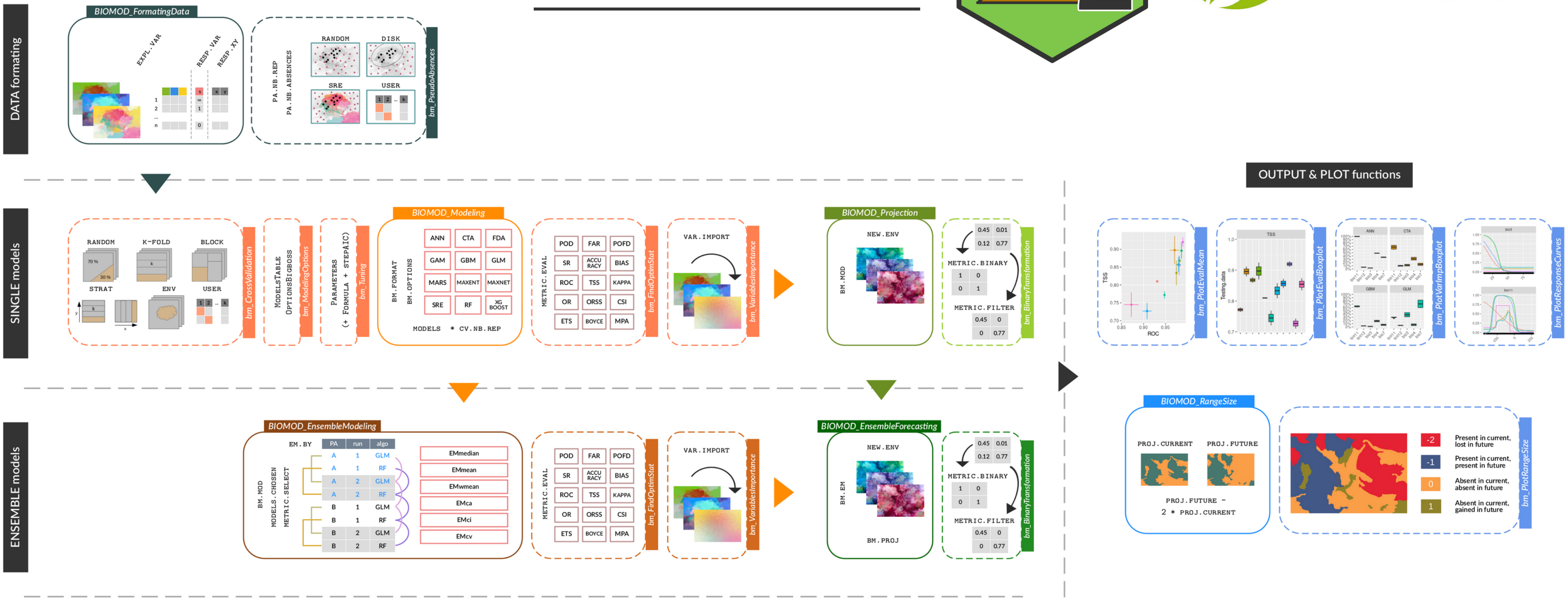
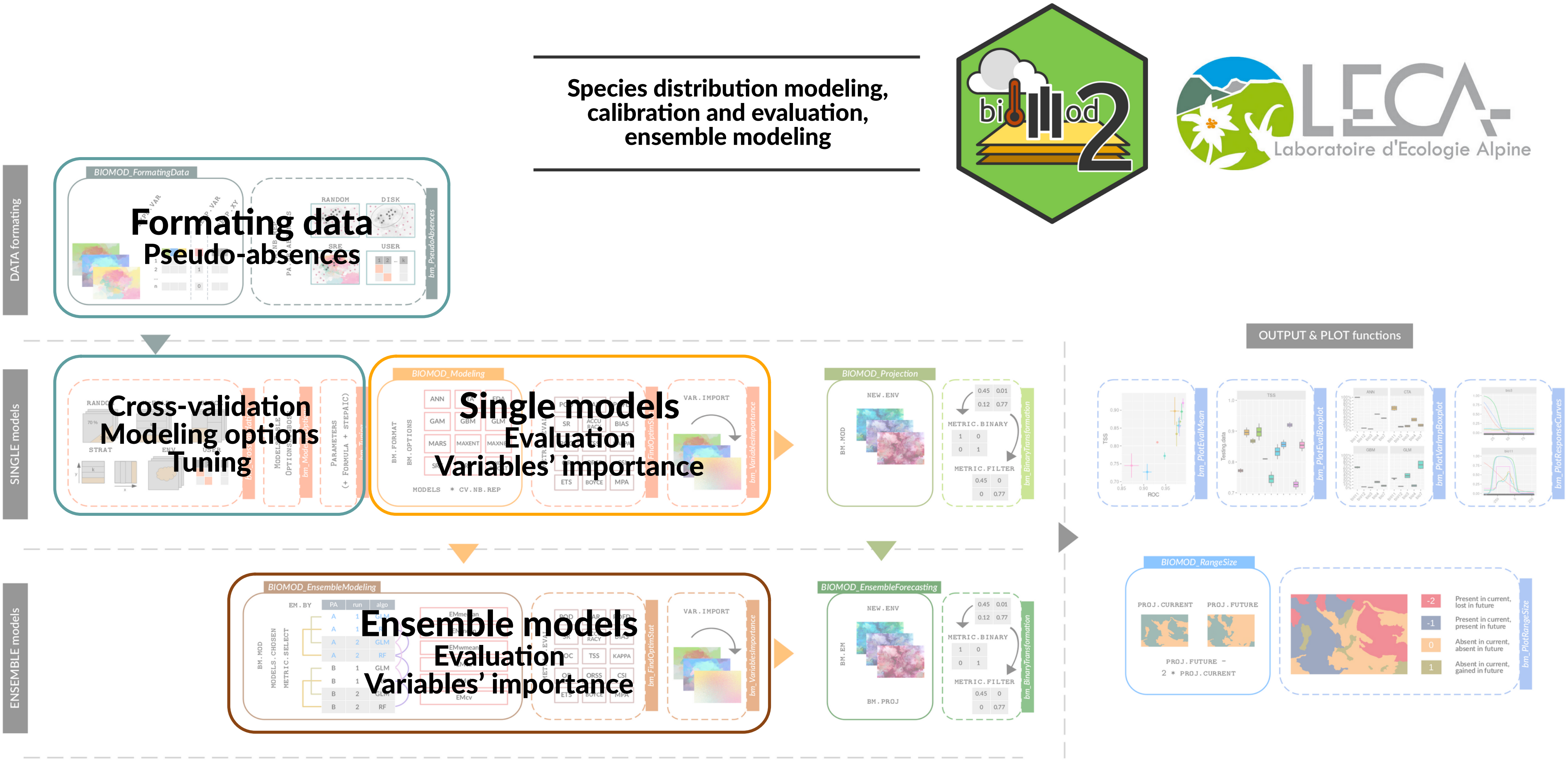
Species distribution modeling, calibration and evaluation, ensemble modeling

Laboratoire d'Ecologie Alpine

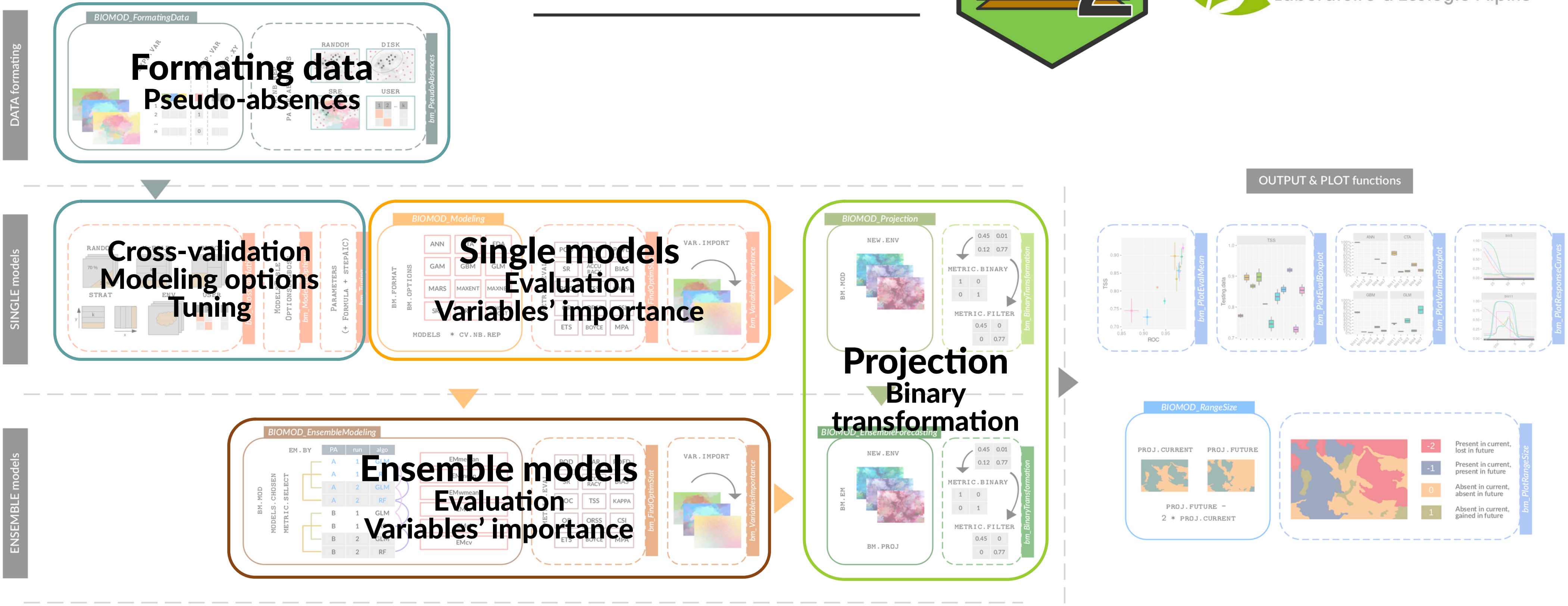


Species distribution modeling, calibration and evaluation, ensemble modeling

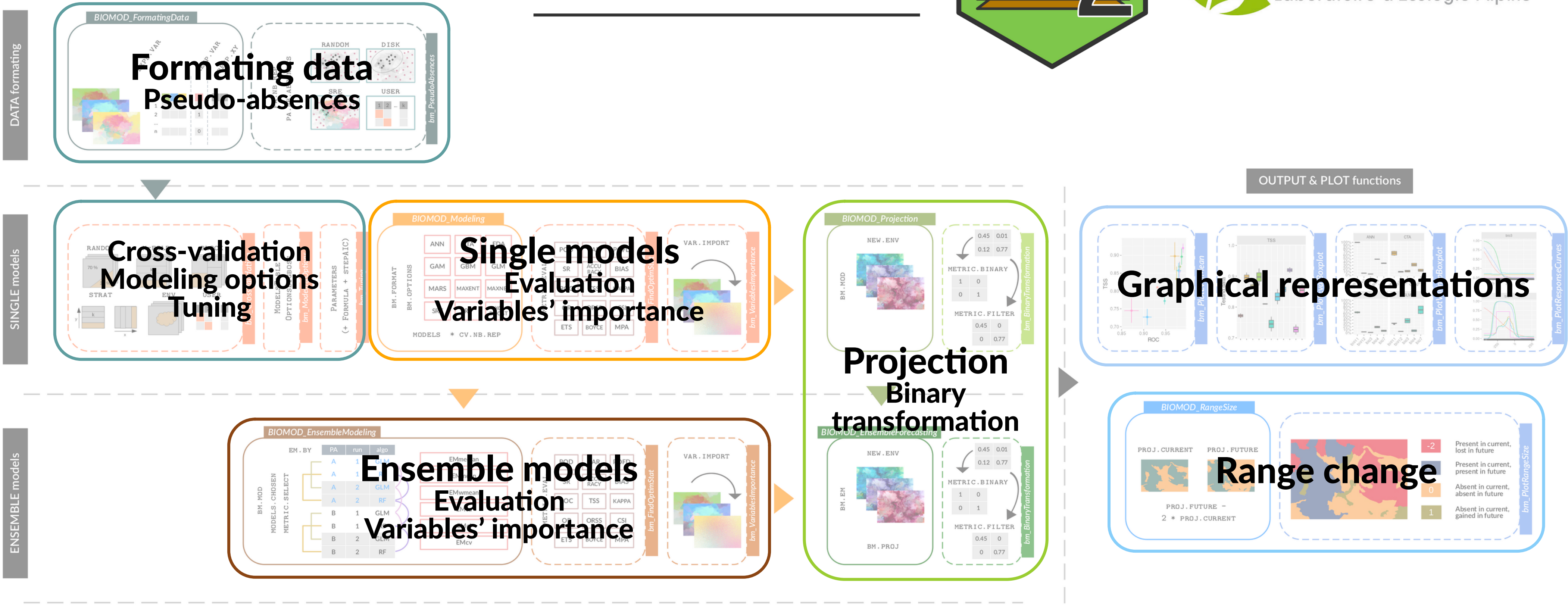
Laboratoire d'Ecologie Alpine


Species distribution modeling,
calibration and evaluation,
ensemble modeling
Laboratoire d'Ecologie Alpine
DATA formating
BIOMOD_FormatingData
Formating data
Pseudo-absences
RANDOM
DISK
SRE
USER
bm_PseudoAbsences
SINGLE models
Cross-validation
Modeling options
Tuning
BIOMOD_Modeling
Single models
Evaluation
Variables' importance
ANN
GAM
GBM
GLM
MARS
MAXENT
BM.FORMAT
BM.OPTIONS
MODELS * CV.NB.REP
VAR.IMPORT
bm_VariablesImportance
BIOMOD_Projection
NEW.ENV
BM.MOD
METRIC.BINARY
METRIC.FILTER
bm_BinaryTransformation
OUTPUT & PLOT functions
bm_PlotEvalMean
bm_PlotEvalBoxplot
bm_PlotVarImpBoxplot
bm_PlotResponseCurves
ENSEMBLE models
BIOMOD_EnsembleModeling
Ensemble models
Evaluation
Variables' importance
EM.BY
BM.MOD
MODELS.CHOSEN
METRIC.SELECT
bm_FindOptimStat
BIOMOD_EnsembleForecasting
BM.EM
BM.PROJ
BIOMOD_RangeSize
PROJ.CURRENT
PROJ.FUTURE
PROJ.FUTURE − 2 * PROJ.CURRENT
Present in current, lost in future
Present in current, present in future
Absent in current, absent in future
Absent in current, gained in future
bm_PlotRangeSize

