a single prediction and each line is giving details of one parameter. For example, if you have 9 different projections, with 3 models and 3 threshold possibilities, your matrix could look like this:

```
     [,1]  [,2]    [,3]  [,4]  [,5]    [,6]  [,7]  [,8]    [,9]
[1,] "GAM" "GAM"   "GAM" "CTA" "CTA"   "CTA" "RF"  "RF"    "RF"
[2,] "Roc" "Kappa" "TSS" "Roc" "Kappa" "TSS" "Roc" "Kappa" "TSS"
```

or like this:

```
     [,1]  [,2]  [,3]  [,4]    [,5]    [,6]    [,7]  [,8]  [,9]
[1,] "GAM" "CTA" "RF"  "GAM"   "CTA"   "RF"    "GAM" "CTA" "RF"
[2,] "Roc" "Roc" "Roc" "Kappa" "Kappa" "Kappa" "TSS" "TSS" "TSS"
```

Do keep in mind that this matrix represents the projections the way you have put them into the projection argument. Sort your matrix the way you have sorted your projections!

Uncertainty Estimation

This function enables an assessment of the variability in predictions. The PDF plot permits a visual assessment but a calculated estimation of each parameter's role in variability is also possible.

uncertainty: if True, the variability due to each parameter entered in the groups argument will be calculated. For 3 or less parameters (i.e. 3 lines in the groups matrix) a table is given. Here is an example with 3 parameters: 9 models, 3 threshold methods, 5 future climate scenarios. The output will be in a standard R format but this presentation is just for making it easier to read.

	Roc	*Kap*	*Prev*	*Sc1*	*Sc2*	*Sc3*	*Sc4*	*Sc5*
GLM	*0.047*	*0.050*	*0.046*	*0.106*	*0.107*	*0.109*	*0.113*	*0.109*
GBM	*0.072*	*0.092*	*0.070*	*0.115*	*0.118*	*0.119*	*0.119*	*0.118*
GAM	*0.068*	*0.074*	*0.067*	*0.100*	*0.098*	*0.098*	*0.100*	*0.097*
CTA	*0.168*	*0.167*	*0.175*	*0.175*	*0.184*	*0.182*	*0.185*	*0.181*
ANN	*0.205*	*0.225*	*0.173*	*0.196*	*0.206*	*0.197*	*0.210*	*0.200*
MDA	*0.138*	*0.136*	*0.140*	*0.139*	*0.154*	*0.149*	*0.144*	*0.150*
MARS	*0.329*	*0.271*	*0.211*	*0.387*	*0.368*	*0.379*	*0.366*	*0.374*
RF	*0.108*	*0.122*	*0.095*	*0.164*	*0.172*	*0.168*	*0.169*	*0.173*
Roc	*NA*	*NA*	*NA*	*0.265*	*0.275*	*0.273*	*0.266*	*0.278*
Kap	*NA*	*NA*	*NA*	*0.234*	*0.266*	*0.249*	*0.233*	*0.263*
Prev	*NA*	*NA*	*NA*	*0.314*	*0.317*	*0.313*	*0.316*	*0.318*

.

You can identify 4 boxes: model/threshold (top left), model/scenarios (top right), threshold/scenarios(bottom right) and threshold/threshold (bottom left which contains NA values meaning 'not available').

Let's take the first value on the top left corner of the matrix. The way to read is as follows: only the projections concerning the GLM with the Roc evaluation method are taken into account. That makes 5 projections, one for each scenario. The standard deviation is measured for each line (i.e. site) of the data across these projections. The value printed in the matrix is the mean of the standard deviations across all lines. It represents the variation due to the different scenarios.

You can see that the effect of the scenarios is more or less constant considering different threshold methods (i.e. the 3 first values of each line) but is more varying across models (i.e. the first 8

values of each column). The impact of different scenarios is the strongest for the MARS, a model known for showing significant discrepancies when making projections. For this model, the threshold method seems to have an even bigger influence.

NOTE: do keep in mind that standard deviations are influenced by how many values you use for the calculation. The more you have, the bigger the chance to have a smoothing of the differences. Also, for example, using extreme future scenarios will bring greater variations than with several middle ones. Be careful when interpeting these values.

0.12.2 Pseudo-absences

The majority of models need information about presences and absences for being able to determine the suitable conditions for a given species. Some data sets, however, do not contain absences but only presences and the construction of virtual absences is therefore needed. This is, for example, the case of bird datasets where determining an absence can be rather tricky. The assumed absences are called pseudo-absences for there is no field verification of this generated information.

These pseudo-absences are created by considering any point where the species was not recorded and where the environmental conditions are known to cause potential absence. Feeding the models with exceeding numbers of absences can significantly disturb the ability of models to discriminate meaningful relationships between climate and species distributions. Moreover, running models on such heavy databases is incredibly time consuming.

