An example of species distribution modelling with `biomod2`

`biomod2` version : 1.0
R version 2.15.1 (2012-06-22)

Damien Georges & Wilfried Thuiller

July 5, 2012

Contents

1 Introduction

This vignette illustrates how to build, evaluate and project a single species distribution model using `biomod2` package. The three main modeling steps, described bellow, are the following :

1. formatting the data
2. computing the models
3. making the projections

The example is deliberately simple (few technicals explanations) to make sure it is easy to transpose to your own data relatively simply.

NOTE 1 :
Several other vignettes will be written soon to help you to go through `biomod2` details and subtleties

2 Formatting the data

In this vignette, we will work (because it is the most common case) with :

- only presences data that we will be extracted from a raster
- environmental raster layers (e.g. Worldclim)

Let's import our data.

R input

```
# load the library
library(biomod2)
# load our species raster
# we consider only the presences of Myocastor coypus species
myResp.ras <- raster( system.file(
                        "external/species/Myocastor_coypus.img",
                        package="biomod2") )
# extract the presences data

# the name
myRespName <- 'Myocastor'
# the XY coordinates of the presence
myRespXY <- xyFromCell(object=myResp.ras,
                       cell=which(myResp.ras[]>0))
# and the presence data
myResp <- extract(x=myResp.ras, y=myRespXY)
# load the environmental raster layers (could be .img, ArcGIS rasters or any supported form

# Environmental variables extracted from Worldclim (bio_3, bio_4,
# bio_7, bio_11 & bio_12)
myExpl = stack( system.file( "external/climat/current/bio3.grd",
                             package="biomod2"),
                system.file( "external/climat/current/bio4.grd",
                             package="biomod2"),
                system.file( "external/climat/current/bio7.grd",
                             package="biomod2"),
                system.file( "external/climat/current/bio11.grd",
                             package="biomod2"),
                system.file( "external/climat/current/bio12.grd",
                             package="biomod2"))
```

NOTE 2 :
You may have community or atlas data for which you have both presence and absence. In this case extract the presences and the absences points and code them by 0/1.

NOTE 3 :

If your environmental data are in matrix/data.frame format, you have to give a species as vector (or a one column Spatial.points.data.frame) having a length that match with the number of rows of your environmental data. That implies to add NA's in all points where you do not have information on species presence/absence.

When your data are correctly loaded, you have to transform them in an appropriate `biomod2` format. This is done using `BIOMOD_FormatingData`. As all models need both presences and absences to run, you may need to add some pseudo-absences (or background data) to your data. That is necessary in the case of presence-only, and may be useful in the case of insufficient absence data. 3 algorithms are now implemented to extract a range of pseudo-absence data: 'random', 'SRE' and 'disk'.
Here, we will create two sets of pseudo-absence data using the random algorithm.

NOTE 4 :
If you have both presence-absence data and a large number of presence (not the case here), it's strongly recommended to split your data.frame into two pieces and to keep a part for evaluating all your models on the same data.set (i.e. eval.xxx args)

NOTE 5 :
The PA.nb.absences arg represents the total number of pseudo-absence extracted for each set of extraction (true absences + selected PA). It must be then higher than the number of true absences (if any). If not, no pseudo-absences are selected.

R input

```
myBiomodData <- BIOMOD_FormatingData(resp.var = myResp,
                                     expl.var = myExpl,
                                     resp.xy = myRespXY,
                                     resp.name = myRespName,
                                     PA.nb.rep = 2,
                                     PA.nb.absences = 200,
                                     PA.strategy = 'random')
```

R output

```
    ! No data has been set aside for modeling evaluation
  > Pseudo Absences Selection checkings...
  > random pseudo absences selection
  > Pseudo absences are selected in explanatory variables
```

At this point, check whether the data are correctly formatted by printing and plotting the created object.

