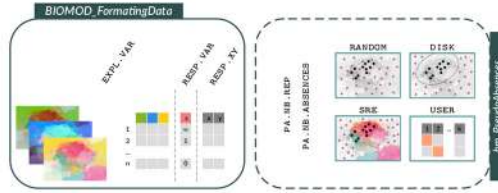


Species distribution modeling, calibration and evaluation, ensemble modeling



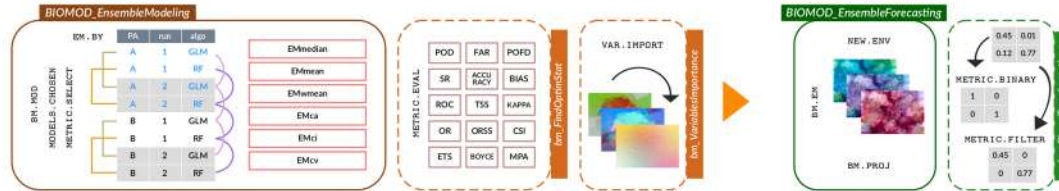
DATA FORMATING



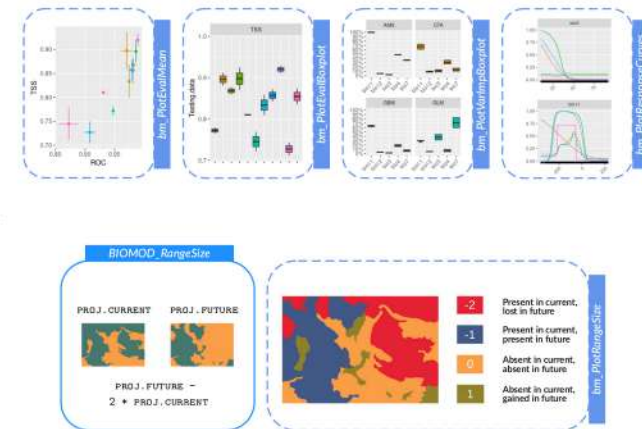
SINGLE models



ENSEMBLE models



OUTPUT & PLOT functions



Species distribution modeling, calibration and evaluation, ensemble modeling



Formating data Pseudo-absences

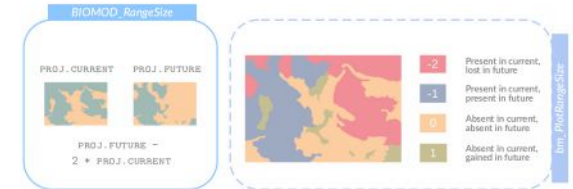
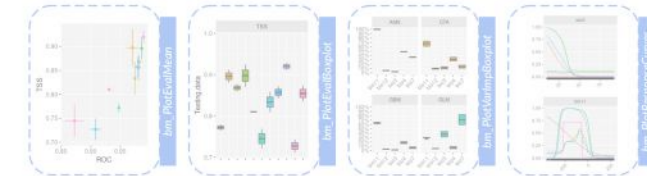
BIOMOD_FormatingData

Cross-validation Modeling options Tuning

Single models Evaluation Variables' importance

Ensemble models Evaluation Variables' importance

OUTPUT & PLOT functions



The logo is a green hexagon with a black border. Inside, there are stylized white clouds at the top left. Below them is a red thermometer with a black scale. To the right of the thermometer are three yellow books stacked on top of each other. The word 'biomod' is written in a black, lowercase, sans-serif font, with the thermometer and books acting as the letter 'i' and 'o' respectively. A large, bold, black number '2' is positioned to the right of the books.



The diagram illustrates the process of forming pseudo-absences from BIOMOD data. It is divided into two main sections by a dashed line.

- Left Section (BIOMOD, FormatingData):** This section shows a grid of environmental variables (e.g., PAR, BIO_01, BIO_02, BIO_03, BIO_04, BIO_05, BIO_06, BIO_07, BIO_08, BIO_09, BIO_10, BIO_11, BIO_12, BIO_13, BIO_14, BIO_15, BIO_16, BIO_17, BIO_18, BIO_19, BIO_20, BIO_21, BIO_22, BIO_23, BIO_24, BIO_25, BIO_26, BIO_27, BIO_28, BIO_29, BIO_30, BIO_31, BIO_32, BIO_33, BIO_34, BIO_35, BIO_36, BIO_37, BIO_38, BIO_39, BIO_40, BIO_41, BIO_42, BIO_43, BIO_44, BIO_45, BIO_46, BIO_47, BIO_48, BIO_49, BIO_50, BIO_51, BIO_52, BIO_53, BIO_54, BIO_55, BIO_56, BIO_57, BIO_58, BIO_59, BIO_60, BIO_61, BIO_62, BIO_63, BIO_64, BIO_65, BIO_66, BIO_67, BIO_68, BIO_69, BIO_70, BIO_71, BIO_72, BIO_73, BIO_74, BIO_75, BIO_76, BIO_77, BIO_78, BIO_79, BIO_80, BIO_81, BIO_82, BIO_83, BIO_84, BIO_85, BIO_86, BIO_87, BIO_88, BIO_89, BIO_90, BIO_91, BIO_92, BIO_93, BIO_94, BIO_95, BIO_96, BIO_97, BIO_98, BIO_99, BIO_100) and a corresponding grid of values (e.g., 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100). Below the grid are three stacked circular plots showing different spatial distributions of data points.
- Right Section:** This section shows four square plots labeled "RANDOM", "DISK", "SKE", and "USER". Each plot displays a different method for generating pseudo-absences. The "RANDOM" plot shows random distribution of points. The "DISK" plot shows points distributed around a central point. The "SKE" plot shows points distributed along a diagonal line. The "USER" plot shows points distributed according to user-defined criteria. To the right of these plots is a vertical bar labeled "Pseudo-absences".

Cross-validation Modeling options Tuning

The diagram illustrates the process of cross-validation and model tuning, organized into three main categories:

- RANDOM**
 - STRAT**: Stratified cross-validation, shown with a 3D cube and a cross-section.
 - EVAL**: Evaluation, shown with a 3D cube and a cross-section.
- Model selection options**
 - Grid search**: A systematic search for the best model.
 - Bayesian optimization**: An optimization technique that uses Bayesian statistics to select the best model.
- PARAMETERS**
 - Formula + stats()**: A formula-based approach to model selection, shown with a 3D cube and a cross-section.

[illegible]

The diagram illustrates the process of binary transformation for image fusion. It is divided into two main sections: the top section for 'BIOMOD_Protection' and the bottom section for 'BIOMOD_Protection'.

Top Section (BIOMOD_Protection):

- Input:** A 3x3 grid of images labeled 'NEW_EMV'.
- Process:** The input is processed by 'EMV_H2D2' to produce a single image labeled 'EMV_H2D2'.
- Output:** The output is a single image labeled 'EMV_H2D2'.

Bottom Section (BIOMOD_Protection):

- Input:** A 3x3 grid of images labeled 'NEW_EMV'.
- Process:** The input is processed by 'EMV_H2D2' to produce a single image labeled 'EMV_H2D2'.
- Output:** The output is a single image labeled 'EMV_H2D2'.

[illegible]

ROC

Box Plot

Heatmap

Line Graph

BIOMOD - RangeSize

PROJ. CURRENT PROJ. FUTURE

PROJ. FUTURE - 2 + PROJ. CURRENT

Range change

Legend:

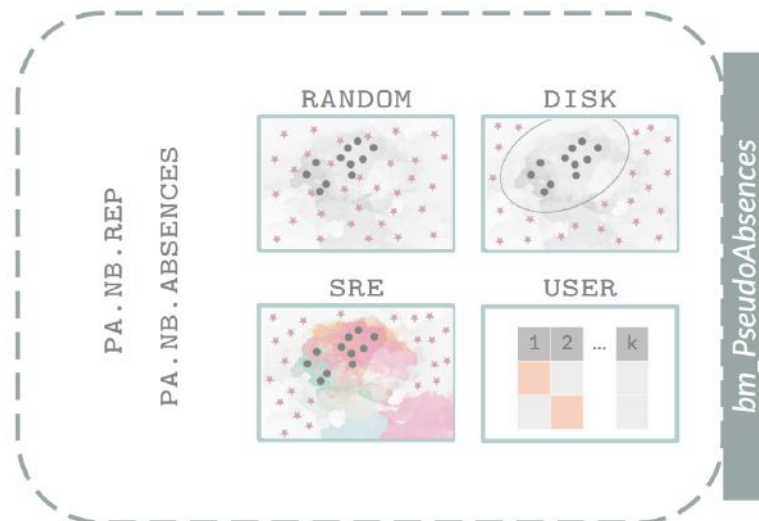
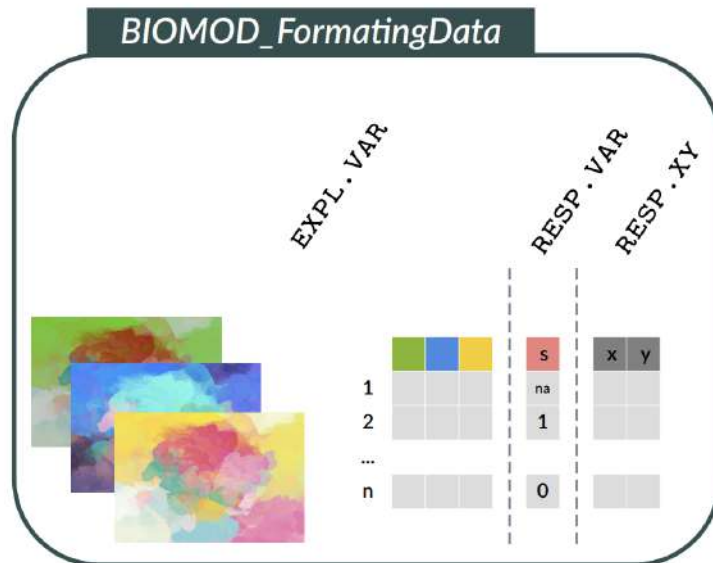
- 2: Present in current, lost in future
- 0: Present in current, present in future
- 1: Absent in current, gained in future

Inv. Proj. RangeSize

OUTPUT & PLOT functions

1. Formating data

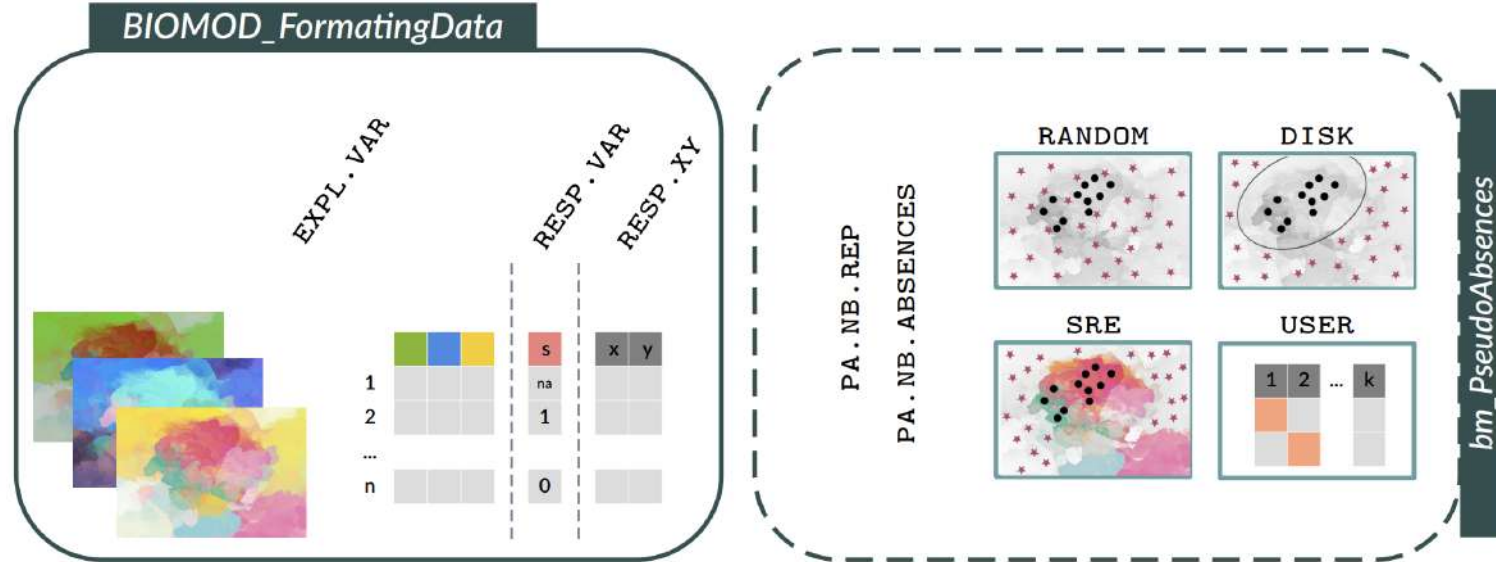
» presences-absences



Species occ
Environment

1. Formating data

- » **presence-only** data
- » *avoid to mix with real absences*



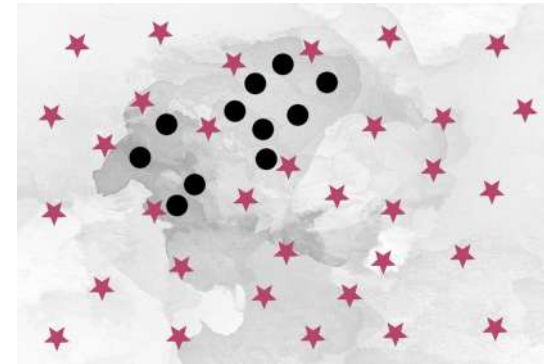
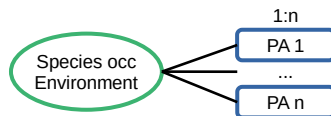
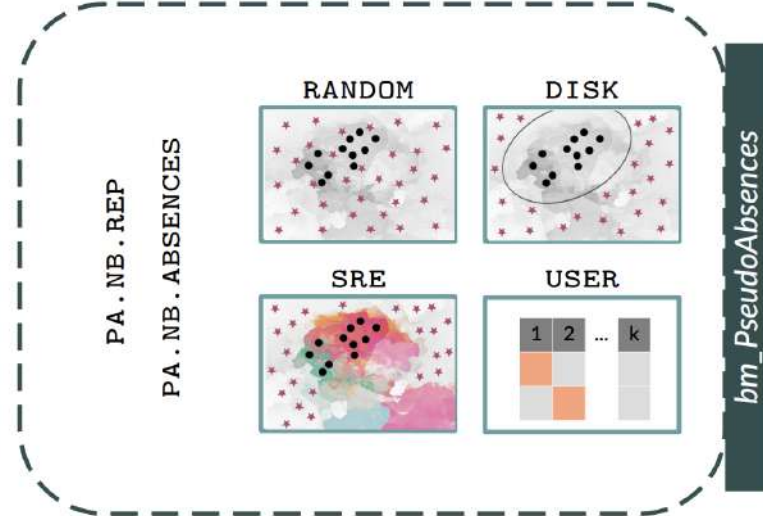
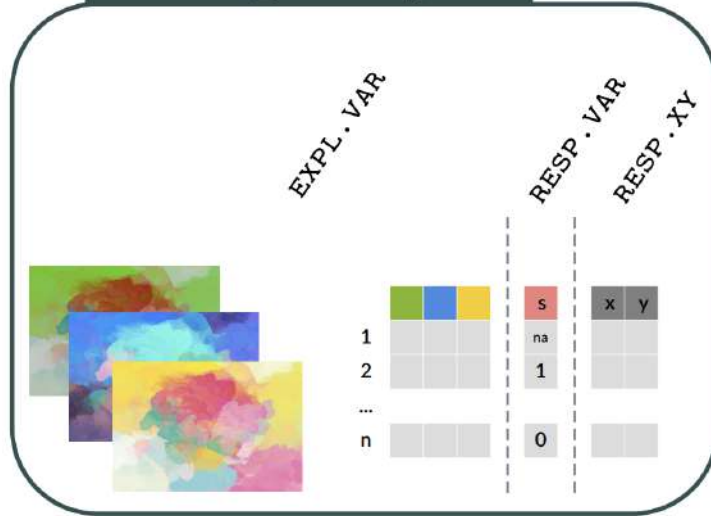
Species occ
Environment



1. Formating data

- » **presence-only** data
- » *avoid to mix with real absences*
- » **random** : sampling potentially biased / non-exhaustive

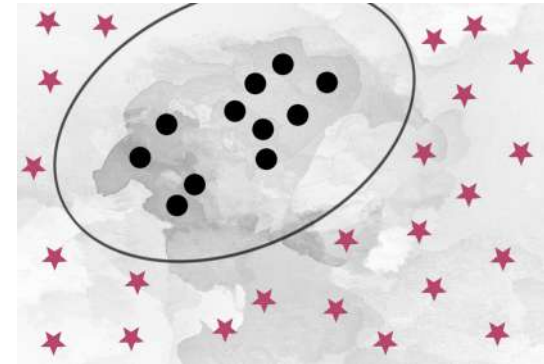
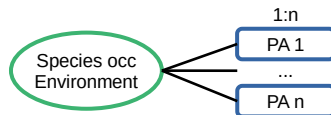
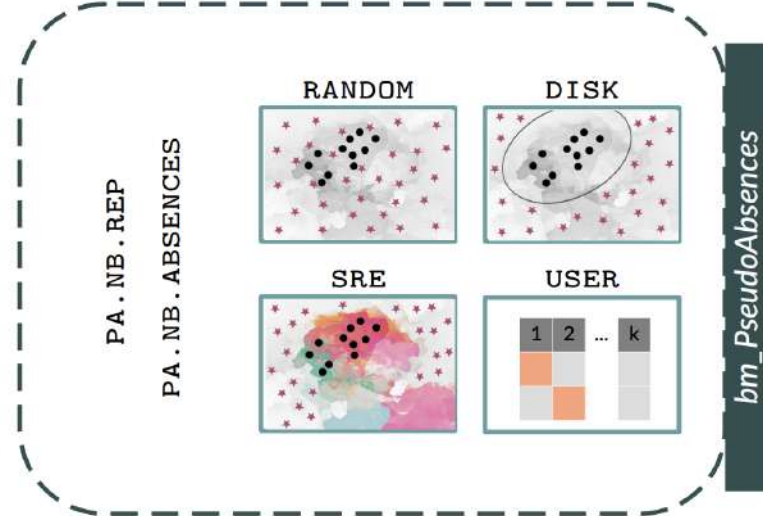
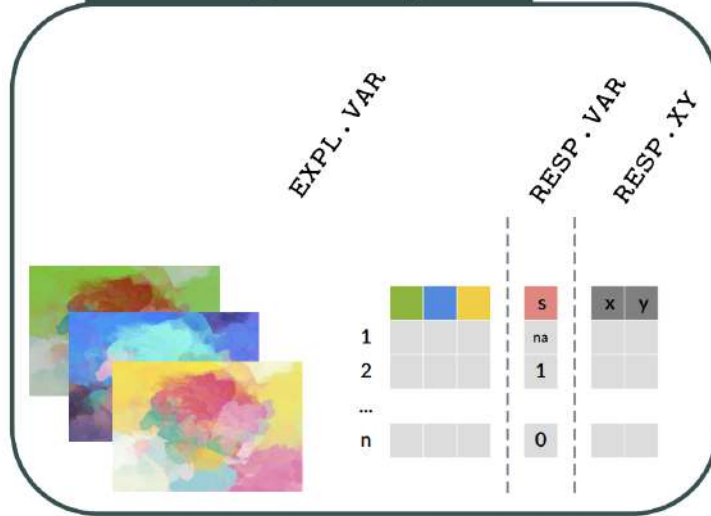
BIOMOD_FormatingData



1. Formating data

- » **presence-only** data
- » *avoid to mix with real absences*
- » **random** : sampling potentially biased / non-exhaustive
- » **disk** : geographic niche well sampled

