Presentation Manual for BIOMOD

Wilfried Thuiller
Bruno Lafourcade, Miguel Araujo

September 16, 2010

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.

What purpose was BIOMOD designed for
BIOMOD was originally created as a platform to gather various existing modelling techniques with this simple question : why stick to a specific technic and on what criteria when several assessed ones exist.

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) and should be installed 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://Program Files//R//R-2.8.0//library. 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.

0.2.1 Biomod Contents

BIOMOD is composed of a series of functions that enables to do species modelling :

running BIOMOD
Initial.State
Models
Projection
Ensemble.Forecasting

further BIOMOD steps
CurrentPred
PredictionBestModel
ProjectionBestModel
Biomod.Turnover
Biomod.RangeSize
Migration

plotting functions
level.plot
multiple.plot
response.plot

ProbDensFunc calculates density probabilities
pseudo.abs generating pseudo-absences
BiomodManual opens the pdf manual from R

0.3 Models

BIOMOD attempts to span the different approaches that can be used in habitat suitability modelling. It does not aim to be exhaustive but it aims to present the most commonly used modelling approaches and the ones considered to be the most interesting and robust and which are implemented in R.

With the rise of new powerful statistical techniques, the development of habitat suitability models has rapidly increased in ecology Guisan & Thuiller 2005; Araújo & Guisan 2006; Elith & Graham 2009. Such models are static and probabilistic in nature, since they statistically relate the distribution of population, species, communities or biodiversity to their contemporary environment. A wide array of models has been developed to cover research aspects as diverse as macroecology, biogeography, conservation biology, climate change, functional ecology and habitat or species management.

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 (RF)
- Flexible Discriminant Analysis (FDA)
- 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.

0.3.1 GLM - Generalised Linear Models

- 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.

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. 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.

0.3.2 GAM - Generalised Additive Models

- 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).

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.

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

- 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.

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.

0.3.4 CTA - Classification Tree Analysis

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

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.

0.3.5 ANN - Artifical Neural Networks

- 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.

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 75% of the number of variables (Venugopal et Baets, 1994)] is selected by 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.

0.3.6 FDA - Flexible Discriminant Analysis

- FDA = T: Run a flexible discriminant analysis using the MARS function for the regression part of the model.

FDA 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. FDA has an implementation in the library mda. Very often, a single Gaussian to model a class, as in LDA, is too restricted. FDA 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.

0.3.7 MARS - Multivariate Adaptive Regression Splines

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

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.

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

- RF = T: Run a random forest model.

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 can be changed directly in the Biomod.Models function) 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.

0.3.9 SRE - Surface Range Envelops

- SRE = T, quant=0.025: Run an rectilinear surface range envelop (=BIOCLIM) using the percentile 0.025 or 0.05 as recommended by Nix or Busby (but any value will do).

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 quant argument allows specifying a broad percentile range (2.5-97.5 % for the 0.025 default value) 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.

In contrary with all other algorithm present in BIOMOD, there is no model produced. Note also that there is no ROC evaluation available, since SRE does not provide probability values but directly the presence-absence prediction of the species.

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.

0.4 The calibration procedure

The next key issue in modelling is the calibration procedure of the models with the constant effort to obtain a reliable estimation of their performance.

Ideally, one should always evaluate the predictive performance of a model using independent data, i.e. data from which the model didn't obtain any information to build itself. this would enable to reliably test its predictive accuracy on a new dataset and certify its efficiency. Unfortunately, this kind of information is rarely accessible in species distribution modelling. An alternative to assess the predictive performance of the models is to split the original data in calibration (training) and evaluation (testing) datasets : one part is used to feed the model, the other, kept aside and therefore new to the model, is used to check the models' efficiency to predict the right value. As a consequence, this method consists of a trade-off between the amount of data used for the construction of the model and the accuracy of the evaluation measure.

0.4.1 Repetitions

This splitting procedure, widely used in the modelling world, nevertheless brings a major issue : the subsequent randomness of the data selection used for calibration and its impact on the modelling quality.

To obtain a reliable way of evaluating the models while not influencing the prediction making by the random splitting of the data, BIOMOD proposes to built a series of models. The above calibration/evaluation procedure is repeated a certain number of times to perform a reliable evaluation as an attempt to free ourselves from the random effect (the mean result is extracted). Then a final model is built without splitting the data, i.e. 100 % of the data available is used, thus using all the information available and not having any random effect in the prediction making.