R input

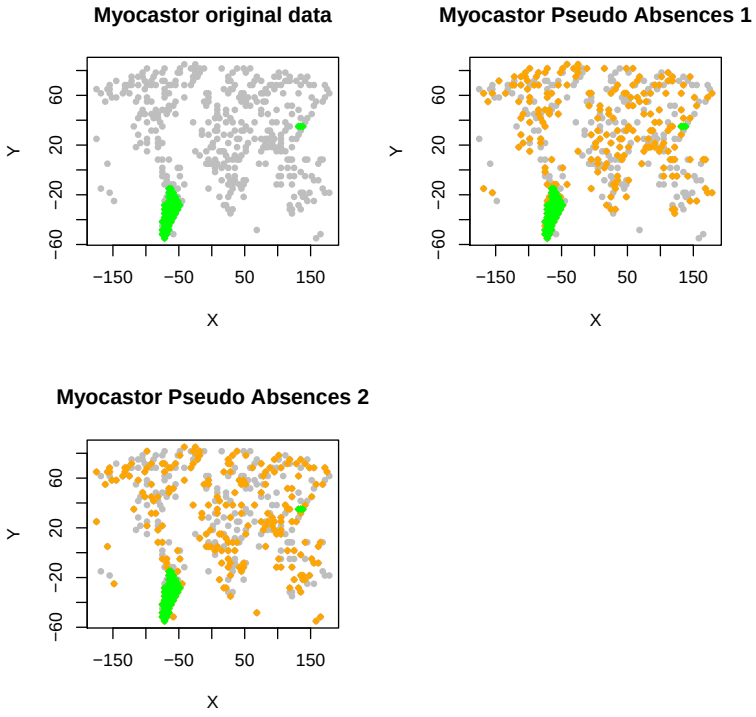
```
myBiomodData
```

R output

```
-=-=-=- 'BIOMOD.formated.data.PA' -=-=-=-
sp.name =  Myocastor
         59 presences,  0 true absences and  383 undifined points in dataset

         5 explanatory variables
     bio_3           bio_4           bio_7
 Min.   : 9.95   Min.   :  103   Min.   : 54.4
 1st Qu.:21.35   1st Qu.: 2368   1st Qu.:180.8
 Median :41.93   Median : 5386   Median :278.1
 Mean   :42.09   Mean   : 6624   Mean   :288.8
 3rd Qu.:56.00   3rd Qu.:10811   3rd Qu.:393.5
 Max.   :91.43   Max.   :21774   Max.   :710.3
     bio_11           bio_12
 Min.   :-426.7   Min.   :   4
 1st Qu.:-151.9   1st Qu.: 259
 Median :  63.9   Median : 620
 Mean   :  15.7   Mean   : 938
 3rd Qu.: 200.4   3rd Qu.:1319
 Max.   : 275.0   Max.   :5133


 2 Pseudo Absences dataset available ( PA1 PA2 ) with  200 absences in each (true abs + pseu
-=-=-=--=-=-=--=-=-=--=-=-=--=-=-=-
```

R input

```
plot(myBiomodData)
```



The colors for this plot match with...

- Presences
- Absences
- Pseudo Absences
- Remaining Backgroud

3 Modeling

3.1 Building models

This step may be considered as the core of the modeling procedure within `biomod2`. Here you have to choose between 10 different algorithms ('GLM', 'GBM', 'GAM', 'CTA', 'ANN', 'SRE', 'FDA', 'MARS', 'RF', 'MAXENT'). Before running the models, you can customize their set of parameters and options using `BIOMOD_ModelingOptions`. The created object is then given to `BIOMOD_Modeling` in the next step. For the sake of simplicity, we keep all default options.

NOTE 6 :

A vignette on models' parametrization will be available soon

R input
```
# 2. Defining Models Options using default options.
myBiomodOption <- BIOMOD_ModelingOptions()
```

We are now ready for running the set of models on our species. As we do not have evaluation data, we will make 1-fold cross-validation (number controlled by NbRunEval argument) of our models by randomly splitting our data set into 2 subsets : DataSplit % for calibrating and training the models and the remainder for testing them. Each model will be tested (and evaluated if any evaluation data is given) according to models.eval.meth evaluation metrics (chosen into 'KAPPA', 'TSS', 'ROC', 'FAR', 'SR', 'ACCURACY', 'BIAS', 'POD', 'CSI' and 'ETS'). To ensure our models will be comparable in term of scale, we decided to rescale them all with a binomial GLM (rescal.all.models). The VarImport argument corresponds to the number of resampling of each explanatory variable to measure the relative importance of each variable for each selected model.

