An example of species distribution modeling with biomod2

Damien Georges & Wilfried Thuiller

April 5, 2013

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.

Here we are going to modeled the current and future (2050) distribution of Gulo Gulo.

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 a quite common case) with :

- presences/absences points data
- environmental raster layers (e.g. Worldclim)

Let's import our data.

R input

```
# load the library
library(biomod2)
# load our species data
DataSpecies <- read.csv(system.file("external/species/mammals_table.csv",
                                    package="biomod2"))
head(DataSpecies)
```

R output

```
  X X_WGS84 Y_WGS84 ConnochaetesGnou GuloGulo PantheraOnca
1 1   -94.5      82                0        0            0
2 2   -91.5      82                0        1            0
3 3   -88.5      82                0        1            0
4 4   -85.5      82                0        1            0
5 5   -82.5      82                0        1            0
6 6   -79.5      82                0        1            0
  PteropusGiganteus TenrecEcaudatus VulpesVulpes
1                 0               0            0
2                 0               0            0
3                 0               0            0
4                 0               0            0
5                 0               0            0
6                 0               0            0
```

R input

```
# the name of studied species
myRespName <- 'GuloGulo'
# the presence/absences data for our species
myResp <- as.numeric(DataSpecies[,myRespName])
# the XY coordinates of species data
myRespXY <- DataSpecies[,c("X_WGS84","Y_WGS84")]
# load the environmental raster layers (could be .img, ArcGIS
# rasters or any supported format by the raster package)

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

NOTE 2:
You may not have absences data. 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.
biomod2 offers some tools to do it more or less automatically. 3 algorithms are now implemented to extract a range of pseudo-absence data: 'random', 'SRE' and 'disk'. A vignette will be written soon to explain how to do. Waiting for this, you can refer to BIOMOD_FormatingData help file

NOTE 3:
If your environmental data are in matrix (or data.frame) format, you have to give a species as vector having a length that match with the number of rows of your environmental dataset. 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.

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)

R input

```
myBiomodData <- BIOMOD_FormatingData(resp.var = myResp,
                                     expl.var = myExpl,
                                     resp.xy = myRespXY,
                                     resp.name = myRespName)
```

R output

```
-=-=-=-=-=-=-=-=-= GuloGulo Data Formating -=-=-=-=-=-=-=-=-=

> No pseudo absences selection !
      ! No data has been set aside for modeling evaluation
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-= Done -=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