This method is also a good way of assessing for uncertainty. While many modellers are satisfied with running only their models once, we propose to build a large number of models to measure the sensitivity of the models to the initial conditions (the input data given). Each model built is kept and can be used to later render projections.

pros : 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.

cons : it lengthens the modelling time needed to build the models (it can be an exceeding amount of time if not done carefully).

main interest : adds variability in the predictions when several runs are made due to the random effect of selecting the data, i.e. each model is not build using the same data, representing the sensibility of the models on the input data.

The combination of the two arguments below will determine in which way the models will be built and tested.

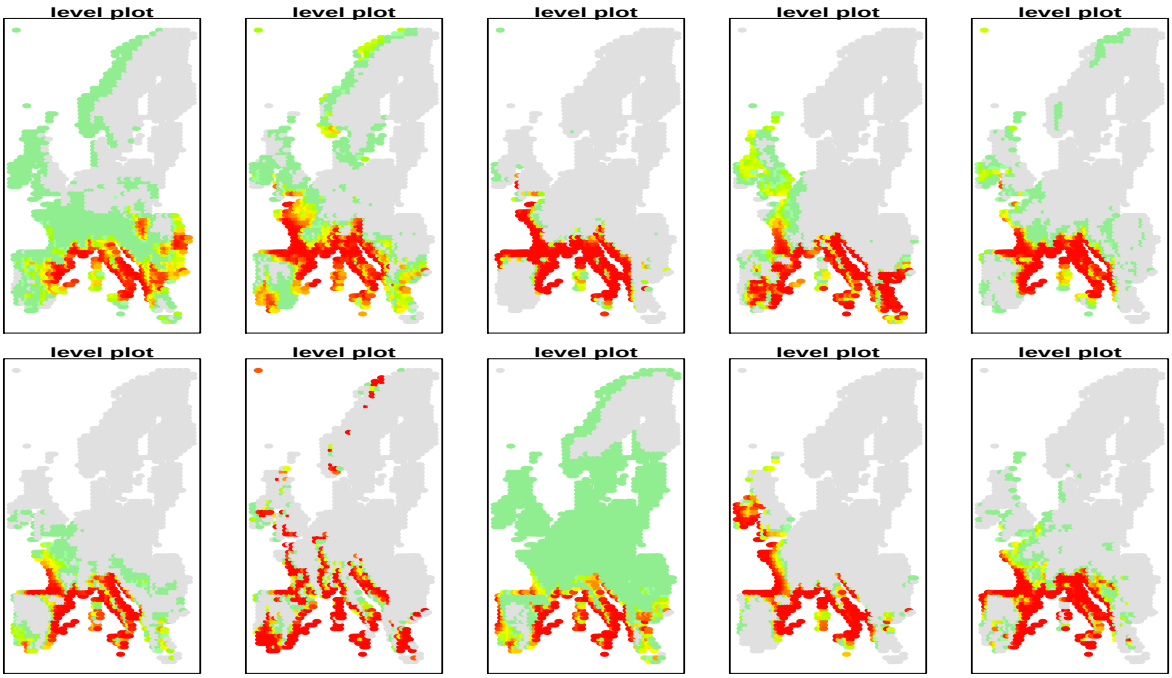
- NbRunEval: number of random data splitting procedure for creating calibration and evaluation datasets ; a model will be built from each one of them. If set to zero, only the final 100 % model is built.

- DataSplit: the ratio used for splitting the original database in calibration and evaluation subsets (value to give is the % awarded for calibration). A 70/30 % partitining is recommended as commonly used (Araújo, et al. 2005b, Guisan and Thuiller 2005).

Example with the fda and species Sp281

Here is an example of the effect of randomness in the prediction making (note that here the prevalence isn't kept, the relative number of presences and absences will vary for each model)

```
> #to call our dataset
> library(BIOMOD)



Design library by Frank E Harrell Jr

Type library(help='Design'), ?DesignOverview, or ?Design.Overview')
to see overall documentation.

Loaded gbm 1.6-3



> data(Sp.Env)
> data(CoorXY)
> store <- matrix(nr=2264, nc=0)
> for(i in 1:10){
 rand <- sample(2264, 100)
 model <- fda("Sp281 ~Var1 + Var2 + Var3 + Var4 + Var5 + Var6 + Var7", data=Sp.Env[rand,], method=mars)
 store <- cbind(store, predict(model, Sp.Env[,4:10], type="post")[,2])
 }



> for(i in 1:10){
 x11()
 par(mar=c(1,1,1,1))
 level.plot(store[,i], CoorXY)
 }



> par(mfrow=c(2,5))
> par(mar=c(1,1,1,1))
> for(i in 1:10) level.plot(store[,i], CoorXY, show.scale=F, cex=0.85)
```



This is the same model (FDA) and the same datasets used, only the initial calibration data is changing. The impact on the geographical patterns can clearly be seen.