NOTE 7 :
No weights are given but some will be automatically generated. Indeed, in the particular case of pseudo-absence selection, we make sure the prevalence is kept to 0.5. It means that the presence data have the same weight than the pseudo-absence data, even if a large number of the latter has been extracted.

R input
```
# 3. Computing the models

myBiomodModelOut <- BIOMOD_Modeling(
                            myBiomodData,
                            models = c('SRE','CTA','RF','MARS','FDA'),
                            models.options = myBiomodOption,
                            NbRunEval=1,
                            DataSplit=80,
                            Yweights=NULL,
                            VarImport=3,
                            models.eval.meth = c('TSS','ROC'),
                            SaveObj = TRUE,
                            rescal.all.models = TRUE)
```

R output
```
Loading required library...

Checking Models arguments...
```

```
Creating suitable Workdir...

-=-=-=- Myocastor Modeling Summary -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
5 environmental variables ( bio_3 bio_4 bio_7 bio_11 bio_12 )
Number of evaluation repetitions : 2
Models selected : SRE CTA RF MARS FDA
Total number of model runs : 20
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-


-=-=-=- Run :  Myocastor_PA1


-=-=-=-=-=-=-=- Myocastor_PA1_RUN1

Model=Surface Range Envelop
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Classification tree
         5 Fold Cross-Validation
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Breiman and Cutler's random forests for classification and regression
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Multiple Adaptive Regression Splines
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Flexible Discriminant Analysis
        Evaluating Model stuff...
        Evaluating Predictor Contributions...


-=-=-=-=-=-=-=- Myocastor_PA1_Full

Model=Surface Range Envelop
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Classification tree
         5 Fold Cross-Validation
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Breiman and Cutler's random forests for classification and regression
```

```
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Multiple Adaptive Regression Splines
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Flexible Discriminant Analysis
        Evaluating Model stuff...
        Evaluating Predictor Contributions...


-=-=-=- Run :  Myocastor_PA2


-=-=-=-=-=-=-=- Myocastor_PA2_RUN1

Model=Surface Range Envelop
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Classification tree
         5 Fold Cross-Validation
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Breiman and Cutler's random forests for classification and regression
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Multiple Adaptive Regression Splines
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Flexible Discriminant Analysis
        Evaluating Model stuff...
        Evaluating Predictor Contributions...


-=-=-=-=-=-=-=- Myocastor_PA2_Full

Model=Surface Range Envelop
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Classification tree
         5 Fold Cross-Validation
        Evaluating Model stuff...
        Evaluating Predictor Contributions...
```

```
Model=Breiman and Cutler's random forests for classification and regression
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Multiple Adaptive Regression Splines
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

Model=Flexible Discriminant Analysis
        Evaluating Model stuff...
        Evaluating Predictor Contributions...

-=-=-=- Done -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

When this step is over, have a look at some outputs :

- modeling summary

R input
```
myBiomodModelOut
```

R output
```
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

BIOMOD.models.out
Specie modelised : Myocastor
Considered variables : bio_3 bio_4 bio_7 bio_11 bio_12

Computed Models :  Myocastor_PA1_RUN1_SRE Myocastor_PA1_RUN1_CTA Myocastor_PA1_RUN1_RF

Failed Models :  none

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

- models evaluations

R input
```
# get all models evaluation
myBiomodModelEval <- getModelsEvaluations(myBiomodModelOut)
# print the dimnames of this object
dimnames(myBiomodModelEval)
```

R output
```
[[1]]
[1] "TSS" "ROC"

[[2]]
```

```
[1] "Testing.data" "Cutoff"       "Sensitivity"
[4] "Specificity"

[[3]]
[1] "SRE"  "CTA"  "RF"   "MARS" "FDA"

[[4]]
[1] "RUN1" "Full"

[[5]]
Myocastor_PA1 Myocastor_PA2
        "PA1"         "PA2"
```