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

R input

```
myBiomodData
```

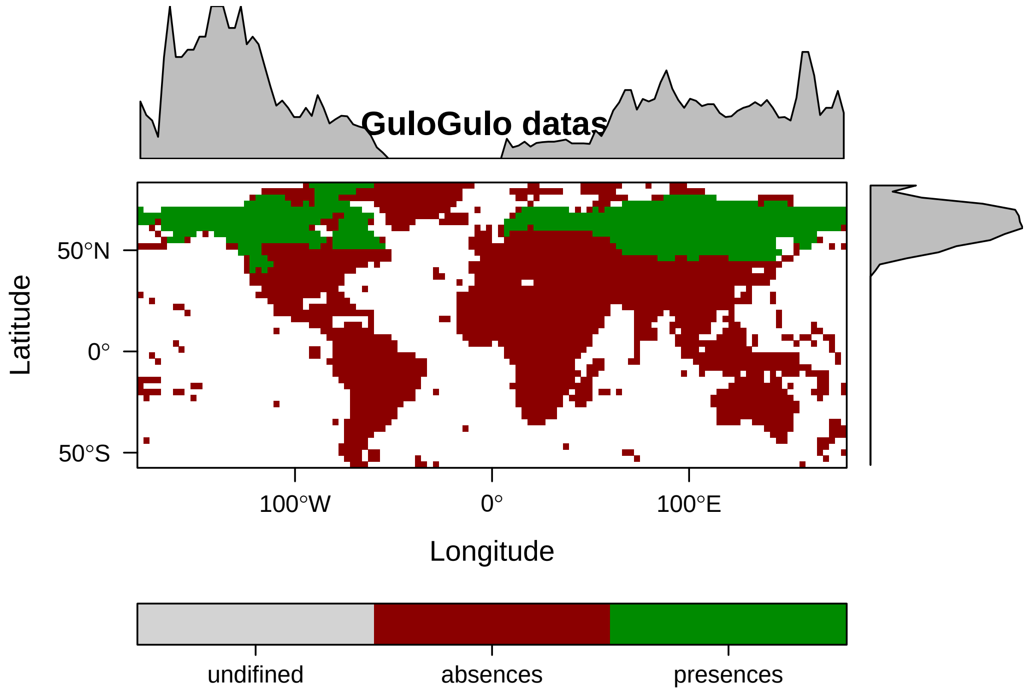
R output

```
-=-=-=-=-=-=-=-=-= 'BIOMOD.formated.data' -=-=-=-=-=-=-=-=-=

sp.name =  GuloGulo

         661 presences,  1827 true absences and  0
undifined points in dataset
```

```
         5 explanatory variables

      bio3           bio4            bio7      
 Min.   :10.2   Min.   :   72   Min.   : 54.5  
 1st Qu.:21.2   1st Qu.: 2641   1st Qu.:186.0  
 Median :35.0   Median : 6682   Median :306.2  
 Mean   :40.3   Mean   : 7358   Mean   :310.9  
 3rd Qu.:56.4   3rd Qu.:11752   3rd Qu.:424.6  
 Max.   :92.0   Max.   :22314   Max.   :718.0  
     bio11             bio12     
 Min.   :-447.7   Min.   :   0  
 1st Qu.:-184.3   1st Qu.: 276  
 Median :  24.2   Median : 563  
 Mean   :  -2.6   Mean   : 854  
 3rd Qu.: 196.3   3rd Qu.:1201  
 Max.   : 283.0   Max.   :5431  

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

R input

```
plot(myBiomodData)
```



The colors for this plot match with...

- Presences
- Absences

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 5:
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 3-fold cross-validation (number controlled by "NbRunEval" argument) of our models by randomly splitting our data set into 2 subsets : "DataSplit"

R input

```
# 3. Computing the models

myBiomodModelOut <- BIOMOD_Modeling(
                            myBiomodData,
                            models = c('SRE','CTA','RF','MARS','FDA'),
                            models.options = myBiomodOption,
                            NbRunEval=3,
                            DataSplit=80,
                            Prevalence=0.5,
                            VarImport=3,
                            models.eval.meth = c('TSS','ROC'),
                            SaveObj = TRUE,
                            rescal.all.models = TRUE,
                            do.full.models = FALSE,
                            modeling.id = paste(myRespName,"FirstModeling",sep=""))
```

R output

```
Loading required library...

Checking Models arguments...

Creating suitable Workdir...

        > Automatic weights creation to rise a 0.5 prevalence


-=-=-=-=-=-=-=-= GuloGulo Modeling Summary -=-=-=-=-=-=-=-=

 5  environmental variables ( bio3 bio4 bio7 bio11 bio12 )
Number of evaluation repetitions : 3
Models selected : SRE CTA RF MARS FDA