In addition, some of the chosen absences might unfortunately represent true presences (this is particularlyl likely in the case of incomplete samples) and therefore the pseudo-absence data gives false information for the estimation of the species-climate relationship. Hence, we propose various strategies that seek to remove the spurious effects of using poorly seelcted pseudo-absences ***before*** *running the models.*

Use the pseudo.abs function as in the example below.

```
> pseudo.abs(coor=data[,1:2], status=data[,3], strategy='per', env=data[,4:16], distance=10000, plot=F,
    species.name= 'Sp1', acol='grey80', pcol='red', add.pres=T)
```

coor: a 2 columns matrix giving the coordinates of the points - presences and the whole set of potential absences.
status: a vector containing the presence-absence (1-0) information for the coor data. Any point for which a "1" is not given will be taken as zero by default, thus considered as an absence.
strategy: (examples on the figure below)
- random: the absences will be taken at random from the whole set of potential absences
- per: stands for the perimeter to be drawn around the presences as a whole.
- perind: same as per but the perimeter is drawn individually around each presence. For this strategy, information is needed on the distance wanted (distance argument)
- sre: sites where the environment is considered to be possibly favourable to the species (according to the SRE model) are unselected as candidate sites for drawing pseudo-absences. For this strategy,

the env argument must be given.

distance: only used for the "perind" strategy. The unit is the one of the coor data.

env: needed for the "sre" strategy. A matrix giving information on the environment as a set of variables (just like the one needed to run any model).

species.name: The output will be stored under the name given by this argument, plus the strategy chosen separated by a dot. For example, if you give "larix" in this argument and choose the sre strategy, then the output is stored in a new object named: "larix.sre".

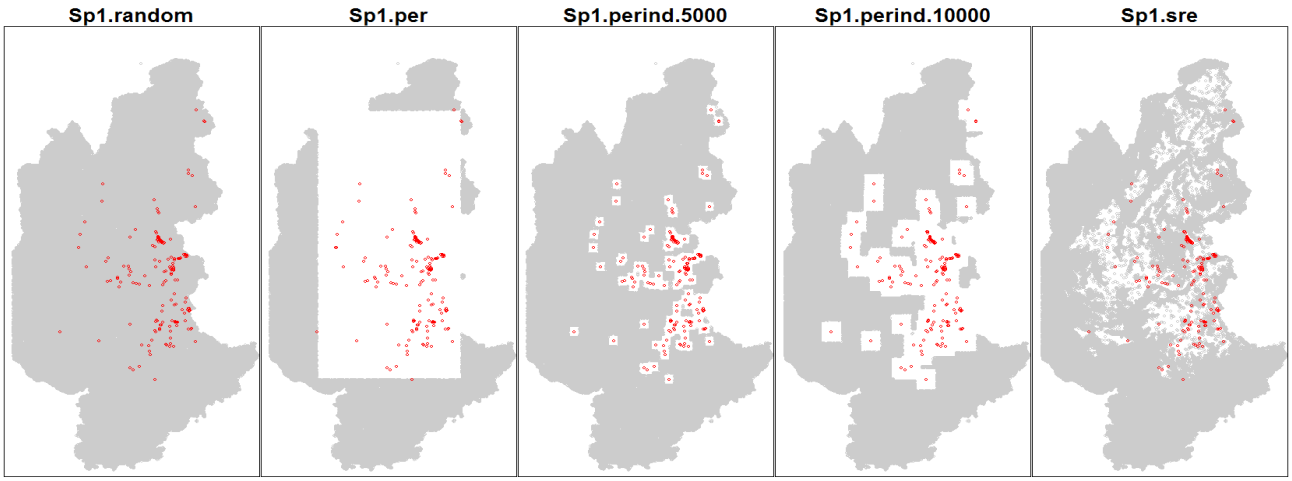
nb.points: an option for selecting only a limited number of absences at random. The default (nb.points=NULL) keeps all the possible absences according to the strategy selected.

add.pres: if True, the output will be an object also containing the presence information (see section below for further explanations).

plot: an option for plotting the outup set of presences and absences obtained.

acol and pcol: the colours wanted to plot the absences and presences respectively.

Example of the 4 available strategies in the region of the French Alps for Larix decidua miller. The presences are in red and the pseudo-absences selected by each strategy are in grey.



How to correctly use the *pseudo.abs* function output

The output of this function is an object containing the rows of the absences selected by a strategy (and presences if add.pres was set to True) from the original full presence-absence dataset. Mind that it will only contain a limited number of absences if you have used the nb.points argument. The way to use the output correctly is the following.