R input

```
# let's print the TSS scores of Random Forest
myBiomodModelEval["TSS","Testing.data","RF",,]
```

R output

```
      PA1   PA2
RUN1 0.775 0.867
Full 1.000 1.000
```

R input

```
# let's print the ROC scores of all selected models
myBiomodModelEval["ROC","Testing.data",,,]
```

R output

```
, , PA1

      RUN1  Full
SRE  0.733 0.816
CTA  0.821 0.988
RF   0.926 1.000
MARS 0.948 0.980
FDA  0.948 0.960

, , PA2

      RUN1  Full
SRE  0.867 0.796
CTA  0.859 0.994
RF   0.982 1.000
MARS 1.000 0.979
FDA  0.992 0.975
```

R input

- Relative importance of the explanatory variables

R input

```
# print variable importances
getModelsVarImport(myBiomodModelOut)
```

R output

```
, , RUN1, PA1

       SRE   CTA    RF  MARS   FDA
Var1 0.039 0.874 0.440 0.714 0.851
Var2 0.039 0.263 0.122 0.460 0.000
Var3 0.089 0.005 0.138 0.121 0.052
Var4 0.106 0.209 0.348 0.504 0.667
Var5 0.106 0.136 0.161 0.159 0.148

, , Full, PA1

       SRE   CTA    RF  MARS   FDA
Var1 0.010 0.855 0.381 0.567 0.823
Var2 0.010 0.370 0.142 0.528 0.000
Var3 0.058 0.038 0.085 0.325 0.018
Var4 0.097 0.239 0.342 0.382 0.575
Var5 0.043 0.404 0.179 0.119 0.142

, , RUN1, PA2

       SRE   CTA    RF  MARS   FDA
Var1 0.045 0.901 0.500 0.517 0.902
Var2 0.036 0.004 0.199 0.551 0.762
Var3 0.103 0.222 0.105 0.344 0.576
Var4 0.148 0.534 0.439 0.422 0.544
Var5 0.062 0.106 0.071 0.294 0.199

, , Full, PA2

       SRE   CTA    RF  MARS   FDA
Var1 0.018 0.904 0.445 0.492 0.910
Var2 0.009 0.432 0.200 0.045 0.697
Var3 0.092 0.096 0.119 0.578 0.873
Var4 0.122 0.315 0.329 0.643 0.563
Var5 0.032 0.181 0.093 0.303 0.150
```

3.2 Ensemble modeling

Here comes one of the most interesting features of `biomod2`. `BIOMOD_EnsembleModeling` combines individual models to build some kind of meta-model. In the following example, we decide to exclude all models having a TSS score lower than 0.85.

NOTE 8 :
Models are now combined by repetition, other way to combine them (e.g. by Models, all together...) will be available soon

R input

```
myBiomodEM <- BIOMOD_EnsembleModeling(
                      modeling.output = myBiomodModelOut,
                      chosen.models = 'all',
                      eval.metric = c('TSS'),
                      eval.metric.quality.threshold = c(0.85),
                      prob.mean = T,
                      prob.cv = T,
                      prob.ci = T,
                      prob.ci.alpha = 0.05,
                      prob.median = T,
                      committee.averaging = T,
                      prob.mean.weight = T,
                      prob.mean.weight.decay = 'proportional' )
```

R output

```
! all models available will be included in ensemble.modeling
> Evaluation & Weighting methods summary :
   TSS over 0.85

> PA1

> TSS
> models kept :  Myocastor_PA1_Full_CTA, Myocastor_PA1_Full_RF, Myocastor_PA1_Full_MARS
> Mean of probabilities...
> Coef of variation of probabilities...
> Median of ptobabilities...
> Confidence Interval...
   > 2.5 %
   > 97.5 %
>  Comittee averaging...
> Prababilities wegthing mean...
> PA2

> TSS
```

```
> models kept :  Myocastor_PA2_RUN1_RF, Myocastor_PA2_RUN1_MARS, Myocastor_PA2_RUN1_FDA,
> Mean of probabilities...
> Coef of variation of probabilities...
> Median of ptobabilities...
> Confidence Interval...
   > 2.5 %
   > 97.5 %
>  Comittee averaging...
> Prababilities wegthing mean...
```

You can easily access to the data and outputs of `BIOMOD_Modeling` using some specific functions to make your life easier.
Let's see the meta-models evaluation scores.