Total number of model runs : 15

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=


-=-=-=- Run :  GuloGulo_AllData


-=-=-=-=-=-=-=- GuloGulo_AllData_RUN1

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

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

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

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

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


-=-=-=--=-=-=- GuloGulo_AllData_RUN2

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

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

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

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

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


-=-=-=--=-=-=- GuloGulo_AllData_RUN3

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

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

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

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

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

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

R input

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

- modeling summary

R input

```
myBiomodModelOut
```

R output

```
-=-=-=-=-=-=-=-=-=-=-= BIOMOD.models.out -=-=-=-=-=-=-=-=-=-=-=

Modeling id : GuloGuloFirstModeling

Species modeled : GuloGulo

Considered variables : bio3 bio4 bio7 bio11 bio12


Computed Models :  GuloGulo_AllData_RUN1_SRE
GuloGulo_AllData_RUN1_CTA GuloGulo_AllData_RUN1_RF
GuloGulo_AllData_RUN1_MARS GuloGulo_AllData_RUN1_FDA
GuloGulo_AllData_RUN2_SRE GuloGulo_AllData_RUN2_CTA
GuloGulo_AllData_RUN2_RF GuloGulo_AllData_RUN2_MARS
GuloGulo_AllData_RUN2_FDA GuloGulo_AllData_RUN3_SRE
GuloGulo_AllData_RUN3_CTA GuloGulo_AllData_RUN3_RF
GuloGulo_AllData_RUN3_MARS GuloGulo_AllData_RUN3_FDA


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" "RUN2" "RUN3"

[[5]]
GuloGulo_AllData
       "AllData"
```

R input

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

R output

```
 RUN1  RUN2  RUN3
0.914 0.915 0.874
```

R input

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

R output

```
      RUN1  RUN2  RUN3
SRE  0.871 0.846 0.883
CTA  0.948 0.950 0.938
RF   0.987 0.989 0.981
MARS 0.978 0.975 0.970
FDA  0.974 0.964 0.959
```

R input

```
```

- Relative importance of the explanatory variables

R input

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

R output

```
, , RUN1, AllData
```

```
        SRE   CTA    RF  MARS   FDA
bio3  0.469 0.071 0.048 0.248 0.000
bio4  0.352 0.591 0.154 0.520 0.951
bio7  0.295 0.080 0.065 0.236 0.094
bio11 0.450 0.613 0.478 0.640 0.292
bio12 0.317 0.092 0.057 0.070 0.021

, , RUN2, AllData

        SRE   CTA    RF  MARS   FDA
bio3  0.455 0.129 0.048 0.005 0.000
bio4  0.363 0.640 0.190 0.852 1.000
bio7  0.304 0.116 0.079 0.156 0.092
bio11 0.441 0.664 0.583 0.690 0.233
bio12 0.351 0.152 0.068 0.099 0.025

, , RUN3, AllData

        SRE   CTA    RF  MARS   FDA
bio3  0.447 0.055 0.032 0.017 0.000
bio4  0.357 0.398 0.137 0.649 0.932
bio7  0.342 0.105 0.074 0.203 0.100
bio11 0.457 0.708 0.539 0.710 0.309
bio12 0.325 0.123 0.060 0.057 0.027
```

NOTE 6:
Relative importance of variable returned are raw data. It may be usefull to normalise them to make them comparable one to another

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 0.7.

NOTE 7:
You can controle the way formal models are combined with em.by argument. The vignette ”EnsembleModelingAssembly” illustrate the offered possibilities

R input

```
myBiomodEM <- BIOMOD_EnsembleModeling(
                     modeling.output = myBiomodModelOut,
                     chosen.models = 'all',
                     em.by='all',
                     eval.metric = c('TSS'),
                     eval.metric.quality.threshold = c(0.7),
                     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

```
-=-=-=-=-=-=-=-=-= Build Ensemble Models -=-=-=-=-=-=-=-=-=

   ! all models available will be included in ensemble.modeling
   > Evaluation & Weighting methods summary :
      TSS over 0.7