Let's say your original full data is stored in an object called "fulldata" and you want to use the sre strategy for selecting pseudo-absences. Run the pseudo.abs function:

```
> pseudo.abs(coor=data[,1:2], status=data[,3], strategy='sre', env=data[,4:16],
    species.name= 'first.species', add.pres=T)
```

An object called "first.species.sre" will be produced containing all the possible absences but also the presences (because I asked for it in the function call). The new data set will be called by:

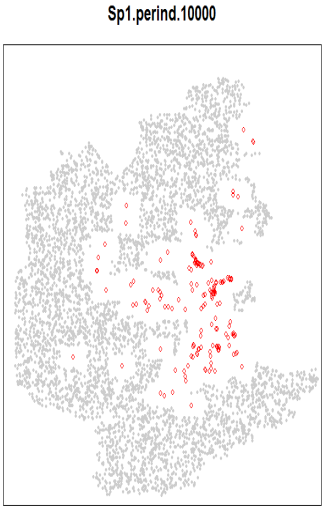
```
> new.data.set <- fulldata[first.species.sre, ]
```

The appropriate lines of the original dataset are called, building a new dataset that was here store under a new name. If you want to pick only 5,000 points from the absences strategy-selected (supposedly that you have more available) or you don't want the presences, the way to proceed is exactly the same by setting the arguments with the appropriate values.

An example :

```
> pseudo.abs(coor=data[,1:2], status=data[,3], strategy='perind', distance=10000, plot=T,
    species.name= 'Sp1', nb.points=5000, add.pres=T)
```

And your dataset will look like this.



0.13 Models' description

0.13.1 GLM - Generalised Linear Models

This provides a less restrictive form than classic multiple regressions by providing error distributions for the dependent variable other than normal and non-constant variance functions. If the response with a predictor variable is not linear, then a transformation can be included where such polynomial terms allow for the simulation of skewed and bimodal responses, -functions or hierarchical sets of models. The associated shortcoming is that the nature of the relationship between species and environmental gradients has to be known a priori. Furthermore, GLM is not always flexible enough to approximate the true regression surface adequately. To select for the most parsimonious model, BIOMOD uses an automatic stepwise model selection. The stepAIC function of Splus (library MASS) builds models by sequentially adding new terms and testing how much they improve the fit, and by dropping terms that do not degrade the fit to a significant amount. The statistical criteria used for selection of models of increasing fit could be either the Akaike Information Criterion (AIC) or the Bayesian Information Criteria (BIC). The stepwise procedure allows the removal of redundancy in variables and reduces multicolinearity (not always).

Three kinds of GLM can be run:
GLM Simple: Used only linear terms.
$Y1 = X1 + X2 + X3 + (X1 * X2) + (X2 * X3)$
GLM Quad: Used linear, 2nd and 3rd order.
$Y1 = X1 + X1^2 + X1^3 + X2^2 + X3^3$
GLM Poly: Use ordinary polynomial terms.
$Y1 = f(X1 + X1^2 + X1^3) + f(X2 + X2^2 + X2^3) +$

If you select GLM, just type GLM = T inside the function call.
If you want to use polynomial terms, type TypeGLM = "poly", or quadratics, TypeGLM = "quad", or using only linear terms, type TypeGLM = "simple" If you want to use the AIC as a selection criteria, just type Test = "AIC", or if you want to use the BIC, just type Test = "BIC".

Key reference.
McCullagh, P. and Nelder, J.A. (1989) Generalized linear models Chapman and Hall.

Key reference in ecology/biogeography.
Austin, M.P. and Meyers, J.A. (1996) Current approaches to modelling the environmental niche of eucalypts: implication for management of forest biodiversity. Forest Ecology and Management, 85, 95-106.
Elith, J., Graham, C.H., Anderson, R.P., Dudik, M., Ferrier, S., Guisan, A., Hijmans, R.J., Huettman, F., Leathwick, J.R., Lehmann, A., Li, J., Lohmann, L., Loiselle, B.A., Manion, G., Moritz, C., Nakamura, M., Nakazawa, Y., Overton, J.M., Peterson, A.T., Phillips, S., Richardson, K., Schachetti Pereira, R., Schapire, R.E., Soberón, J., Williams, S.E., Wisz, M., and Zimmermann, N.E. (2006) Novel methods improve predictions of species' distributions from occurrence data. Ecography, 29, 129-151.
Guisan, A. and Thuiller, W. (2005) Predicting species distribution: offering more than simple habitat models. Ecology Letters, 8, 993-1009.
Guisan, A. and Zimmermann, N.E. (2000) Predictive habitat distribution models in Ecology. Eco-

logical Modelling, 135, 147-186.
Thuiller, W., Araújo, M.B., and Lavorel, S. (2003) Generalized models versus classification tree analysis: a comparative study for predicting spatial distributions of plant species at different scales. Journal of Vegetation Science, 14, 669-680.