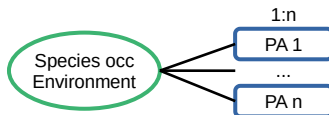
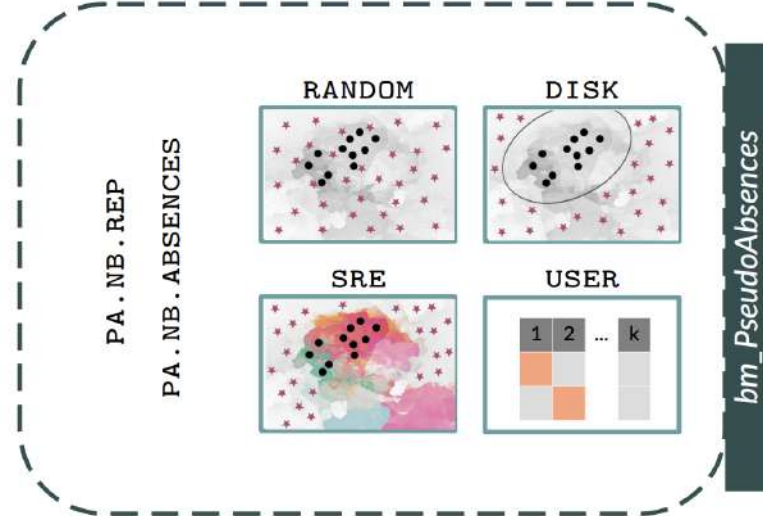
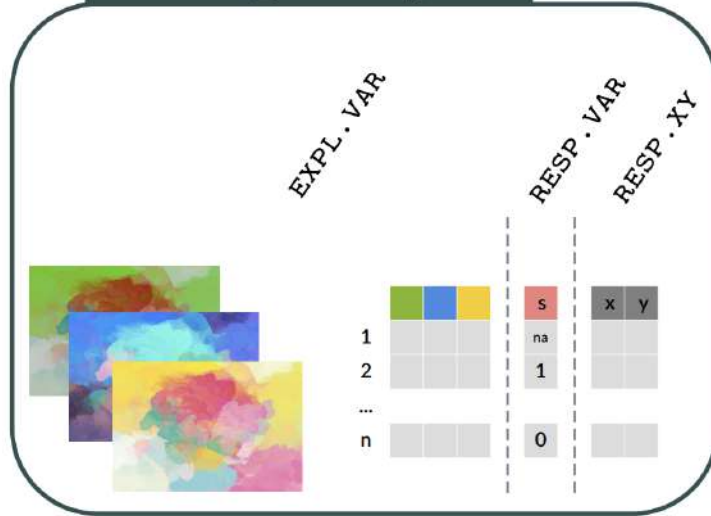
BIOMOD_FormatingData



1. Formating data

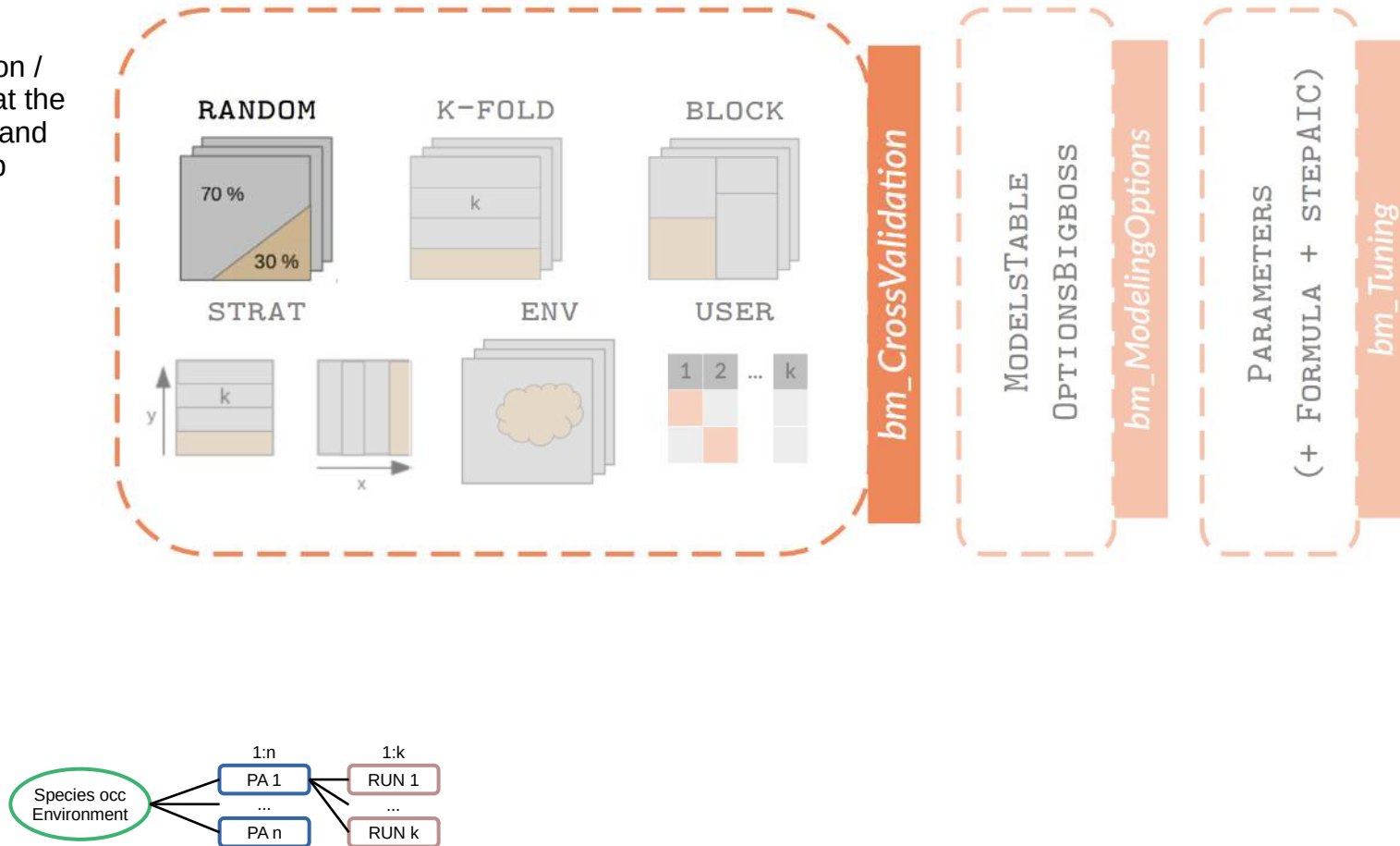
- » **presence-only** data
- » *avoid to mix with real absences*
- » **random** : sampling potentially biased / non-exhaustive
- » **disk** : geographic niche well sampled
- » **SRE** : environmental niche well sampled

BIOMOD_FormatingData



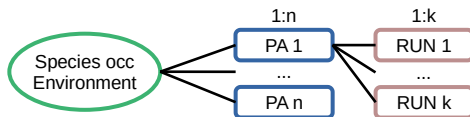
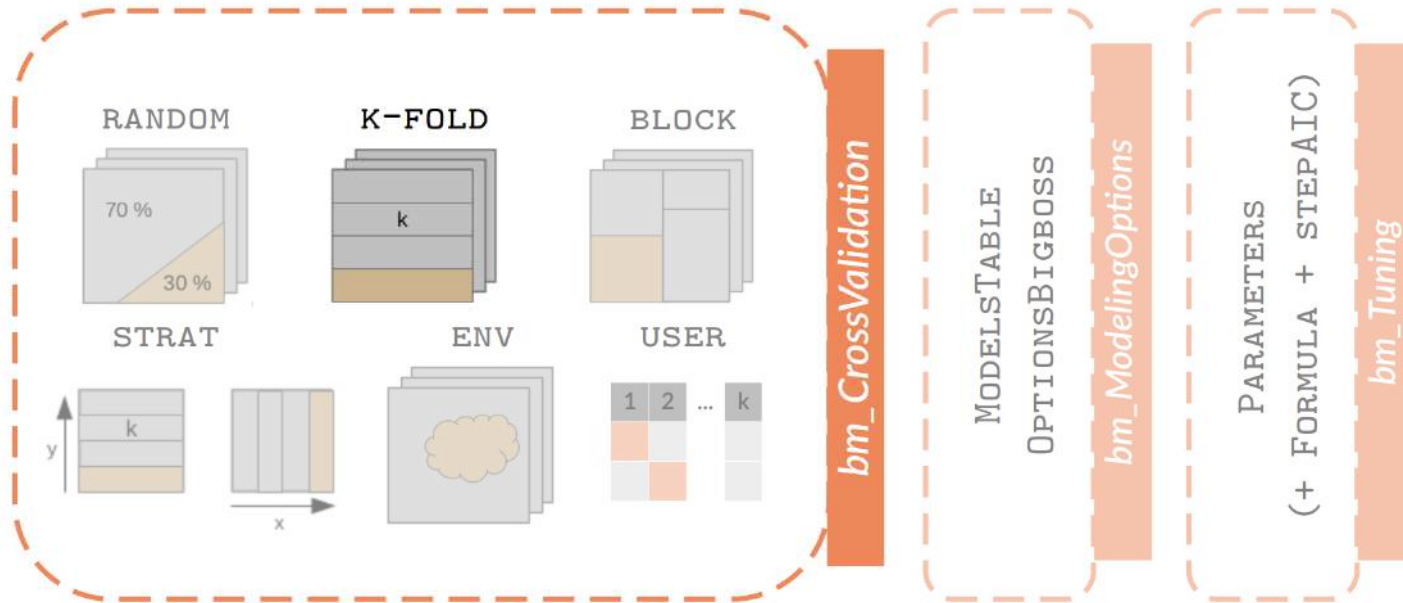
1. Formating data

- » simple calibration / validation split at the modeling step, and repeated nb.rep times



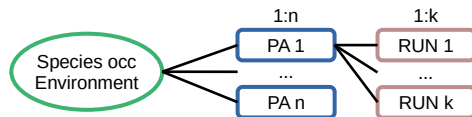
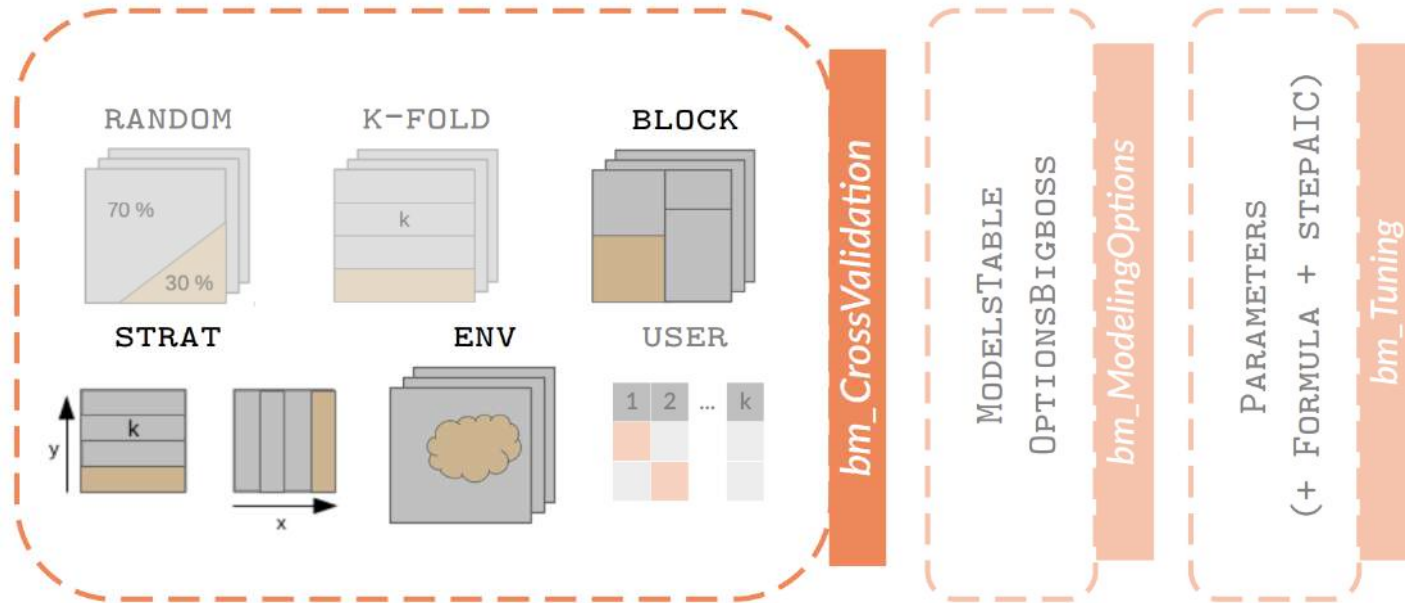
1. Formating data

- » simple calibration / validation split at the modeling step, and repeated nb.rep times
- » **k-fold** : partition data into k sub-dataset, and repeated nb.rep times



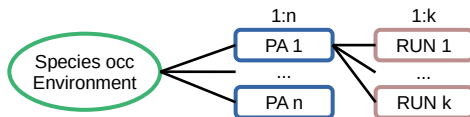
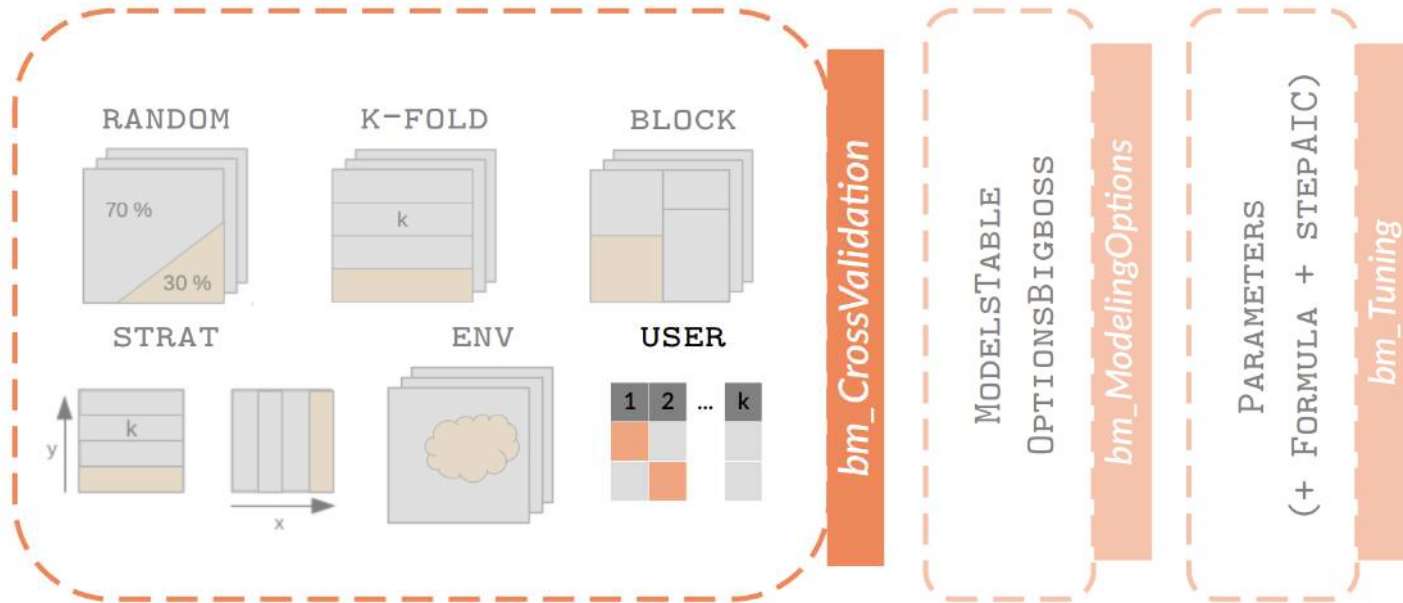
1. Formating data

- » simple calibration / validation split at the modeling step, and repeated nb.rep times
- » **k-fold** : partition data into k sub-dataset, and repeated nb.rep times
- » **stratified** : partition data into k sub-dataset (x, y, both, block, env)



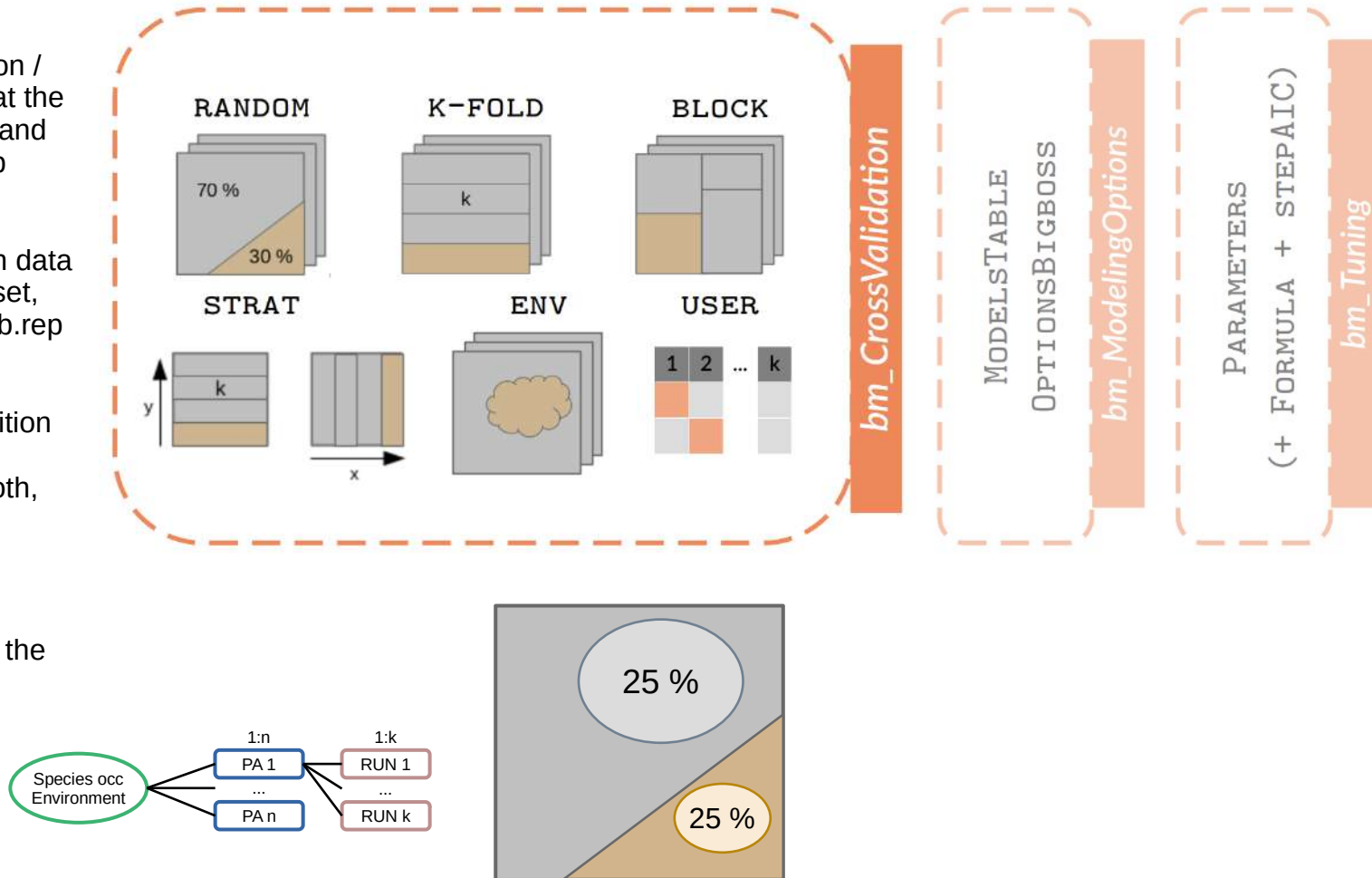
1. Formating data

- › simple calibration / validation split at the modeling step, and repeated nb.rep times
- › **k-fold** : partition data into k sub-dataset, and repeated nb.rep times
- › **stratified** : partition data into k sub-dataset (x, y, both, block, env)
- › **user defined**