Species distribution modeling, calibration and evaluation, ensemble modeling

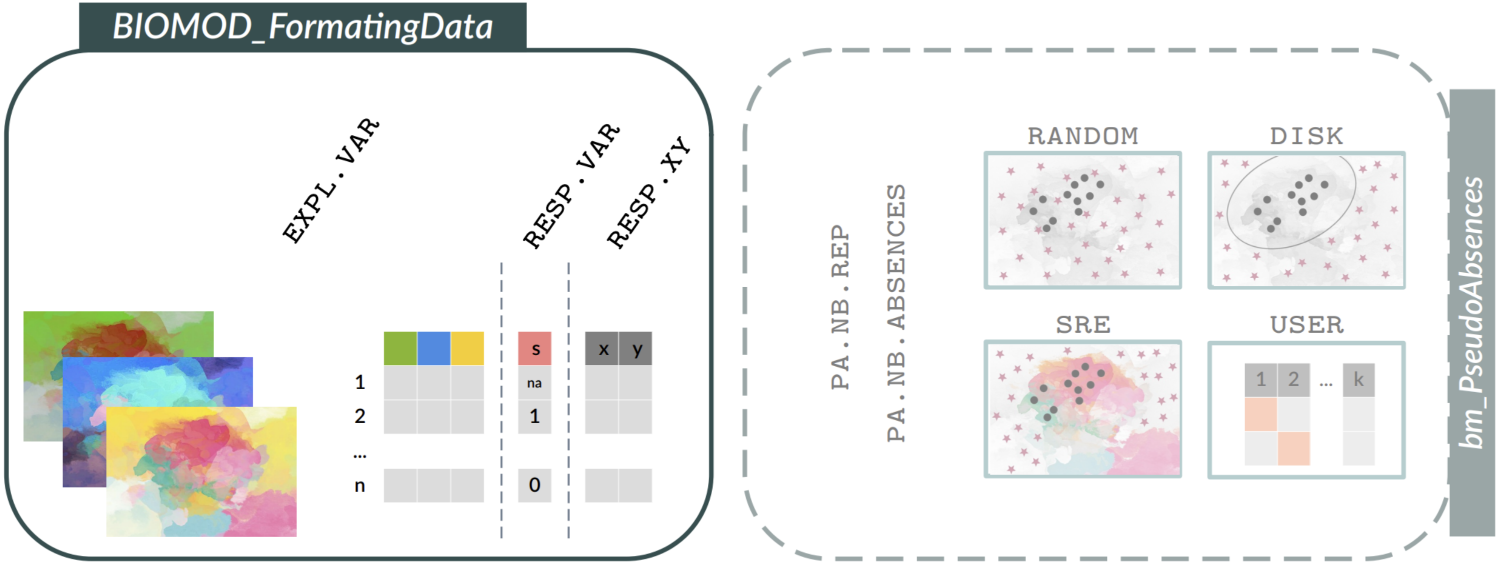


Species distribution modeling, calibration and evaluation, ensemble modeling



1. Formating data

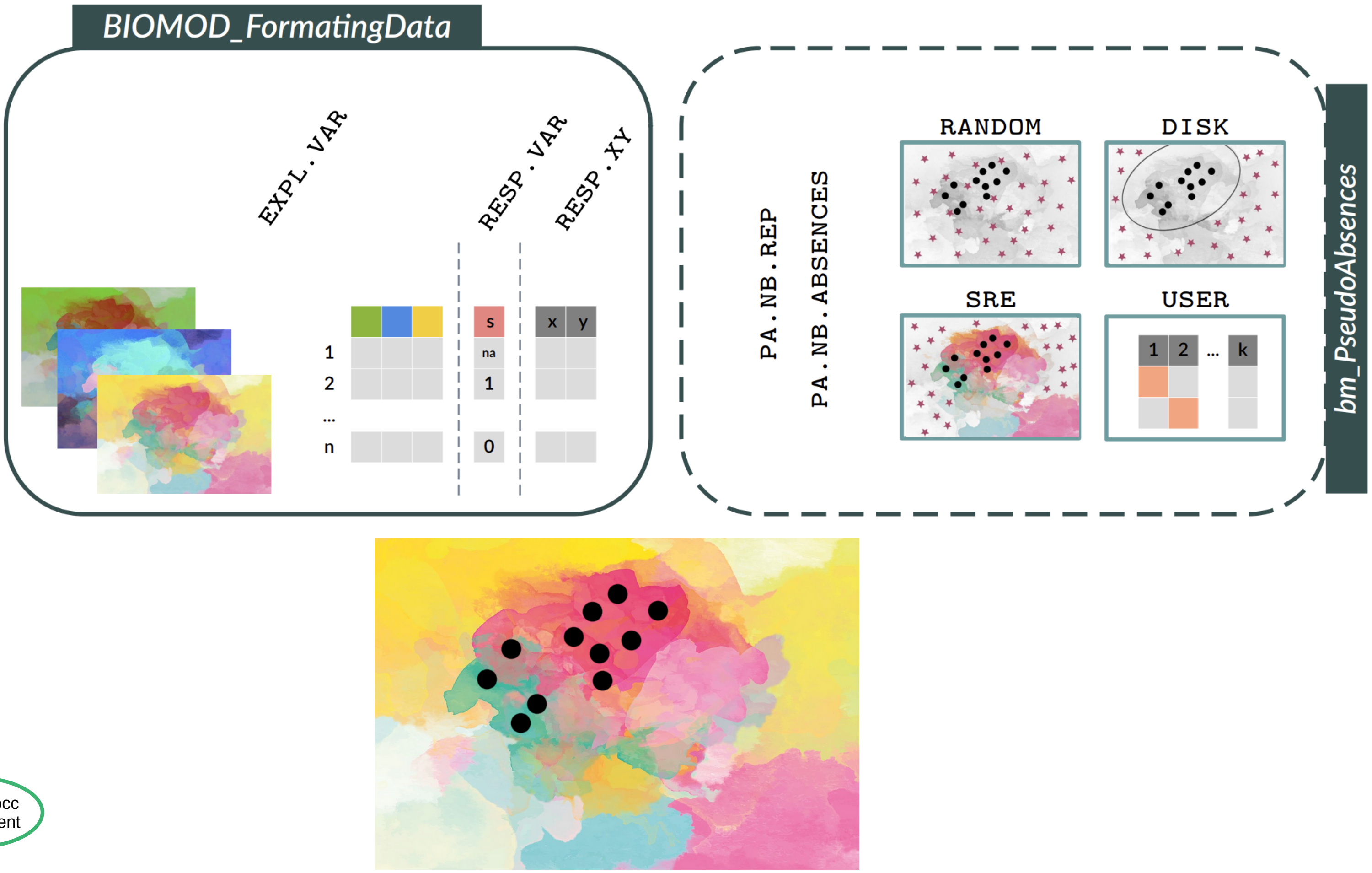
- **presences-absences**



Species occ
Environment

1. Formating data

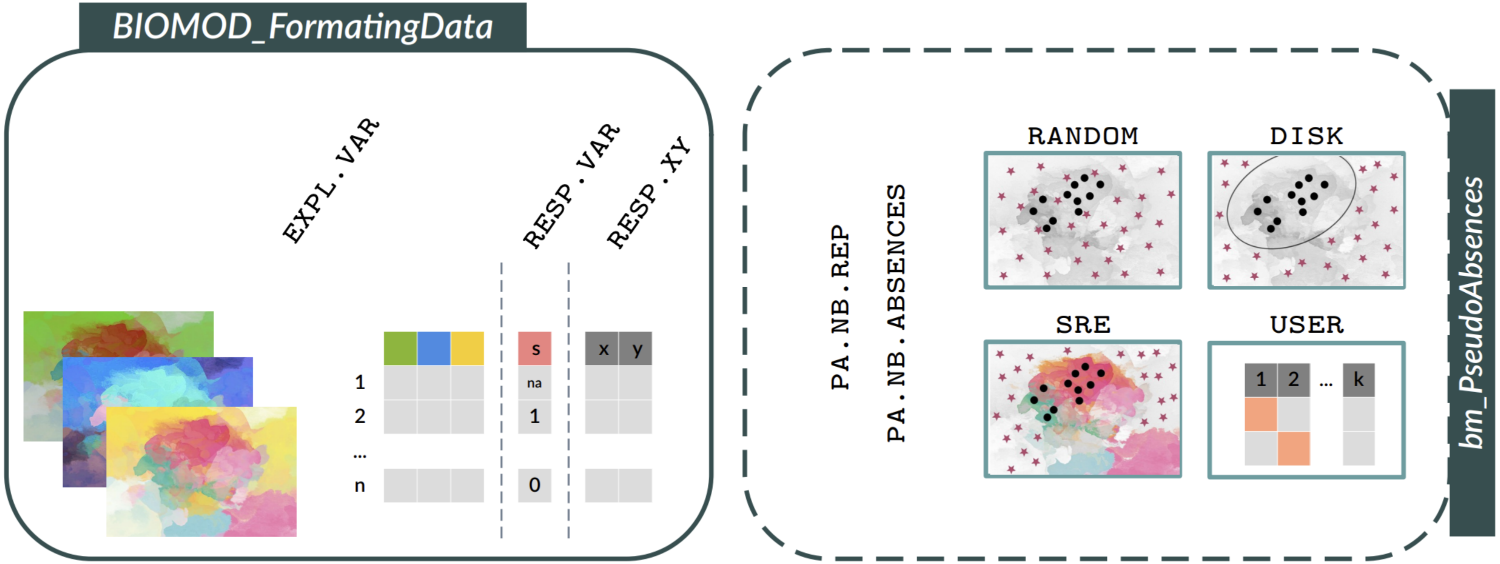
- **presence-only** data
- *avoid to mix with real absences*

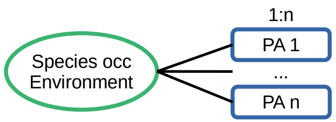


Species occ
Environment

1. Formating data

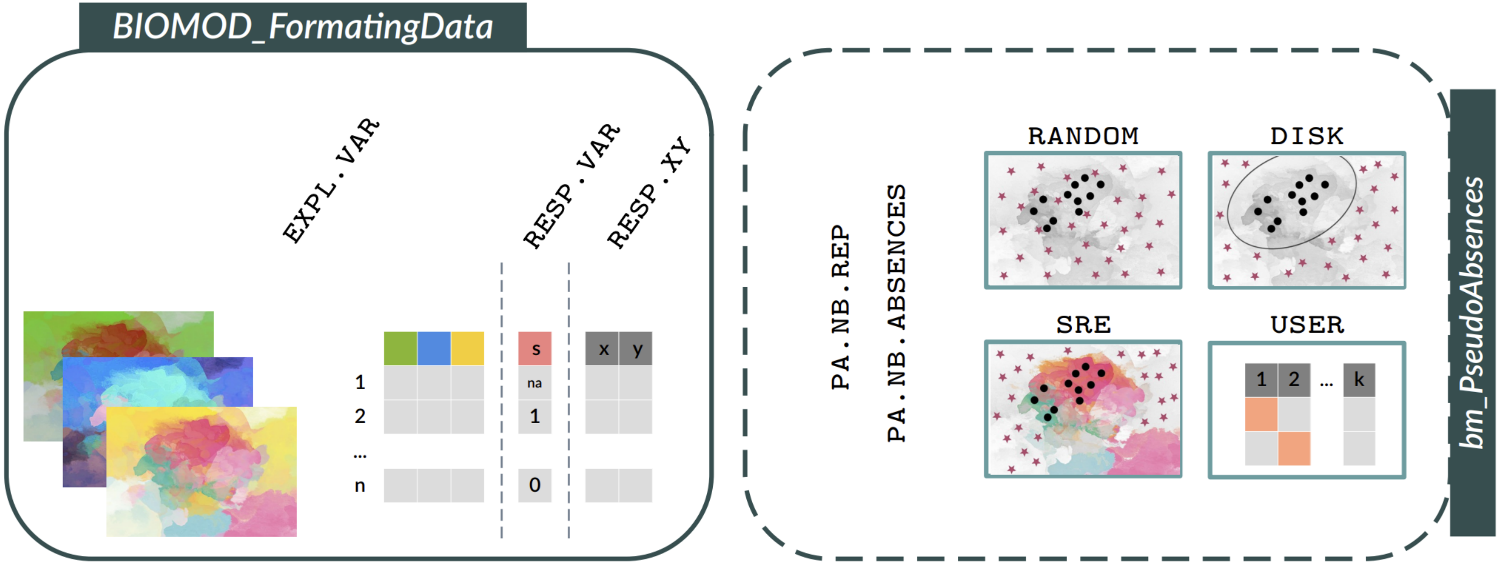
- **presence-only** data
- *avoid to mix with real absences*
- **random** : sampling potentially biased / non-exhaustive

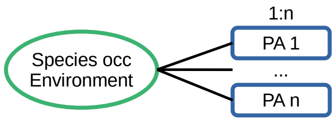


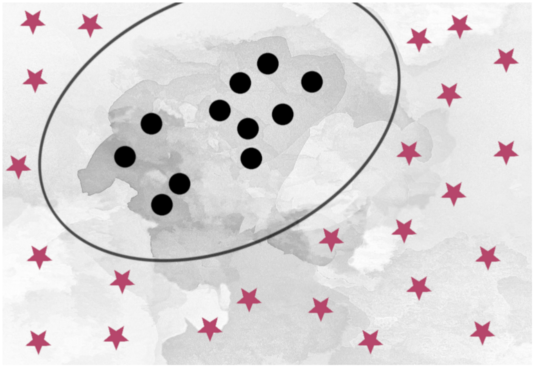


1. Formating data

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

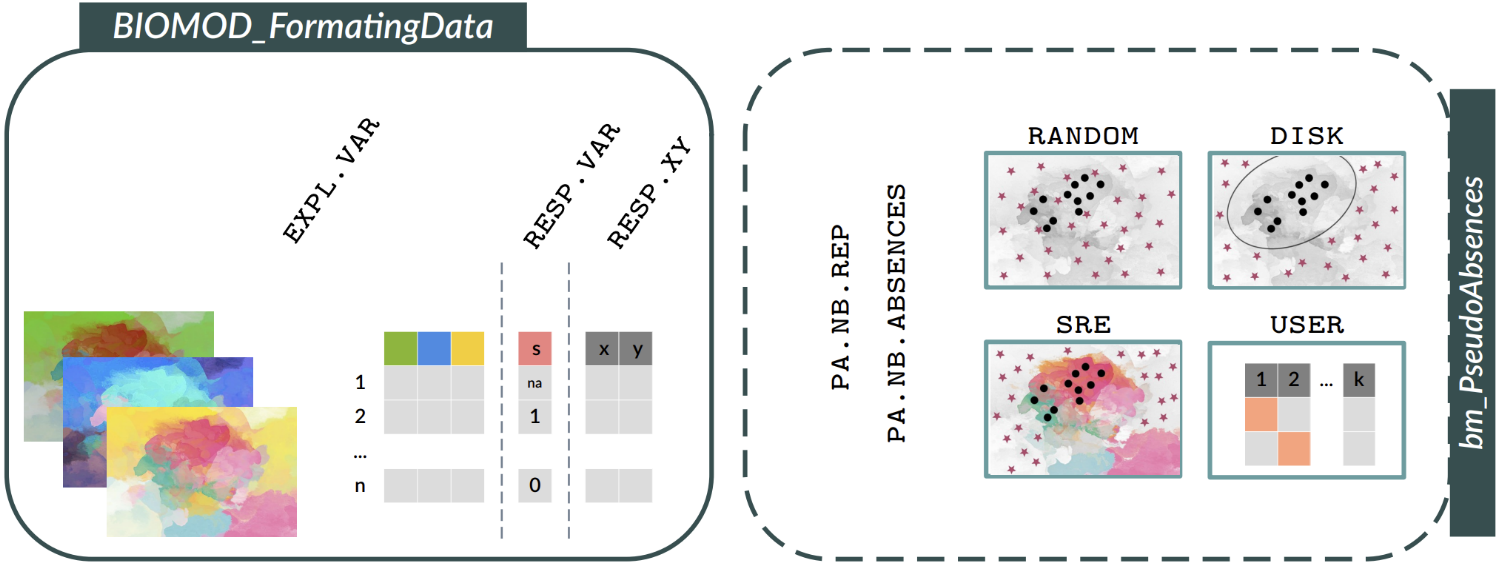


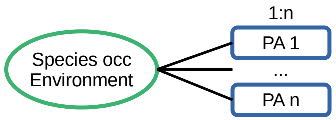




1. Formating data

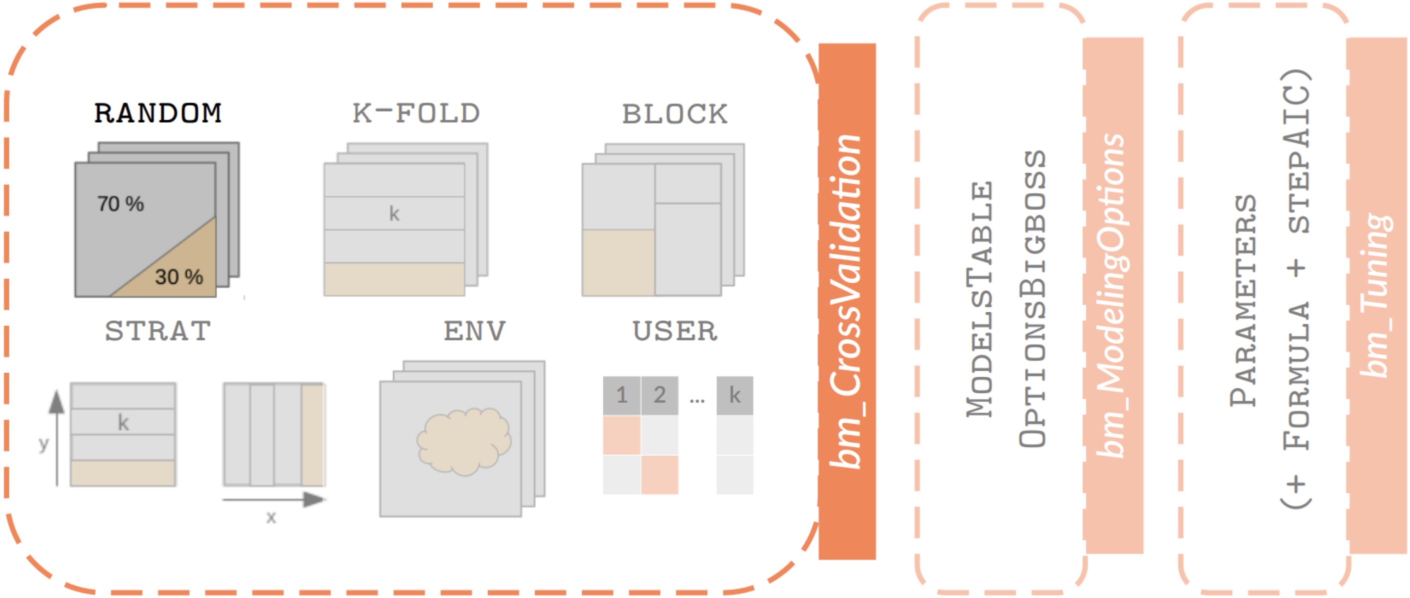
- **presence-only** data
- *avoid to mix with real absences*
- **random** : sampling potentially biased / non-exhaustive
- **disk** : geographic niche well sampled
- **SRE** : environmental niche well sampled

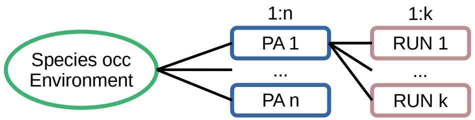




1. Formating data

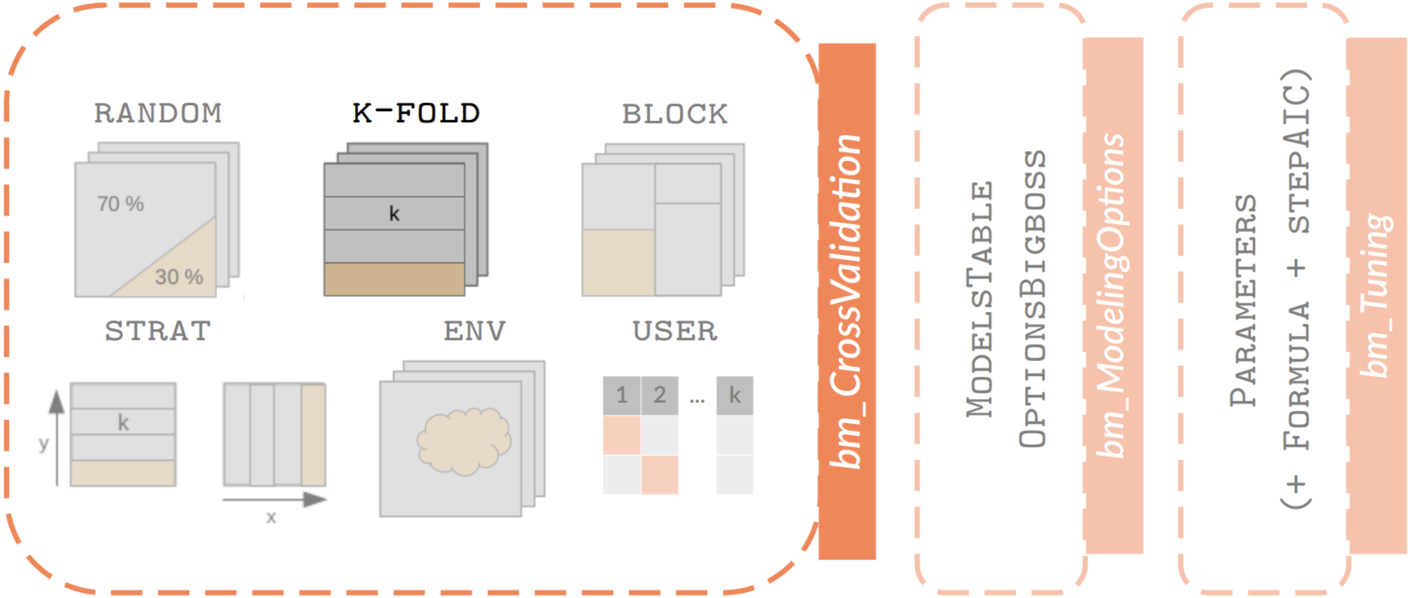
- simple calibration / validation split at the modeling step, and repeated nb.rep times