NOTE 9 :
We decide to evaluate all meta-models produced even the CV (Coefficient of Variation) one which is quite hard to interpret. You may consider it as: higher my score is, more the variation is localised where my species is forecasted as present.

R input

```
# print summary
myBiomodEM
```

R output

```
-=-=-=- 'BIOMOD.EnsembleModeling.out' -=-=-=-
sp.name : Myocastor
expl.var.names : bio_3 bio_4 bio_7 bio_11 bio_12

models computed: Myocastor_PA1_AllRun_EM.TSS, Myocastor_PA2_AllRun_EM.TSS
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

R input

```
# get evaluation scores
getEMeval(myBiomodEM)
```

R output

```
$Myocastor_PA1_AllRun_EM.TSS
, , em.mean

    Testing.data Cutoff Sensitivity Specificity
TSS        0.985  483.0      100.00        98.5
ROC        0.999  522.8       98.31        98.5
```

```
, , em.cv

    Testing.data Cutoff Sensitivity Specificity
TSS        0.000  1.000       0.000         6.0
ROC        0.002  0.878       1.695         1.5

, , em.ci.inf

    Testing.data Cutoff Sensitivity Specificity
TSS        0.881  49.17       88.14         100
ROC        0.941 100.91       88.14         100

, , em.ci.sup

    Testing.data Cutoff Sensitivity Specificity
TSS         0.96    965         100          96
ROC         0.98   1000         100          96

, , em.median

    Testing.data Cutoff Sensitivity Specificity
TSS        0.975  289.1      100.00        97.5
ROC        0.998  483.0       96.61        97.5

, , em.ca

    Testing.data Cutoff Sensitivity Specificity
TSS        0.975  494.9         100        97.5
ROC        0.999  667.0         100        97.5

, , em.pmw

    Testing.data Cutoff Sensitivity Specificity
TSS        0.995    527         100        99.5
ROC        1.000    533         100        99.5


$Myocastor_PA2_AllRun_EM.TSS
, , em.mean

    Testing.data Cutoff Sensitivity Specificity
TSS        0.995  433.0         100        99.5
ROC        1.000  444.8         100        99.5

, , em.cv

    Testing.data Cutoff Sensitivity Specificity
TSS        0.000  0.000     100.000           0
ROC        0.001  0.939       1.695           2
```

```
, , em.ci.inf

    Testing.data Cutoff Sensitivity Specificity
TSS         0.99  24.00      100.00          99
ROC         1.00  68.04       98.31          99

, , em.ci.sup

    Testing.data Cutoff Sensitivity Specificity
TSS        0.995  846.0         100        99.5
ROC        1.000  852.5         100        99.5

, , em.median

    Testing.data Cutoff Sensitivity Specificity
TSS        0.985  293.2      100.00        98.5
ROC        1.000  389.2       98.31        98.5

, , em.ca

    Testing.data Cutoff Sensitivity Specificity
TSS        0.980  580.8         100          98
ROC        0.999  667.0         100          98

, , em.pmw

    Testing.data Cutoff Sensitivity Specificity
TSS        0.995  442.5         100        99.5
ROC        1.000  457.6         100        99.5
```

4 Projection

Once the models are calibrated and evaluated, we might want to project the potential distribution of the species over space and time. This is made using `BIOMOD_Projection`

NOTE 10 :
All projections are stored directly on your hard drive

First let's project the individual models on our current conditions (the globe) to visualize them.

R input

```
# projection over the globe under current conditions
myBiomomodProj <- BIOMOD_Projection(
                          modeling.output = myBiomodModelOut,
```