  > TotalConsensus ensemble modeling
   > TSS
   > models kept :  GuloGulo_AllData_RUN1_SRE, GuloGulo_AllData_RUN1_CTA, GuloGulo_AllData_RUN1_RF, Gul
   ! Models projections for whole zonation required...
        > Projecting GuloGulo_AllData_RUN1_SRE ...
        > Projecting GuloGulo_AllData_RUN1_CTA ...
        > Projecting GuloGulo_AllData_RUN1_RF ...
        > Projecting GuloGulo_AllData_RUN1_MARS ...
        > Projecting GuloGulo_AllData_RUN1_FDA ...
        > Projecting GuloGulo_AllData_RUN2_CTA ...
        > Projecting GuloGulo_AllData_RUN2_RF ...
        > Projecting GuloGulo_AllData_RUN2_MARS ...
        > Projecting GuloGulo_AllData_RUN2_FDA ...
        > Projecting GuloGulo_AllData_RUN3_SRE ...
        > Projecting GuloGulo_AllData_RUN3_CTA ...
        > Projecting GuloGulo_AllData_RUN3_RF ...
        > Projecting GuloGulo_AllData_RUN3_MARS ...
        > Projecting GuloGulo_AllData_RUN3_FDA ...

   > Mean of probabilities...
   > Coef of variation of probabilities...
   > Median of ptobabilities...
   > Confidence Interval...
      > 2.5 %
      > 97.5 %
   >  Comittee averaging...
   > Prabahilities wegthing mean...
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-= Done -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

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 8:
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 : GuloGulo

expl.var.names : bio3 bio4 bio7 bio11 bio12


models computed: GuloGulo_TotalConsensus_EMbyTSS

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

R input

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

R output

```
$GuloGulo_TotalConsensus_EMbyTSS
, , em.mean

    Testing.data Cutoff Sensitivity Specificity
TSS        0.919    417       97.73       94.14
ROC        0.992    421       97.73       94.20

, , em.cv

    Testing.data Cutoff Sensitivity Specificity
TSS        0.000      0         100           0
ROC        0.016     NA          NA          NA

, , em.ci.inf

    Testing.data Cutoff Sensitivity Specificity
TSS        0.917    188       97.73       93.76
ROC        0.992    191       97.73       93.98

, , em.ci.sup

    Testing.data Cutoff Sensitivity Specificity
TSS        0.921  664.0       97.43       94.64
ROC        0.991  670.5       97.28       94.96

, , em.median

    Testing.data Cutoff Sensitivity Specificity
TSS        0.905    623       93.34       97.15
ROC        0.990    624       93.34       97.15

, , em.ca
```

```
    Testing.data Cutoff Sensitivity Specificity
TSS        0.904    394       97.58       92.78
ROC        0.989    393       97.58       92.78

, , em.pmw

    Testing.data Cutoff Sensitivity Specificity
TSS        0.926    436       97.73       94.86
ROC        0.993    436       97.73       94.86
```

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 9:
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
 myBiomodProj <- BIOMOD_Projection(
                          modeling.output = myBiomodModelOut,
                          new.env = myExpl,
                          proj.name = 'current',
                          selected.models = 'all',
                          binary.meth = 'TSS',
                          compress = 'xz',
                          clamping.mask = F,
                          output.format = '.grd')
```

R output

```
-=-=-=-=-=-=-=-=-= Do Models Projections -=-=-=-=-=-=-=-=-=

        > Building clamping mask

        > Projecting GuloGulo_AllData_RUN1_SRE ...
        > Projecting GuloGulo_AllData_RUN1_CTA ...
        > Projecting GuloGulo_AllData_RUN1_RF ...
        > Projecting GuloGulo_AllData_RUN1_MARS ...
        > Projecting GuloGulo_AllData_RUN1_FDA ...
        > Projecting GuloGulo_AllData_RUN2_SRE ...
        > Projecting GuloGulo_AllData_RUN2_CTA ...
        > Projecting GuloGulo_AllData_RUN2_RF ...
        > Projecting GuloGulo_AllData_RUN2_MARS ...
        > Projecting GuloGulo_AllData_RUN2_FDA ...
        > Projecting GuloGulo_AllData_RUN3_SRE ...
        > Projecting GuloGulo_AllData_RUN3_CTA ...
        > Projecting GuloGulo_AllData_RUN3_RF ...
        > Projecting GuloGulo_AllData_RUN3_MARS ...
        > Projecting GuloGulo_AllData_RUN3_FDA ...