0.13.2 GAM - Generalised Additive Models

This has been recently used in ecology to deal with various species response shapes to environmental variables. GAMs are designed to capitalise on the strengths of GLMs without requiring the problematic steps of postulating a response curve shape or specific parametric response function. They use a class of equations called "smoothers" that attempt to generalise data into smooth curves by local fitting to subsections of the data. GAMs are therefore useful when the relationship between the variables are expected to be of a more complex form, not easily fitted by standard linear or non-linear models, or where there is no a priori reason for using a particular model. The idea is to 'plot' the value of the dependent variables (occurrences) along a single environmental variable, and then to calculate a smooth curve that fits the data as closely as possible while being parsimonious. The algorithm fits a smooth curve to each variable and then combines the results additively.
BIOMOD uses a cubic spline smoother, which is a collection of polynomials of degree less than or equal to 3, defined on subintervals. A separate polynomial is fitted for each neighbourhood, thus enabling the fitted curve to join all of the points. Similarly to GLM, BIOMOD uses an automated stepwise process to select the most significant variables for each species.
$Y = s(X1, 4) + s(X2, 4) + s(X3, 4)$.

The user needs to select the number of degree of freedom. By default, the value is 4. Just type Spline = 4. In order words, 4 degrees of freedom is similar to a polynomial of degree 3.

Key reference.
Hastie, T.J. and Tibshirani, R. (1990) Generalized additive models Chapman and Hall, London.

Key reference in ecology/biogeography.
Austin, M.P. and Meyers, J.A. (1996) Current approaches to modelling the environmental niche of eucalypts: implication for management of forest biodiversity. Forest Ecology and Management, 85, 95-106.
Elith, J., Graham, C.H., Anderson, R.P., Dudik, M., Ferrier, S., Guisan, A., Hijmans, R.J., Huettman, F., Leathwick, J.R., Lehmann, A., Li, J., Lohmann, L., Loiselle, B.A., Manion, G., Moritz, C., Nakamura, M., Nakazawa, Y., Overton, J.M., Peterson, A.T., Phillips, S., Richardson, K., Schachetti Pereira, R., Schapire, R.E., Soberón, J., Williams, S.E., Wisz, M., and Zimmermann, N.E. (2006) Novel methods improve predictions of species' distributions from occurrence data. Ecography, 29, 129-151.
Guisan, A. and Thuiller, W. (2005) Predicting species distribution: offering more than simple habitat models. Ecology Letters, 8, 993-1009.
Guisan, A. and Zimmermann, N.E. (2000) Predictive habitat distribution models in Ecology. Ecological Modelling, 135, 147-186.
Thuiller, W., Araújo, M.B., and Lavorel, S. (2003) Generalized models versus classification tree analysis: a comparative study for predicting spatial distributions of plant species at different scales. Journal of Vegetation Science, 14, 669-680.
Yee, T.W. and Mitchell, N.D. (1991) Generalized additive models in plant ecology. Journal of Vegetation Science, 2, 587-602.

0.13.3 CTA - Classification Tree Analysis

This provides a good alternative to regression techniques. Like GAM, they do not rely on a priori hypotheses about the relationship between independent and dependent variables. This method consists of recursive partitions of the dimensional space defined by the predictors into groups that are as homogeneous as possible in terms of response. The tree is built by repeatedly splitting the data, defined by a simple rule based on a single explanatory variable. At each split, the data are partitioned into two exclusive groups, each of which is as homogeneous as possible. The algorithm seeks to decrease the variance within the subset as much as possible. The heterogeneity of a node can be interpreted as a deviance of a Gaussian model (regression tree) or of a multinomial model (classification tree). The result is a graph representing the deviance function of the cost-complexity parameter. The best tree is a trade-off between a high decrease of deviance and the smallest number of leaves. BIOMOD uses the rpart library to run the classification tree analysis. To control the length of the tree, the program builds a nested sequence of sub-trees by recursively snipping off the less important splits in terms of explained deviance. BIOMOD uses a procedure running X-fold cross-validations to select the best trade-off between the number of leaves of the tree and the explained deviance. The user can specify the number of cross-validation required.
If you want to use classification tree analysis model, just type Tree = TRUE. Then select the number of cross-validation typing CV.tree = 10.
There is no optimal number of cross-validation. Note that high number increases the memory demand.

Key reference.
Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984) Classification and regression trees Chapman and Hall, New York.

Key reference in ecology/biogeography.
De'Ath, G. and Fabricius, K.E. (2000) Classification and regression trees: a powerful yet simple technique for ecological data analysis. Ecology, 81, 3178-3192.
Thuiller, W., Vaydera, J., Pino, J., Sabaté, S., Lavorel, S., and Gracia, C. (2003) Large-scale environmental correlates of forest tree distributions in Catalonia (NE Spain). Global Ecology and Biogeography, 12, 313-325.
Vayssières, M.P., Plant, R.E., and Allen-Diaz, B.H. (2000) Classification trees: an alternative non-parametric approach for predicting species distributions. Journal of Vegetation Science, 11, 679-694.