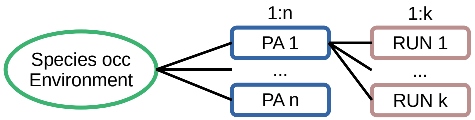




1. Formating data

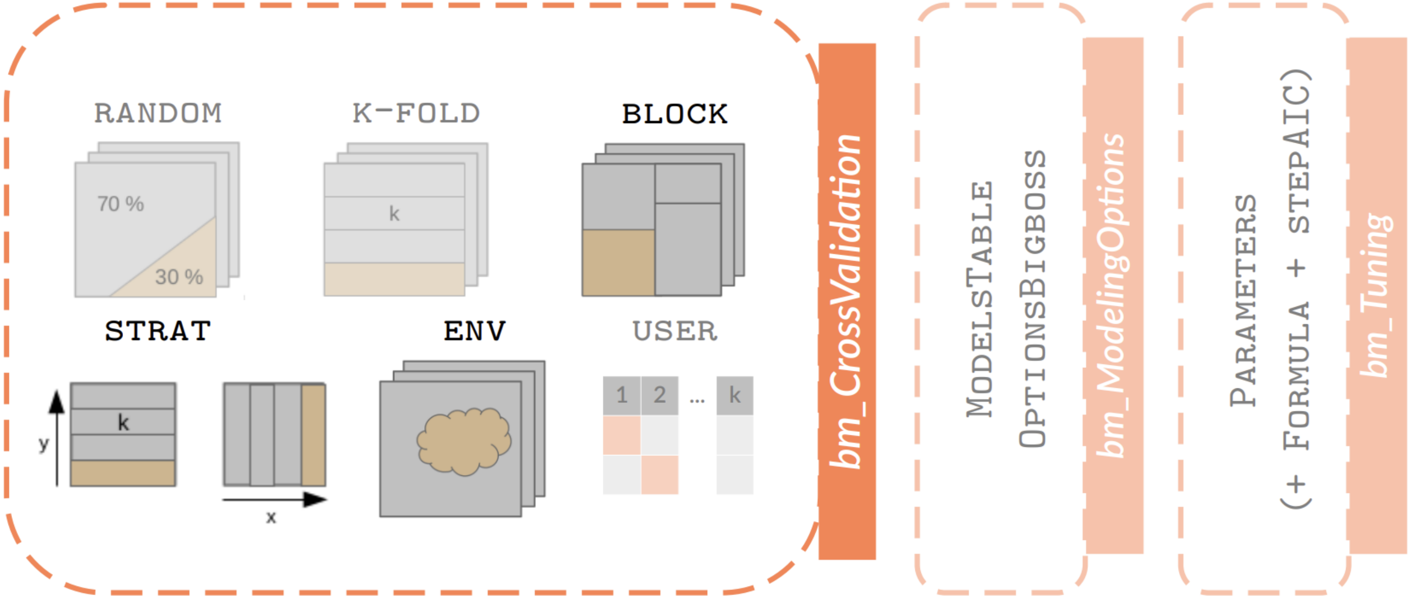
- simple calibration / validation split at the modeling step, and repeated nb.rep times
- **k-fold** : partition data into k sub-dataset, and repeated nb.rep times

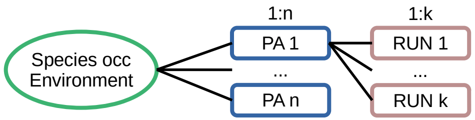




1. Formating data

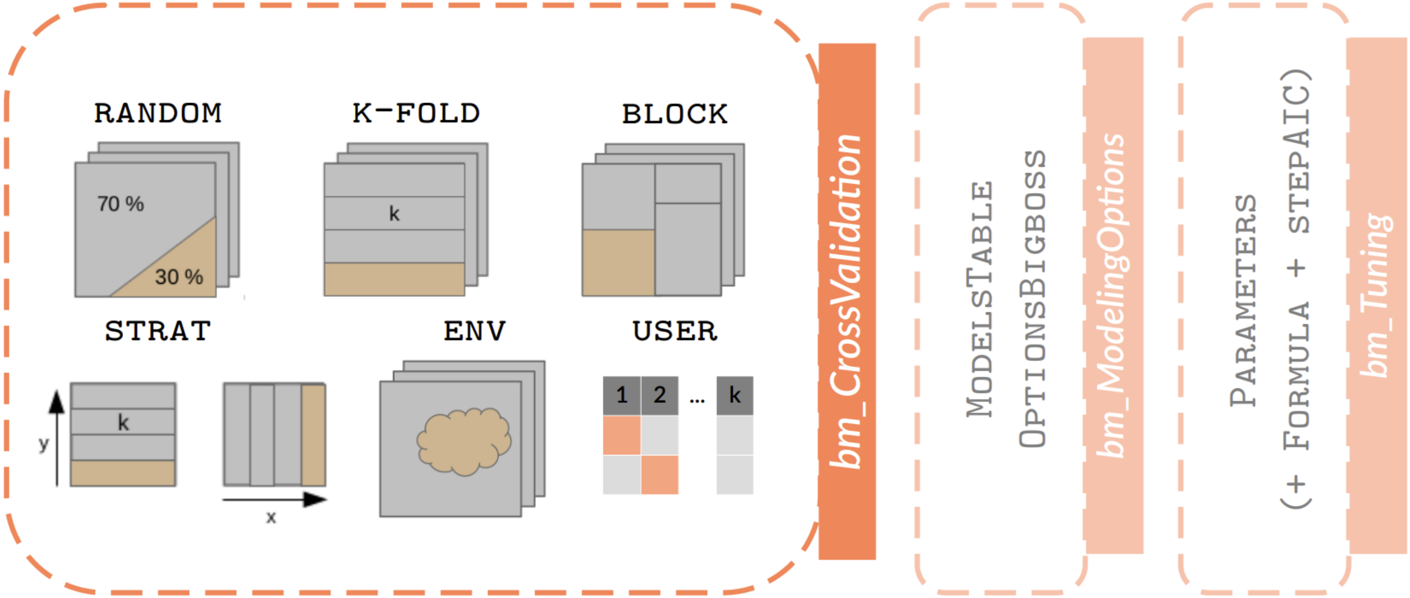
- simple calibration / validation split at the modeling step, and repeated nb.rep times
- **k-fold** : partition data into k sub-dataset, and repeated nb.rep times
- **stratified** : partition data into k sub-dataset (x, y, both, block)

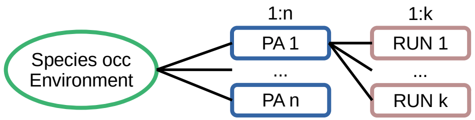


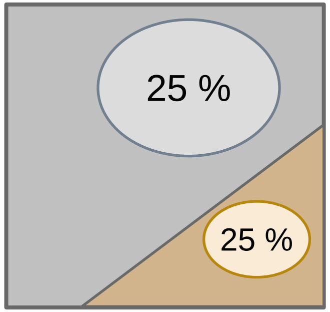


1. Formating data

- 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)
- **balance** : keep the prevalence of presences (*or absences*) in sub-dataset

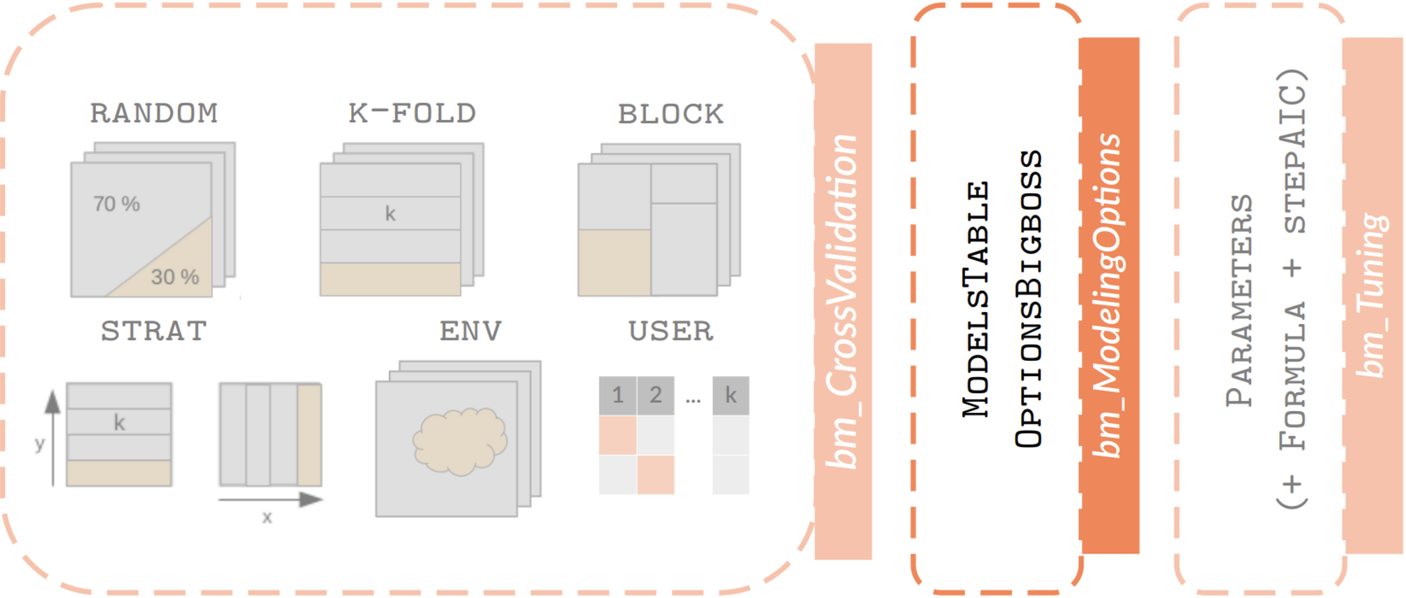


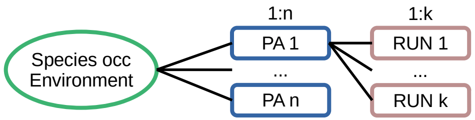




1. Formating data

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



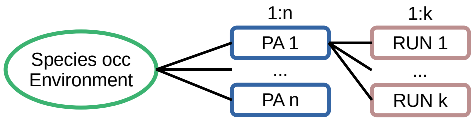



```
> ModelsTable
     model   type       package        func       train
1      ANN binary          nnet        nnet      avNNet
2      CTA binary         rpart       rpart       rpart
3      FDA binary           mda         fda         fda
4      GAM binary           gam         gam    gamLoess
5      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 XGBOOST 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, ...)
```



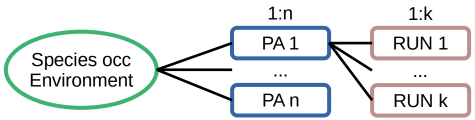

```
> ModelsTable
     model   type      package         func       train
1      ANN binary         nnet         nnet      avNNet
2      CTA binary        rpart        rpart       rpart
3      FDA binary          mda          fda         fda
4      GAM binary          gam          gam    gamLoess
5      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
```

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 )
```



\> 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	XGBOOST	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 )
```

ModelsTable

OptionsBigBoss

bm_ModelingOptions

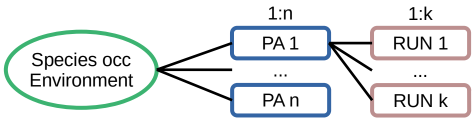
Parameters

(+ formula + stepAIC)

bm_Tuning

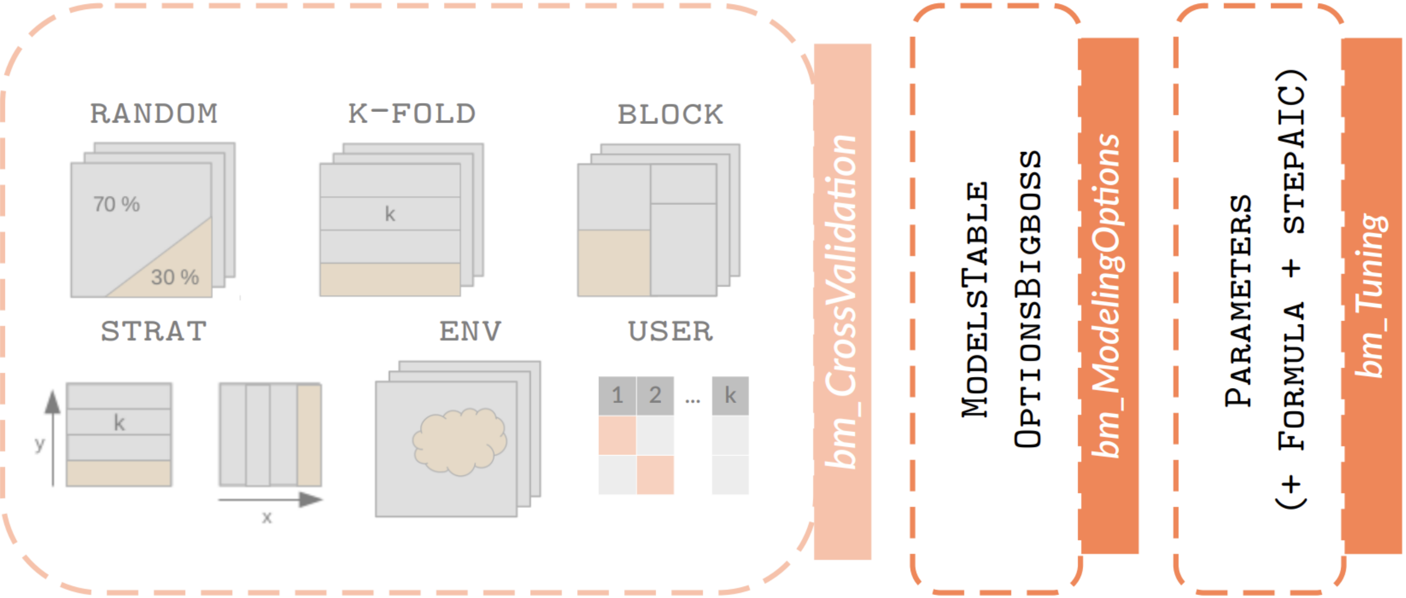
```
> ModelsTable
     model   type      package         func        train
1      ANN binary         nnet         nnet       avNNet
2      CTA binary        rpart        rpart        rpart
3      FDA binary          mda          fda          fda
4      GAM binary          gam          gam     gamLoess
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 XGBOOST binary      xgboost      xgboost      xgbTree
```

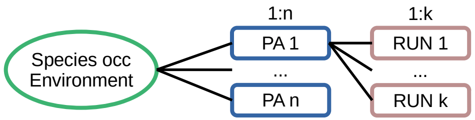
```
user.ANN = list('_allData_allRun' =
     list( size = 5,
           decay = 0.5,
           trace = FALSE,
           rang = 0.1,
           maxit = 500))
```



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

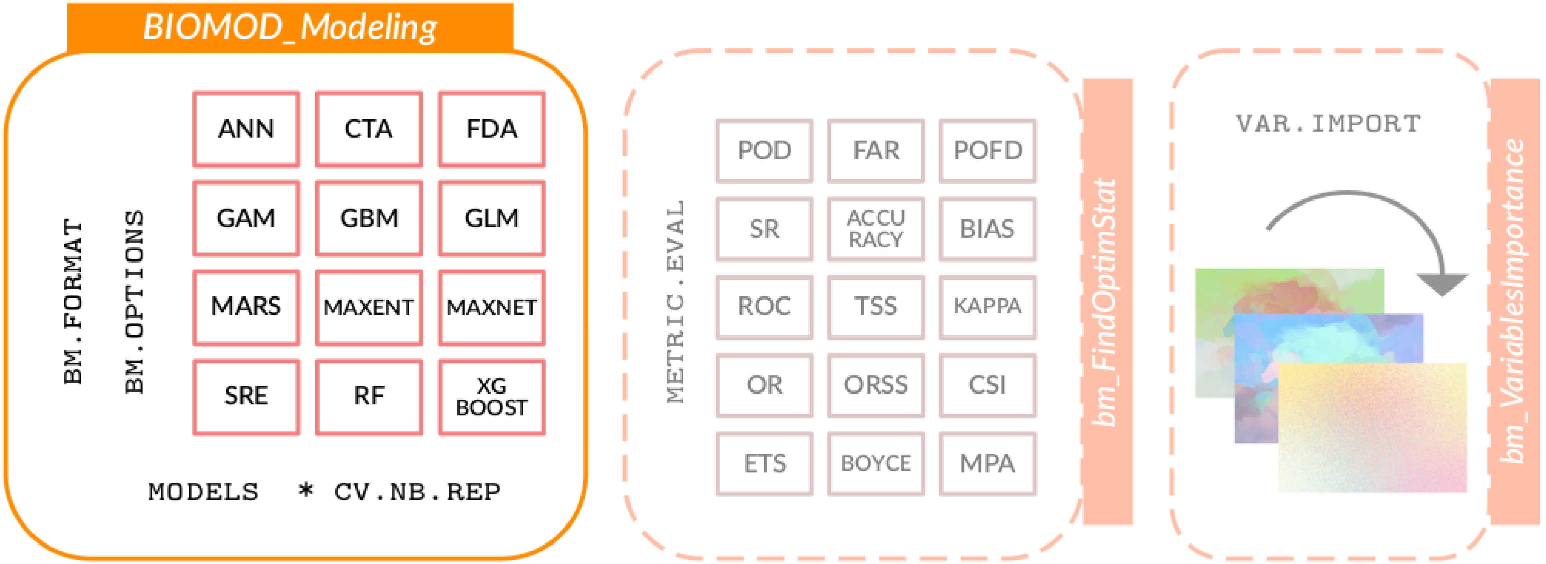


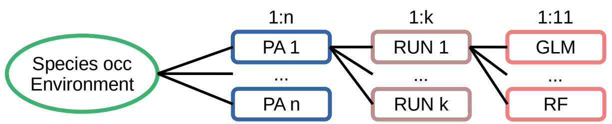



```
> ModelsTable
    model   type       package        func       train
1     ANN binary          nnet        nnet      avNNet
2     CTA binary         rpart       rpart       rpart
3     FDA binary           mda         fda         fda
4     GAM binary           gam         gam    gamLoess
5     GAM 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 XGBOOST binary      xgboost     xgboost     xgbTree
```