        > Building TSS binaries
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-= Done -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

R input

```
# summary of crated oject
myBiomodProj
```

R output

```
-=-=-=-=-=-=-=-=-=-= 'BIOMOD.projection.out' -=-=-=-=-=-=-=-=-=-=

Projection directory : GuloGulo/current


sp.name : GuloGulo

expl.var.names : bio3 bio4 bio7 bio11 bio12


modeling id : GuloGuloFirstModeling (
GuloGulo/GuloGulo.GuloGuloFirstModeling.models.out )

models projected :
GuloGulo_AllData_RUN1_SRE, GuloGulo_AllData_RUN1_CTA, GuloGulo_AllData_RUN1_RF, GuloGulo_AllData_RUN1_M

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

R input

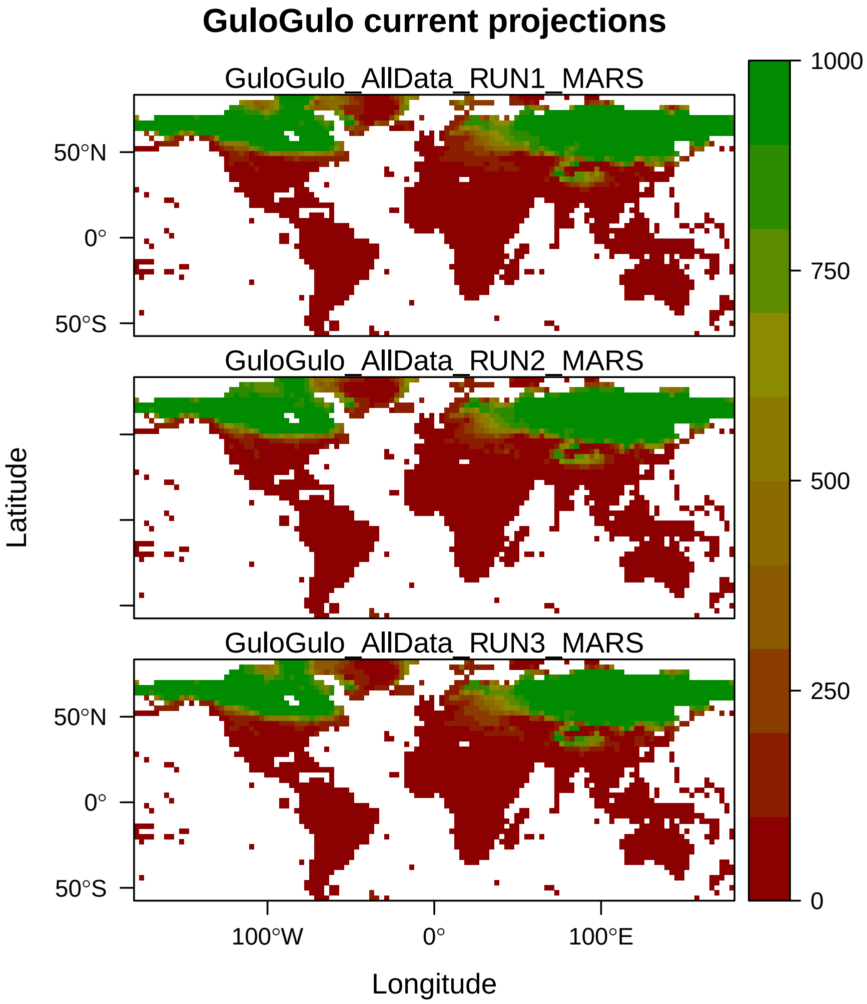
```
# files created on hard drive
list.files("GuloGulo/proj_current/")
```

R output