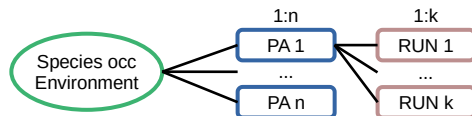
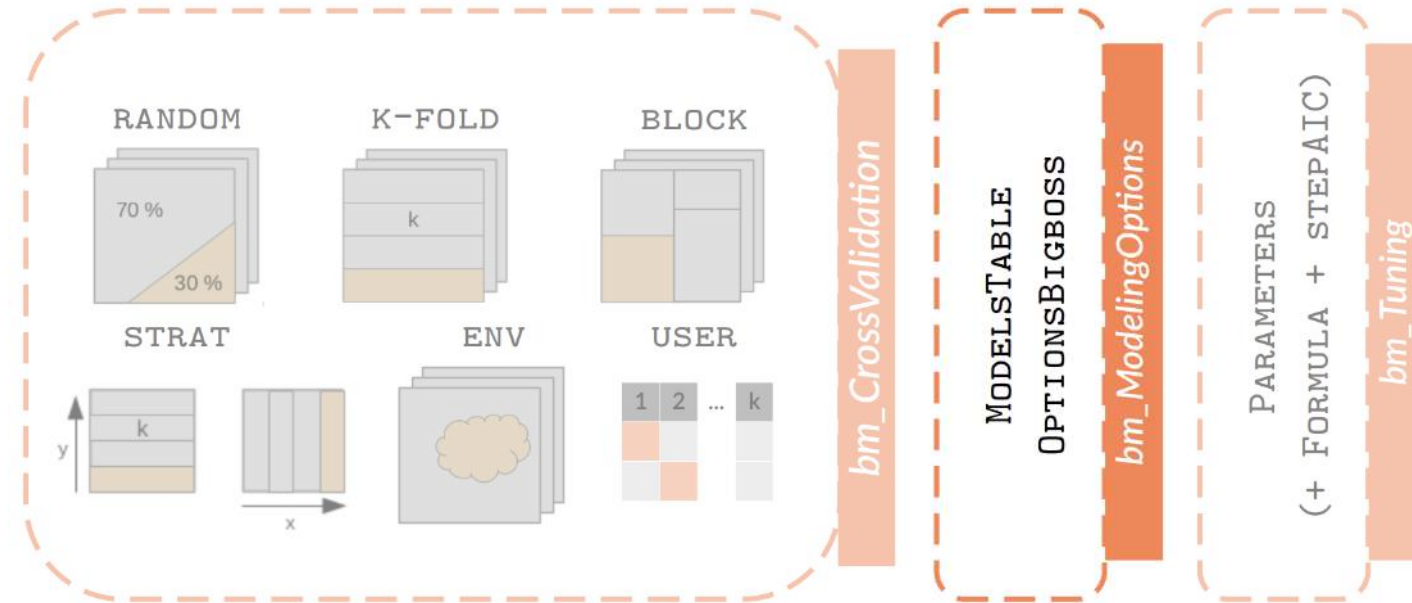
1. Formating data

- » simple calibration / validation split at the modeling step, and repeated nb.rep times
- » **k-fold** : partition data into k sub-dataset, and repeated nb.rep times
- » **stratified** : partition data into k sub-dataset (x, y, both, block, env)
- » **user defined**
- » **balance** : keep the prevalence of presences (or absences) in sub-dataset



1. Formating data

- » 11 types of models, 14 single models
- » 1 coded in biomod2, 1 external software, 12 other R packages



```
> ModelsTable
```

	model	type	package	func	train
1	ANN	binary	nnet	nnet	avNNet
2	CTA	binary	rpart	rpart	rpart
3	FDA	binary	mda	fda	fda
4	GAM	binary	gam	gam	gamLoess
5	GAM	binary	mgcv	bam	bam
6	GAM	binary	mgcv	gam	gam
7	GBM	binary	gbm	gbm	gbm
8	GLM	binary	stats	glm	glm
9	MARS	binary	earth	earth	earth
10	MAXENT	binary	MAXENT	MAXENT	ENMevaluate
11	MAXNET	binary	maxnet	maxnet	maxnet
12	RF	binary	randomForest	randomForest	rf
13	SRE	binary	biomod2	bm_SRE	bm_SRE
14	XGB00ST	binary	xgboost	xgboost	xgbTree

1. Formating data

- » 11 types of models, 14 single models
- » 1 coded in biomod2, 1 external software, 12 other R packages
- » **default** : extracted from functions

```
nnet                                package:nnet                                R Documentation

Fit Neural Networks

Description:

  Fit single-hidden-layer neural network, possibly with skip-layer
  connections.

Usage:

  nnet(x, ...)

## S3 method for class 'formula'
nnet(formula, data, weights, ...,
      subset, na.action, contrasts = NULL)

## Default S3 method:
nnet(x, y, weights, size, Wts, mask,
      linout = FALSE, entropy = FALSE, softmax = FALSE,
      censored = FALSE, skip = FALSE, rang = 0.7, decay = 0,
      maxit = 100, Hess = FALSE, trace = TRUE, MaxNWts = 1000,
      abstol = 1.0e-4, reltol = 1.0e-8, ...)
```

MODELSTABLE

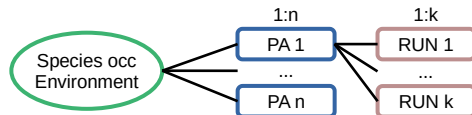
OPTIONSBIGBOSS

bm_ModelingOptions

PARAMETERS

(+ FORMULA + STEPAIC)

bm_Tuning

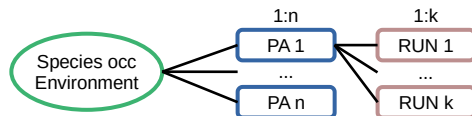


```
> ModelsTable
  model type package func train
1 ANN binary nnet nnet avNNet
2 CTA binary rpart rpart rpart
3 FDA binary mda fda fda
4 GAM binary gam gam gamLoess
5 GAM binary mgcv bam bam
6 GAM binary mgcv gam gam
7 GBM binary gbm gbm gbm
8 GLM binary stats glm glm
9 MARS binary earth earth earth
10 MAXENT binary MAXENT MAXENT ENMevaluate
11 MAXNET binary maxnet maxnet maxnet
12 RF binary randomForest randomForest rf
13 SRE binary biomod2 bm_SRE bm_SRE
14 XGB00ST binary xgboost xgboost xgbTree
```

1. Formating data

- » 11 types of models, 14 single models
- » 1 coded in biomod2, 1 external software, 12 other R packages
- » **default** : extracted from functions
- » **bigboss** : redefined by biomod2 team

```
nnet package:nnet R Documentation
Fit Neural Networks
Description:
  Fit single-hidden-layer neural network, possibly with skip-layer
  connections.
Usage:
  nnet(x, ...)
  ## S3 method for class 'formula'
  nnet(formula, data, weights, ...,
        subset, na.action, contrasts = NULL)
  ## Default S3 method:
  nnet(x, y, weights, size, Wts, mask,
        linout = FALSE, entropy = FALSE, softmax = FALSE,
        censored = FALSE, skip = FALSE, rang = 0.7, decay = 0,
        maxit = 100, Hess = FALSE, trace = TRUE, MaxNWts = 1000,
        abstol = 1.0e-4, reltol = 1.0e-8, ...)
> ANN options (datatype: binary , package: nnet , function: nnet ) :
( dataset: allData allRun )
- size = 5 (default: 2 )
- decay = 5 (default: NULL )
- trace = FALSE (default: NULL )
- rang = 0.1 (default: NULL )
- maxit = 200 (default: NULL )
```

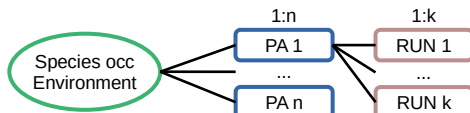


	model	type	package	func	train
1	ANN	binary	nnet	nnet	avNNet
2	CTA	binary	rpart	rpart	rpart
3	FDA	binary	mda	fda	fda
4	GAM	binary	gam	gam	gamLoess
5	GAM	binary	mgcv	bam	bam
6	GAM	binary	mgcv	gam	gam
7	GBM	binary	gbm	gbm	gbm
8	GLM	binary	stats	glm	glm
9	MARS	binary	earth	earth	earth
10	MAXENT	binary	MAXENT	MAXENT	ENMevaluate
11	MAXNET	binary	maxnet	maxnet	maxnet
12	RF	binary	randomForest	randomForest	rf
13	SRE	binary	biomod2	bm_SRE	bm_SRE
14	XGB00ST	binary	xgboost	xgboost	xgbTree

1. Formating data

- » 11 types of models, 14 single models
- » 1 coded in biomod2, 1 external software, 12 other R packages
- » **default** : extracted from functions
- » **bigboss** : redefined by biomod2 team
- » **user-defined**

```
nnet package:nnet R Documentation
Fit Neural Networks
Description:
  Fit single-hidden-layer neural network, possibly with skip-layer
  connections.
Usage:
  nnet(x, ...)
  ## S3 method for class 'formula'
  nnet(formula, data, weights, ...,
        subset, na.action, contrasts = NULL)
  ## Default S3 method:
  nnet(x, y, weights, size, Wts, mask,
        linout = FALSE, entropy = FALSE, softmax = FALSE,
        censored = FALSE, skip = FALSE, rang = 0.7, decay = 0,
        maxit = 100, Hess = FALSE, trace = TRUE, MaxNWts = 1000,
        abstol = 1.0e-4, reltol = 1.0e-8, ...)
> ANN options (datatype: binary , package: nnet , function: nnet ) :
( dataset allData allRun )
- size = 5 (default: 2 )
- decay = 5 (default: NULL )
- trace = FALSE (default: NULL )
- rang = 0.1 (default: NULL )
- maxit = 200 (default: NULL )
```

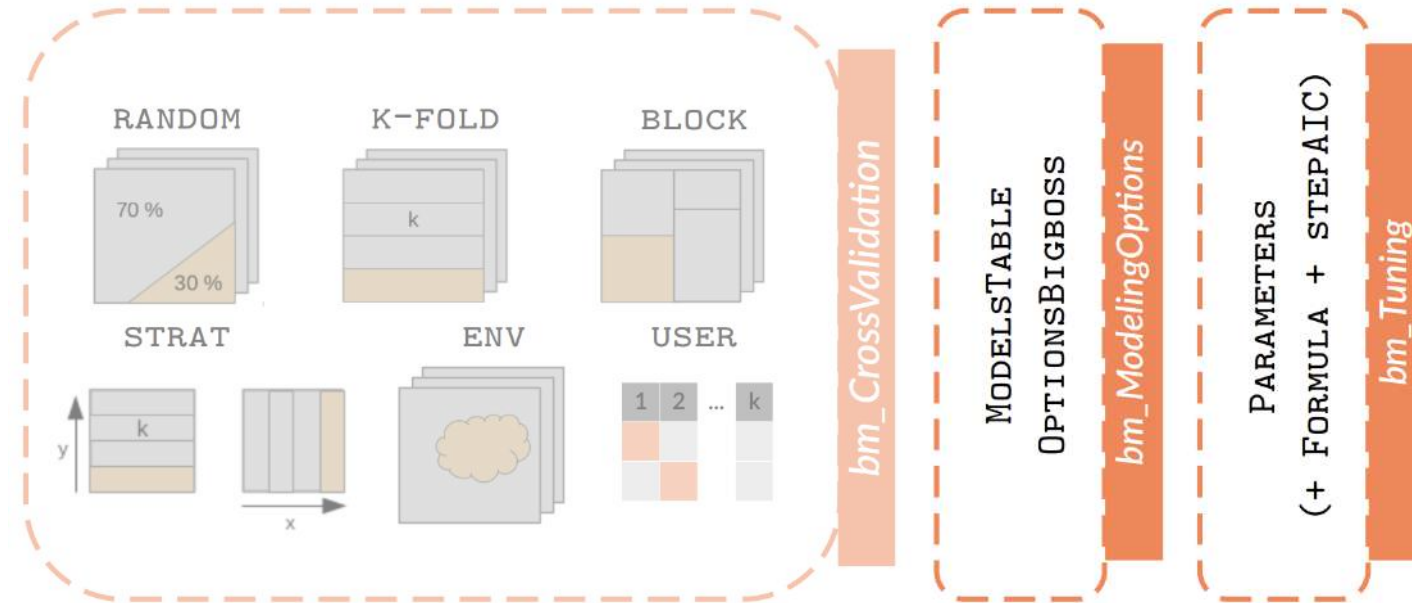
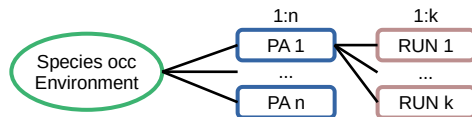


	model	type	package	func	train
1	ANN	binary	nnet	nnet	avNNet
2	CTA	binary	rpart	rpart	rpart
3	FDA	binary	mda	fda	fda
4	GAM	binary	gam	gam	gamLoess
5	GAM	binary	mgcv	bam	bam
6	GAM	binary	mgcv	gam	gam
7	GBM	binary	gbm	gbm	gbm
8	GLM	binary	stats	glm	glm
9	MARS	binary	earth	earth	earth
10	MAXENT	binary	MAXENT	MAXENT	ENMevaluate
11	MAXNET	binary	maxnet	maxnet	maxnet
12	RF	binary	randomForest	randomForest	rf
13	SRE	binary	biomod2	bm_SRE	bm_SRE
14	XGBOOST	binary	xgboost	xgboost	xgbTree

user.ANN = list('_allData_allRun' =
list(size = 5,
decay = 0.5,
trace = FALSE,
rang = 0.1,
maxit = 500))

1. Formating data

- » 11 types of models, 14 single models
- » 1 coded in biomod2, 1 external software, 12 other R packages
- » **default** : extracted from functions
- » **bigboss** : redefined by biomod2 team
- » **user-defined**
- » **tuned** : with *train* function from *caret* package

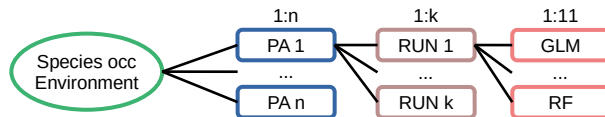
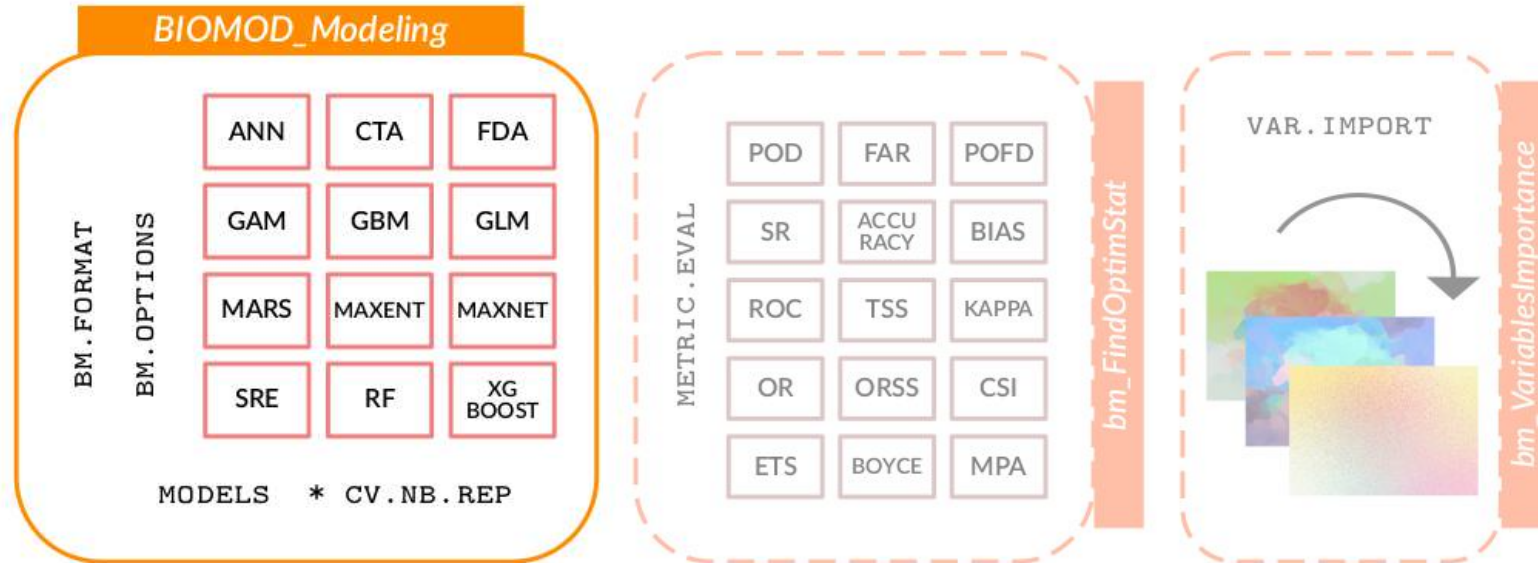


```

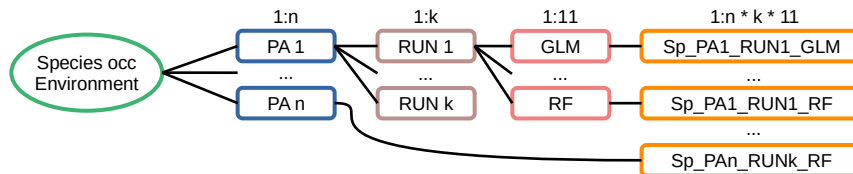
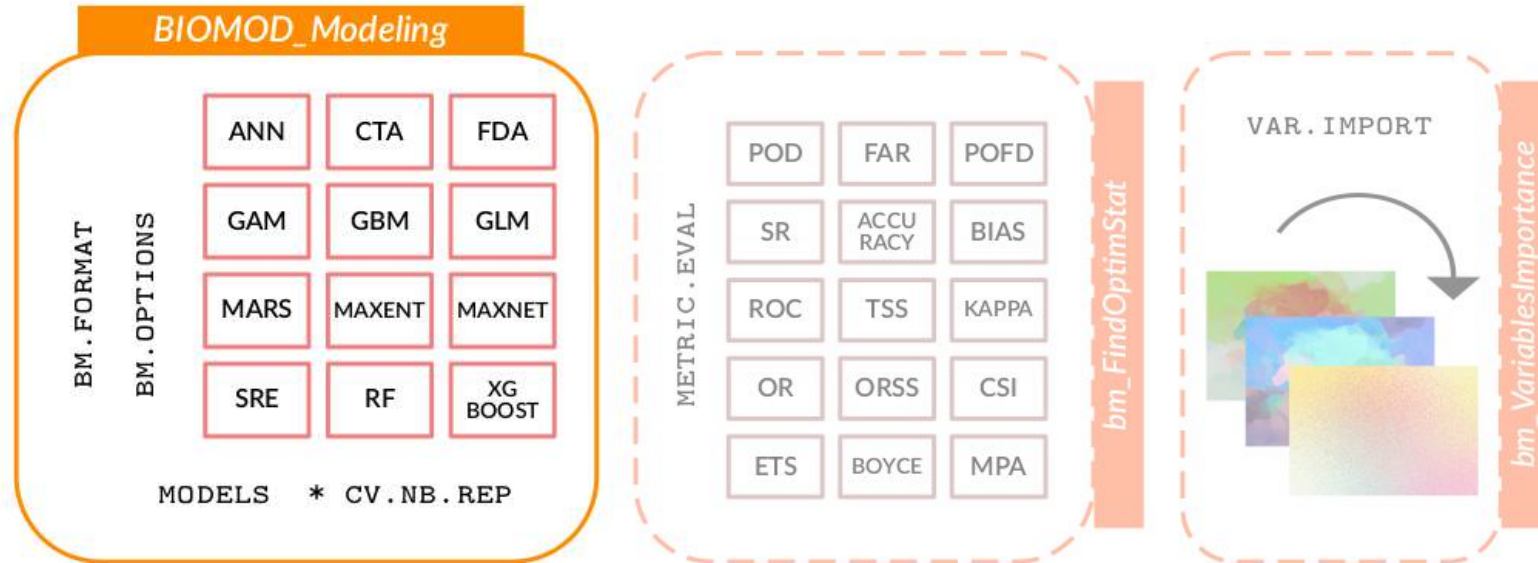
> ModelsTable
  model type package func train
1 ANN binary nnet nnet avNNet
2 CTA binary rpart rpart rpart
3 FDA binary mda fda fda
4 GAM binary gam gam gamLoess
5 GAM binary mgcv bam bam
6 GAM binary mgcv gam gam
7 GBM binary gbm gbm gbm
8 GLM binary stats glm glm
9 MARS binary earth earth earth
10 MAXENT binary MAXENT MAXENT ENMevaluate
11 MAXNET binary maxnet maxnet maxnet
12 RF binary randomForest randomForest rf
13 SRE binary biomod2 bm_SRE bm_SRE
14 XGB00ST binary xgboost xgboost xgbTree
    
```