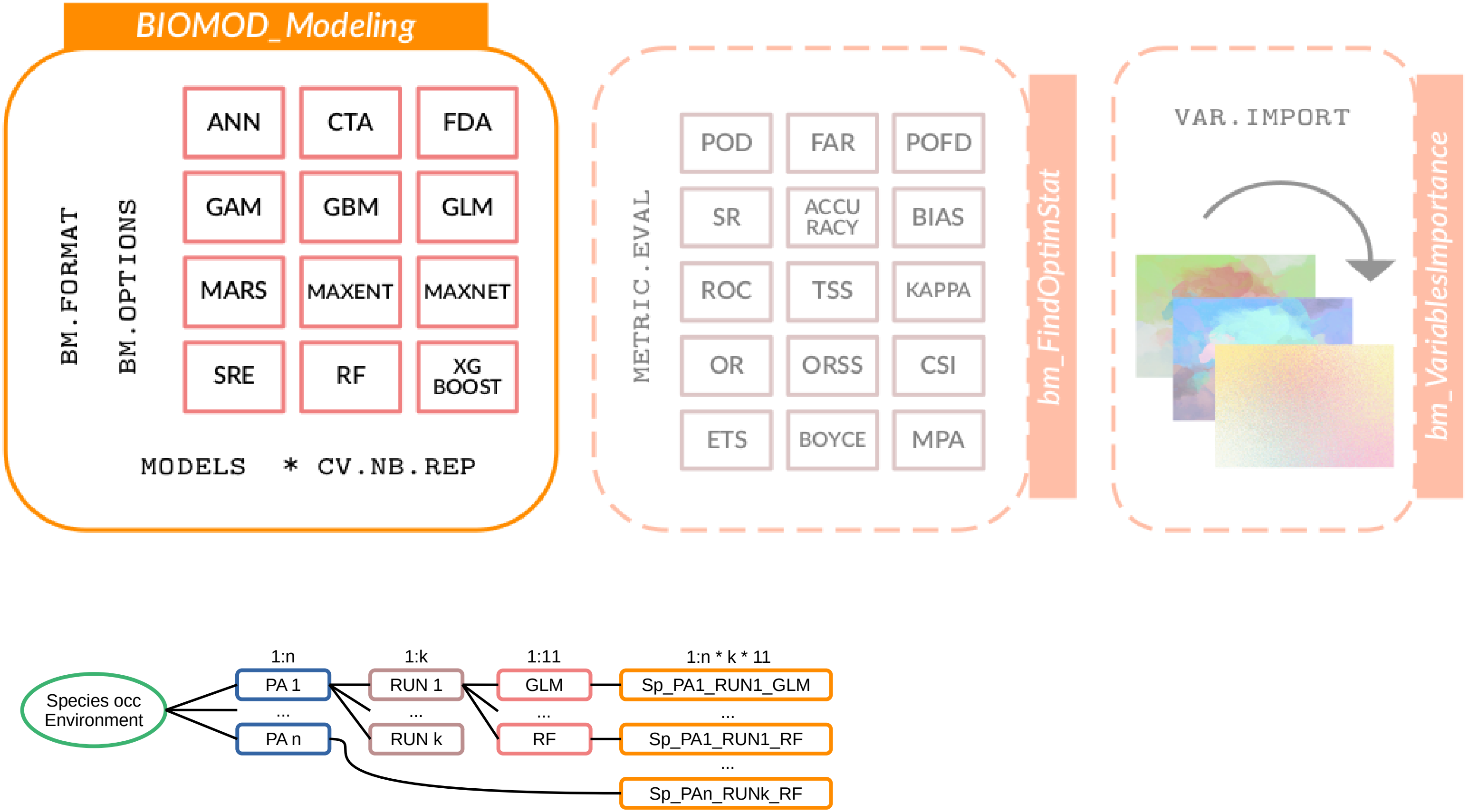
- test a bunch of parameters, and try to keep the « best » according to some evaluation metrics (TSS or ROC)

2.a Single models



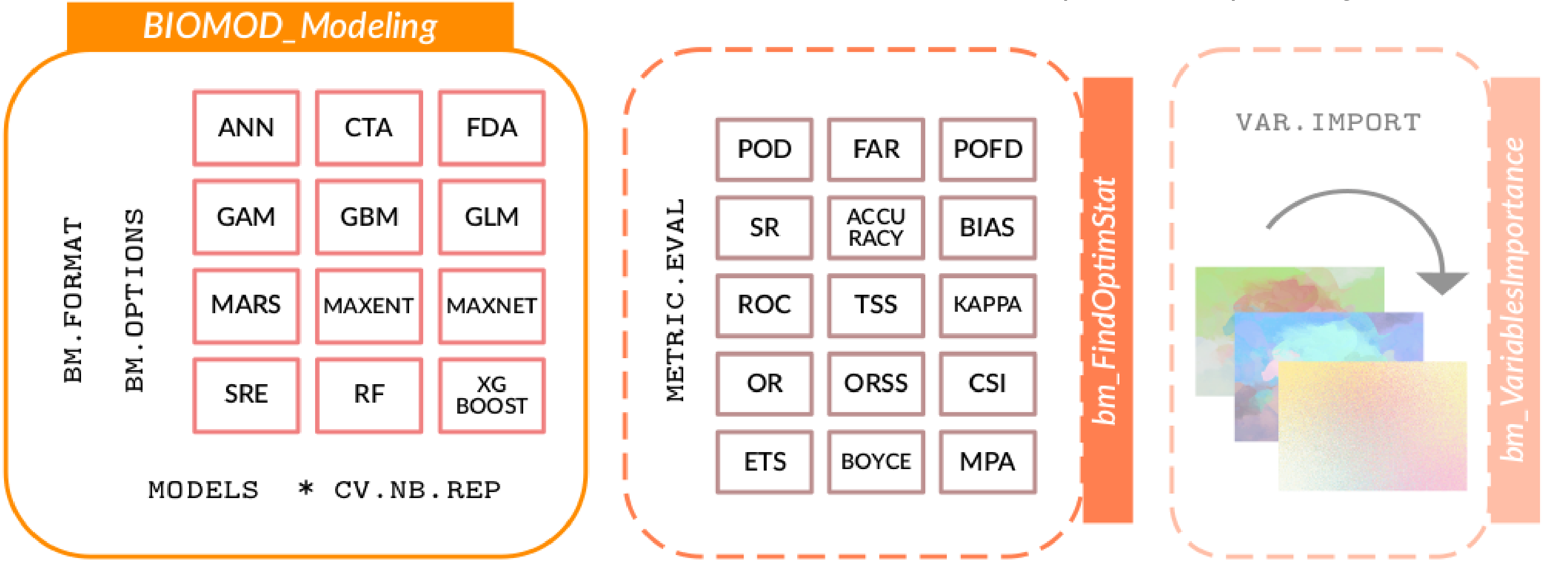


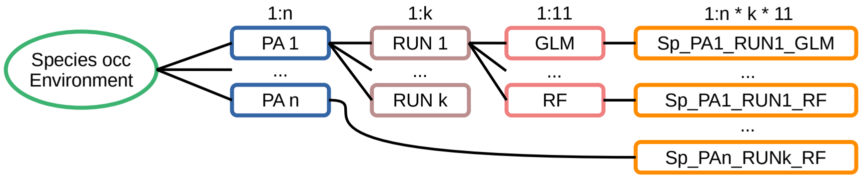
2.a Single models



2.a Single models

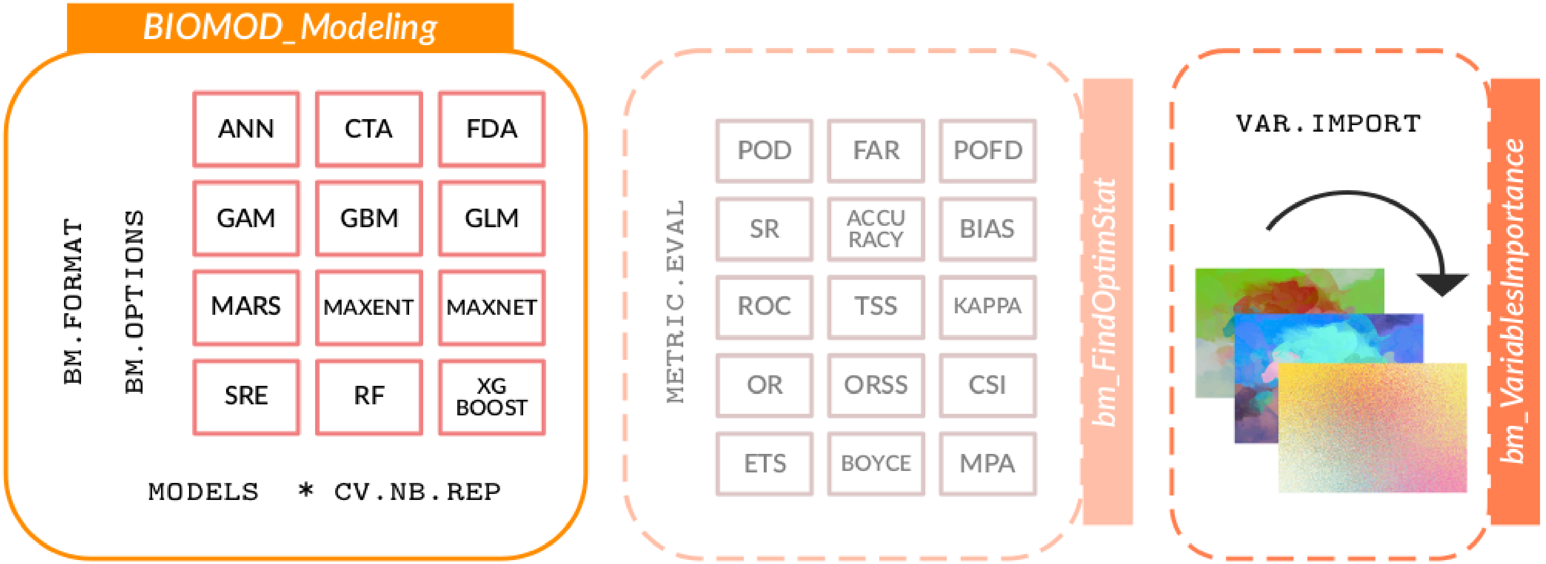
- except ROC, all evaluation metrics obtained from contingency table (*containing TP, FP, TN, FN*)
- require a **binary transformation** :
  - range of thresholds tested
  - keep thresold optimising the evaluation metric

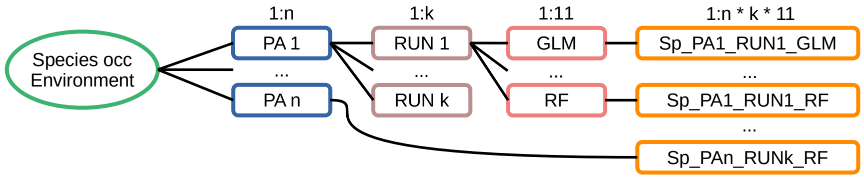




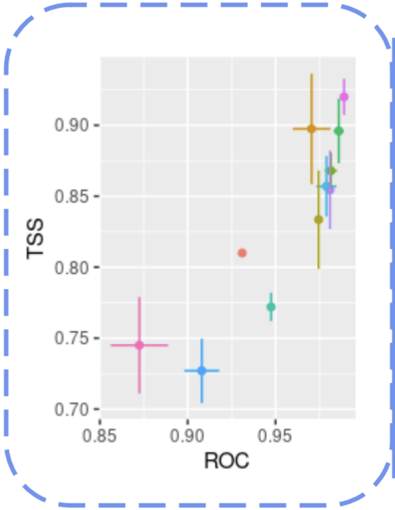
2.a Single models

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

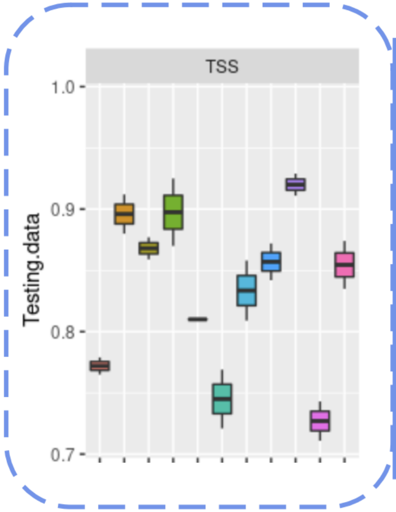




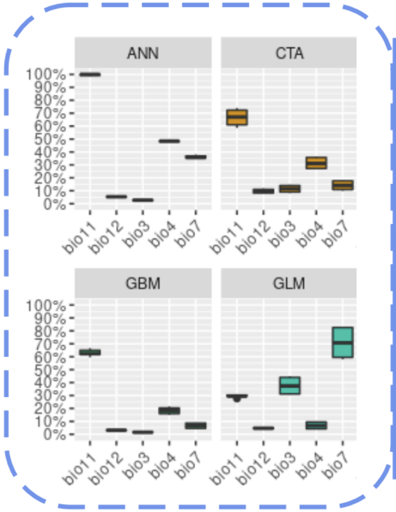
3.a Exploring single models



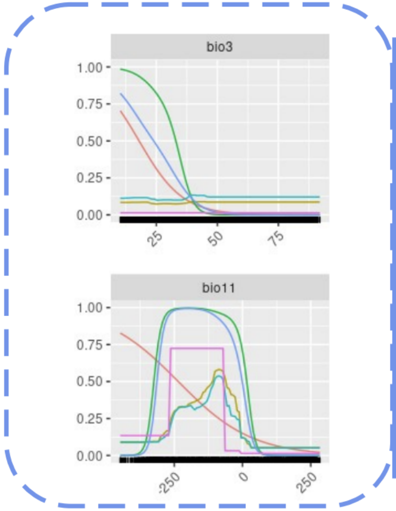
bm_PlotEvalMean



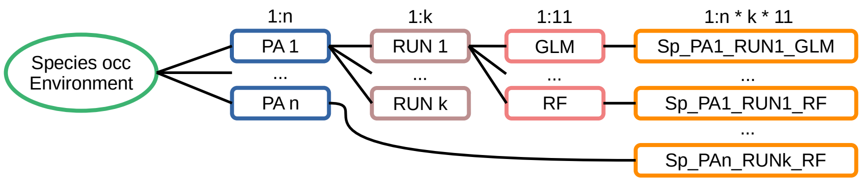
bm_PlotEvalBoxplot



bm_PlotVarImpBoxplot

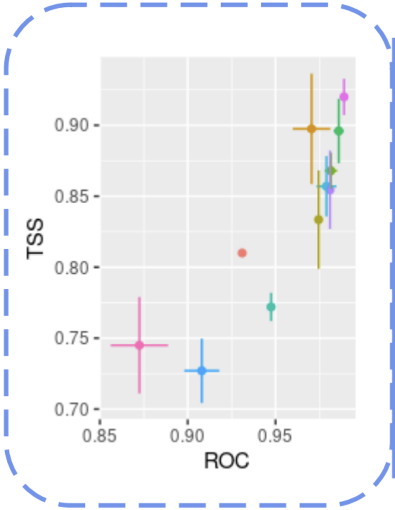


bm_PlotResponseCurves

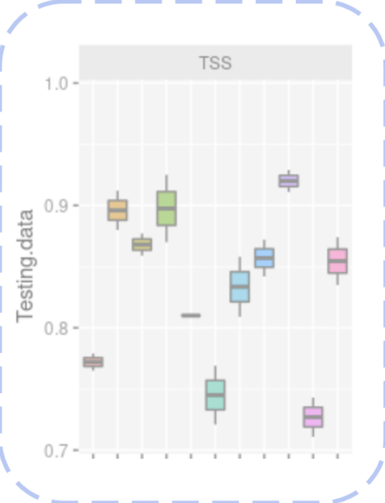


3.a Exploring single models

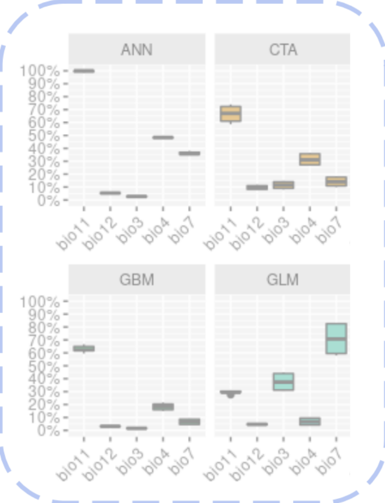
- « evaluation space »
- visualize the metrics consistency between models