```
 [1] "GuloGulo.current.projection.out"
 [2] "proj_current_ClampingMask.grd"
 [3] "proj_current_ClampingMask.gri"
 [4] "proj_current_GuloGulo.grd"
 [5] "proj_current_GuloGulo.gri"
 [6] "proj_current_GuloGulo_TotalConsensus_EMbyTSS.grd"
 [7] "proj_current_GuloGulo_TotalConsensus_EMbyTSS.gri"
 [8] "proj_current_GuloGulo_TotalConsensus_EMbyTSS_TSSbin.grd"
 [9] "proj_current_GuloGulo_TotalConsensus_EMbyTSS_TSSbin.gri"
[10] "proj_current_GuloGulo_TSSbin.grd"
[11] "proj_current_GuloGulo_TSSbin.gri"
```

R input

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


GuloGulo current projections
GuloGulo_AllData_RUN1_MARS
GuloGulo_AllData_RUN2_MARS
GuloGulo_AllData_RUN3_MARS
50°N
0°
50°S
Latitude
100°W
0°
100°E
Longitude
1000
750
500
250
0

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

```
R output
class       : RasterStack
dimensions  : 47, 120, 5640, 15  (nrow, ncol, ncell, nlayers)
resolution  : 3, 3  (x, y)
extent      : -180, 180, -57.5, 83.5  (xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
names       : GuloGulo_AllData_RUN1_SRE, GuloGulo_AllData_RUN1_CTA, GuloGulo_AllData_RUN1_RF, GuloGulo_
min values  :                         0,                        25,                        4,
max values  :                      1000,                       970,                     1000,
```

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.
myExplFuture = stack( system.file( "external/bioclim/future/bio3.grd",
                                   package="biomod2"),
                     system.file( "external/bioclim/future/bio4.grd",
                                   package="biomod2"),
                     system.file( "external/bioclim/future/bio7.grd",
                                   package="biomod2"),
                     system.file( "external/bioclim/future/bio11.grd",
                                   package="biomod2"),
                     system.file( "external/bioclim/future/bio12.grd",
                                   package="biomod2"))
myBiomodProjFuture <- BIOMOD_Projection(
                                modeling.output = myBiomodModelOut,
                                new.env = myExplFuture,
                                proj.name = 'future',
                                selected.models = 'all',
                                binary.meth = 'TSS',
                                compress = 'xz',
                                clamping.mask = T,
                                output.format = '.grd')
```