0.13.4 ANN - Artifical Neural Networks

Feed forward neural networks provide a flexible way to generalize linear regression functions. They are non-linear regression models but with so many parameters that they are extremely flexible; flexible enough to approximate any smooth function. The accuracy of ANN is mainly controlled by two parameters: the amount of weight decay and the number of hidden unit. BIOMOD uses the library nnet. As different runs can provide different results, the best amount of weight decay and the number of units in the hidden layer [either equals to the number of variables (see Wierenga et Kluytmans, 1994) or 75by using N-fold cross-validation (3 by default). The user can also select the number of cross-validation. Note than ANN is very time-consuming so avoid excessive number of cross-validations.

If you want to use ANN model, simply type ANN = T. Then select the number of cross-validation typing CV.ann = 3.

Key reference.

Ripley, B.D. (1996) Pattern Recognition and Neural Networks Cambridge.

Key references in ecology/biogeography

Lek, S., Delacoste, M., Baran, P., Dimopoulos, I., Lauga, J., and Aulagnier, S. (1996) Application of neural networks to modelling nonlinear relationships in ecology. Ecological Modelling, 90, 39-52.
Luoto, M. and Hjort, J. (2005) Evaluation of current statistical approaches for predictive geomorphological mapping. Geomorphology, 67, 299-315.
Moisen, G.G. and Frescino, T.S. (2002) Comparing five modelling techniques for predicting forest characteristics. Ecological Modelling, 157, 209-225.
Pearson, R.G., Dawson, T.P., Berry, P.M., and Harrison, P.A. (2002) SPECIES: A Spatial Evaluation of Climate Impact on the Envelope of Species. Ecological Modelling, 154, 289-300.
Segurado, P. and Araújo, M.B. (2004) Evaluation of methods for modelling species probabilities of occurrence. Journal of Biogeography, 31, 1555-1568.

0.13.5 MDA - Mixture Discriminant Analysis

MDA is a method for classification (supervised) based on mixture models. It is an extension of the well-known linear discriminant analysis. The mixture of normals is used to obtain a density of estimation for each class. MDA has an implementation in the library mda. Very often, a single Gaussian to model a class, as in LDA, is too restricted. MDA extends to a mixture of Gaussians. Different regression methods can be used in the optimal scaling process. R-BIOMOD used mars (see below) to increase the predictive power of the models.

Key reference.

Hastie, T., Tibshirani, R and Buja, A. (1994) Flexible Disriminant Analysis by Optimal Scoring, JASA, 1255-1270.

Hastie, T. J., Buja, A., and Tibshirani, R. (1995) Penalized Discriminant Analysis. Annals of Statistics.

Hastie, T. and Tibshirani, R. (1996) Discriminant Analysis by Gaussian Mixtures. JRSSB.

Key references in ecology/biogeography

Manel, D., Dias, J. M., Buckton, S. T. and Ormerod, S. J. (1999) Alternative methods for predicting species distribution: an illustration with Himalayan river birds. Journal of Applied Ecology. 36, 734-747.

0.13.6 MARS - Multivariate Adaptive Regression Splines

A major assumption of any linear process is that the coefficients are stable across all levels of the explanatory variables and, in the case of a time series model, across all time periods. The MARS model is a very useful method of analysis when it is suspected that the model's coefficients have different optimal values across different levels of the explanatory variables. There are many theoretical reasons consistent with this possibility occurring in many different applications including energy, finance, economics, social science, and manufacturing. The MARS approach introduced by Friedman (1991) will systematically identify and estimate a model whose coefficients differ based on the levels of the explanatory variables. The breakpoints or thresholds that define a change in a model coefficient is termed a spline knot and can be thought of similar to a piecewise regression. An advantage of the MARS approach is that the spline knots are determined automatically by the procedure. In addition, complex nonlinear interactions between variables can also be specified. The MARS procedure is particularly powerful in situations where there are large numbers of right-hand variables and low-order interaction effects. The equation switching model, in which the slope of the model suddenly changes for a given value of the X variable, is a special case of the MARS model. The MARS procedure can detect and fit models in situations where there are distinct breaks in the model, such as are found if there is a change in the underlying probability density function of the coefficients and where there are complex variable interactions.
R-BIOMOD uses the mars function from the mda library programmed by Trevor Hastie and Robert Tibshirani. MARS automatically selects the amount of smoothing required for each predictor as well as the interaction order of the predictors. It is considered a projection method where variable selection is not a concern but the maximum level of interaction needs to be determined. Taking a conservative approach, only two-level interactions are specified into R-BIOMOD (this could be changed easily)
There is no specific parameterisation to modify here. More experience user could have a look at the private functions.

Key reference.