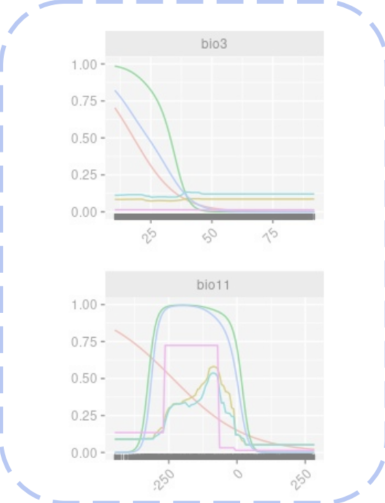
bm_PlotEvalMean



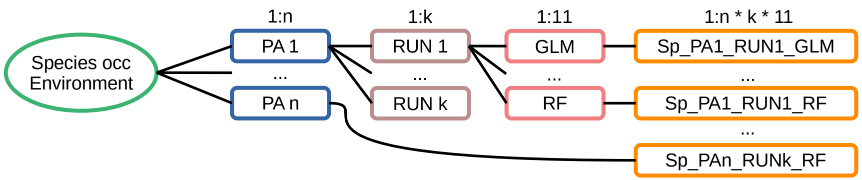
bm_PlotEvalBoxplot



bm_PlotVarImpBoxplot

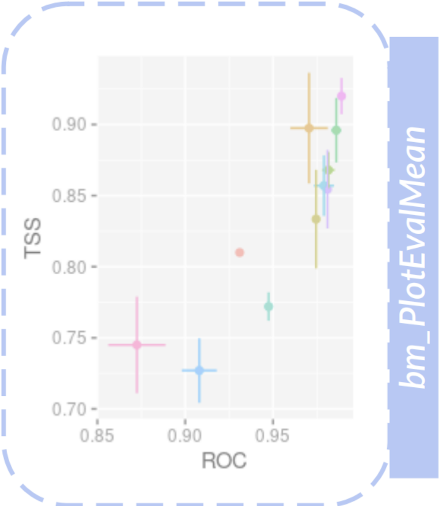


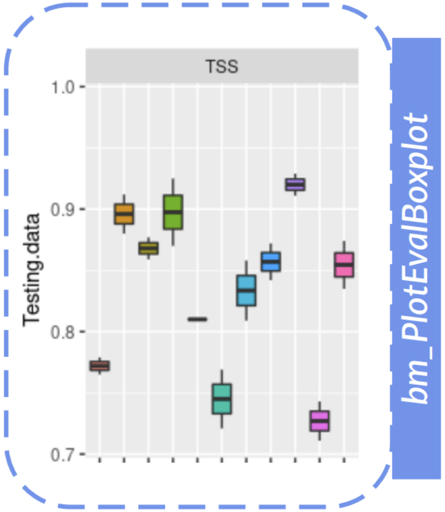
bm_PlotResponseCurves

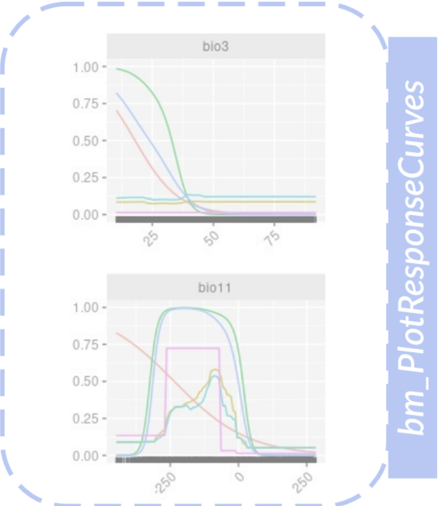


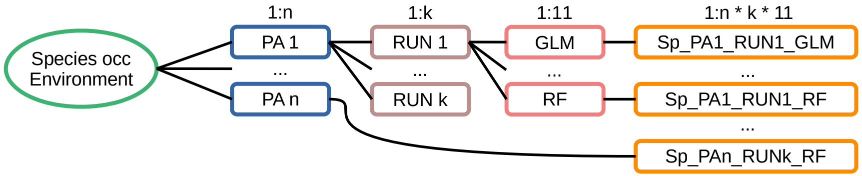
3.a Exploring single models

- more classical view
- visualize the metrics consistency between models



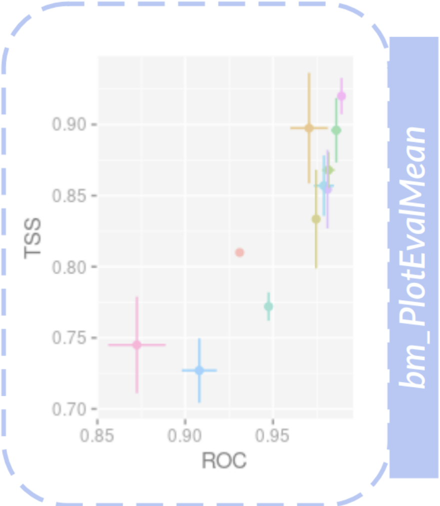


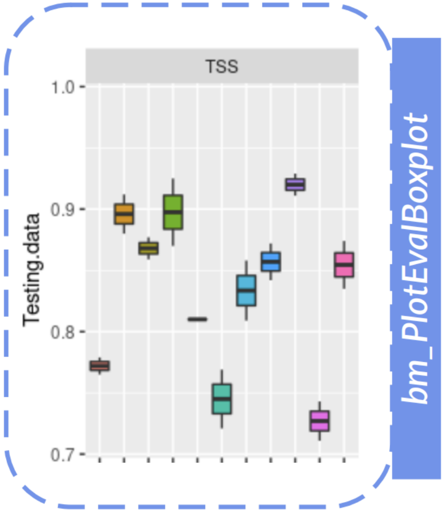


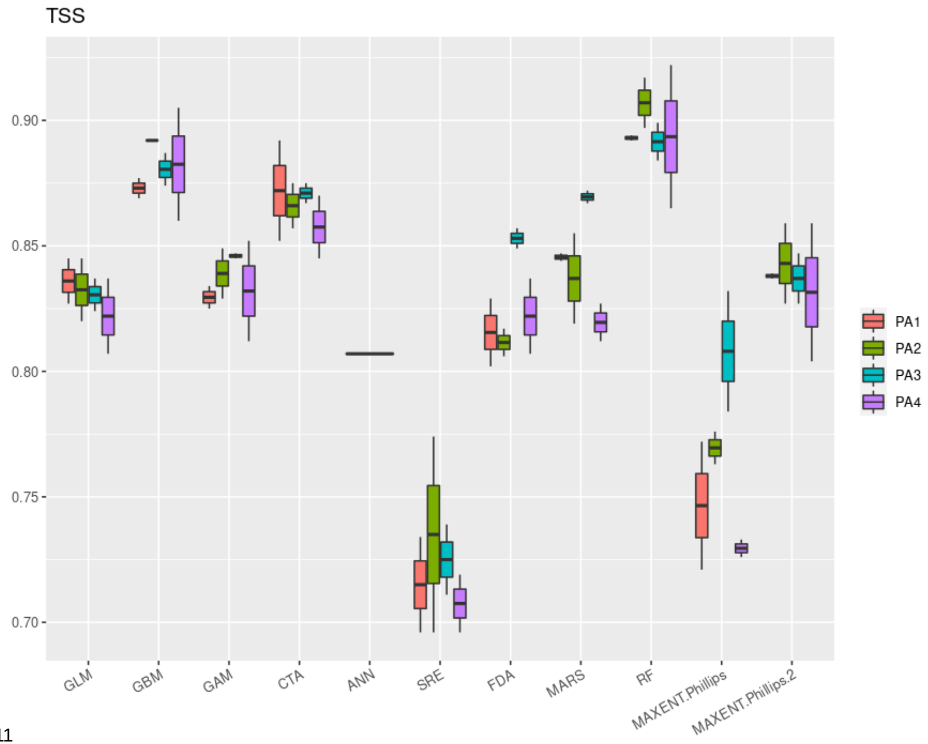


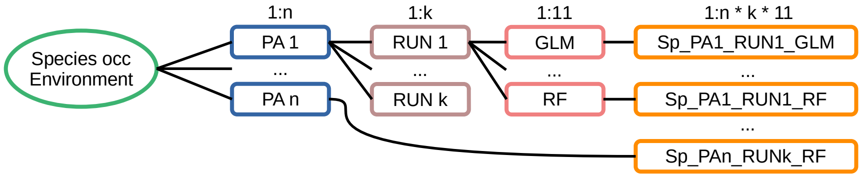
3.a Exploring single models

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



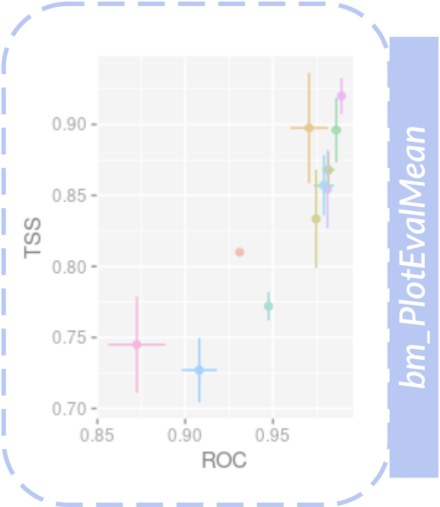


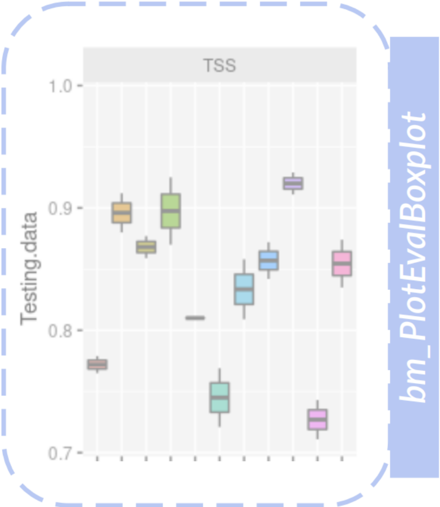


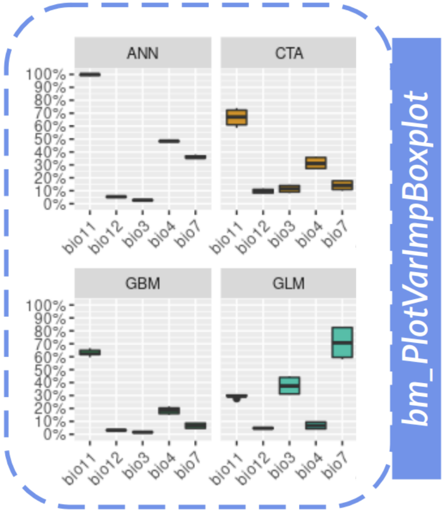


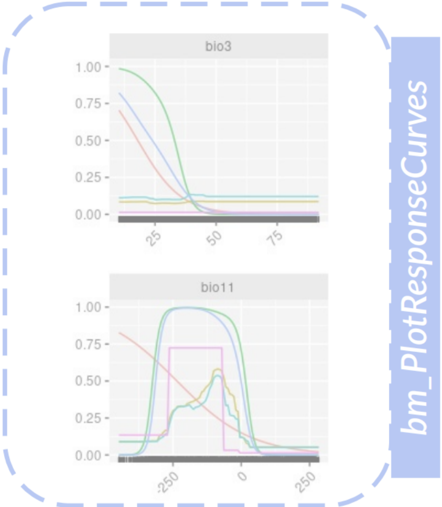
3.a Exploring single models

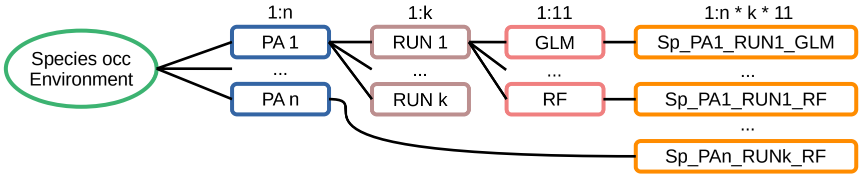
- compare importance of variables between models
- visualize the consistency between models
  (*and different types of models*)





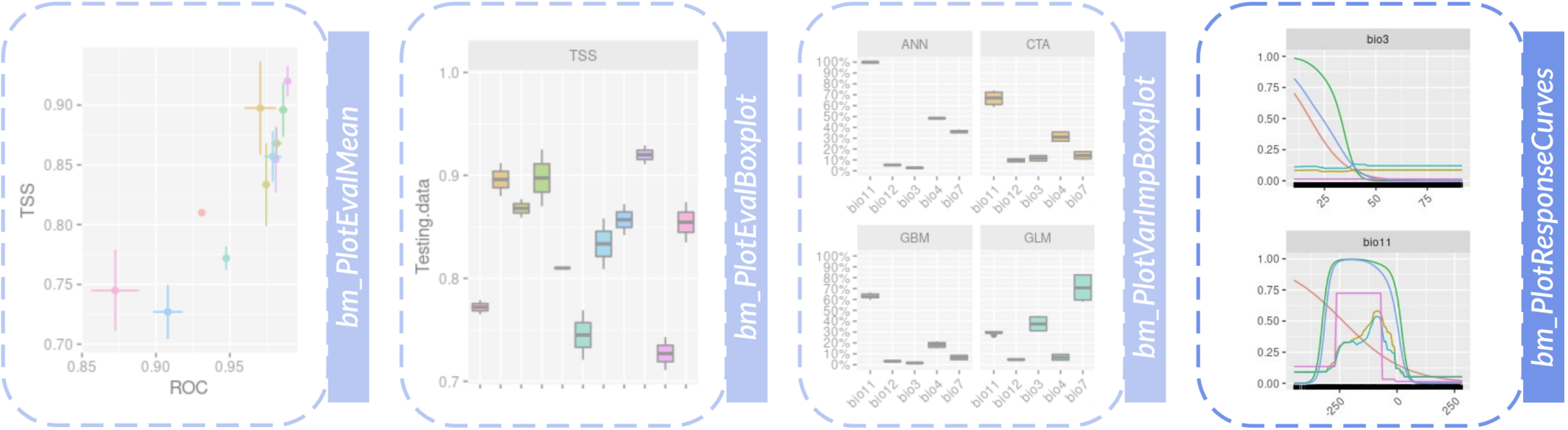


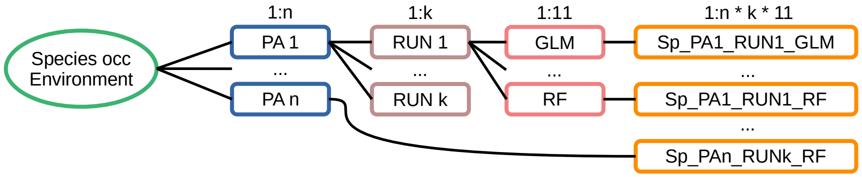




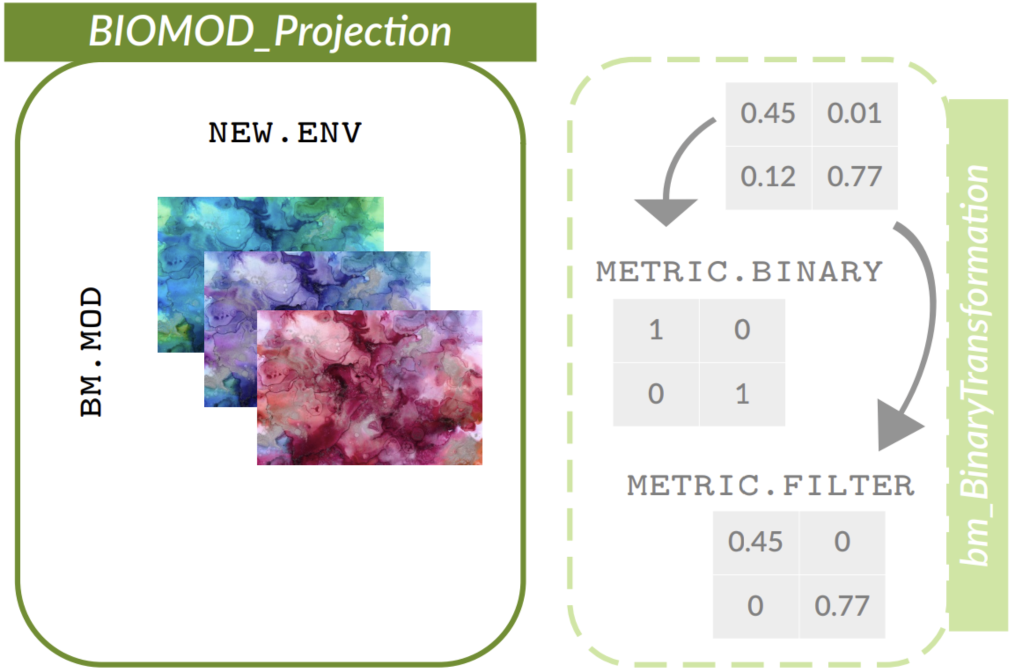
3.a Exploring single models

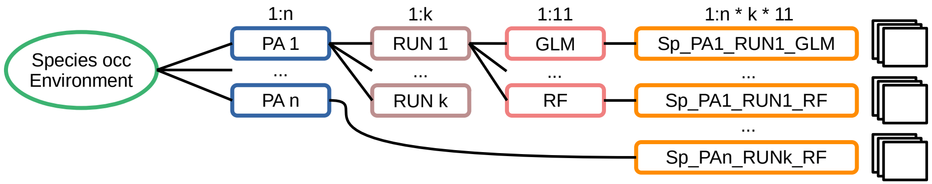
- better understand the effect of each variable along its gradient onto the probability of presence
- visualize the consistency between models (*and different types of models*)