NOTE : Another issue that has shown an influence on the prediction is the prevalence of the data, i.e. the ratio between the total number of presences and the number of absences. In all procedures, BIOMOD ensures that the prevalence of the original data is conserved in the calibration and evaluation datasets.

0.4.2 Pseudo-absences

All the models in BIOMOD need information about presences and absences for being able to determine the suitable conditions for a given species. Some datasets, 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 particularly likely in the case of incomplete samples) and therefore the pseudo-absence data gives

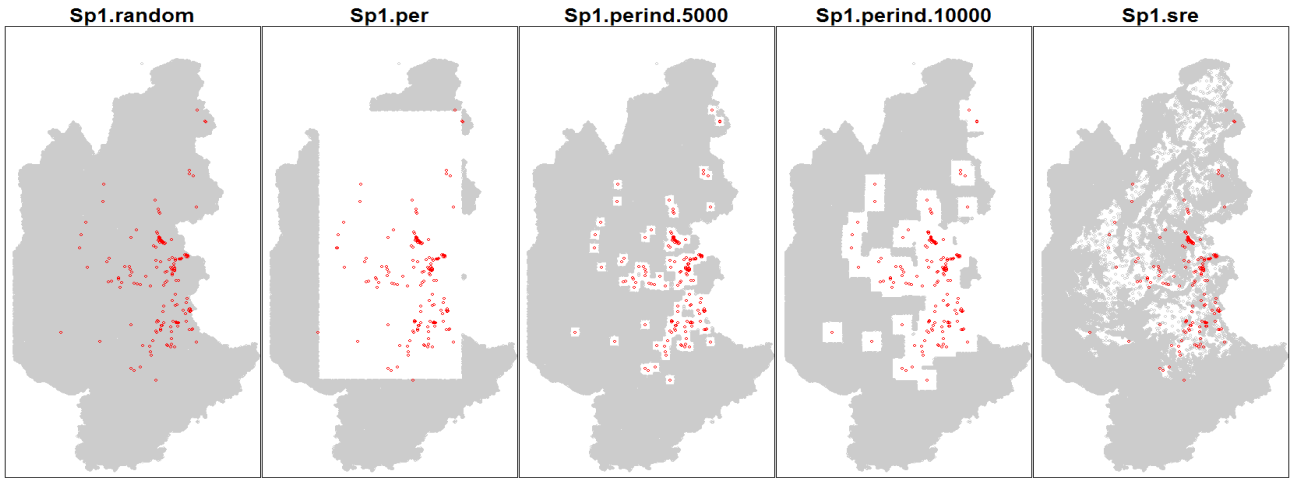
false information for the estimation of the species-climate relationship. Hence, we propose various strategies that seek to remove the spurious effects of using poorly selected pseudo-absences before running the models.

You can use the function manually or choose to run it within the *Models()* function. the *pseudo.abs* function as in the example below.

```
> #use it individually
> 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)
> #or in Models()
> Models(...,
        NbRepPA=2, strategy="circles", coor=CoorXY, distance=2, nb.absences=1000)
```

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)

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.



- 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).