J. Friedman, "Multivariate Additive Regression Splines". Annals of Statistics, 1991

Key references in ecology/biogeography

Elith, J., Graham, C.H., Anderson, R.P., Dudik, M., Ferrier, S., Guisan, A., Hijmans, R.J., Huettman, F., Leathwick, J.R., Lehmann, A., Li, J., Lohmann, L., Loiselle, B.A., Manion, G., Moritz, C., Nakamura, M., Nakazawa, Y., Overton, J.M., Peterson, A.T., Phillips, S., Richardson, K., Schachetti Pereira, R., Schapire, R.E., Soberón, J., Williams, S.E., Wisz, M., and Zimmermann, N.E. (2006) Novel methods improve predictions of species' distributions from occurrence data. Ecography, 29, 129-151.
Luoto, M. and Hjort, J. (2005) Evaluation of current statistical approaches for predictive geomorphological mapping. Geomorphology, 67, 299-315.
Moisen, G.G. and Frescino, T.S. (2002) Comparing five modelling techniques for predicting forest characteristics. Ecological Modelling, 157, 209-225.

0.13.7 GBM - Generalised Boosting Models (or boosting regression trees, BRT)

Explanation adapted from Greg Ridgeway

Boosting: basic explanations Whereas GLM seeks to fit the single most parsimonious model that best explains the relationship between species distribution and a set of ecological predictors, boosting methods fit a large number of relatively simple models whose predictions are then combined to give more robust estimates of the response. The algorithm used by BIOMOD is a boosted regression tree (BRT, Friedman 2001, Ridgeway 1999) where each of the individual models consists of a simple classification or regression trees, i.e. a rule based classifier that consists of recursive partitions of the dimensional space defined by the predictors into groups that are as homogeneous as possible in terms of response. The tree is built by repeatedly splitting the data, defined by a simple rule based on a single explanatory variable. At each split, the data are partitioned into two exclusive groups, each of which is as homogeneous as possible. Ordinary generalised linear models have the form: where the algorithm seeks to estimate the ?j throughout various optimisation procedures (often maximum likelihood estimation). Special cases of basis expansions like generalised additive models (GAM) have also been using the same form: where h(x) is a non parametric function (e.g. spline). These methods have so far fixed the hjs and then found ?j using standard techniques (e.g. ordinary least squares regression - OLS). Regression trees also have this form where the hjs are indicator functions indicating whether x falls into a particular "box" and ? is just the terminal node means. Regression trees do not preselect the hjs nor J, rather they are estimated iteratively through the recursive partitioning algorithm. GBM makes each hj take the form of a regression tree. They are fitted incrementally so that h1(x) is the single best tree, h2(x) is the best tree that predicts the residuals of h1(x), and so on (Friedman, et al. 2000). By this way, the BRT uses an iterative method for developing a final model progressively adding trees to the model, while re-weighting the data to emphasises cased poorly predicted by the previous trees.

In BIOMOD, the user has the possibility to set up the number of cross-validation to identify an optimal number of trees that maximises the ability of a model to make accurate predictions to new, independent sites while avoiding excessive model complexity. The user has also to define the maximum number of trees which are going to be fitted. There is no way to know a priori what is the best. Between 2000 and 5000 is a good compromise. More importantly, BRT allowed the estimation of the relative importance of each variable in the model. BIOMOD uses a permutation method, which randomly permutes each predictor variable independently, and computes the associated reduction in predictive performance.
For more details:
http://www.salford-systems.com/friedmankdd.php
www.i-pensieri.com/gregr/ ModernPrediction/L9boosting.pdf

R-BIOMOD uses the gbm library programmed by Greg Ridgeway. This package implements the generalized boosted modelling framework. This implementation closely follows Friedman's Gradient Boosting Machine (Friedman, 2001). The interaction depth and the learning rate are set-up to 4 and 0.001 respectively (but could be easily changed).

Key reference.

Friedman, J.H. (2001) Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29, 1189-1232.
Friedman, J.H., Hastie, T.J., and Tibshirani, R. (2000) Additive logistic regression: a statistical

view of boosting. Annals of Statistics, 28, 337-374.
Ridgeway, G. (1999) The state of boosting. Computing Science and Statistics, 31, 172-181.