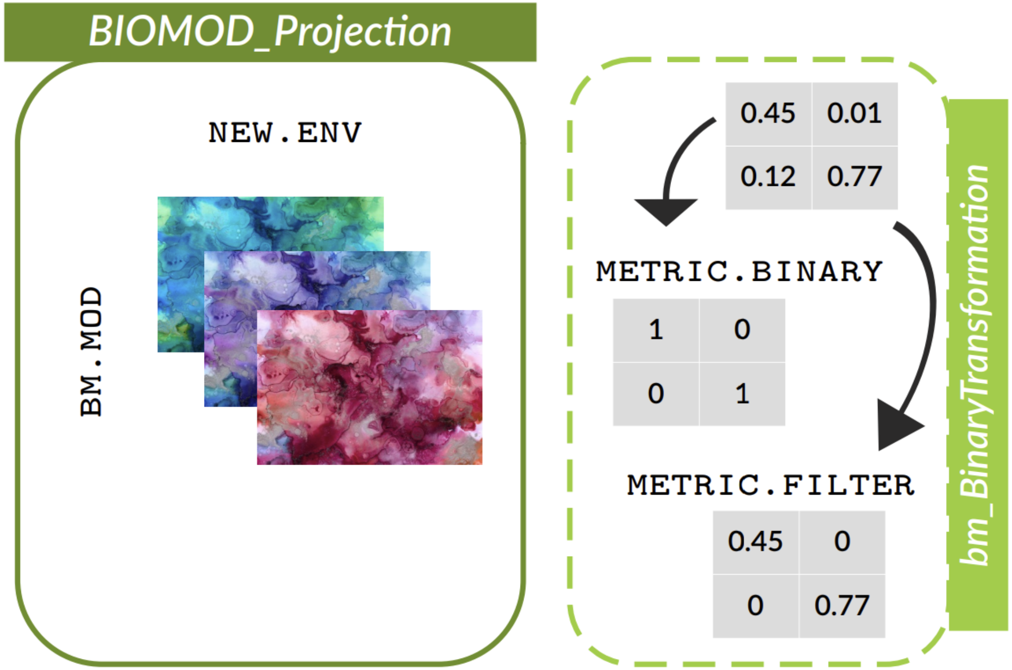
4.a Projecting single models

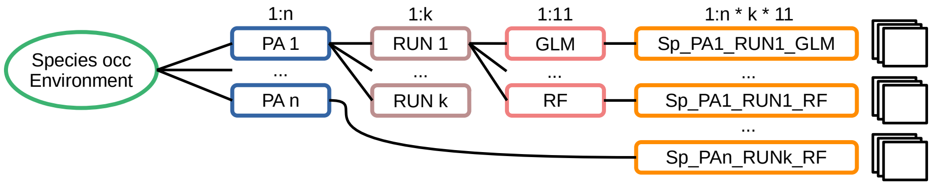




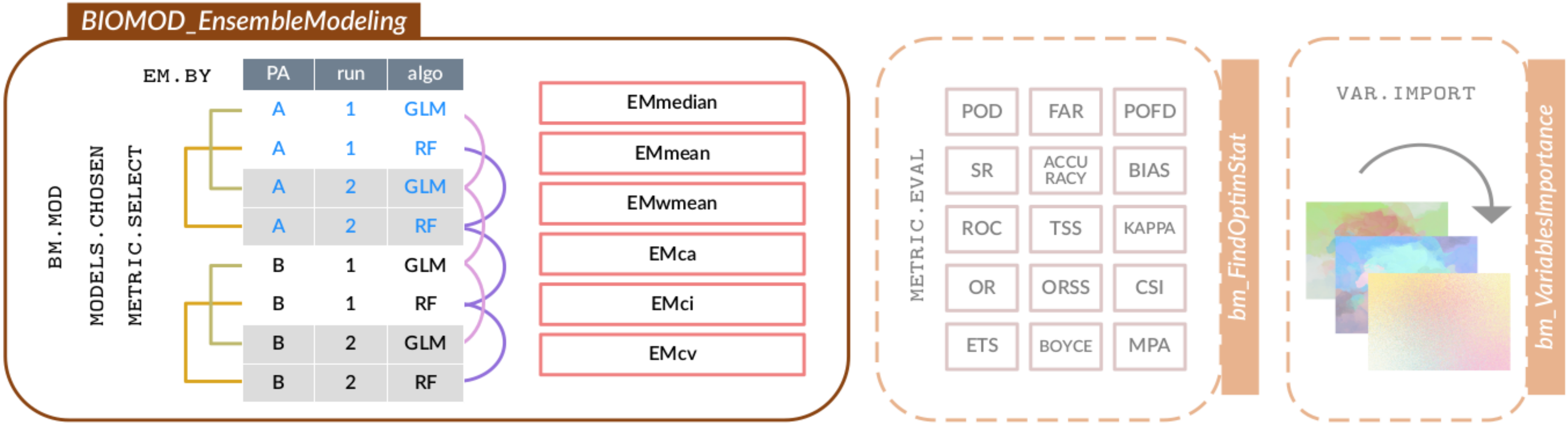
4.a Projecting single models

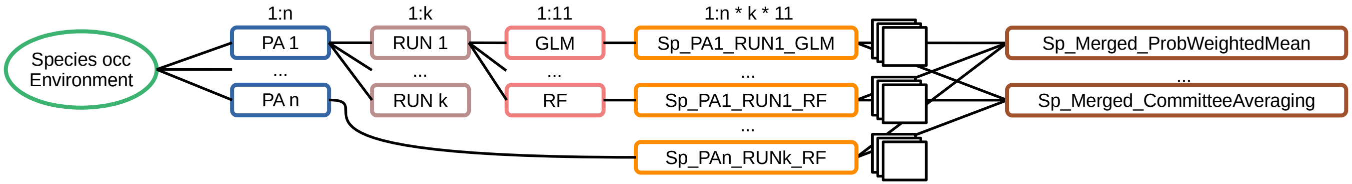
- transformation associated to one evaluation metric (*one map created for each metric selected*)
- use the threshold maximising the chosen metric





2.b Ensemble models

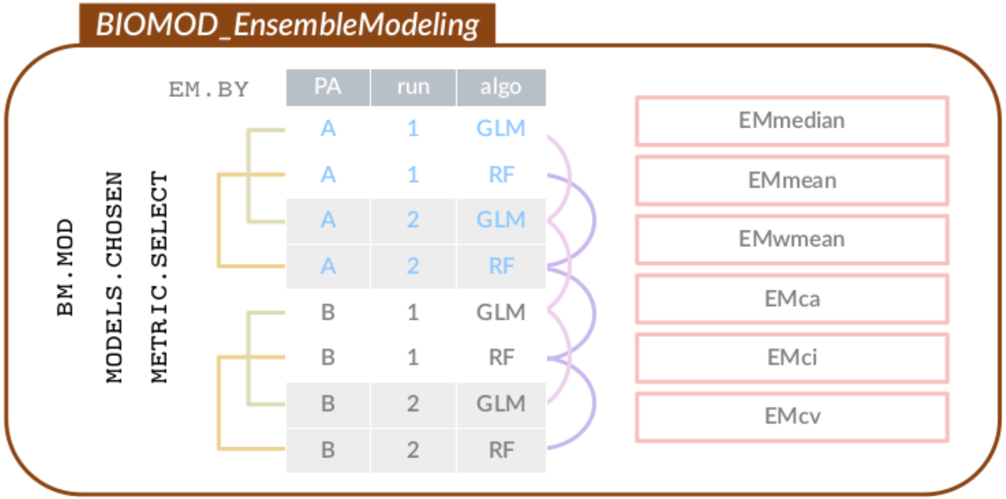


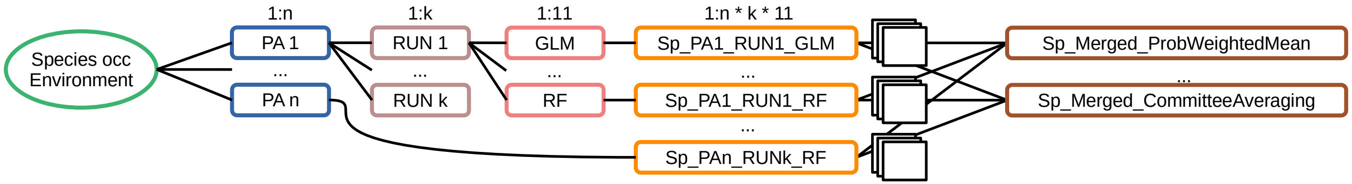


2.b Ensemble models

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

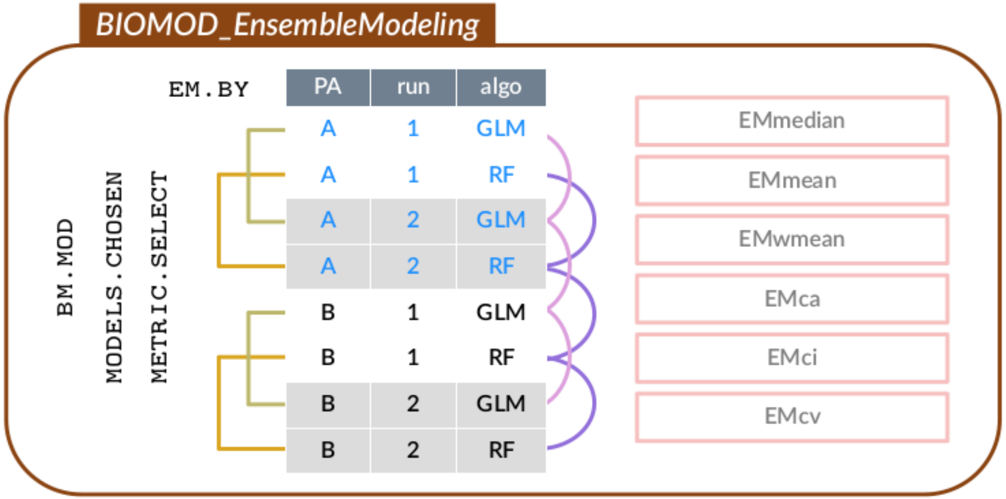




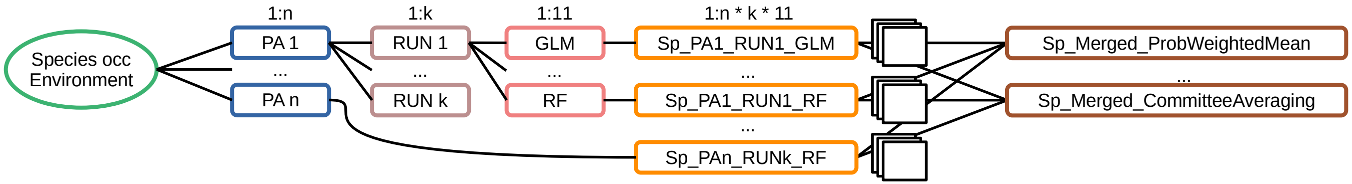
2.b Ensemble models

Step 1 : filter single models
Step 2 : gather single models

- different ways of **combining** single models together :
  - *all*
  - *algo*
  - *PA*
  - *PA+algo*
  - *PA+run*



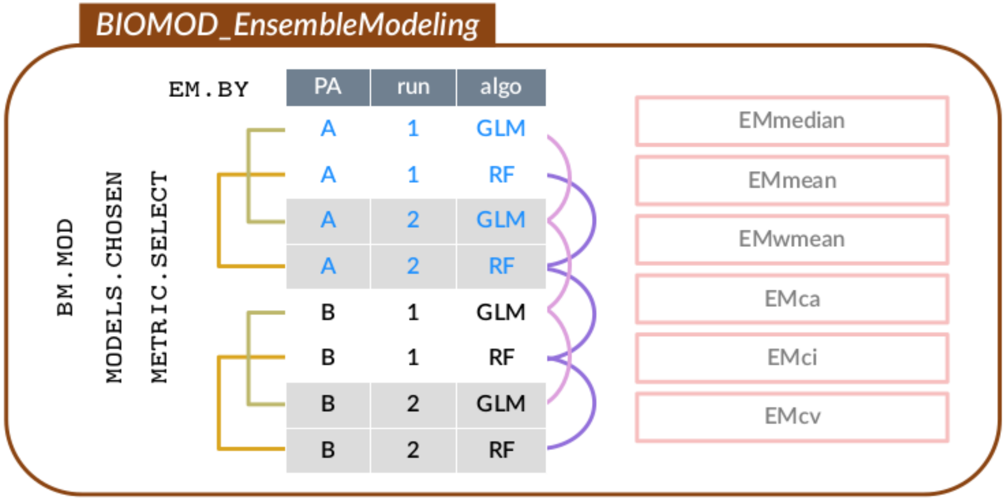
PA1_RUN1_GLM	PA1_RUN1_RF
PA1_RUN2_GLM	PA1_RUN2_RF
PA1_RUN3_GLM	PA1_RUN3_RF
PA2_RUN1_GLM	PA2_RUN1_RF
PA2_RUN2_GLM	PA2_RUN2_RF
PA2_RUN3_GLM	PA2_RUN3_RF



2.b Ensemble models

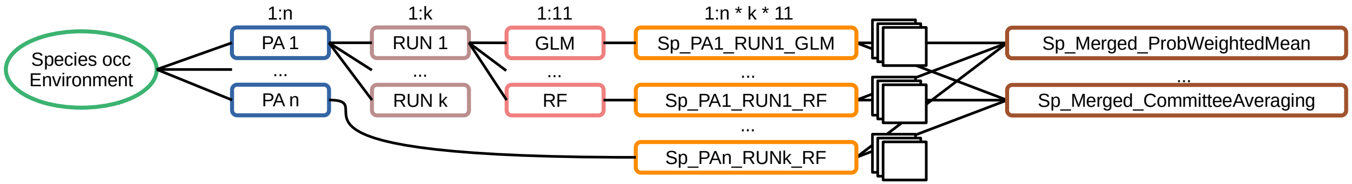
Step 1 : filter single models
Step 2 : gather single models

- 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

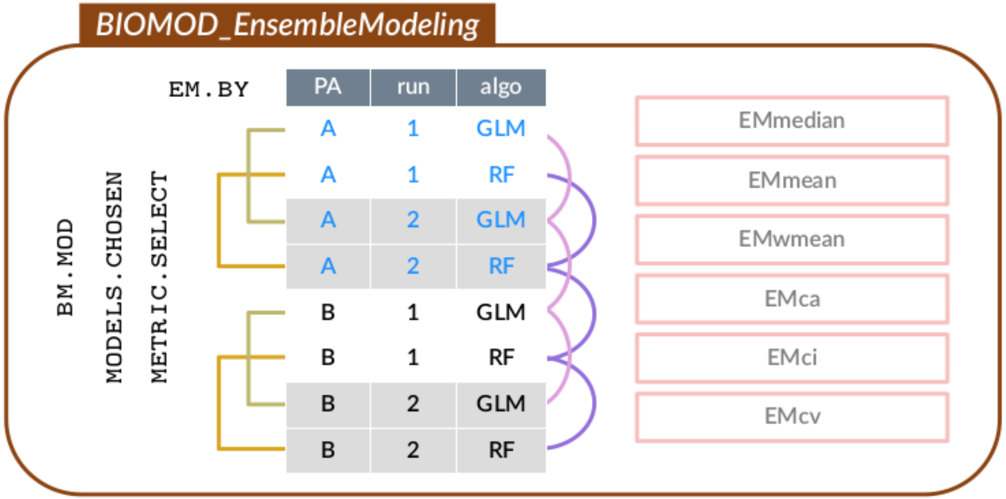
All models

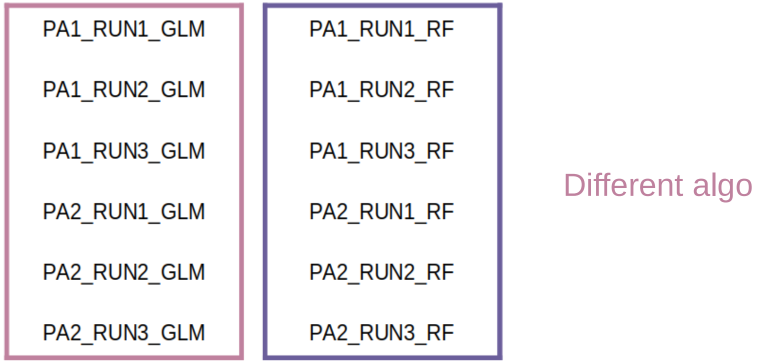


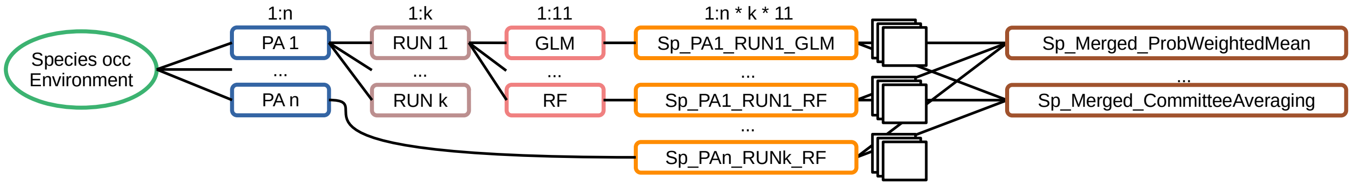
2.b Ensemble models

Step 1 : filter single models
Step 2 : gather single models

- different ways of **combining** single models together :
  - *all*
  - *algo*
  - *PA*
  - *PA+algo*
  - *PA+run*