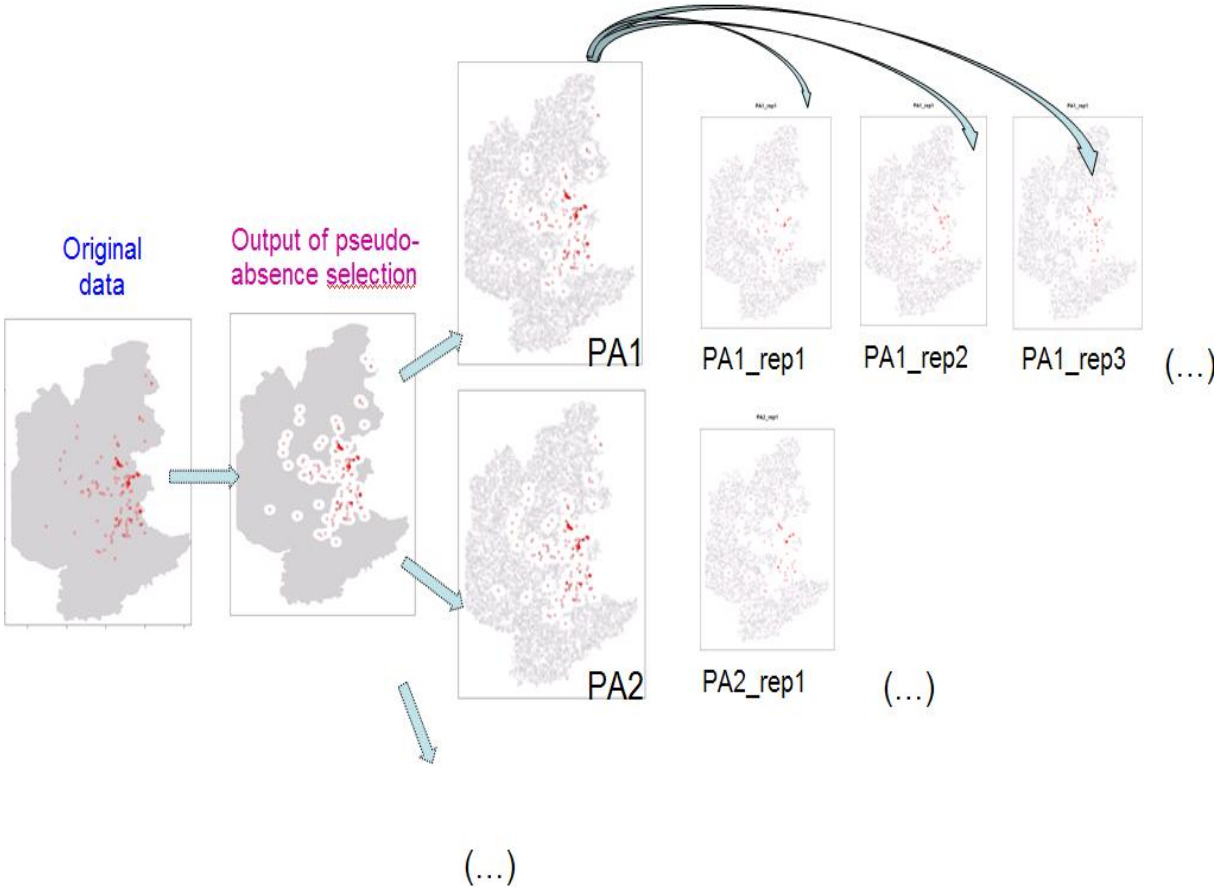
plot: an option for plotting the outup set of presences and absences obtained.

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

Usage inside the Models() function

There are less arguments to inform as some information is already known by BIOMOD (status, env, species.name) and others useless (plotting arguments). There is nevertheless a new argument *NbRepPA*.

This argument is to be correlated with the usage of repetitions for the calibration : once the pool of potential pseudo-absences has benn definied by the strategy selected, a user-definied number (Nb.absences argument) is randomly selected from this pool. We therefore have a random effect in the calibration process coming from the creation of pseudo-absences for our data. The NbRepPA argument will define a number of repetitions for randomly withdrawing absences to constitute the calibration datasets. Do consider that the total number of repetitions will be a multiplication of the two repetion arguments :