```
                        new.env = myExpl,
                        proj.name = 'current',
                        selected.models = 'all',
                        binary.meth = 'ROC',
                        compress = 'xz',
                        clamping.mask = F)
```

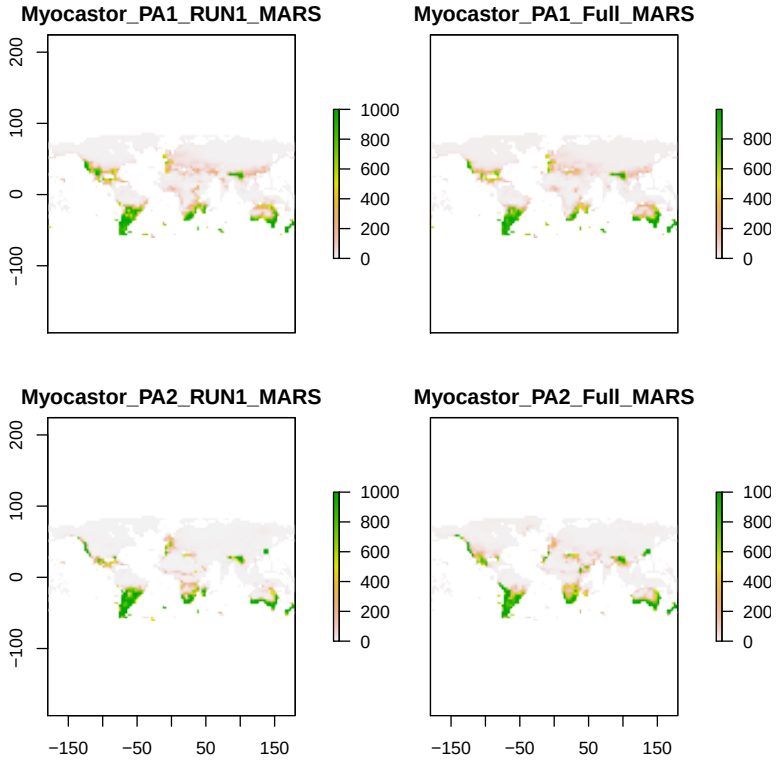
R output

```
-=-=-=- BIOMOD Projection Stuff -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Loading required library...
Doing Models Projections...
*** Myocastor_PA1_RUN1_SRE
*** Myocastor_PA1_RUN1_CTA
*** Myocastor_PA1_RUN1_RF
*** Myocastor_PA1_RUN1_MARS
*** Myocastor_PA1_RUN1_FDA
*** Myocastor_PA1_Full_SRE
*** Myocastor_PA1_Full_CTA
*** Myocastor_PA1_Full_RF
*** Myocastor_PA1_Full_MARS
*** Myocastor_PA1_Full_FDA
*** Myocastor_PA2_RUN1_SRE
*** Myocastor_PA2_RUN1_CTA
*** Myocastor_PA2_RUN1_RF
*** Myocastor_PA2_RUN1_MARS
*** Myocastor_PA2_RUN1_FDA
*** Myocastor_PA2_Full_SRE
*** Myocastor_PA2_Full_CTA
*** Myocastor_PA2_Full_RF
*** Myocastor_PA2_Full_MARS
*** Myocastor_PA2_Full_FDA
Binary transformations...

-=-=-=- Done -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

R input

R input

```
# make some plots sub-selected by str.grep argument
plot(myBiomomodProj, str.grep = 'MARS')
```



R input

```
# if you want to make custom plots, you can also get the projected map
myCurrentProj <- getProjection(myBiomomodProj)
myCurrentProj
```

R output

```
class       : RasterStack
dimensions  : 45, 108, 4860, 20  (nrow, ncol, ncell, nlayers)
resolution  : 3.333, 3.333  (x, y)
extent      : -180, 180, -60, 90  (xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0
min values  :  0 14  2  0 45  0  2  0  0 49 ...
max values  : 1000  772  995 1000  976 1000  922 1000  998  988 ...
```

Then we can project the potential distribution of the species over time, i.e. into the future.