R output

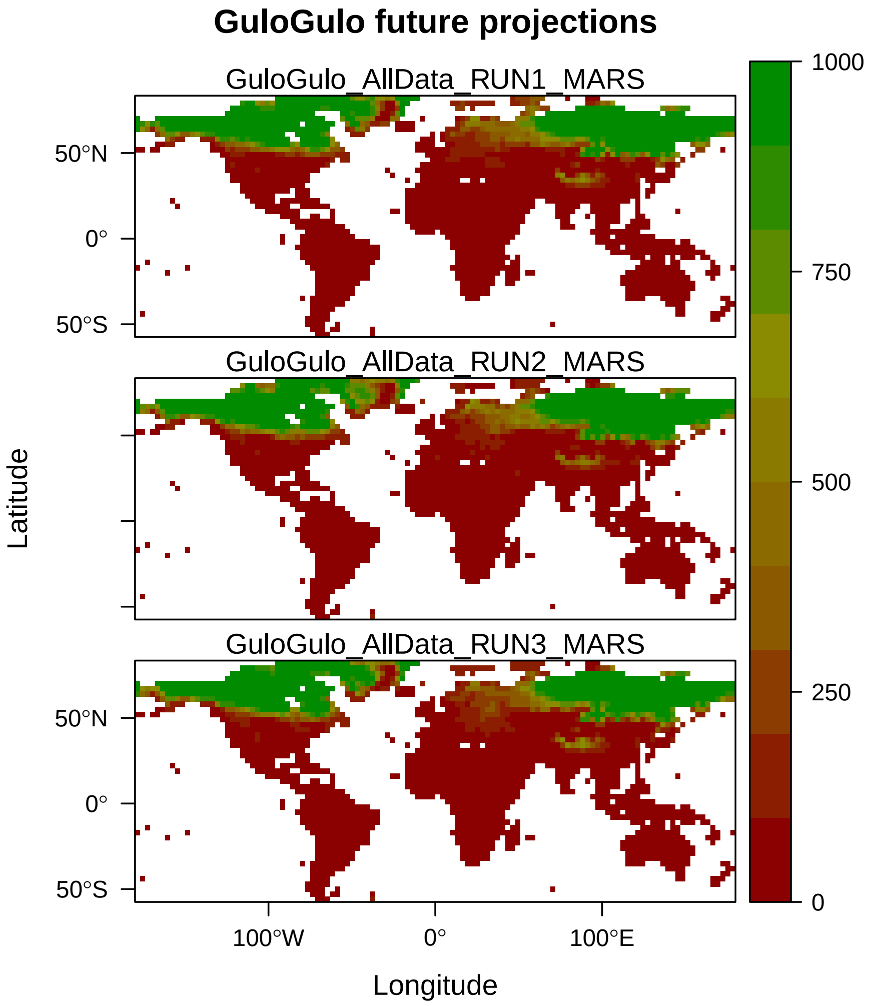
```
-=-=-=-=-=-=-=-=-= Do Models Projections -=-=-=-=-=-=-=-=-=

        > Building clamping mask

        > Projecting GuloGulo_AllData_RUN1_SRE ...
        > Projecting GuloGulo_AllData_RUN1_CTA ...
        > Projecting GuloGulo_AllData_RUN1_RF ...
        > Projecting GuloGulo_AllData_RUN1_MARS ...
        > Projecting GuloGulo_AllData_RUN1_FDA ...
        > Projecting GuloGulo_AllData_RUN2_SRE ...
        > Projecting GuloGulo_AllData_RUN2_CTA ...
        > Projecting GuloGulo_AllData_RUN2_RF ...
        > Projecting GuloGulo_AllData_RUN2_MARS ...
        > Projecting GuloGulo_AllData_RUN2_FDA ...
        > Projecting GuloGulo_AllData_RUN3_SRE ...
        > Projecting GuloGulo_AllData_RUN3_CTA ...
        > Projecting GuloGulo_AllData_RUN3_RF ...
        > Projecting GuloGulo_AllData_RUN3_MARS ...
        > Projecting GuloGulo_AllData_RUN3_FDA ...

        > Building TSS binaries
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-= Done -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

R input

```
```

R input

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


GuloGulo future projections
GuloGulo_AllData_RUN1_MARS
GuloGulo_AllData_RUN2_MARS
GuloGulo_AllData_RUN3_MARS
50°N
0°
50°S
Latitude
100°W
0°
100°E
Longitude
1000
750
500
250
0

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 = myBiomodProj,
                        EM.output = myBiomodEM,
                        binary.meth = 'TSS')
```