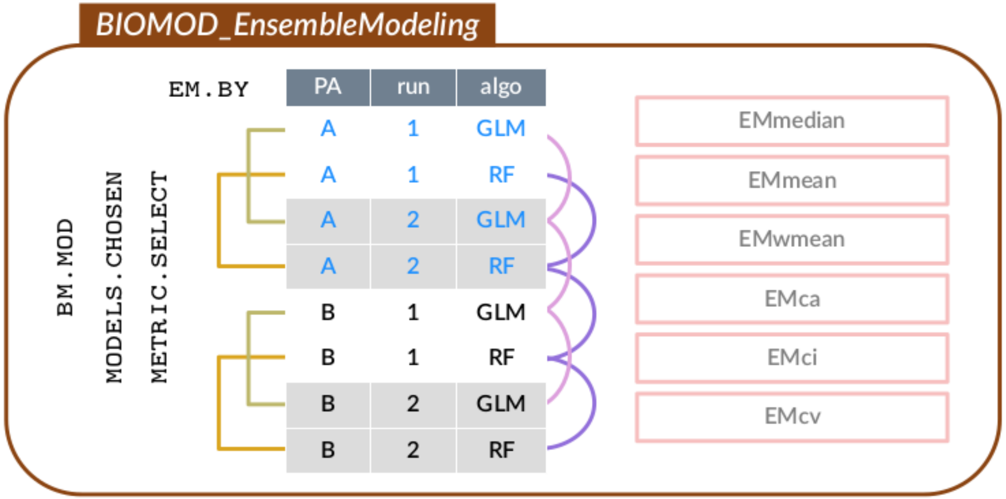


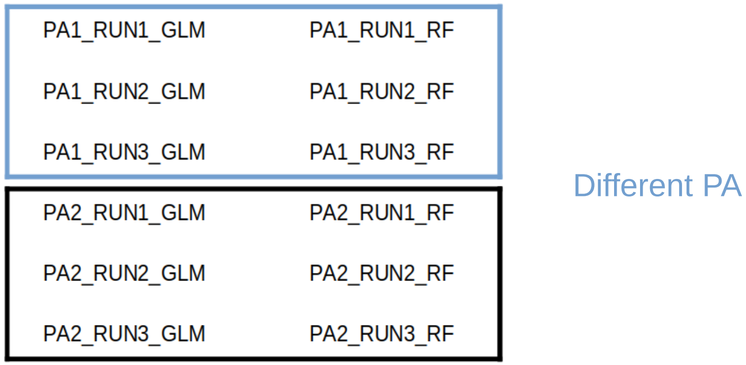


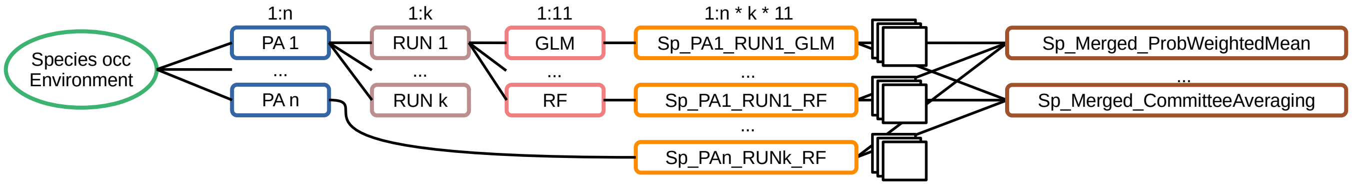
2.b Ensemble models

Step 1 : filter single models
Step 2 : gather single models

- different ways of **combining** single models together :
  - *all*
  - *algo*
  - *PA*
  - *PA+algo*
  - *PA+run*



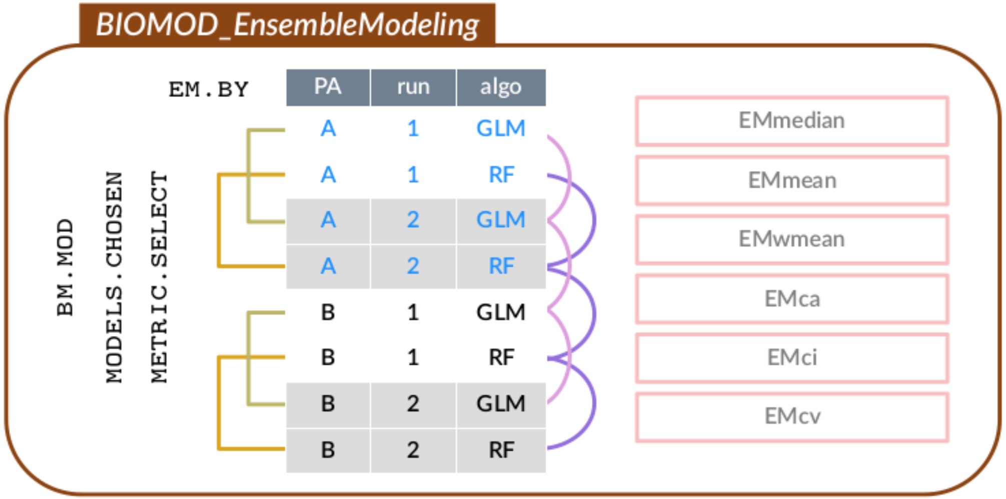


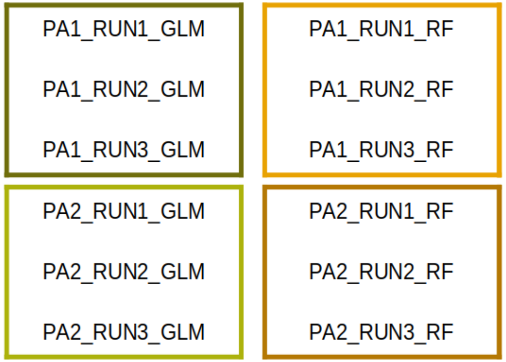


2.b Ensemble models

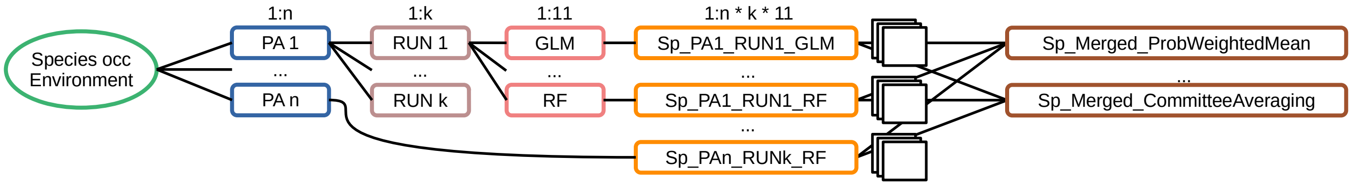
Step 1 : filter single models
Step 2 : gather single models

- different ways of **combining** single models together :
  - *all*
  - *algo*
  - *PA*
  - *PA+algo*
  - *PA+run*





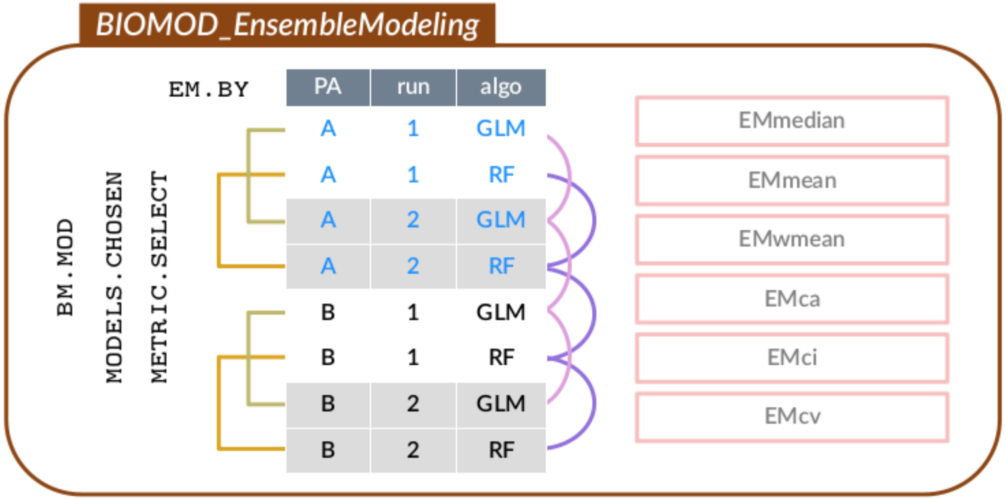
Different PA and algo



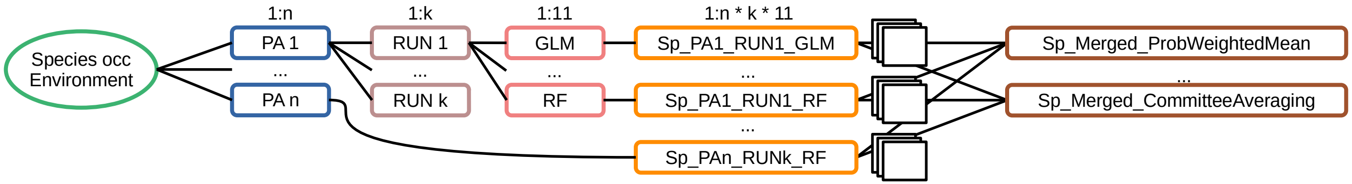
2.b Ensemble models

Step 1 : filter single models
Step 2 : gather single models

- different ways of **combining** single models together :
  - *all*
  - *algo*
  - *PA*
  - *PA+algo*
  - *PA+run*



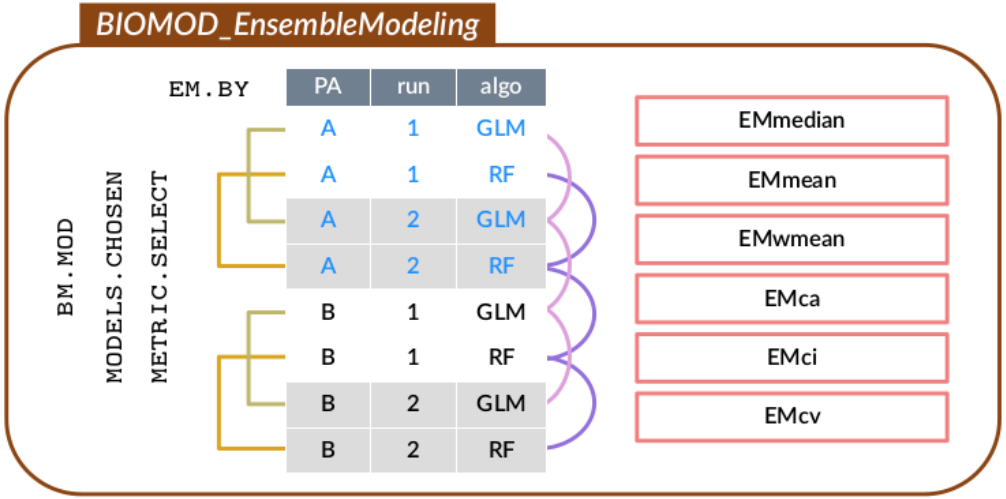
Different PA and run

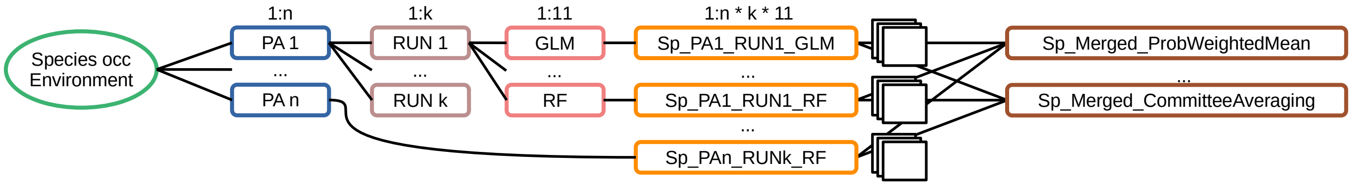


2.b Ensemble models

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

Step 1 : filter single models
Step 2 : gather single models
Step 3 : build ensemble models