R input

```
# load environmental variables for the future.
myExpl2050 = stack( system.file( "external/climat/future/bio3.grd",
                                 package="biomod2"),
                    system.file( "external/climat/future/bio4.grd",
                                 package="biomod2"),
                    system.file( "external/climat/future/bio7.grd",
```

```
                                   package="biomod2"),
                      system.file( "external/climat/future/bio11.grd",
                                   package="biomod2"),
                      system.file( "external/climat/future/bio12.grd",
                                   package="biomod2"))
myBiomomodProj2050 <- BIOMOD_Projection(
                                modeling.output = myBiomodModelOut,
                                new.env = stack(myExpl2050),
                                proj.name = 't2050',
                                selected.models = 'all',
                                binary.meth = 'ROC',
                                compress = 'xz',
                                clamping.mask = T)
```

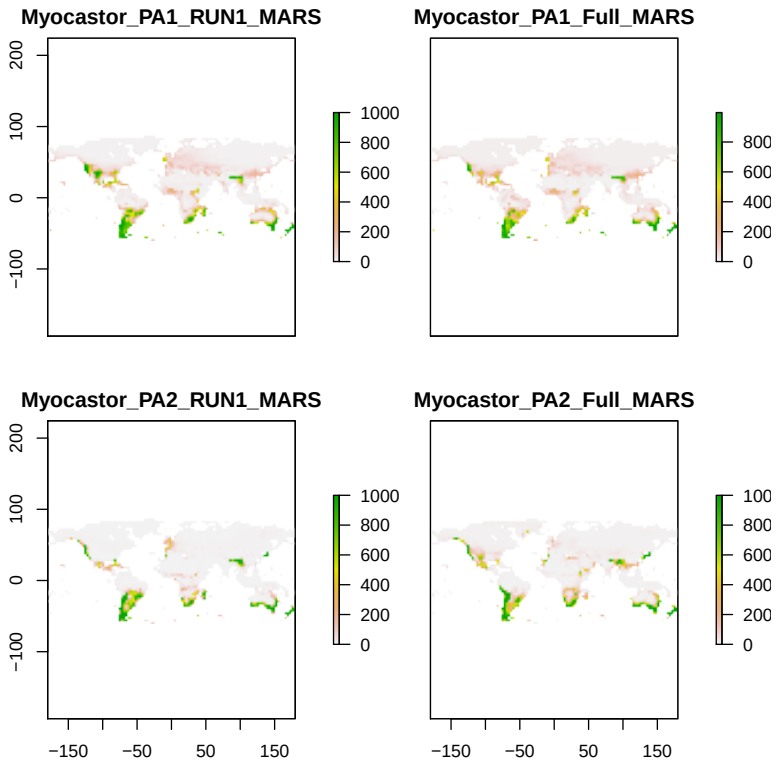
R output

```
-=-=-=- BIOMOD Projection Stuff -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
   > defining clamping mask

Loading required library...
Doing Models Projections...
*** Myocastor_PA1_RUN1_SRE
*** Myocastor_PA1_RUN1_CTA
*** Myocastor_PA1_RUN1_RF
*** Myocastor_PA1_RUN1_MARS
*** Myocastor_PA1_RUN1_FDA
*** Myocastor_PA1_Full_SRE
*** Myocastor_PA1_Full_CTA
*** Myocastor_PA1_Full_RF
*** Myocastor_PA1_Full_MARS
*** Myocastor_PA1_Full_FDA
*** Myocastor_PA2_RUN1_SRE
*** Myocastor_PA2_RUN1_CTA
*** Myocastor_PA2_RUN1_RF
*** Myocastor_PA2_RUN1_MARS
*** Myocastor_PA2_RUN1_FDA
*** Myocastor_PA2_Full_SRE
*** Myocastor_PA2_Full_CTA
*** Myocastor_PA2_Full_RF
*** Myocastor_PA2_Full_MARS
*** Myocastor_PA2_Full_FDA
Binary transformations...

-=-=-=- Done -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

R input

R input