Manual usage : 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, ]
```

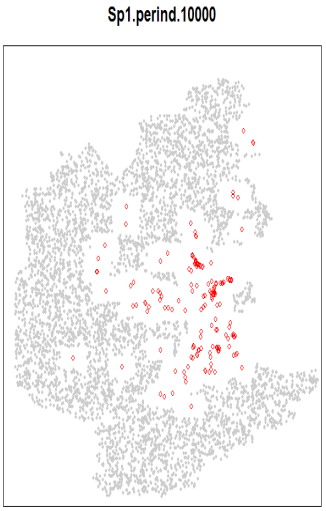
The appropriate lines of the original dataset are called, building a new dataset that was here store under a new name. If you want to pick only 5,000 points from the absences strategy-selected

(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.



0.4.3 Weights

The Yweights arguments enables the user to set extra information 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.

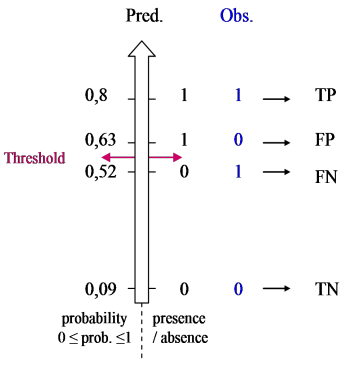
0.5 Evaluation of the predictive performance

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

- 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, it is possible to find the optimal threshold maximising 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.

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.

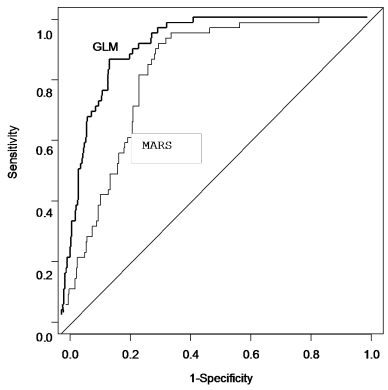
Sensitivity can be described as the ratio of positive sites (presence) correctly predicted over the number of positive sites in the sample. Specificity is the ratio of negatives sites (absence) correctly predicted over the number of negative sites in the sample. False positive and false negative fractions 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	

0.5.1 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.



0.5.2 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.

0.5.3 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.

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

Index for classifying model prediction accuracy.

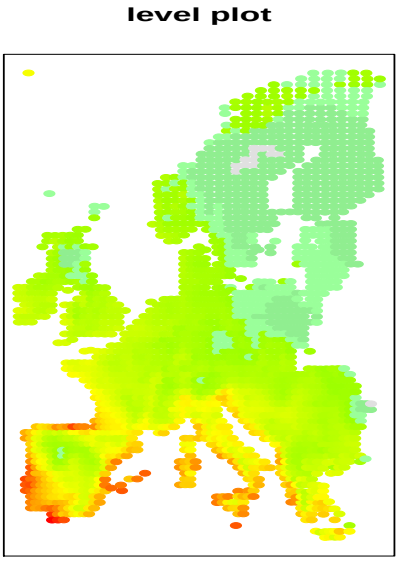
0.5.4 Importance of each variable

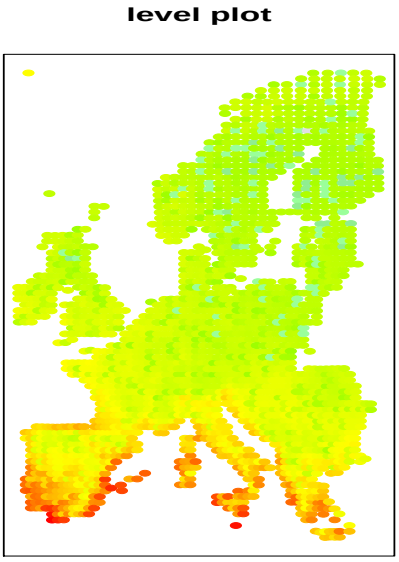
It is always difficult to compare predictions from different models as they do not rely on the same algorithms, techniques and assumptions about the expected relationship between the reponse and the variables, i.e. the species distributions and the environment. With a permutation procedure, BIOMOD proposes another way to examine the importance of the variables in the models. We extract a measure of relative importance of each variable that is independent of the model. Note that the importance of the variables is only calculated for the final model.

Procedure: 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 :

```
> model <- glm(Sp281 ~ Var1 + Var2 + Var3 + Var4 + Var5 + Var6 + Var7, data=Sp.Env)
> Pred <- predict(model, Sp.Env[,4:10], type="response")
```

```
> Sp.Env2 <- Sp.Env
> Sp.Env2[,4] <- sample(Sp.Env[,4])
> Pred2 <- predict(model, Sp.Env2[,4:10], type="response")
> par(mfrow=c(1,2))
> level.plot(Pred, CoorXY, show.scale=F, cex=0.8)
> level.plot(Pred2, CoorXY, show.scale=F, cex=0.8)
```

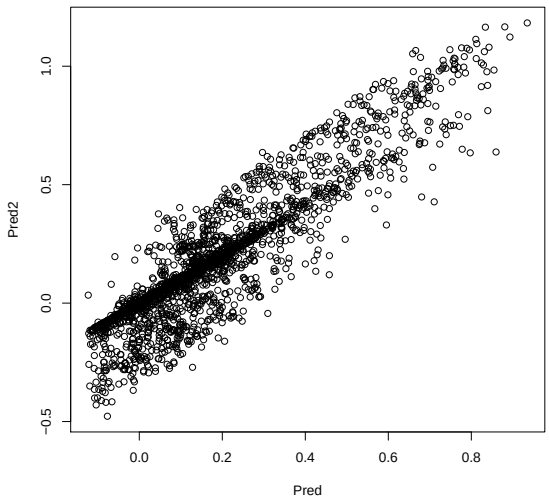