- » test a bunch of parameters, and try to keep the « best » according to some evaluation metrics (TSS or ROC)

2.a Single models



2.a Single models

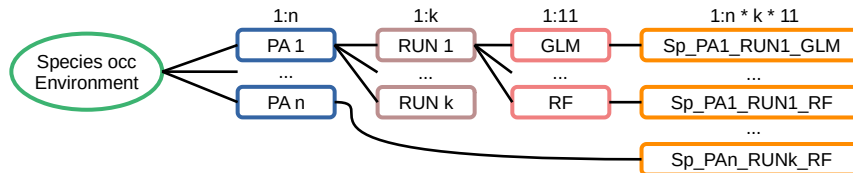
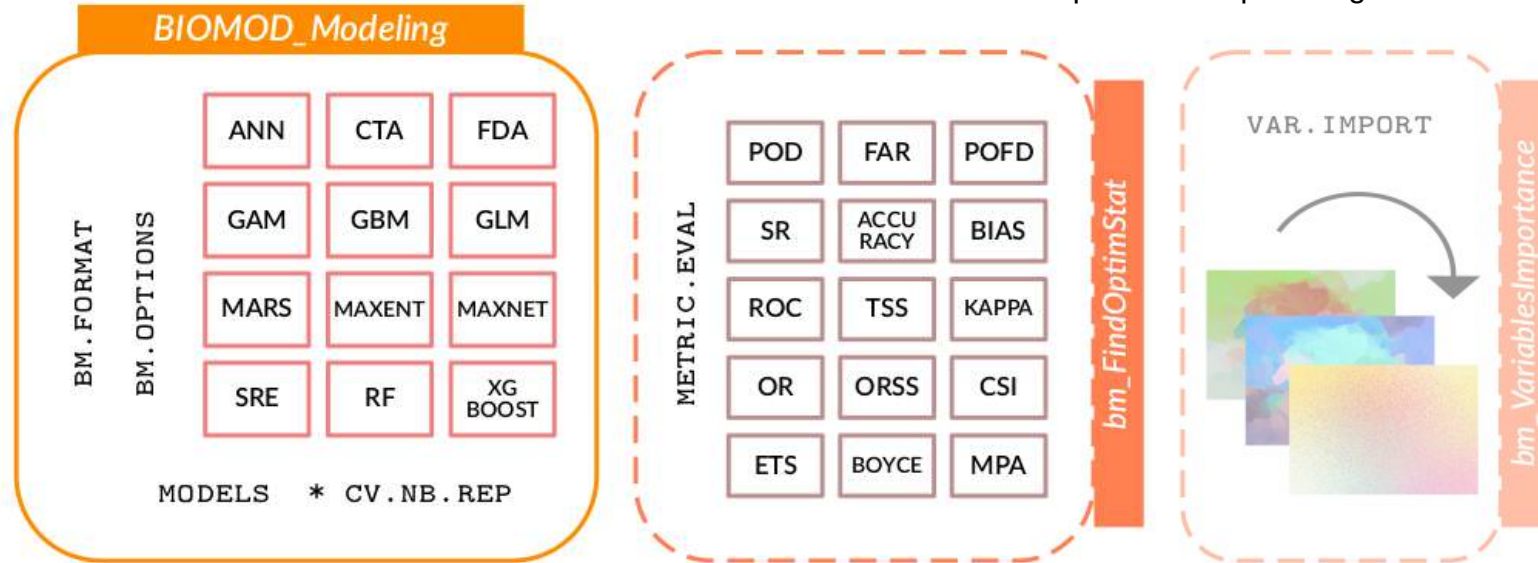


2.a Single models

» except ROC, all evaluation metrics obtained from contingency table (*containing TP, FP, TN, FN*)

» require a **binary transformation** :

- range of thresholds tested
- keep threshold optimising the evaluation metric

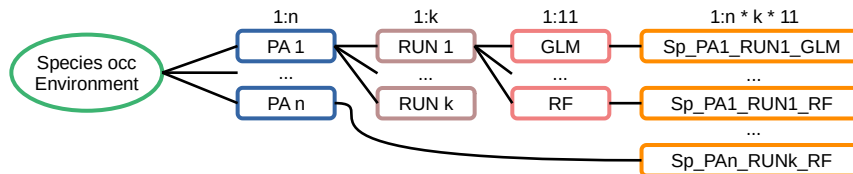
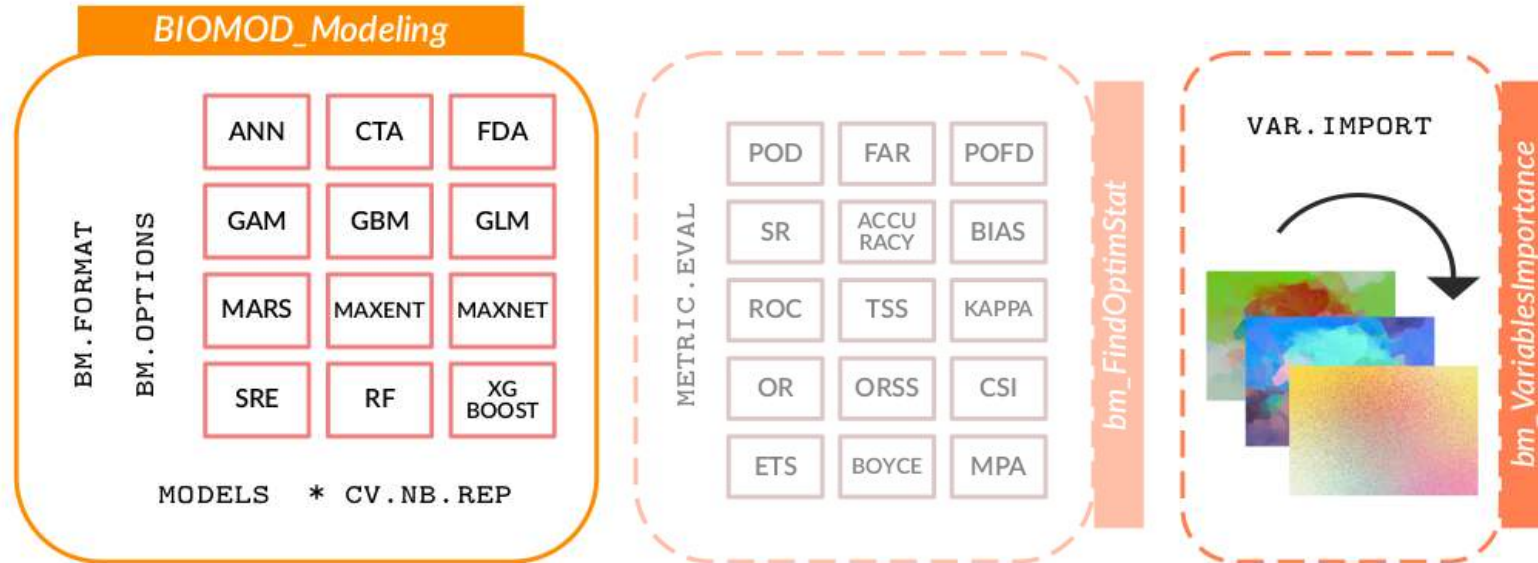


2.a Single models

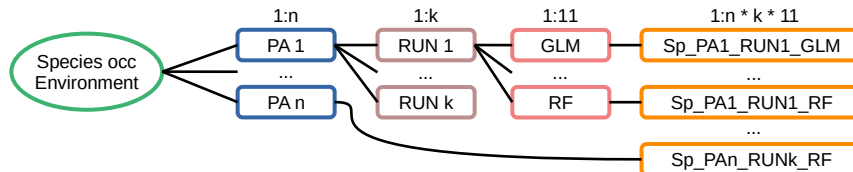
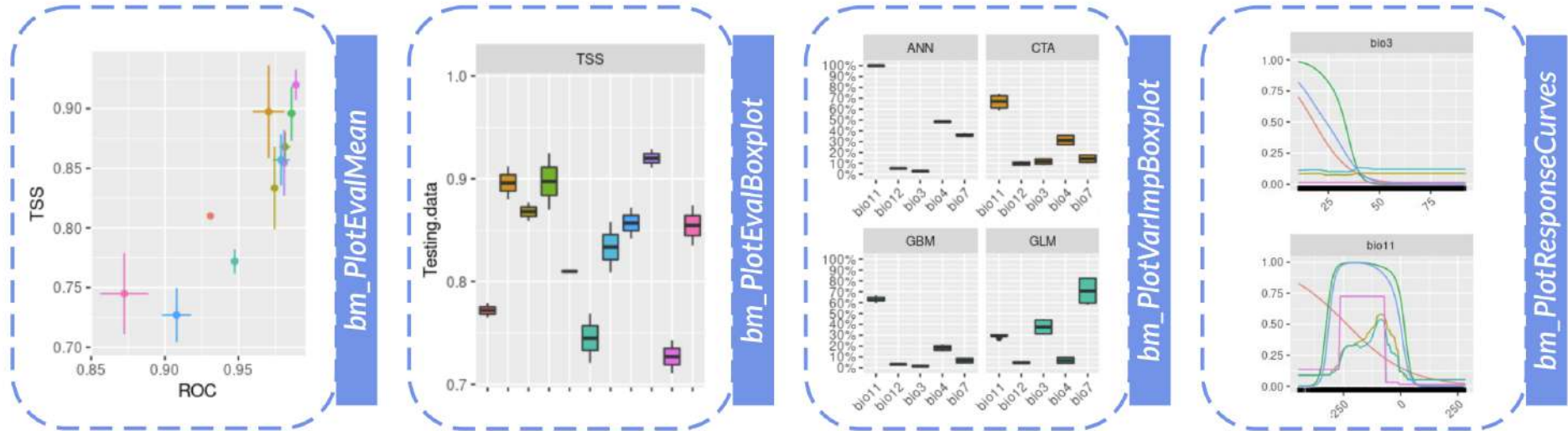
» comparison of importance of variables between models

» **Pearson correlation** between :

- normal prediction
- prediction with 1 variable randomised

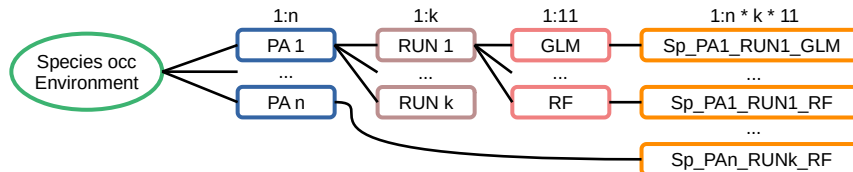
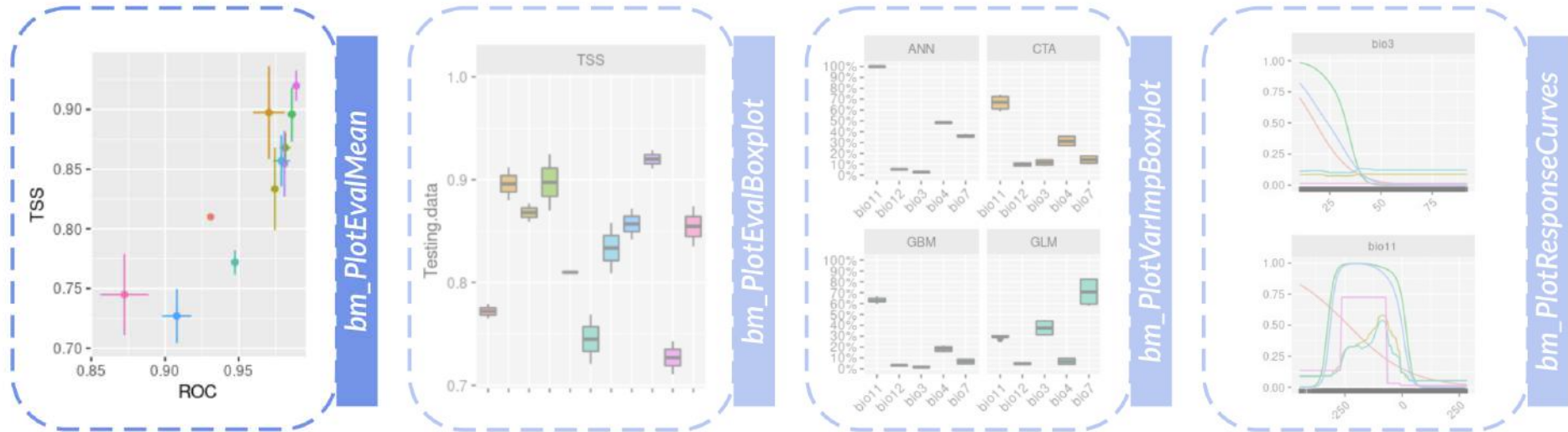


3.a Exploring single models



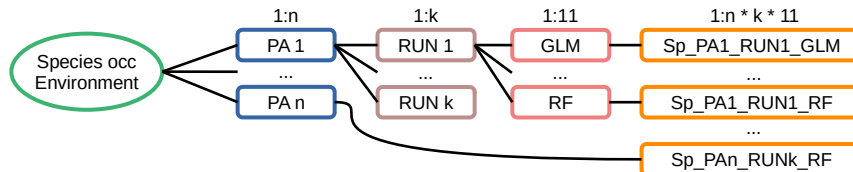
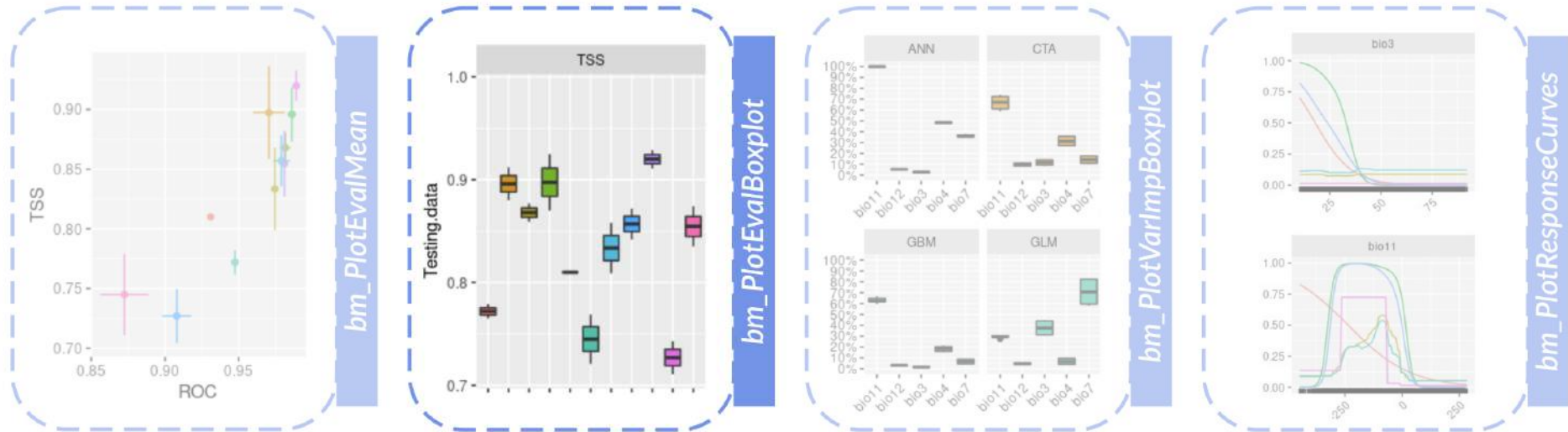
3.a Exploring single models

- » « evaluation space »
- » visualize the metrics consistency between models



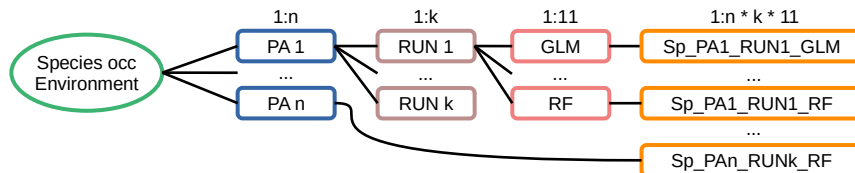
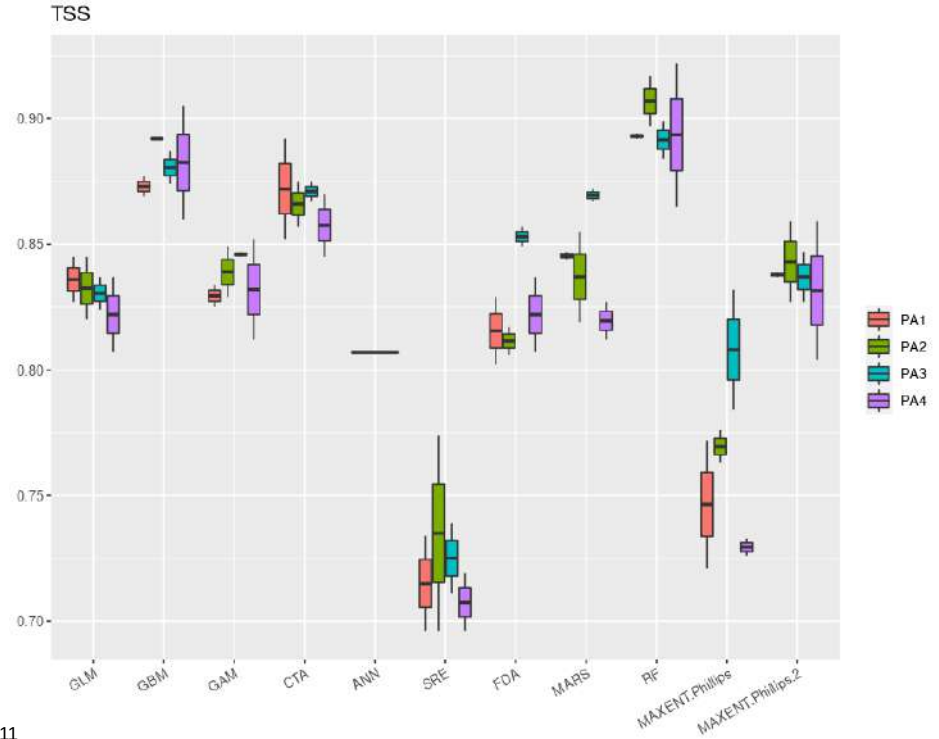
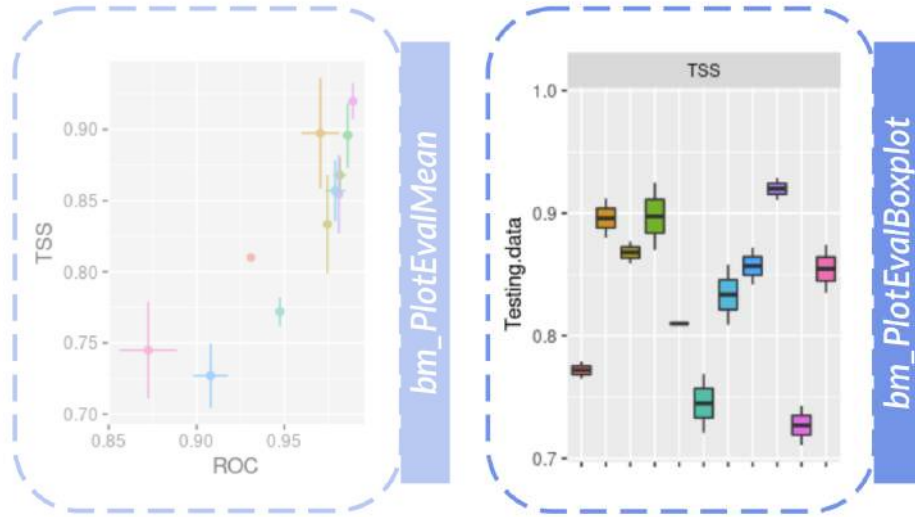
3.a Exploring single models