```
# make some plots, sub-selected by str.grep argument
plot(myBiomomodProj2050, str.grep = 'MARS')
```



The last step of this vignette is to make Ensemble Forcasting, that means to project the meta-models you have created with `BIOMOD_EnsembleModeling`. `BIOMOD_EnsembleForecasting` required the output of `BIOMOD_EnsembleModeling` and `BIOMOD_Projection`. It will combine the projections made according to models ensemble rules defined at the ensemble modelling step.

R input

```
myBiomodEF <- BIOMOD_EnsembleForecasting(
                        projection.output = myBiomomodProj2050,
                        EM.output = myBiomodEM )
```

R output

```
*** Myocastor_PA1_AllRun_EM.TSS ...
   > em.mean
   > em.cv
   > em.ci.inf
   > em.ci.sup
   > em.median
   > em.ca
```

```
   > em.pmw

*** Myocastor_PA2_AllRun_EM.TSS ...
   > em.mean
   > em.cv
   > em.ci.inf
   > em.ci.sup
   > em.median
   > em.ca
   > em.pmw
```

Nothing is returned but some additional files have been created in your projection folder (RasterStack or array depending on your projection type). This file contains your meta-models projections.

R input

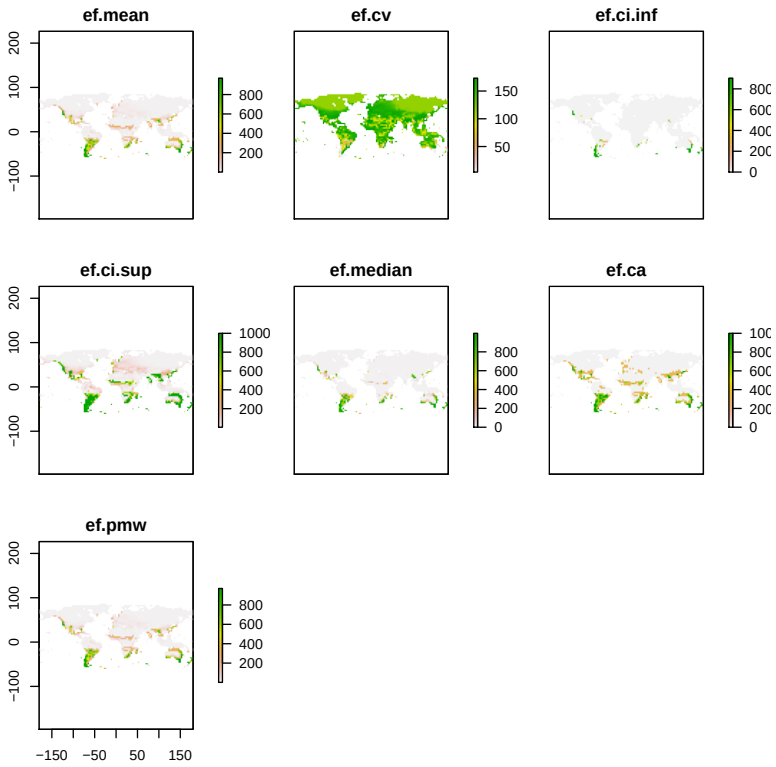
```
load("Myocastor/proj_t2050/Myocastor_PA1_AllRun_EM.TSS")
Myocastor_PA1_AllRun_EM.TSS
```

R output

```
class       : RasterStack
dimensions  : 45, 108, 4860, 7  (nrow, ncol, ncell, nlayers)
resolution  : 3.333, 3.333  (x, y)
extent      : -180, 180, -60, 90  (xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0
min values  : 0.67 4.06 0.00 3.00 0.00 0.00 1.00
max values  :  969  173  903 1000  997 1000  970
```

R input

```
plot(Myocastor_PA1_AllRun_EM.TSS)
```


ef.mean
ef.cv
ef.ci.inf
ef.ci.sup
ef.median
ef.ca
ef.pmw
200
100
0
−100
800
600
400
200
150
100
50
1000
−150
−50
50
150