```
> cor(Pred, Pred2)
```

```
[1] 0.9124
```

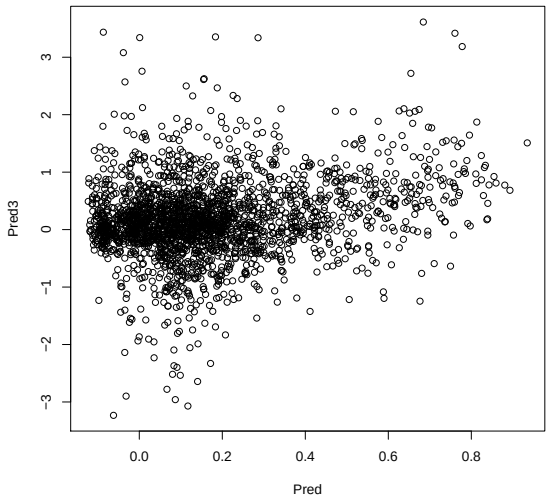
```
> plot(Pred, Pred2)
```



A good correlation score between the two predictions, i.e. they only slightly differ, shows that the randomized variable has little influence on the prediction making and is considered not important for the model in its prediction.

```
> Sp.Env2 <- Sp.Env
> Sp.Env2[,6] <- sample(Sp.Env[,6])
> Pred3 <- predict(model, Sp.Env2[,4:10], type="response")
> plot(Pred, Pred3)
> cor(Pred, Pred3)
```