- » more classical view
- » visualize the metrics consistency between models



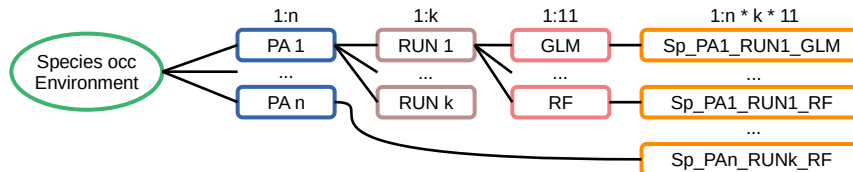
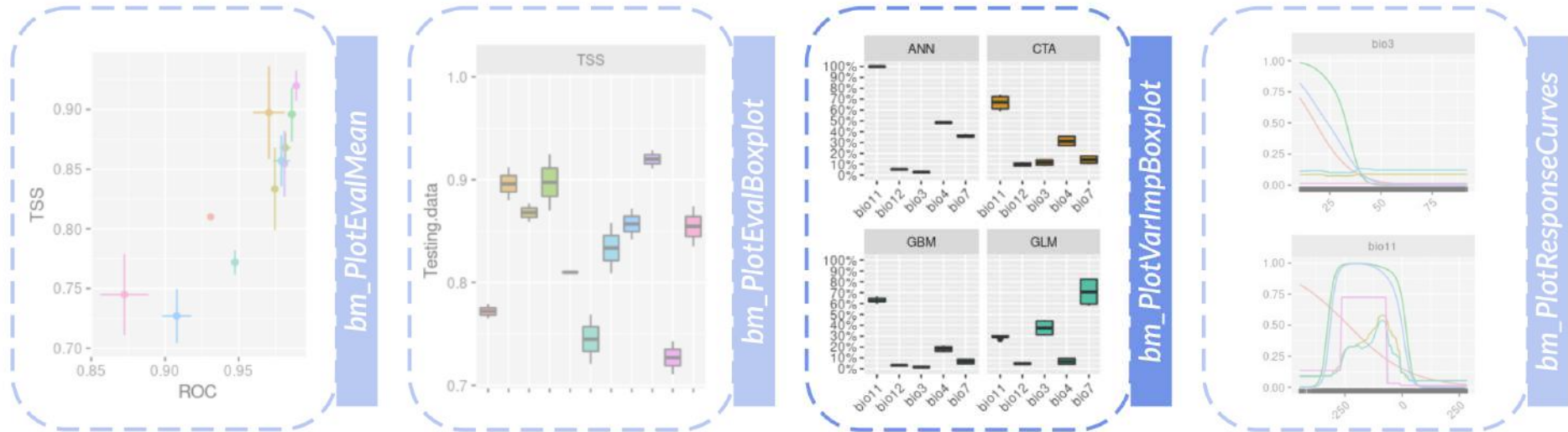
3.a Exploring single models

- » more classical view
- » visualize the metrics consistency between models
 - explore the different levels of subsets



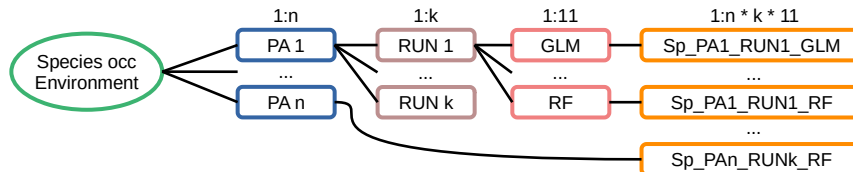
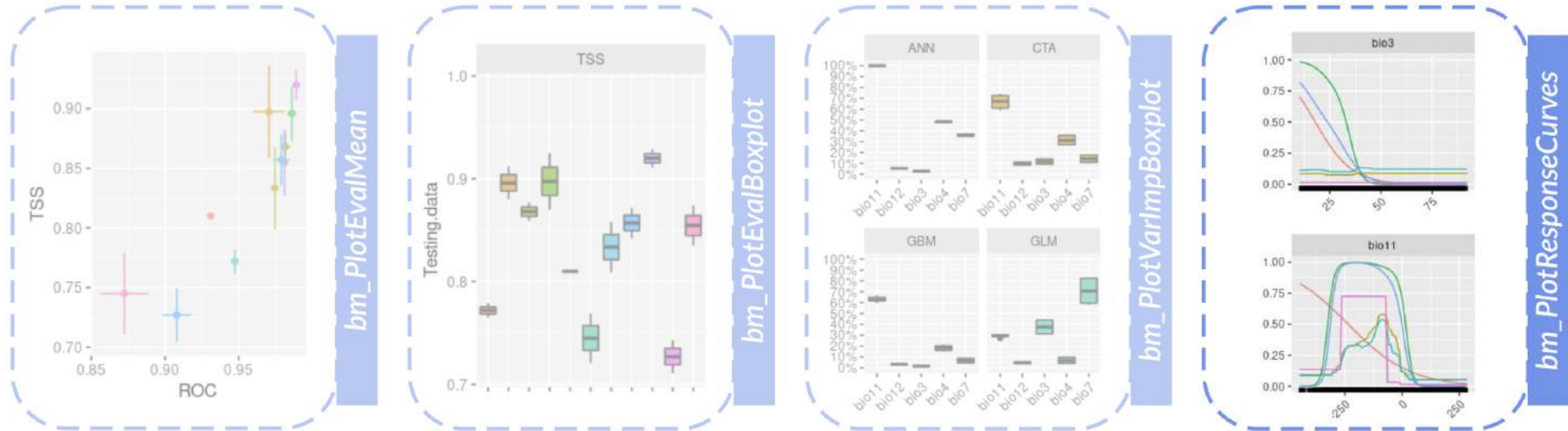
3.a Exploring single models

- » compare importance of variables between models
- » visualize the consistency between models
(and different types of models)

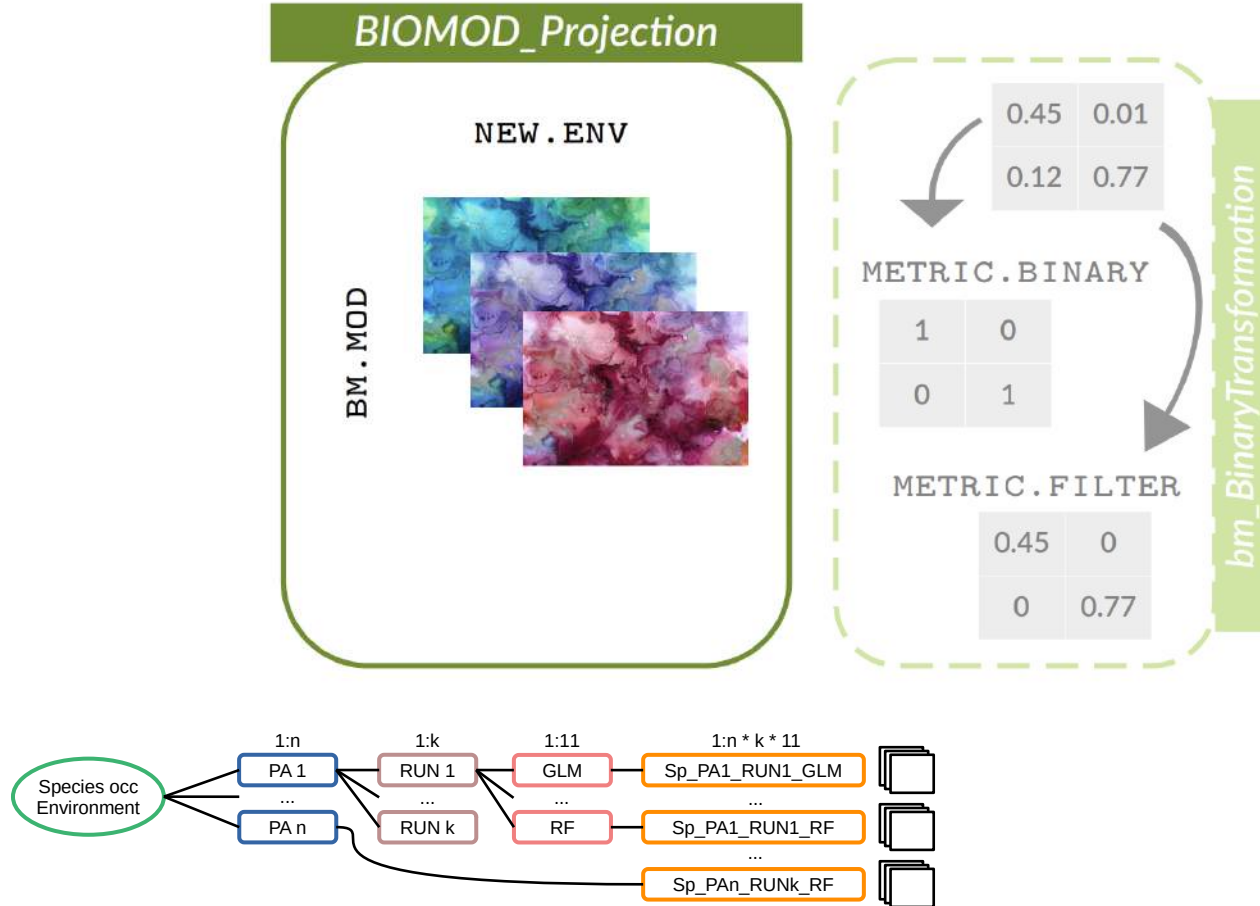


3.a Exploring single models

- » better understand the effect of each variable along its gradient onto the probability of presence
- » visualize the consistency between models (and different types of models)

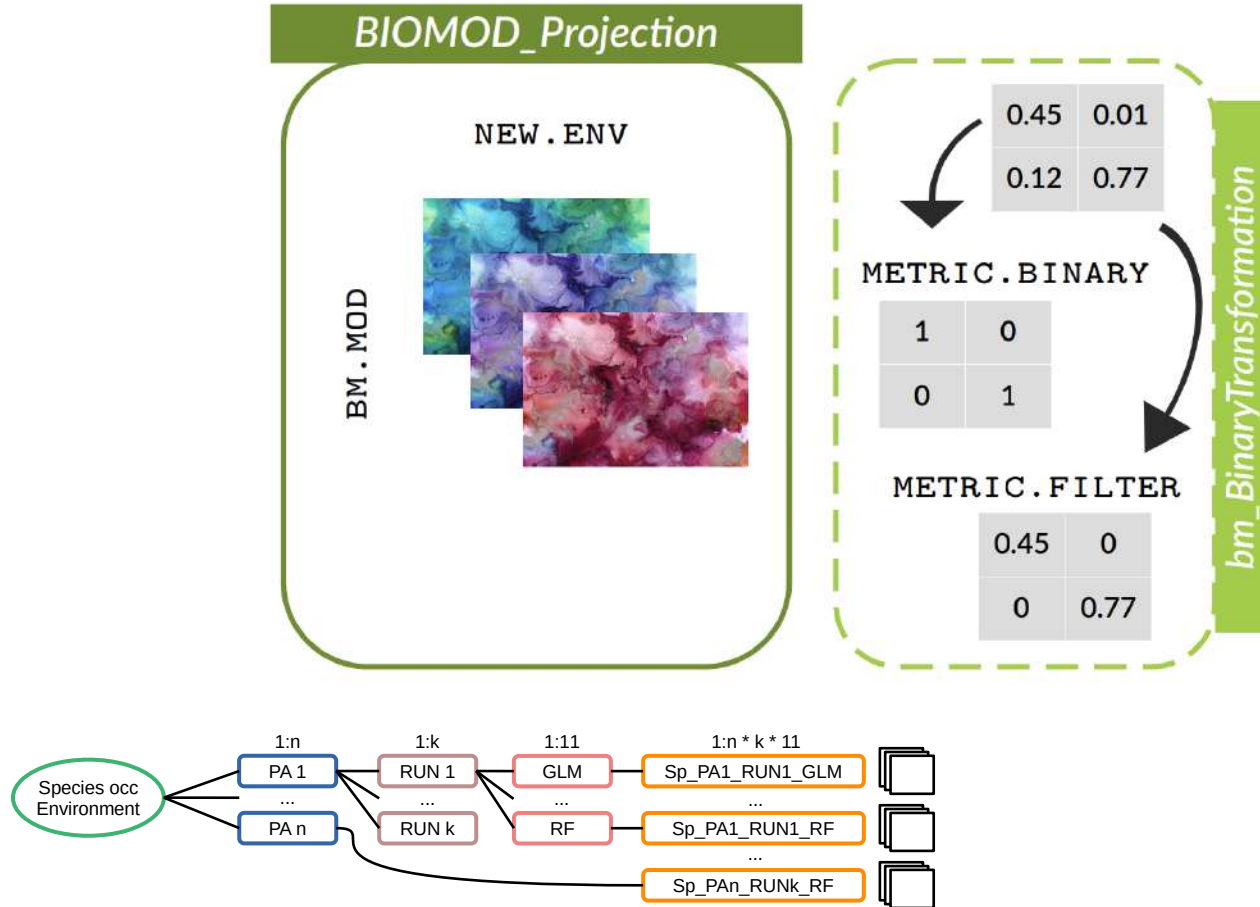


4.a Projecting single models

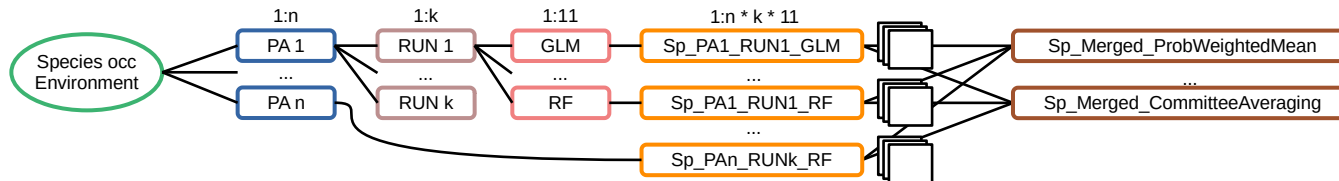
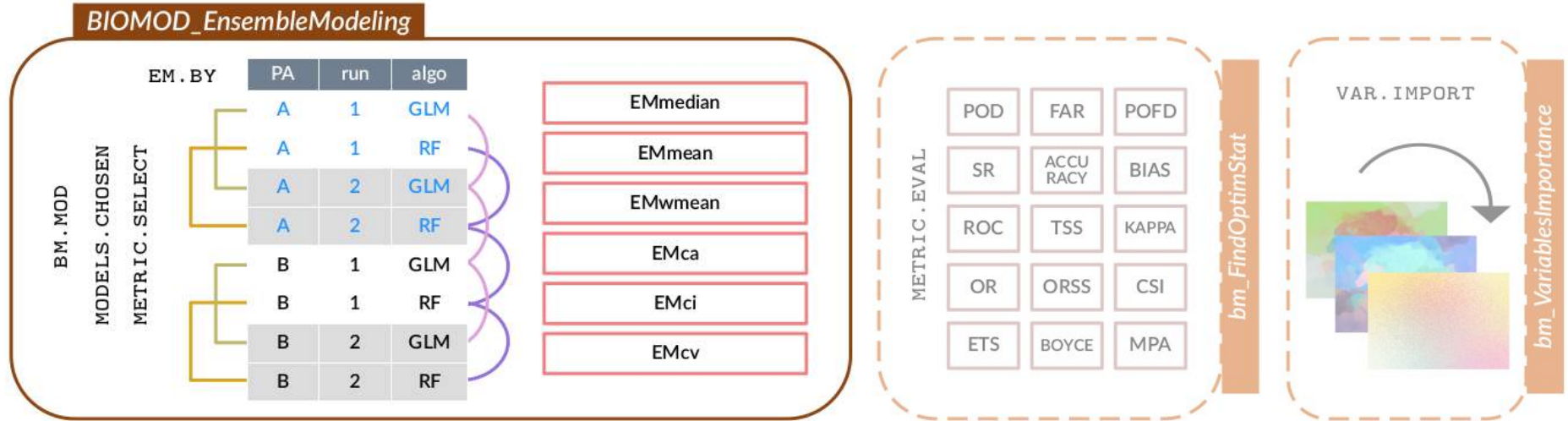


4.a Projecting single models

- » transformation associated to one evaluation metric
(one map created for each metric selected)
- » use the threshold maximising the chosen metric



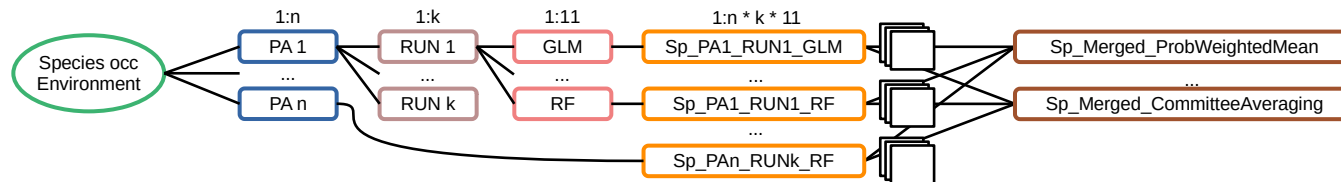
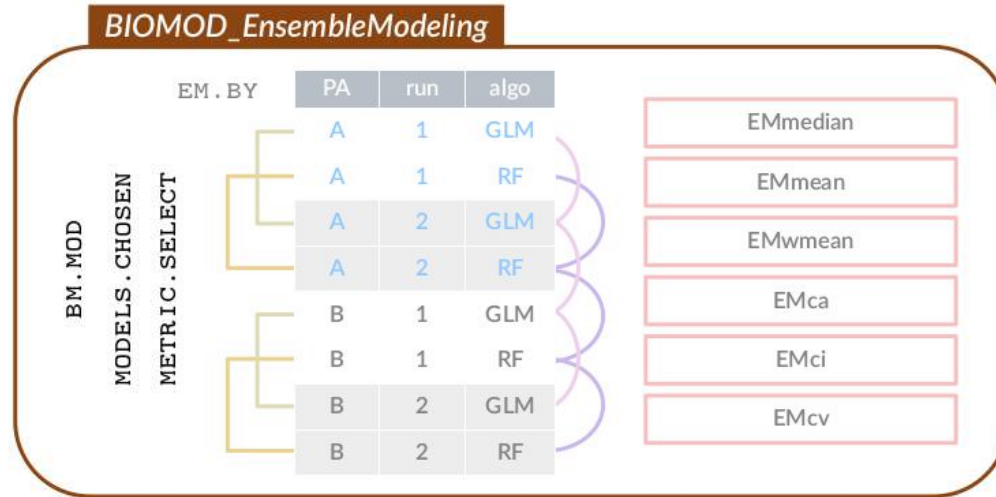
2.b Ensemble models



2.b Ensemble models

Step 1 : filter single models

- » **filtering** associated to one evaluation metric
(one set of ensemble models created for each metric selected)
- » use a threshold to keep single models



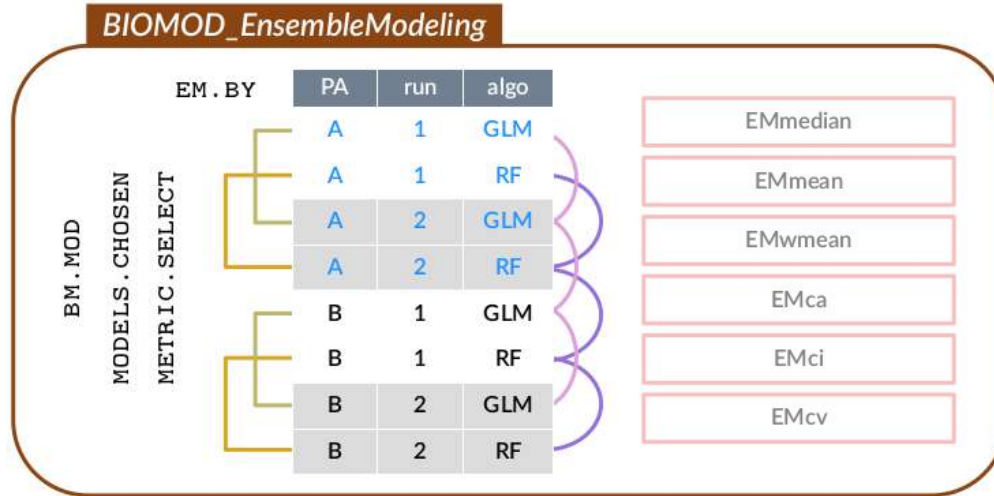
2.b Ensemble models

Step 1 : filter single models

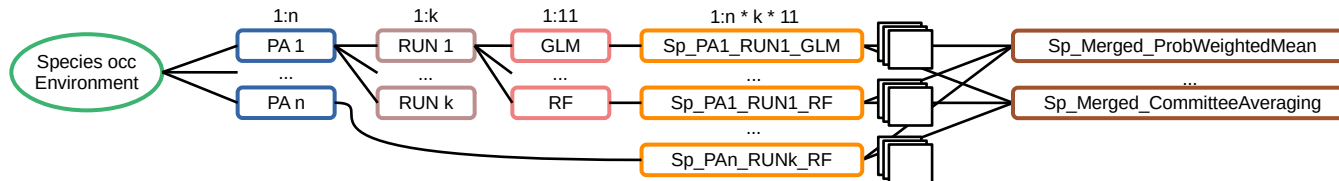
Step 2 : gather single models

» different ways of **combining** single models together :