Key references in ecology/biogeography

Elith, J., Graham, C. H., Anderson, R. P., Dudik, M., Ferrier, S., Guisan, A., Hijmans, R. J., Huettman, F., Leathwick, J. R., Lehmann, A., Li, J., Lohmann, L., Loiselle, B. A., Manion, G., Moritz, C., Nakamura, M., Nakazawa, Y., Overton, J. M., Peterson, A. T., Phillips, S., Richardson, K., Schachetti Pereira, R., Schapire, R. E., Soberón, J., Williams, S. E., Wisz, M. and Zimmermann, N. E. (2006) Novel methods improve predictions of species' distributions from occurrence data. Ecography. 29, 129-151.
Leathwick, J.R., Elith, J., Francis, M.P., Hastie, T.J., and Taylor, P. (2006) Variation in demersal fish species richness in the oceans surroundings New Zealand: an analysis using boosted regression trees. Marine Ecology Progress Series, In press.
Thuiller, W., Midgley, G.F., Rouget, M., and Cowling, R.M. (2006) Predicting patterns of plant species richness in megadiverse South Africa. Ecography, 29, 733-744

0.13.8 randomForest - Breiman and Cutler's random forest for classification and regression

The model randomForest implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. It is implemented into the "randomForest" library programmed by Andy Liaw and Matthew Wiener.
Random Forests grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).
Each tree is grown as follows:
If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree. If there are M input variables, a number $m << M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing. Each tree is grown to the largest extent possible. There is no pruning.
In the original paper on random forests, it was shown that the forest error rate depends on two things:
- The correlation between any two trees in the forest. Increasing the correlation increases the forest error rate.
- The strength of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.
Reducing m reduces both the correlation and the strength. Increasing it increases both. Somewhere in between is an "optimal" range of m - usually quite wide. Using the oob error rate (see below) a value of m in the range can quickly be found. This is the only adjustable parameter to which random forests is somewhat sensitive.

Features of Random Forests.
It runs efficiently on large data bases.
It can handle thousands of input variables without variable deletion.
It gives estimates of what variables are important in the classification.
It generates an internal unbiased estimate of the generalization error as the forest building progresses.
It has methods for balancing error in class population unbalanced data sets.
It offers an experimental method for detecting variable interactions.

How random forests work. To understand and use the various options, further information about how they are computed is useful. Most of the options depend on two data objects generated by random forests. When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample. This oob (out-of-bag) data is used to get a running unbiased estimate of the classification error as trees are added to the forest. It is also used to get estimates of variable importance.

The out-of-bag (oob) error estimate In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally, during the run, as follows: Each tree is constructed using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the kth tree. Put each case left out in the construction of the kth tree down the kth tree to get a classification. In this way, a test set classification is obtained for each case in about one-third of the trees. At the end of the run, take j to be the class that got most of the votes every time case n was oob. The proportion of times that j is not equal to the true class of n averaged over all cases is the

oob error estimate.

Variable importance In every tree grown in the forest, put down the oob cases and count the number of votes cast for the correct class. Now randomly permute the values of variable m in the oob cases and put these cases down the tree. Subtract the number of votes for the correct class in the variable-m-permuted oob data from the number of votes for the correct class in the untouched oob data. The average of this number over all trees in the forest is the raw importance score for variable m. If the values of this score from tree to tree are independent, then the standard error can be computed by a standard computation. The correlations of these scores between trees have been computed for a number of data sets and proved to be quite low, therefore we compute standard errors in the classical way, divide the raw score by its standard error to get a z-score, ands assign a significance level to the z-score assuming normality. For each case, consider all the trees for which it is oob. Subtract the percentage of votes for the correct class in the variable-m-permuted oob data from the percentage of votes for the correct class in the untouched oob data.

R-BIOMOD uses 500 trees (this could be changed easily in BIOMOD-R Private Functions 200X.XX.XX.R) and extracts the importance of each selected variable.

Key References. *Breiman, L. (2001), Random Forests, Machine Learning 45(1), 5-32. Breiman, L (2002), "Manual On Setting Up, Using, And Understanding Random Forests V3.1.*

Key References in ecology/biogeography.

Elith, J., Graham, C.H., Anderson, R.P., Dudik, M., Ferrier, S., Guisan, A., Hijmans, R.J., Huettman, F., Leathwick, J.R., Lehmann, A., Li, J., Lohmann, L., Loiselle, B.A., Manion, G., Moritz, C., Nakamura, M., Nakazawa, Y., Overton, J.M., Peterson, A.T., Phillips, S., Richardson, K., Schachetti Pereira, R., Schapire, R.E., Soberón, J., Williams, S.E., Wisz, M., and Zimmermann, N.E. (2006) Novel methods improve predictions of species' distributions from occurrence data. Ecography, 29, 129-151.

Prasad, A.M., Iverson, L.R., and Liaw, A. (2006) Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems, 9, 181-199.

0.13.9 SRE - Surface Range Envelops

This is a simple surface range envelop, similar to BioClim. The envelop is defined by identifying maximum and minimum values for each input variable from the set of sites containing an observed species' presence. Any site with all variables falling between these maximum and minimum limits is included within the range. This is the simplest method to model the distribution of species or biomes. The Perc025 and Perc05 allow specifying a broad percentile range (2.5-97.5 % or 5-95 % respectively) based on the chosen predictors. It allows removing the extreme presence (those who are close to be outside the envelop) which might be considered as outliers.