```
[1] 0.186
```



In contrary, a low correlation means a significant difference in the prediction making, showing an importance of that variable for the model.

NOTE : in the *VarImportance* output, the values given correspond to 1 minus the correlation score. High values will therefore reveal a high importance of the variable whereas a value close to 0 will reveal no importance.

Score of variable 1 (Pred2) : 1 - cor(Pred, Pred2) = 0,09 meaning low influence
Score of variable 2 (Pred3) : 1 - cor(Pred, Pred3) = 0,77 meaning high influence

This step is repeated n times for each variable independently and the means are kept for each variable.

NOTE : The obtained 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.

0.6 Assessment of uncertainty and Models' optimisation

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 *PredBestModelByX* (with X 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*.

0.7 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.

The function returns a list that is also stored in R's memory. In our case, it will be called *consensus_Future1_results*. It contains all the computational information that has been used to render the ensemble forecasts, for example predictive performance of each method when applied to current predictions (if Test = True), the weights awarded to the models in the weighting process, the model selected by the PCA.median method (if set to True). The forecasts themselves are stored on the hard disk directly in the corresponding folder.

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	FDA
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 dataset 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.

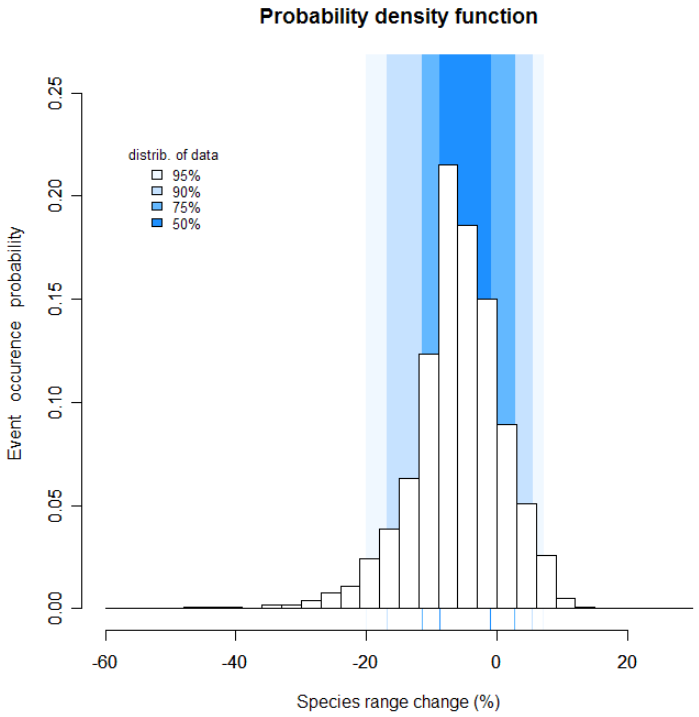
Test: This option will test the efficiency of the consensus method on the data given for calibration. A Roc evaluation is run and the score will be given in the output of the function as the "test.results".

0.8 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, cvsn=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.

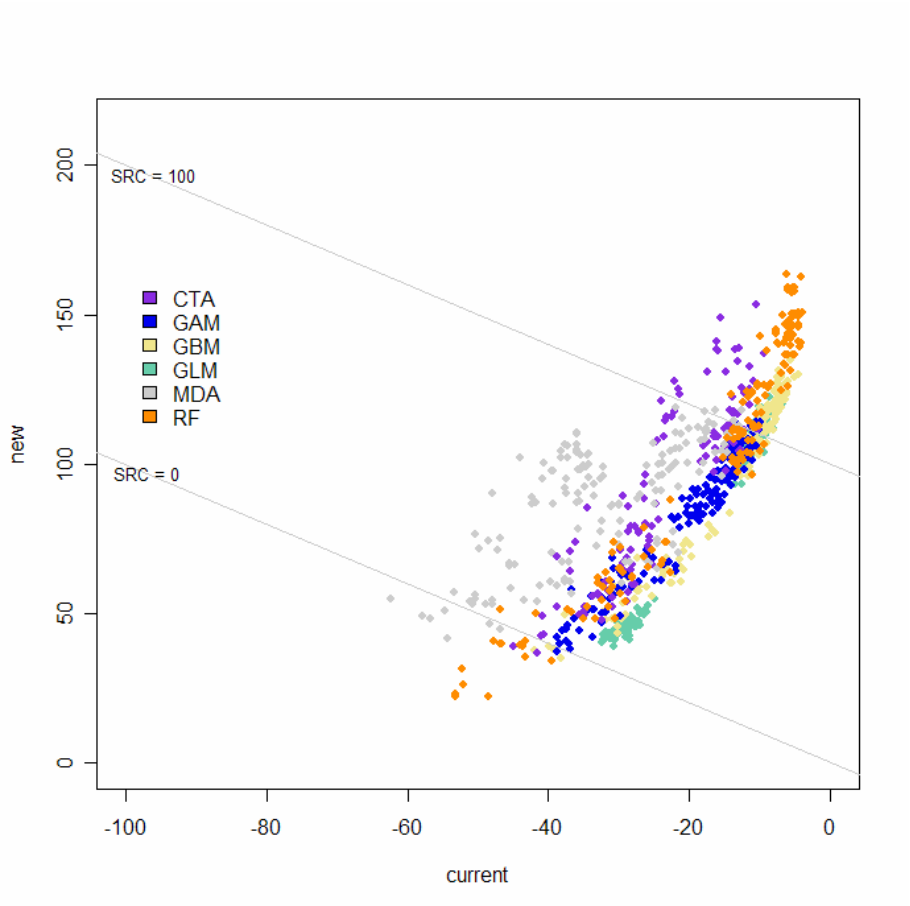
Resolution: the step used for classes of prediction in graphics. The default value is 5.

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!

An example with repetitions

The help file of the ProbDensFunc function provides a full example. It is done with 20 repetitions for half of the models to assess the variability in prediction making when the calibration of the model is done on partial data. Only Sp163 is done. Please look in details the help file for an example of the data preparation you should go through to run the function properly.

```
> example(ProbDensFunc)
```

As you will see on your own R session, it produces a series of plots that represents the variability in the projections obtained.

0.9 Glossary

- AIC = Akaike Information Criterion
- ANN = Artificial Neural Network
- AUC = Area Under the Curve (or Area Under the ROC Curve)
- BIC = Bayesian Information Criterion
- CTA = Classification and regression Tree Analysis
- GAM = Generalized Additive Model
- GBM = Generalized Boosting Model
- GCM = Global Change Model
- GLM = Generalized Linear Model
- PDF = Probability Density Function
- ROC = Receiver Operater Characteristics
- RF = Random Forest
- SRC = Species Range Change
- SRE = Surface Range Enveloppe
- TSS = True Skill Statistics