- *all*
- *algo*
- *PA*
- *PA+algo*
- *PA+run*



PA1_RUN1_GLM	PA1_RUN1_RF
PA1_RUN2_GLM	PA1_RUN2_RF
PA1_RUN3_GLM	PA1_RUN3_RF
PA2_RUN1_GLM	PA2_RUN1_RF
PA2_RUN2_GLM	PA2_RUN2_RF
PA2_RUN3_GLM	PA2_RUN3_RF



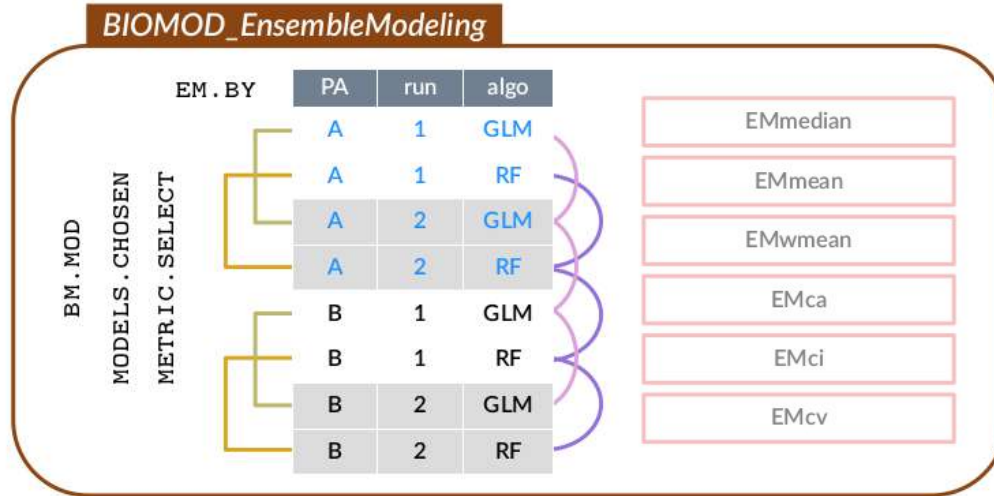
2.b Ensemble models

Step 1 : filter single models

Step 2 : gather single models

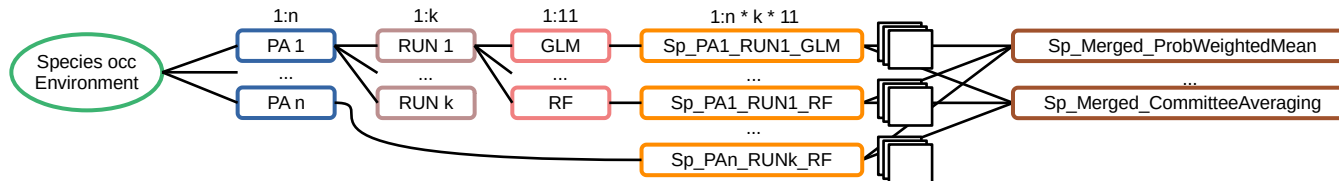
» different ways of **combining** single models together :

- *all*
- *algo*
- *PA*
- *PA+algo*
- *PA+run*



PA1_RUN1_GLM	PA1_RUN1_RF
PA1_RUN2_GLM	PA1_RUN2_RF
PA1_RUN3_GLM	PA1_RUN3_RF
PA2_RUN1_GLM	PA2_RUN1_RF
PA2_RUN2_GLM	PA2_RUN2_RF
PA2_RUN3_GLM	PA2_RUN3_RF

All models



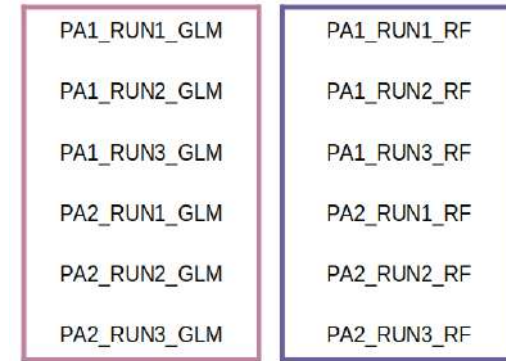
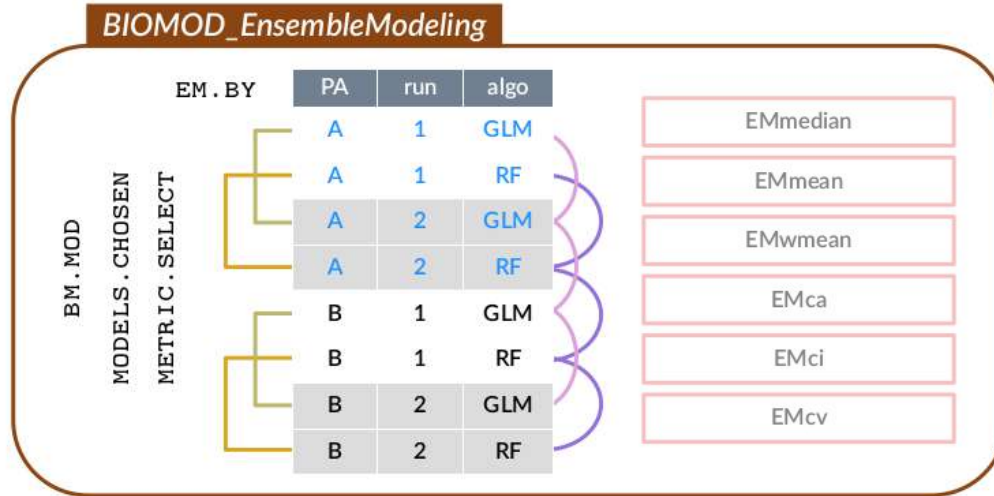
2.b Ensemble models

Step 1 : filter single models

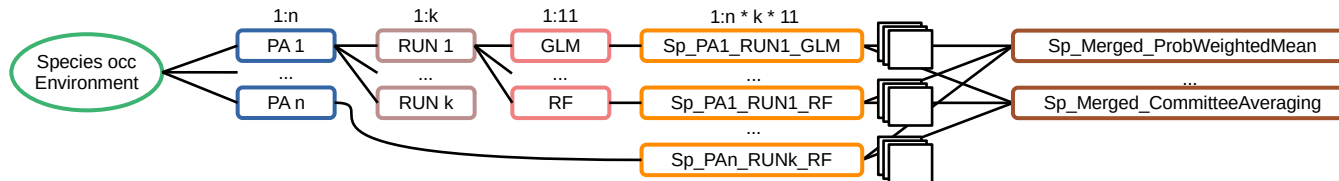
Step 2 : gather single models

» different ways of **combining** single models together :

- *all*
- *algo*
- *PA*
- *PA+algo*
- *PA+run*



Different algo



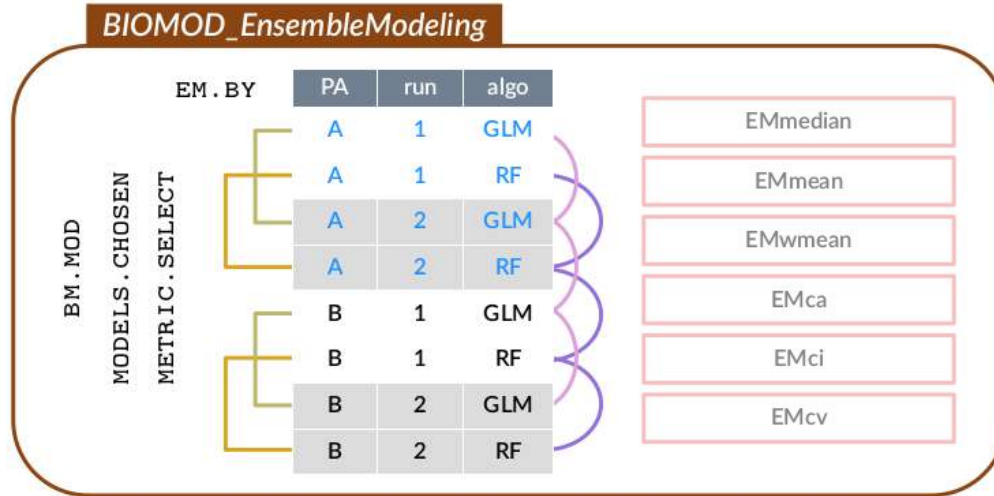
2.b Ensemble models

Step 1 : filter single models

Step 2 : gather single models

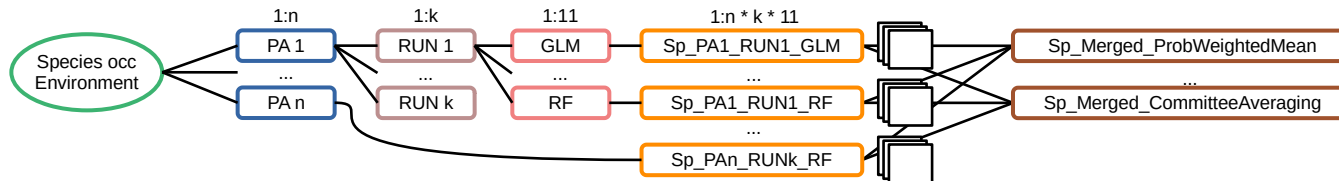
» different ways of **combining** single models together :

- *all*
- *algo*
- *PA*
- *PA+algo*
- *PA+run*



PA1_RUN1_GLM	PA1_RUN1_RF
PA1_RUN2_GLM	PA1_RUN2_RF
PA1_RUN3_GLM	PA1_RUN3_RF
PA2_RUN1_GLM	PA2_RUN1_RF
PA2_RUN2_GLM	PA2_RUN2_RF
PA2_RUN3_GLM	PA2_RUN3_RF

Different PA



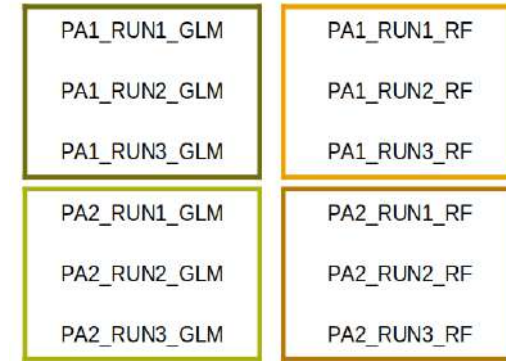
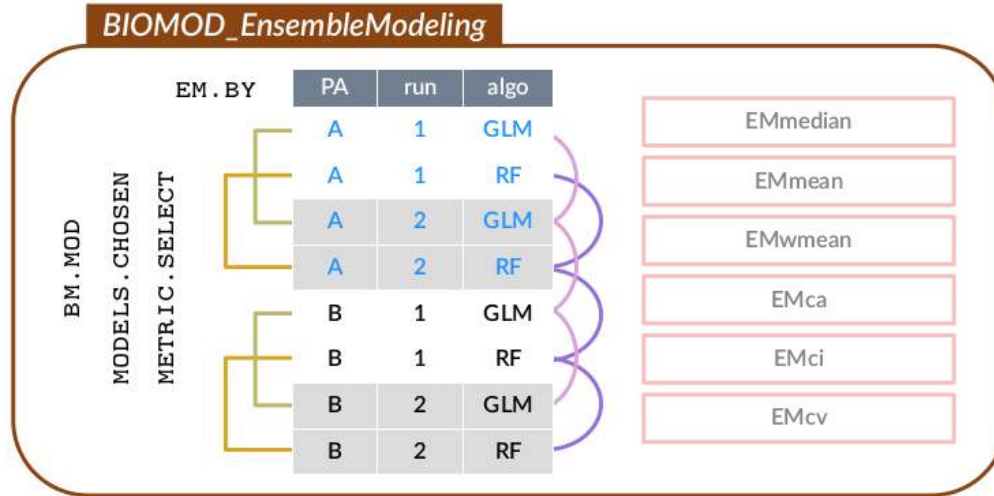
2.b Ensemble models

Step 1 : filter single models

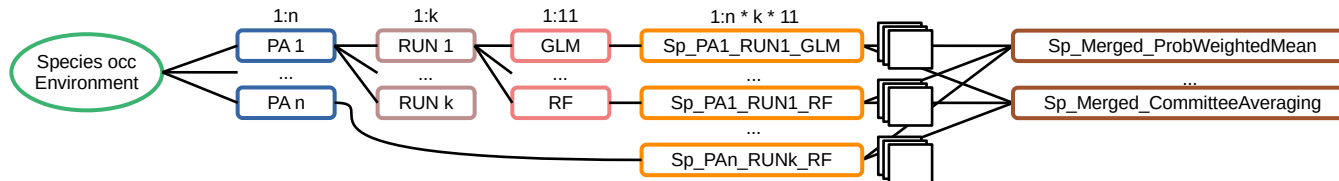
Step 2 : gather single models

» different ways of **combining** single models together :

- *all*
- *algo*
- *PA*
- *PA+algo*
- *PA+run*



Different PA
and algo



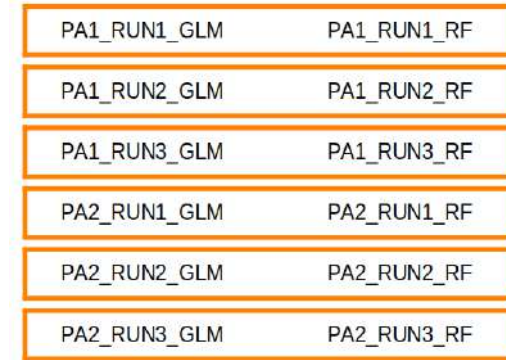
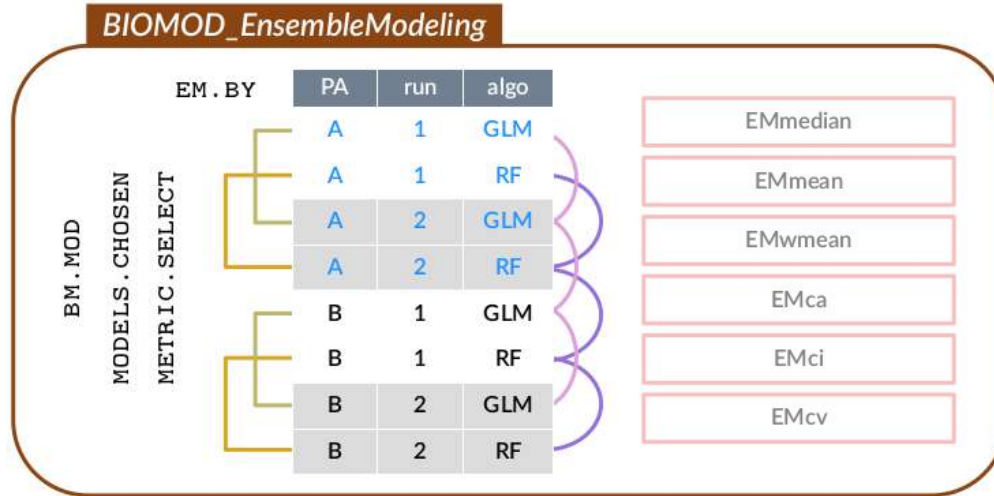
2.b Ensemble models

Step 1 : filter single models

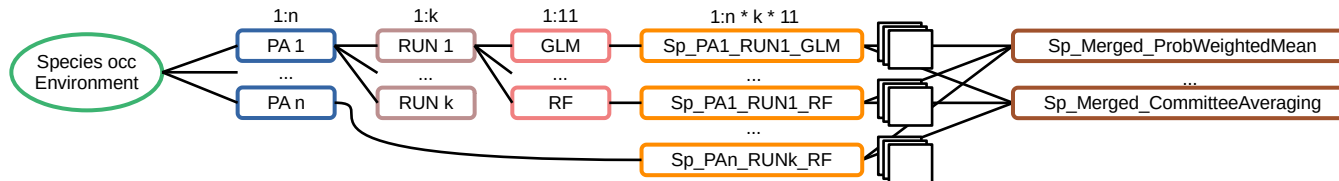
Step 2 : gather single models

» different ways of **combining** single models together :

- *all*
- *algo*
- *PA*
- *PA+algo*
- *PA+run*



Different PA
and run



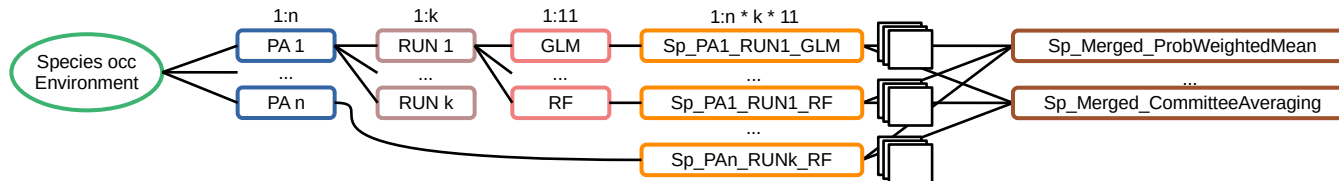
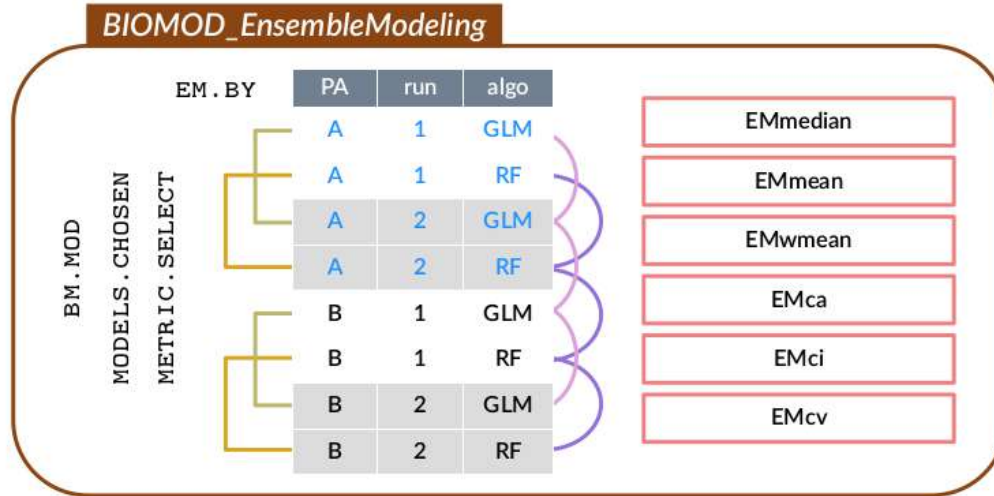
2.b Ensemble models