Key reference.
Busby JR (1991) BIOCLIM - a bioclimate analysis and prediction system. In: Margules CR, Austin MP, editors. Nature Conservation: Cost Effective Biological Surveys and Data Analysis. Canberra, Australia: CSIRO. pp. 64-68.

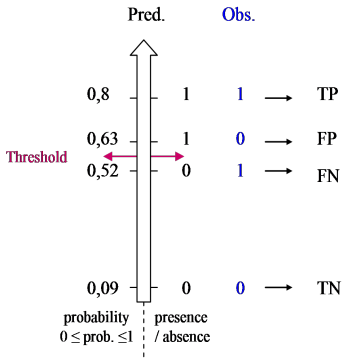
Key References in ecology/biogeography.
Beaumont LJ and Hughes L (2002) Potential changes in the distribution of latitudinally restricted Australian butterfly species in response to climate change. Global Change Biology 8:954-971.

0.14 Predictive performance description

BIOMOD proposes three different evaluation procedures, namely the ROC curve, the True Skill Statistic and the Kappa statistic. Any of them can be used independently but it is advisable to run them all for cross-comparisons.

The accuracy of statistical models is often assessed by studying the agreement between observation and prediction using a confusion matrix (see below). Four fractions can be deduced from this matrix.
- sensitivity (true positive fraction).
- specificity (true negative fraction).
- false positive fraction.
- false negative fraction.

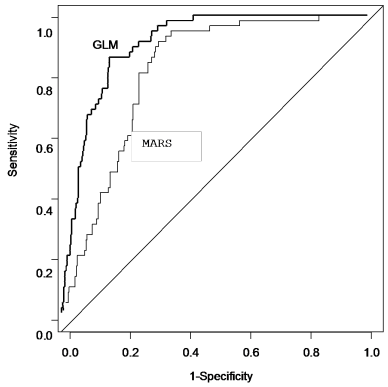
Sensitivity can be described as the ratio of positive sites (presence) correctly predicted over the number of positive sites in the sample. Specificity is the ratio of negatives sites (absence) correctly predicted over the number of negative sites in the sample. False positive and false negative fractions equal 1-specificity and 1-sensitivity respectively. To generate such a matrix and because a very large fraction of the existing models produce predictions as a probability of presence, a probability threshold must be decided to differentiate between a site (or cell) predicted to be occupied and a site (or cell) predicted to be unoccupied.



		Predicted		
		Absence	Presence	Total
Observed	Absence	TN	FP	TN + FP
	Presence	FN	TP	FN + TP
	Total	TN + FN	FP + TP	

BIOMOD is able to compute three different approaches.

Relative Operating Characteristic curve (ROC curve): This is not dependent on the threshold. The ROC curve is a graphical method representing the relationship between the False Positive fraction (1-specificity) and the sensitivity for a range of thresholds. If all predictions were possibly expected by chance, the relation would be a 45ř line. Good model performance is characterised by a curve that maximises sensitivity for low values of (1-specificity), i.e. when the curve passes close to the upper left corner of the plot. The area between the 45ř line and the curve measures discrimination, that is, the ability of the model to correctly classify a species as present or absent in a given plot. This measure is therefore called the area under the curve (AUC). In the example below, the GLM will show a better score than the MARS and is expected to be more reliable.



Cohen's Kappa statistic: This measure expresses the agreement not obtained randomly between two qualitative variables (of which a binary variable is a particular case). Kappa is based on the misclassification matrix which necessitates the calculation of a probability threshold. To do that, BIOMOD calculated Kappa for all thresholds between zero to one. The greatest value was kept as the best Kappa value. This measure expresses the best possible agreement.

The Hanssen-Kuiper Skill Score (KSS) or True Skill Statistic (TSS): This statistic, traditionally used for assessing the accuracy of weather forecasts compares the number of correct forecasts, minus those attributable to random guessing, to that of a hypothetical set of perfect forecasts.
For a 2x2 confusion matrix TSS is defined as:
TSS = sensitivity + specificity - 1

Like kappa, TSS takes into account both omission and commission errors, and success as a result of random guessing, and ranges from -1 to +1, where +1 indicates perfect agreement and values of zero or less indicate a performance no better than random. However, in contrast to kappa, TSS is not affected by prevalence. It can also be seen that TSS is not affected by the size of the validation set, and that two methods of equal performance have equal TSS scores. TSS is a special case of kappa, given that the proportions of presences and absences in the validation set are equal.

Index for classifying model prediction accuracy.

Accuracy	AUC	Kappa/TSS
Excellent or high	0.9 – 1	0.8 – 1
Good	0.8 – 0.9	0.6 – 0.8
Fair	0.7 – 0.8	0.4 – 0.6
Poor	0.6 – 0.7	0.2 – 0.4
Fail or null	0.5 – 0.6	0 – 0.2