R output

```
-=-=-=-=-=-=-= Do Ensemble Models Projections -=-=-=-=-=-=-=


        > Projecting GuloGulo_TotalConsensus_EMbyTSS ...
                > em.mean
                > em.cv
                > em.ci.inf
                > em.ci.sup
                > em.median
                > em.ca
                > em.pmw
                > Writing proj_current_GuloGulo_TotalConsensus_EMbyTSS.grd on hard drive...
                > Writing proj_current_GuloGulo_TotalConsensus_EMbyTSS_TSSbin.grd on hard drive...

Nothing is returned but you can access created projections by loading them with 'load(...)' for '.RData

Available files are :
'GuloGulo/proj_current/proj_current_GuloGulo_TotalConsensus_EMbyTSS.grd'
'GuloGulo/proj_current/proj_current_GuloGulo_TotalConsensus_EMbyTSS_TSSbin.grd'

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

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

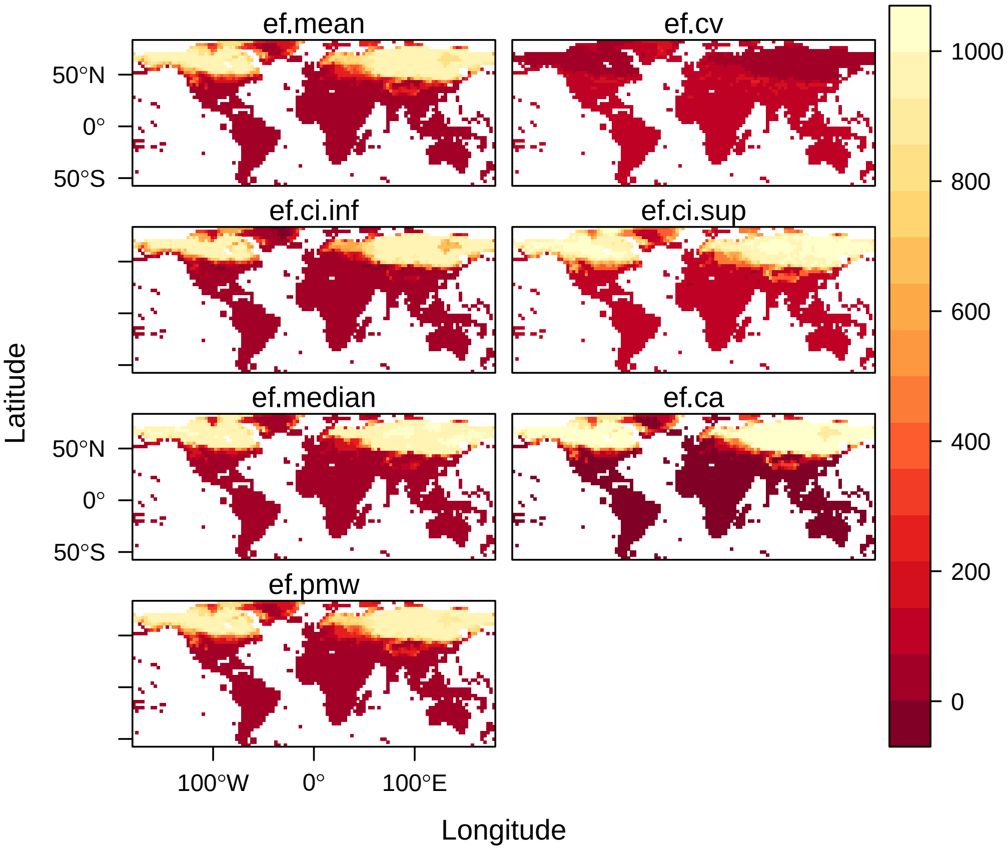
```
proj_current_GuloGulo_TotalConsensus_EMbyTSS <-
   stack("GuloGulo/proj_current/proj_current_GuloGulo_TotalConsensus_EMbyTSS.grd")
proj_current_GuloGulo_TotalConsensus_EMbyTSS
```

R output

```
class        : RasterStack
dimensions   : 47, 120, 5640, 7  (nrow, ncol, ncell, nlayers)
resolution   : 3, 3  (x, y)
extent       : -180, 180, -57.5, 83.5  (xmin, xmax, ymin, ymax)
coord. ref.  : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
names        : GuloGulo_//SS_ef.mean, GuloGulo_//yTSS_ef.cv, GuloGulo_//_ef.ci.inf, GuloGulo_//_ef.ci.su
min values   :                 38.50,                  1.36,                  0.00,                 72.0
max values   :                 992.1,                 239.2,                 984.0,                1000.
```

R input

```
# reduce layer names for plotting convegences
names(proj_current_GuloGulo_TotalConsensus_EMbyTSS) <-
   sapply(strsplit(names(proj_current_GuloGulo_TotalConsensus_EMbyTSS),"_"), tail, n=1)
levelplot(proj_current_GuloGulo_TotalConsensus_EMbyTSS)
```



5 Conclusion

This vignette describes how to build and test a range of models within biomod2 but also how to build ensemble projections under current and future conditions. With few modifications, you should be able to apply the default functions onto your own dataset.