» « simple » ensemble models : **mean** or **median**

Step 1 : filter single models

Step 2 : gather single models

Step 3 : build ensemble models



2.b Ensemble models

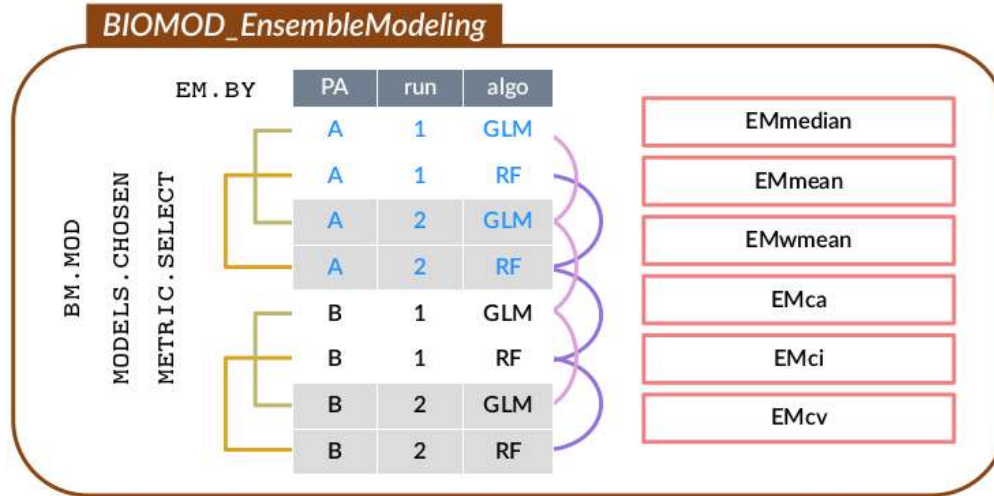
Step 1 : filter single models

Step 2 : gather single models

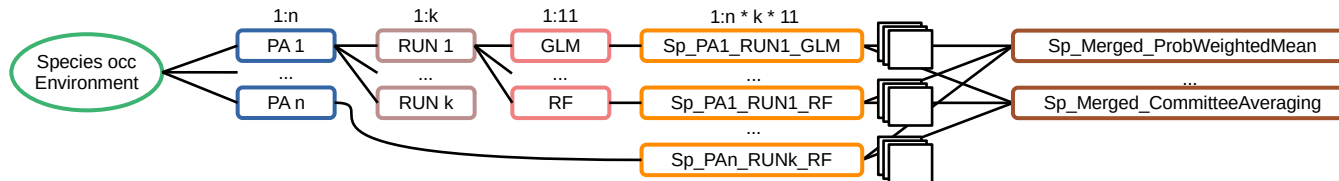
Step 3 : build ensemble models

» « simple » ensemble models : mean or median

» « complex » ensemble models :
 • **probability weighted mean**



$$W1 * \begin{bmatrix} 0,45 & 0,01 \\ 0,12 & 0,77 \end{bmatrix} + W2 * \begin{bmatrix} 0,30 & 0,5 \\ 0,10 & 0,77 \end{bmatrix} + W3 * \begin{bmatrix} 0,15 & 0,23 \\ 0,25 & 0,64 \end{bmatrix}$$



2.b Ensemble models

Step 1 : filter single models

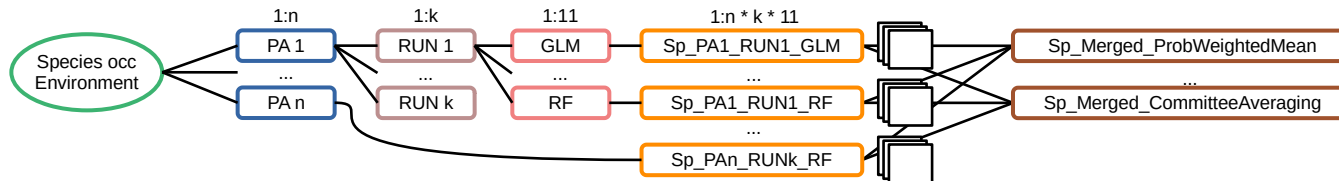
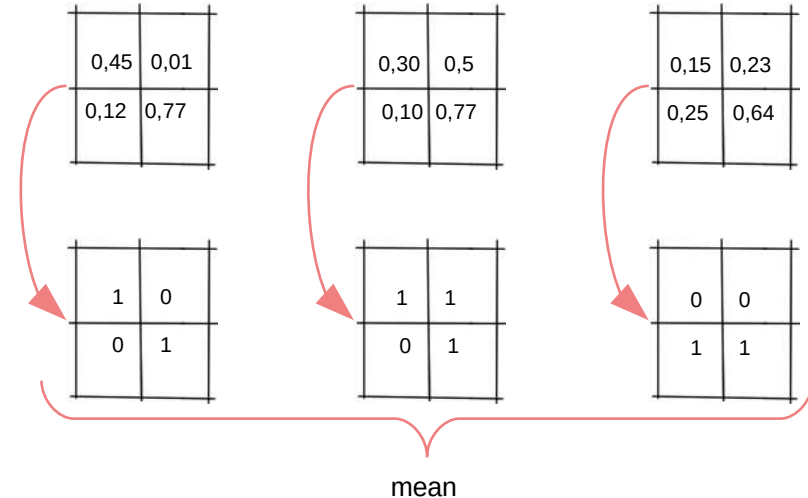
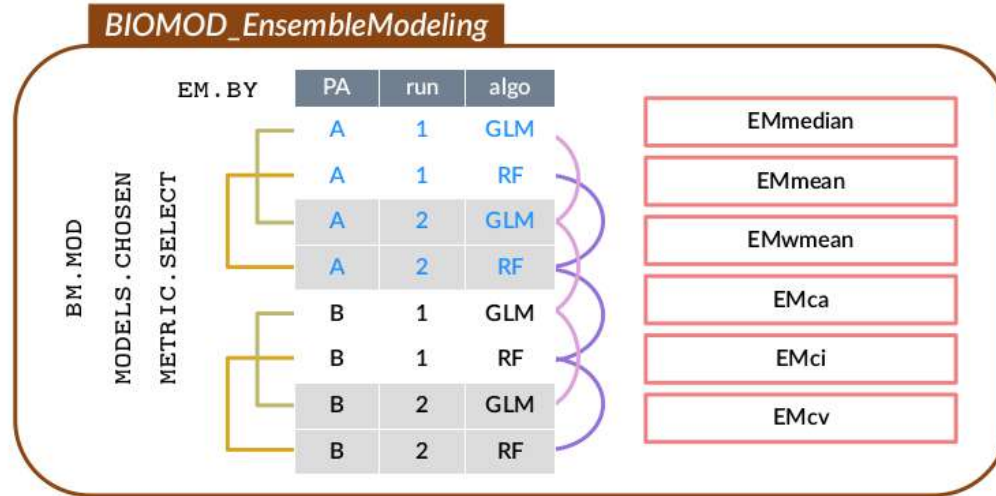
Step 2 : gather single models

Step 3 : build ensemble models

» « simple » ensemble models : mean or median

» « complex » ensemble models :

- probability weighted mean
- **committe averaging**

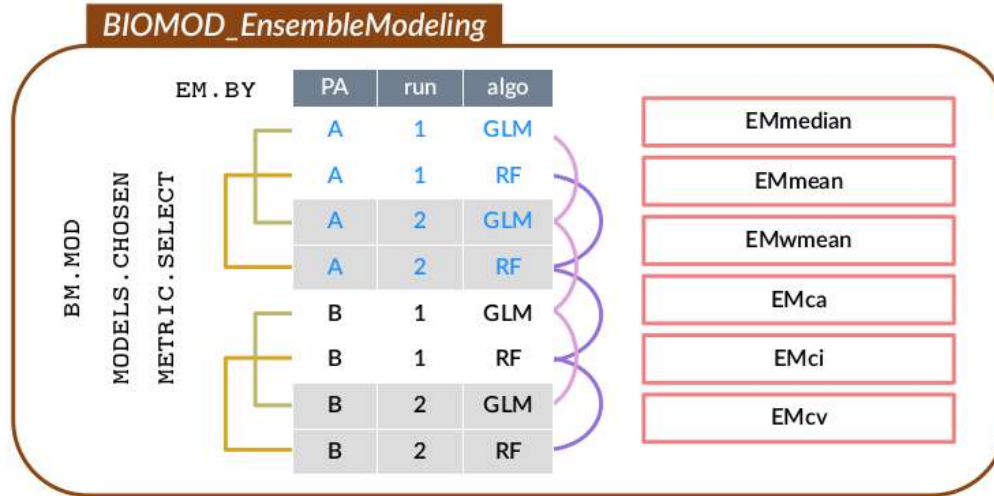


2.b Ensemble models

Step 1 : filter single models

Step 2 : gather single models

Step 3 : build ensemble models



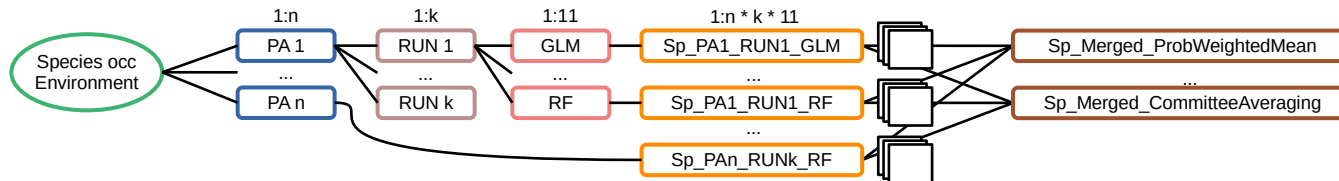
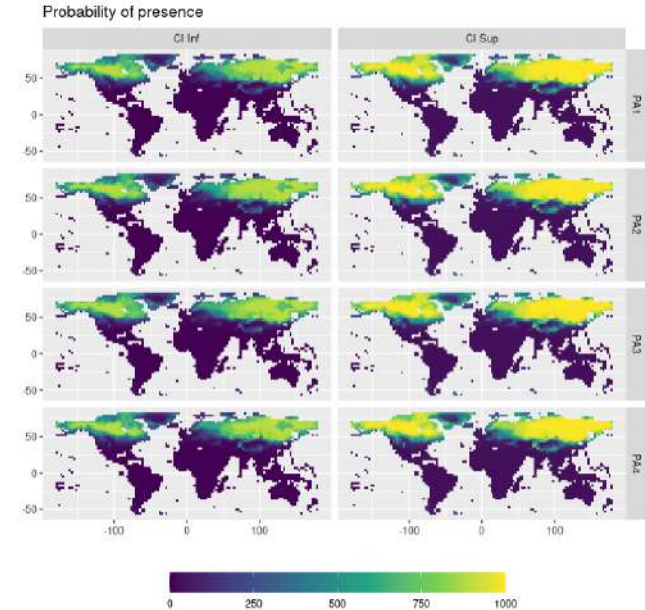
» « simple » ensemble models : mean or median

» « complex » ensemble models :

- probability weighted mean
- committe averaging

» « exploratory » ensemble models :

- **confidence intervals** or **coefficient of variation**

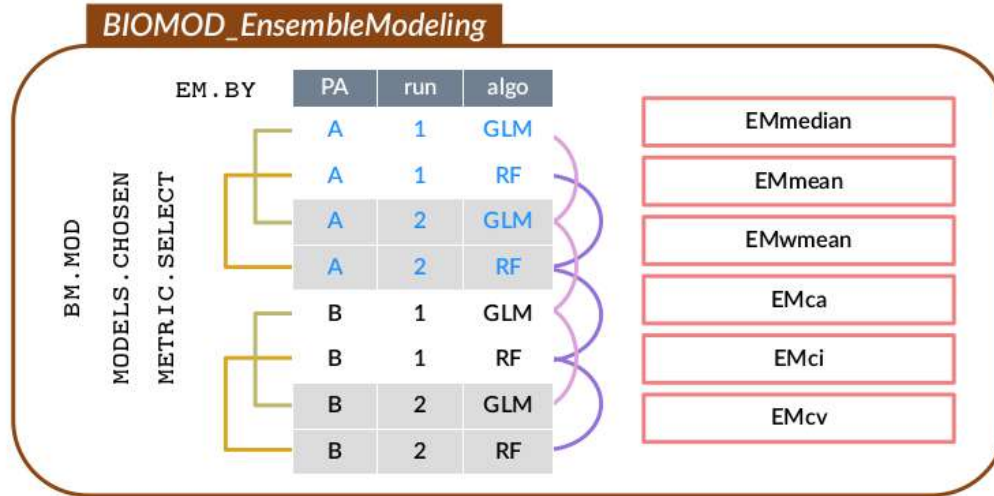


2.b Ensemble models

Step 1 : filter single models

Step 2 : gather single models

Step 3 : build ensemble models



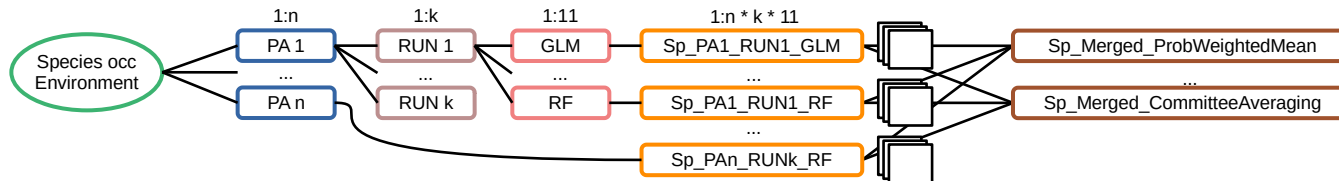
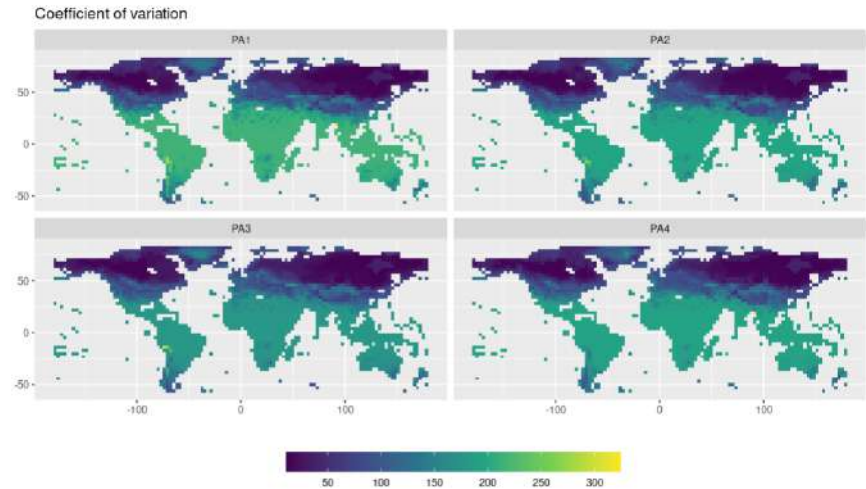
» « simple » ensemble models : mean or median

» « complex » ensemble models :

- probability weighted mean
- committe averaging

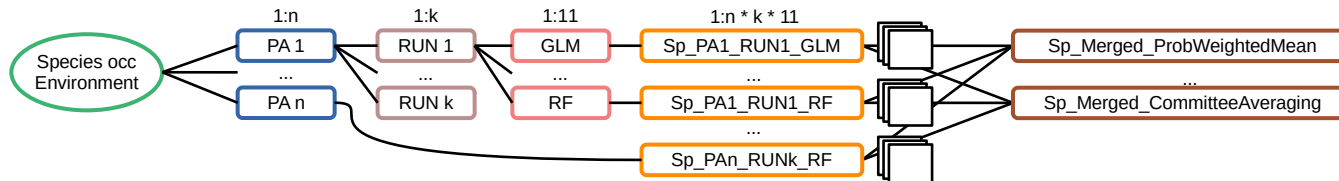
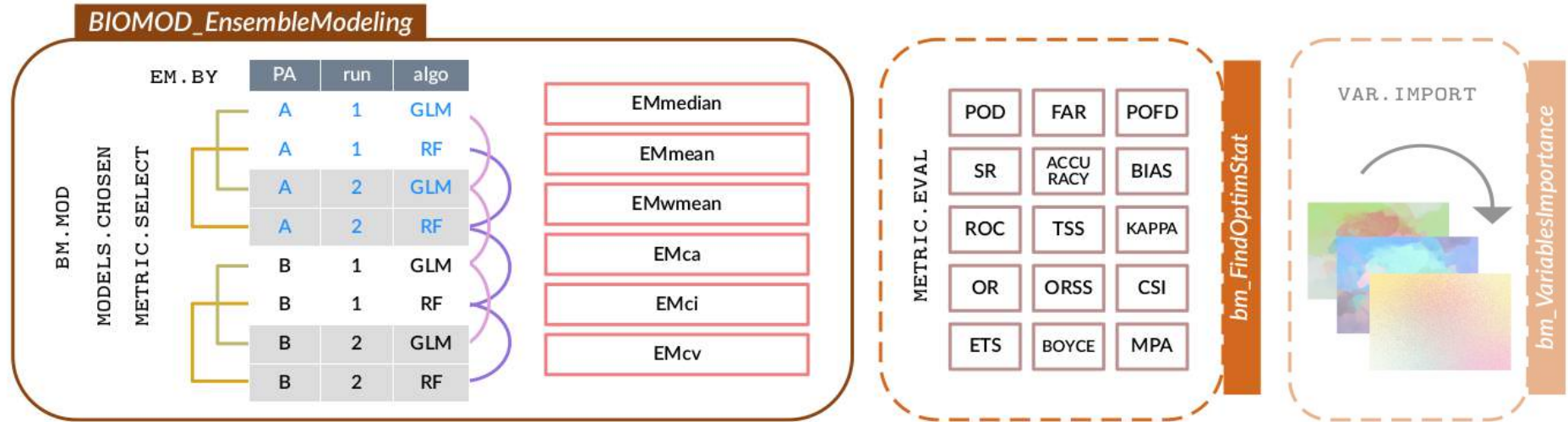
» « exploratory » ensemble models :

- **confidence intervals** or **coefficient of variation**



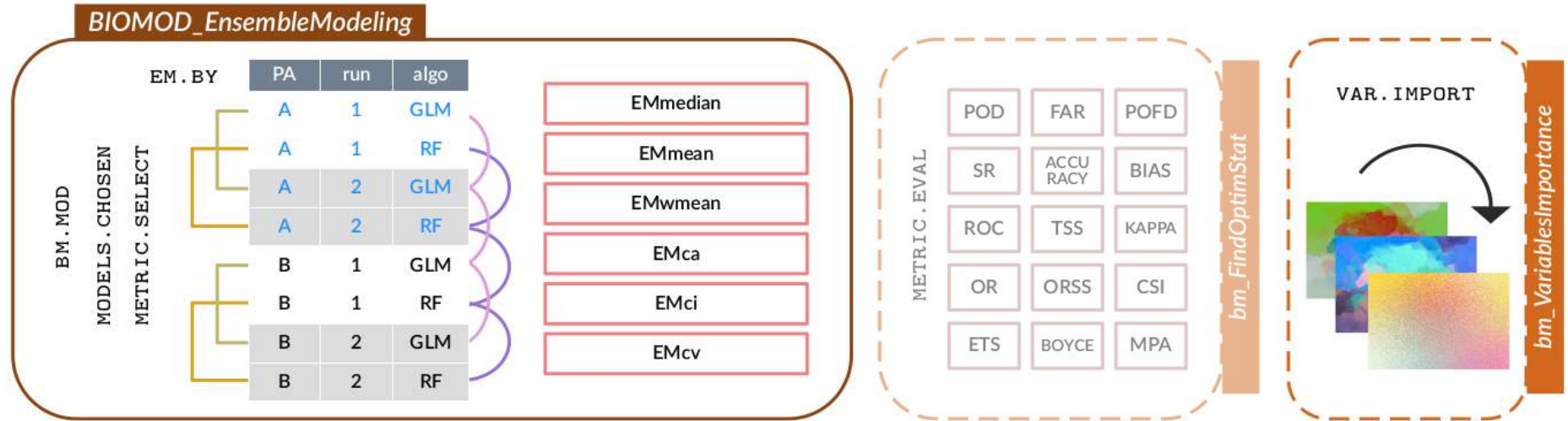
2.b Ensemble models

- » except ROC, all evaluation metrics obtained from contingency table (*containing TP, FP, TN, FN*)
- » require a **binary transformation** :
 - range of thresholds tested
 - keep threshold optimising the evaluation metric



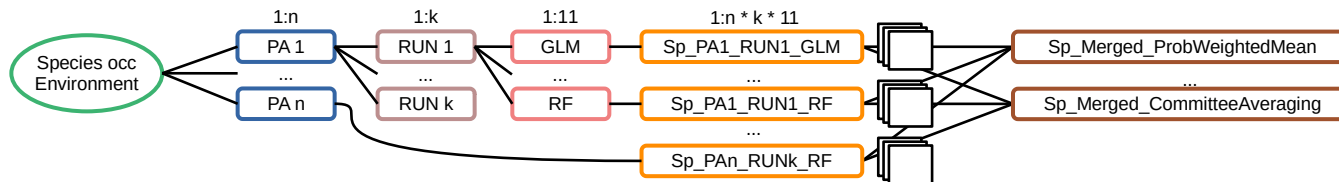
2.b Ensemble models

- » comparison of importance of variables between models
- » **Pearson correlation** between :
 - normal prediction
 - prediction with 1 variable randomised

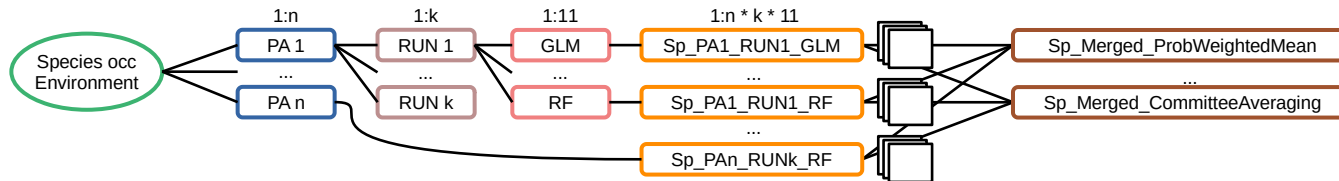
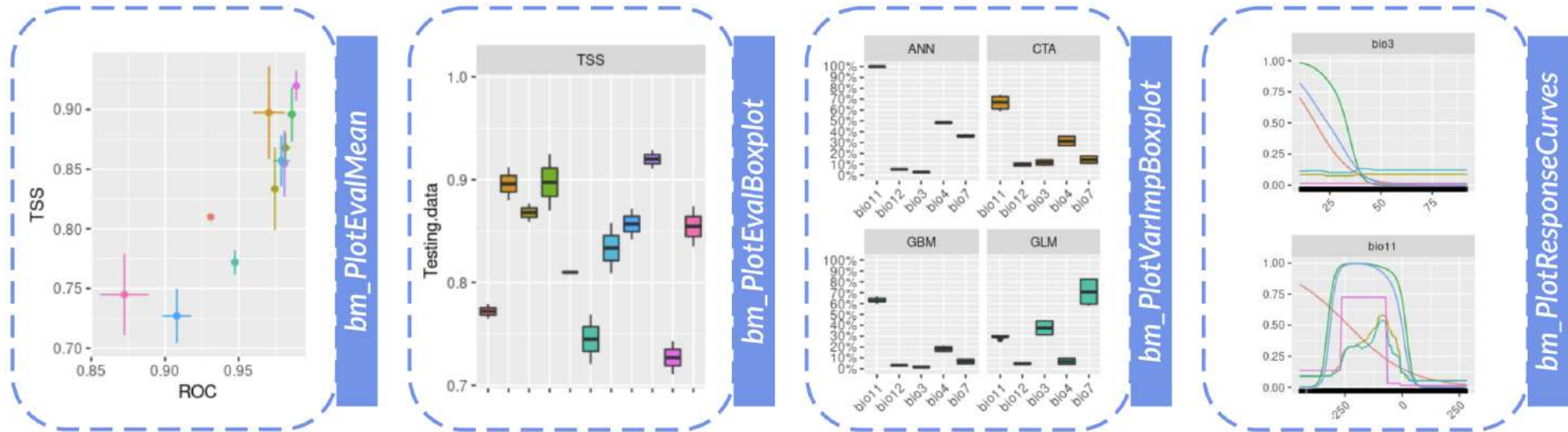


! IMPORTANT

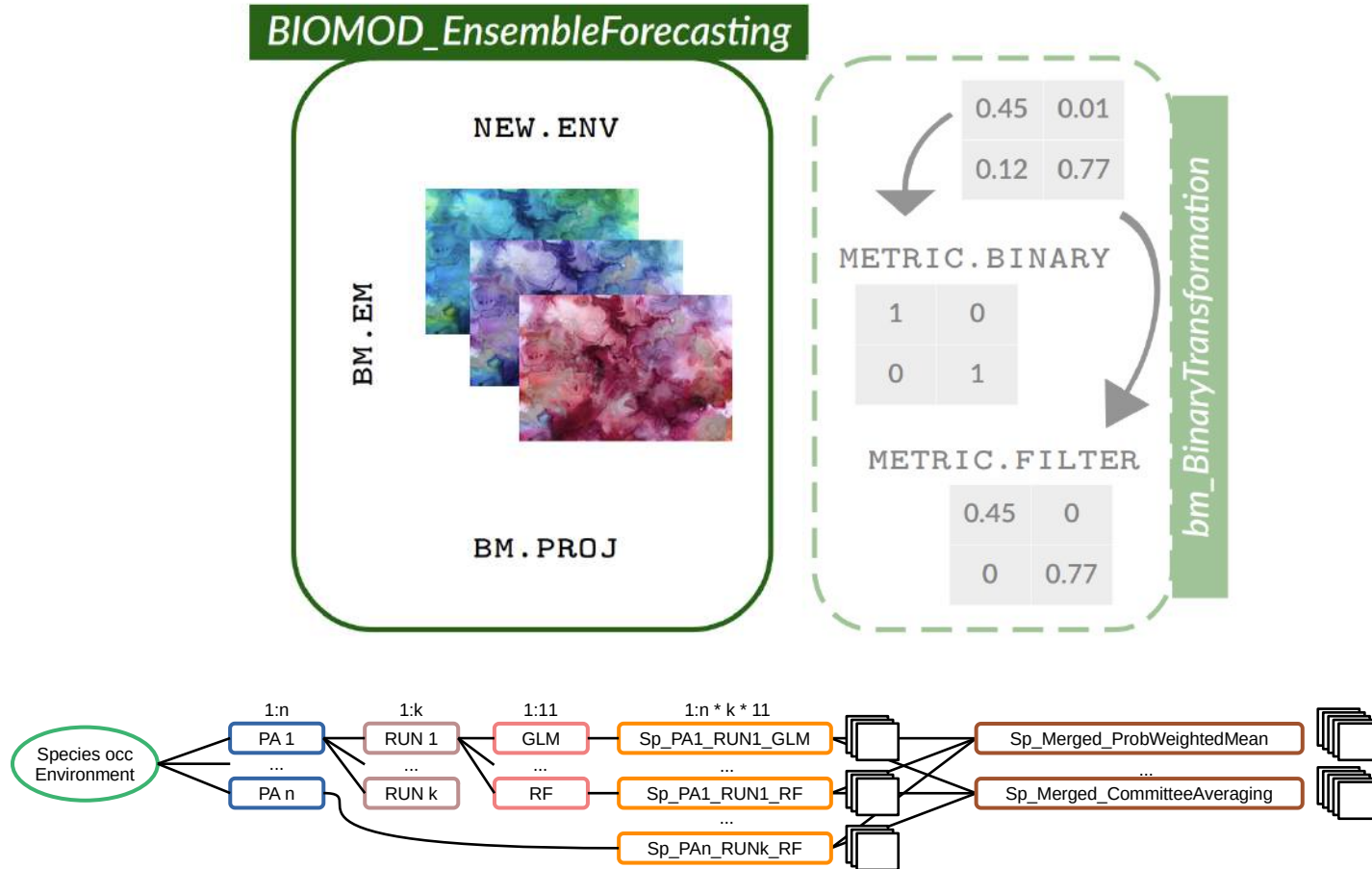
← Takes time ! Has to go through the whole workflow →



3.b Exploring ensemble models

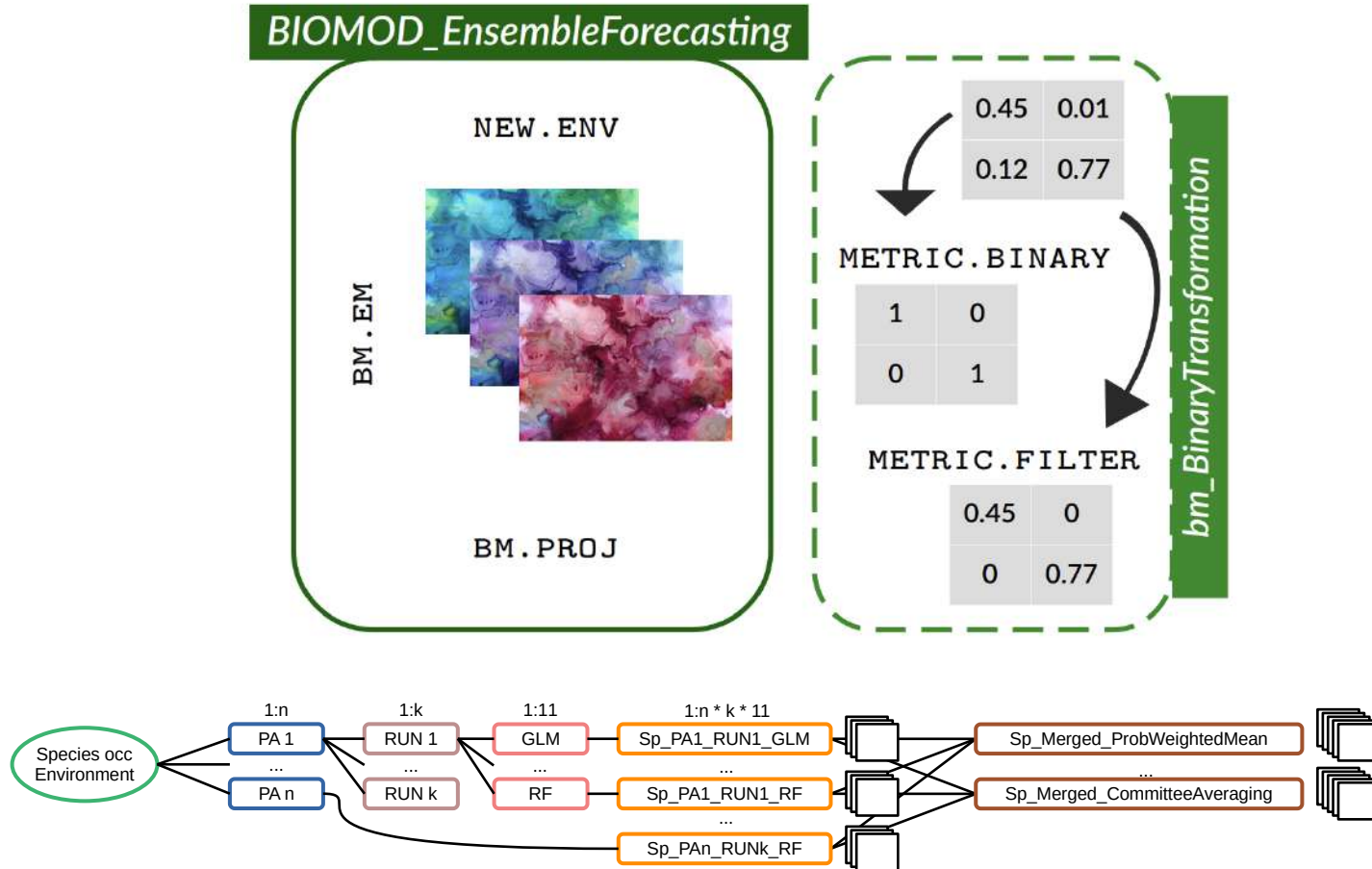


4.b Projecting ensemble models



4.b Projecting ensemble models

- » transformation associated to one evaluation metric (*one map created for each metric selected*)
- » use the threshold maximising the chosen metric



5. Species range change

BIOMOD_RangeSize

PROJ . CURRENT



PROJ . FUTURE



$$\text{PROJ . FUTURE} - 2 * \text{PROJ . CURRENT}$$



-2

Present in current,
lost in future

-1

Present in current,
present in future

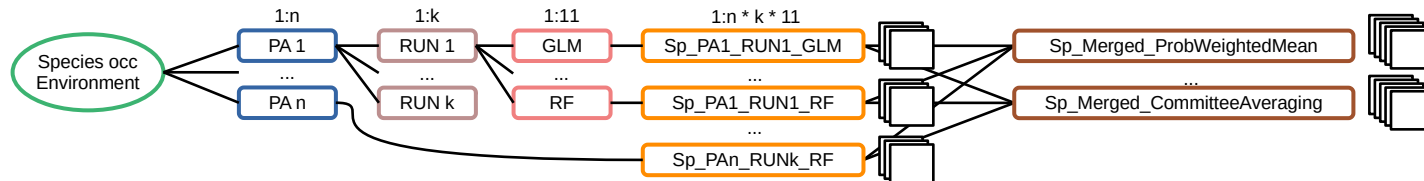
0

Absent in current,
absent in future

1

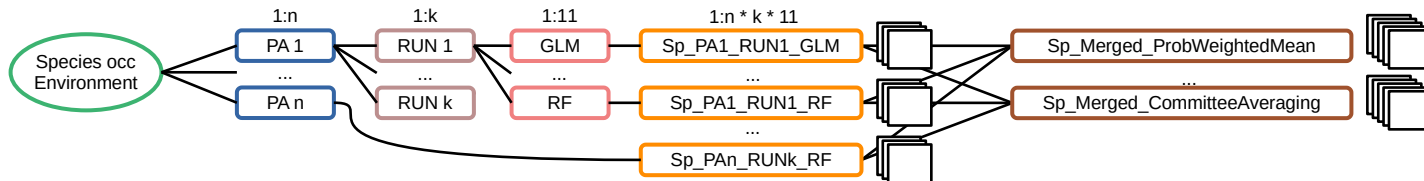
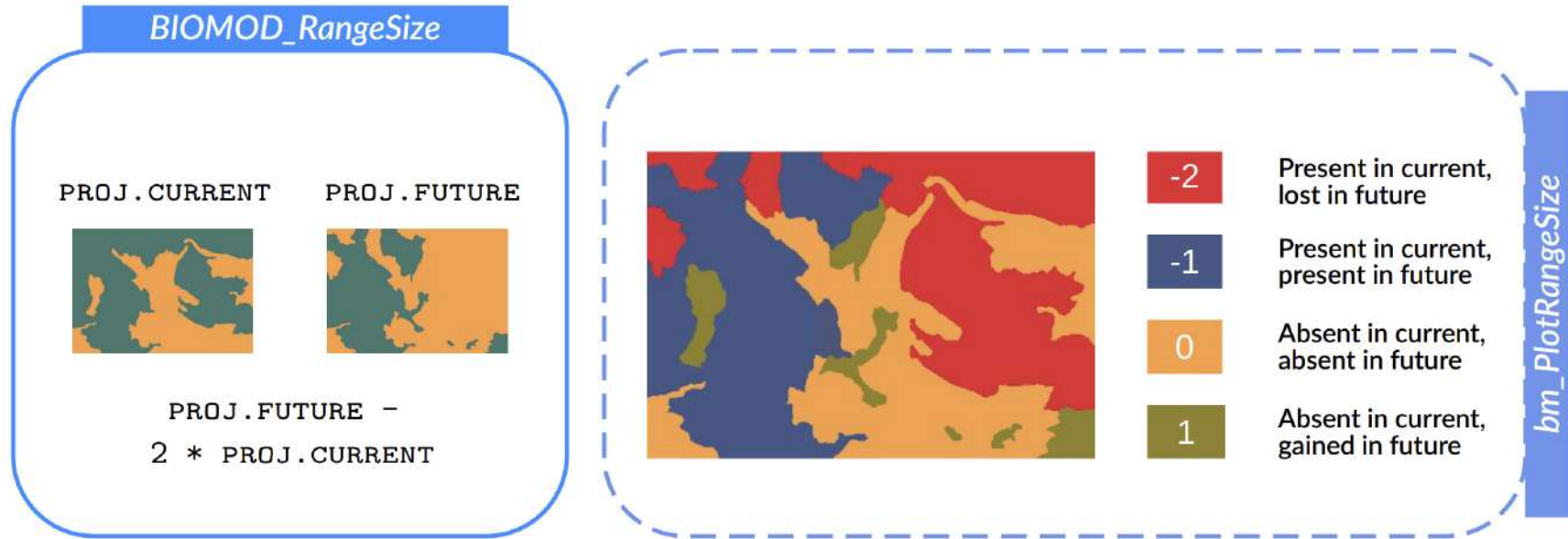
Absent in current,
gained in future

bm_PlotRangeSize



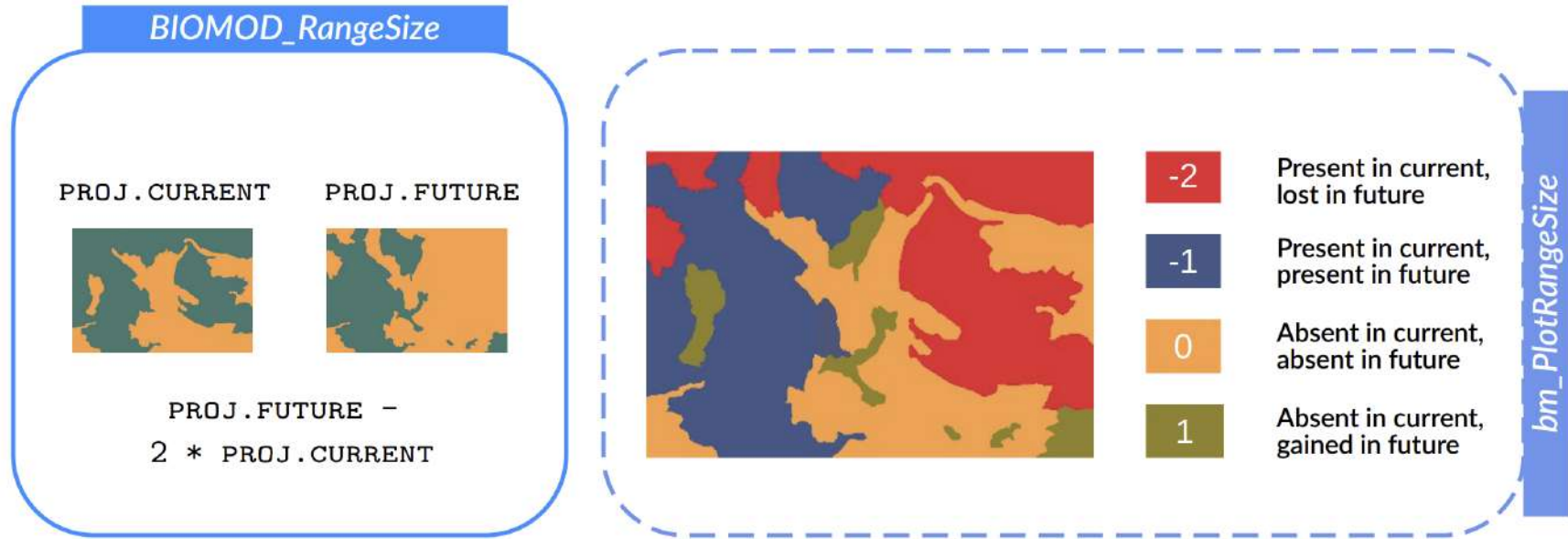
5. Species range change

- » explore **spatially** the difference in predictions
- » provide **summary values** :
 - percentage of loss / gain
 - species range change



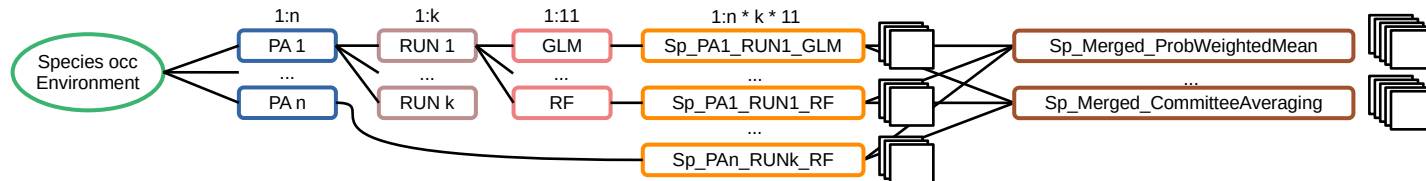
5. Species range change

- » explore **spatially** the difference in predictions
- » provide **summary values** :
 - percentage of loss / gain
 - species range change



! IMPORTANT

Work with binary maps (and not predictions between 0 and 1)



Species distribution modeling,
calibration and evaluation,
ensemble modeling



Thuiller, W. (2003), *BIOMOD – optimizing predictions of species distributions and projecting potential future shifts under global change*. **Global Change Biology**, 9: 1353-1362. <https://doi.org/10.1046/j.1365-2486.2003.00666.x>

Thuiller, W., Lafourcade, B., Engler, R. and Araújo, M.B. (2009), *BIOMOD – a platform for ensemble forecasting of species distributions*. **Ecography**, 32: 369-373. <https://doi.org/10.1111/j.1600-0587.2008.05742.x>

<https://github.com/biomodhub/biomod2/>

<https://biomodhub.github.io/biomod2/>