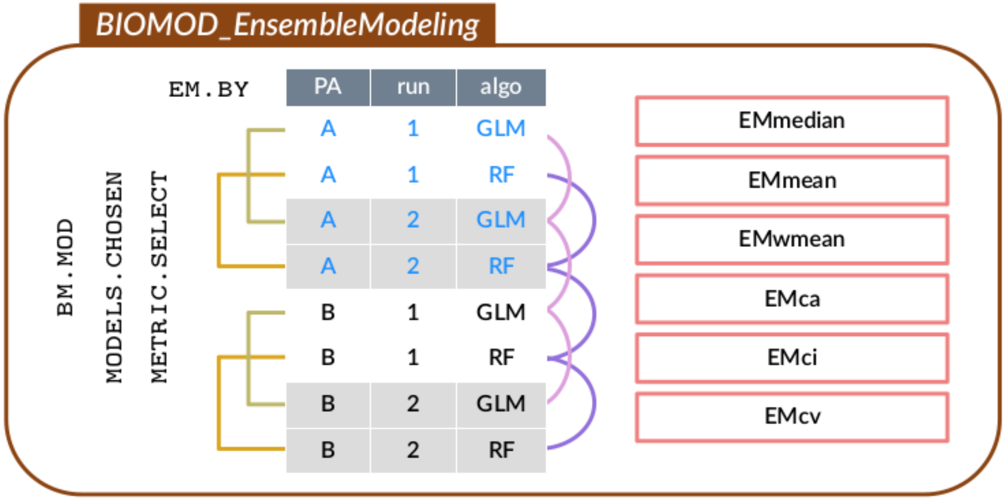


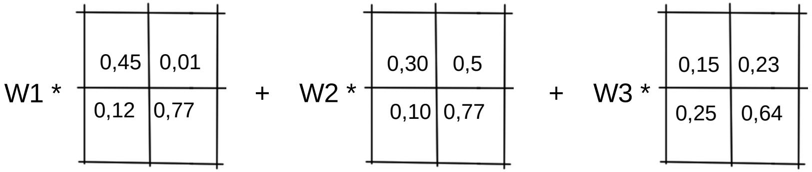


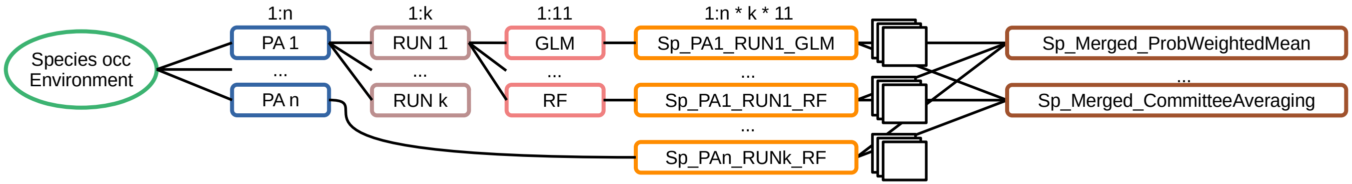
2.b Ensemble models

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**



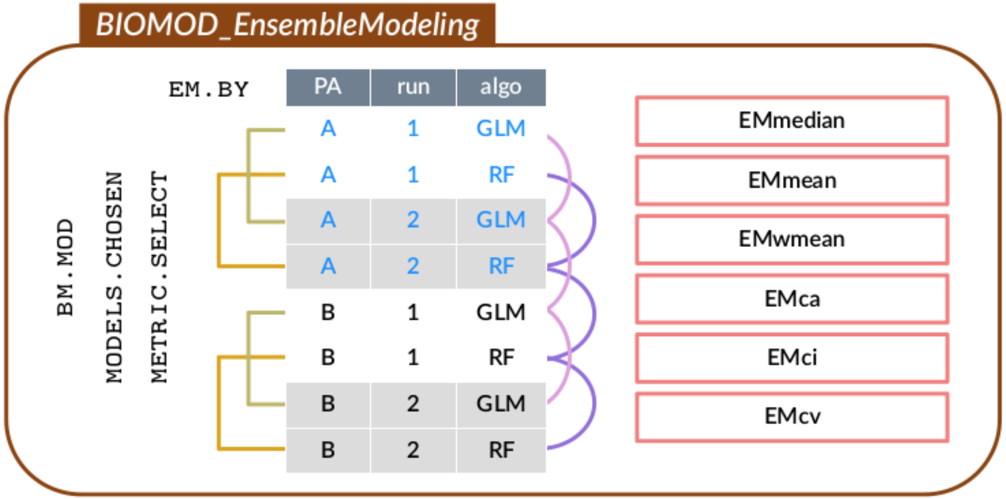


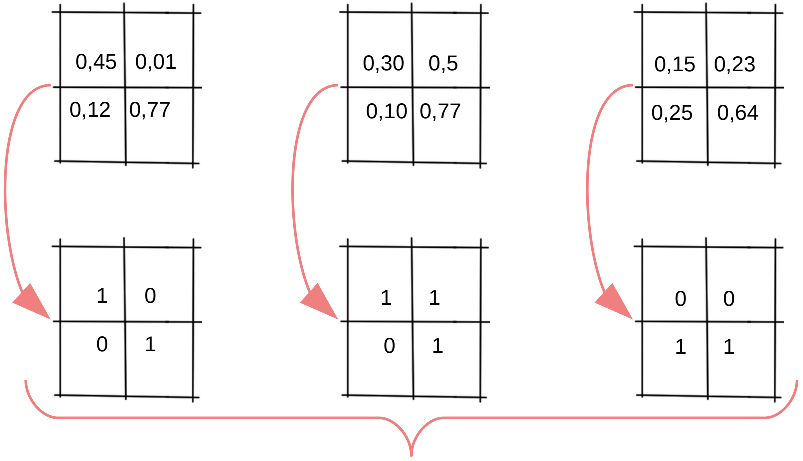


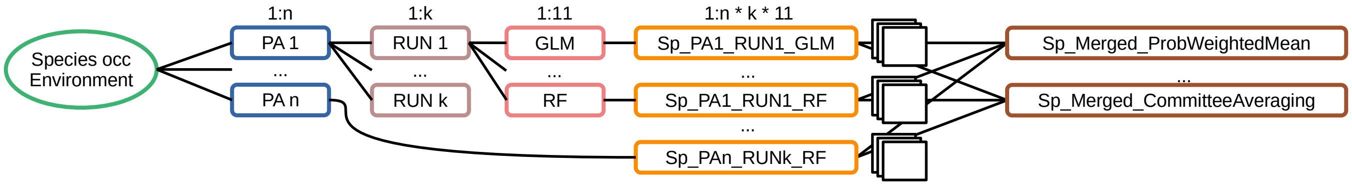
2.b Ensemble models

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**

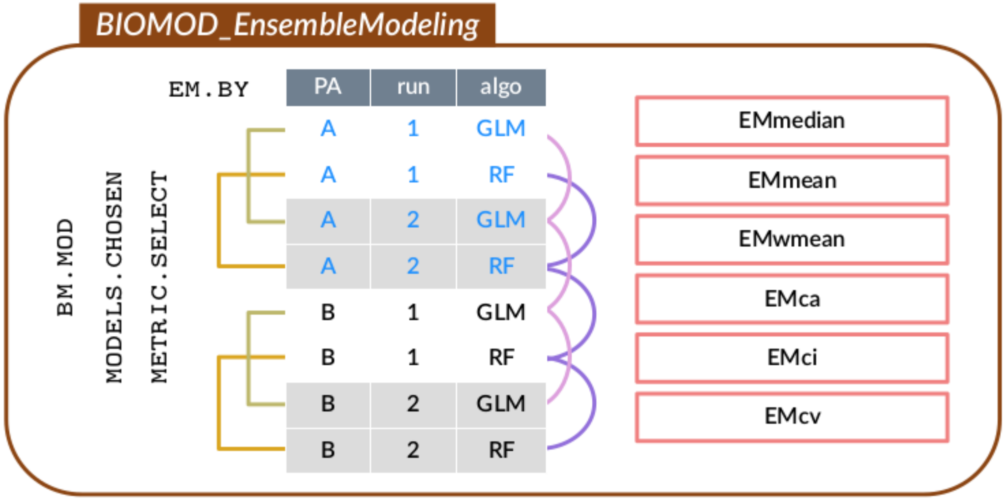




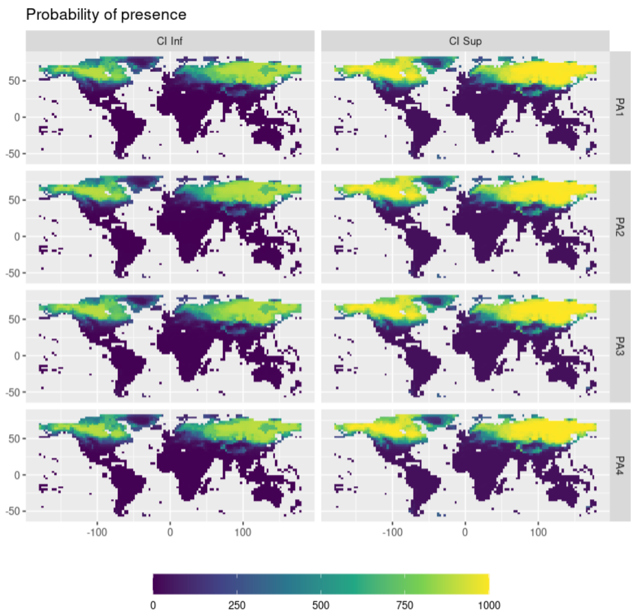


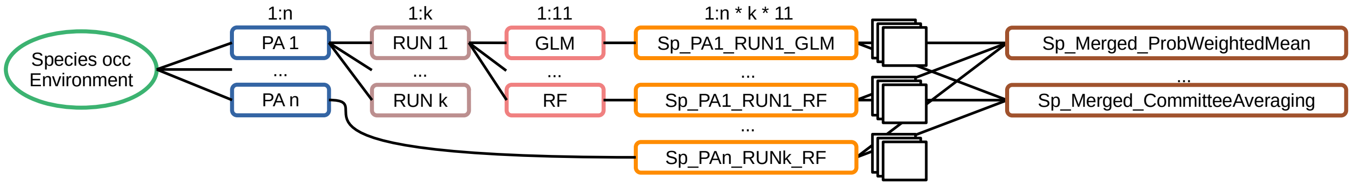
2.b Ensemble models

Step 1 : filter single models
Step 2 : gather single models
Step 3 : build ensemble models



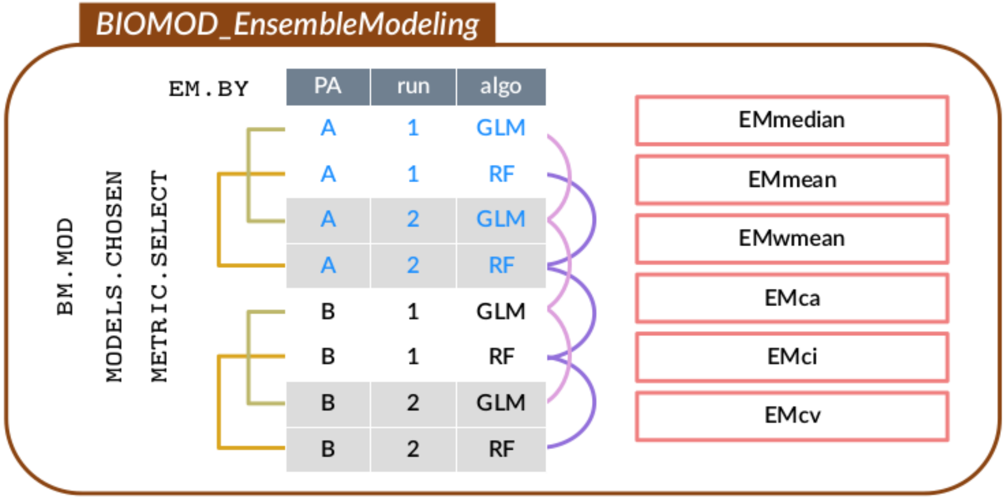
- « simple » ensemble models : mean or median
- « complex » ensemble models :
  - probability weighted mean
  - committe averaging
- « exploratory » ensemble models :
  - **confidence intervals** or **coefficient of variation**



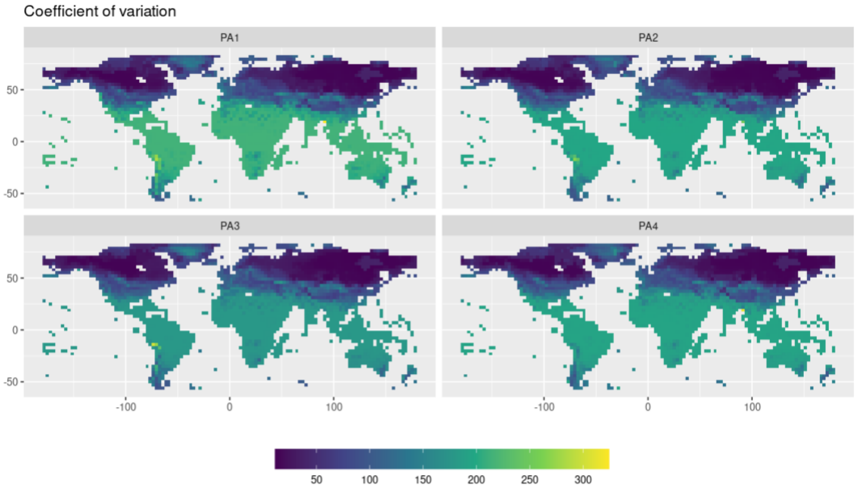


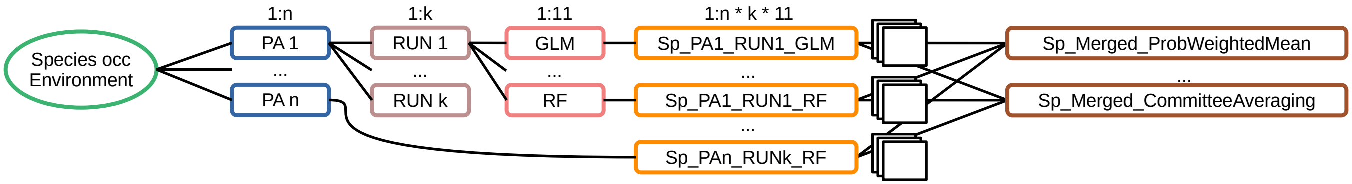
2.b Ensemble models

Step 1 : filter single models
Step 2 : gather single models
Step 3 : build ensemble models



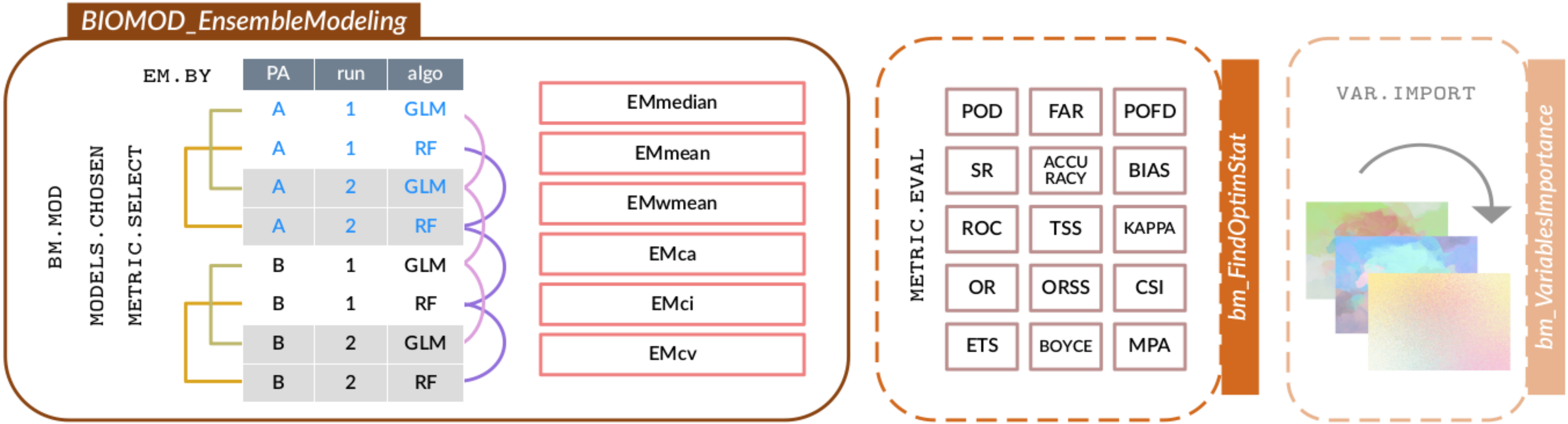
- « simple » ensemble models : mean or median
- « complex » ensemble models :
  - probability weighted mean
  - committe averaging
- « exploratory » ensemble models :
  - **confidence intervals** or **coefficient of variation**

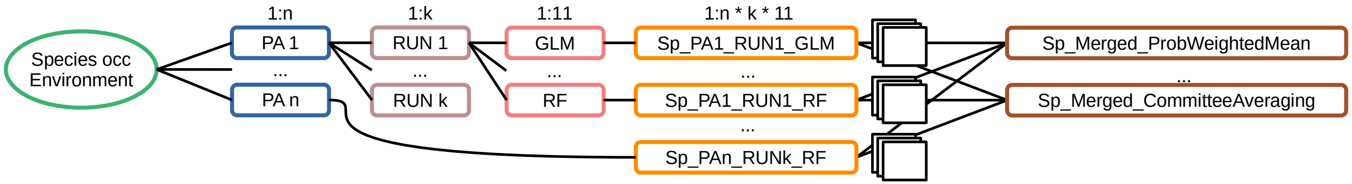




2.b Ensemble models

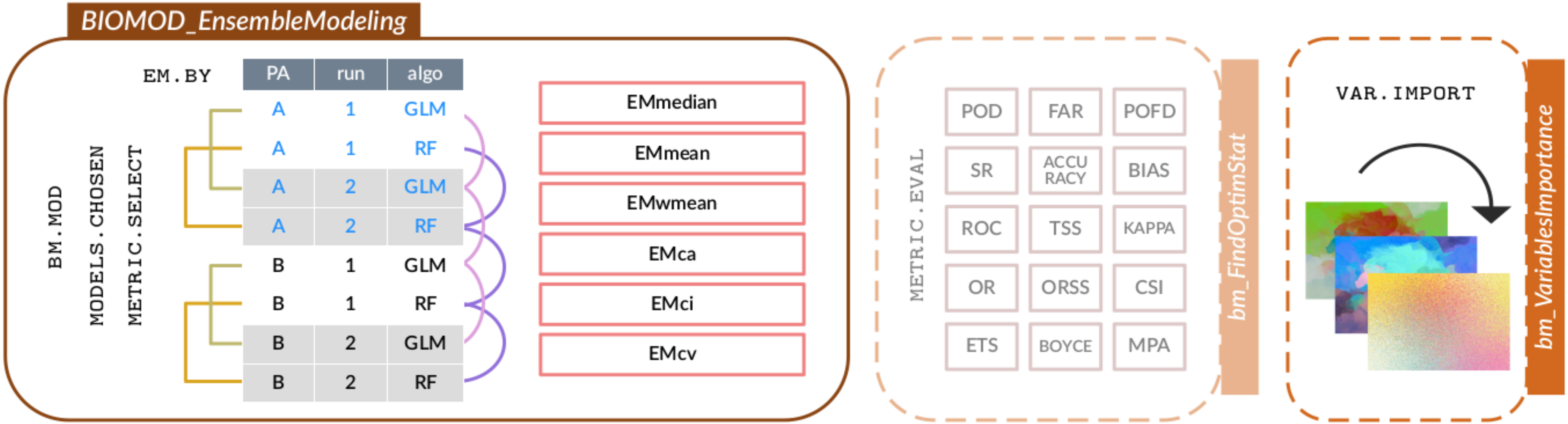
- except ROC, all evaluation metrics obtained from contingency table (*containing TP, FP, TN, FN*)
- require a **binary transformation** :
  - range of thresholds tested
  - keep thresold optimising the evaluation metric





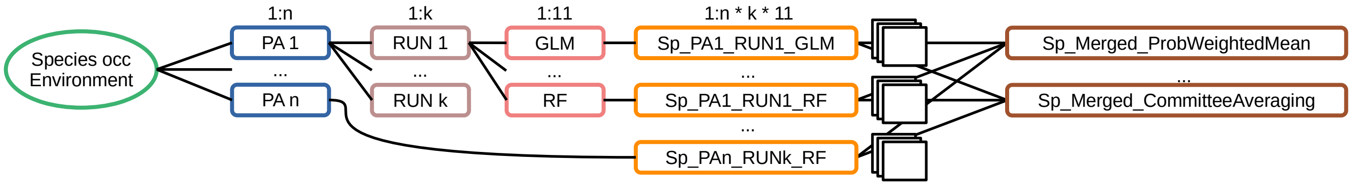
2.b Ensemble models

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

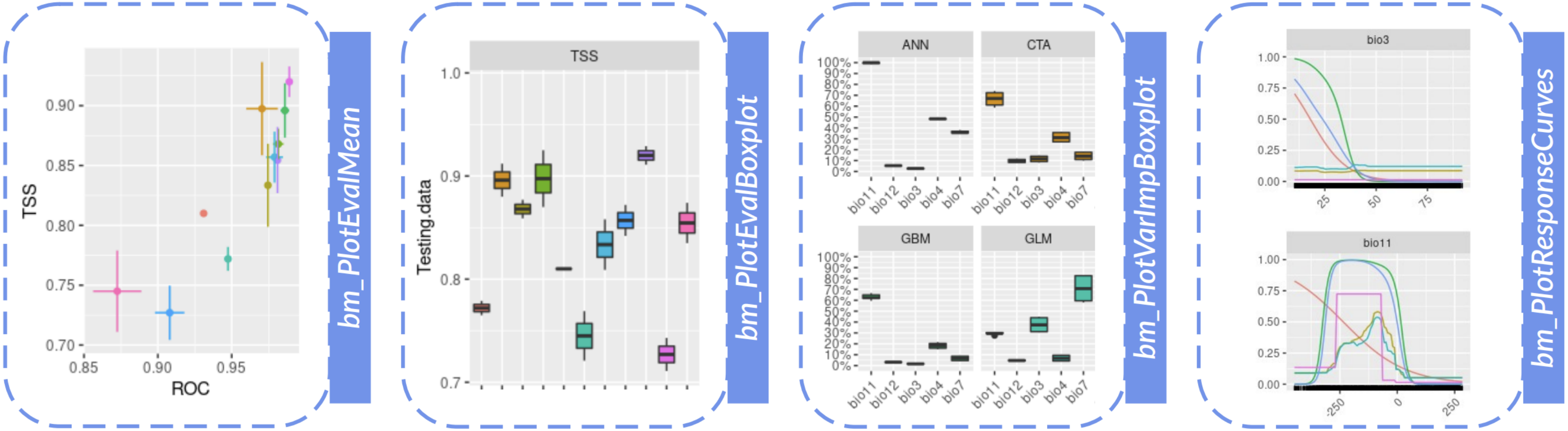


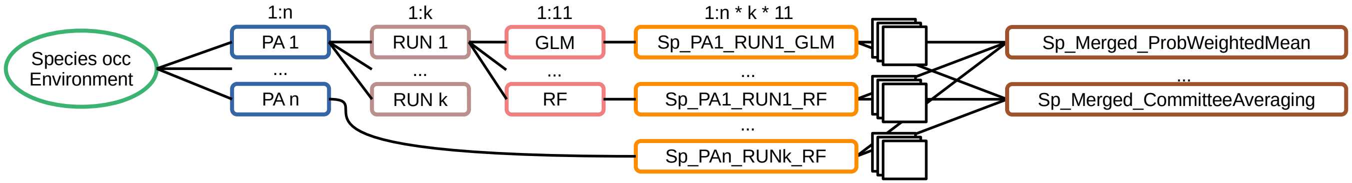
IMPORTANT

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

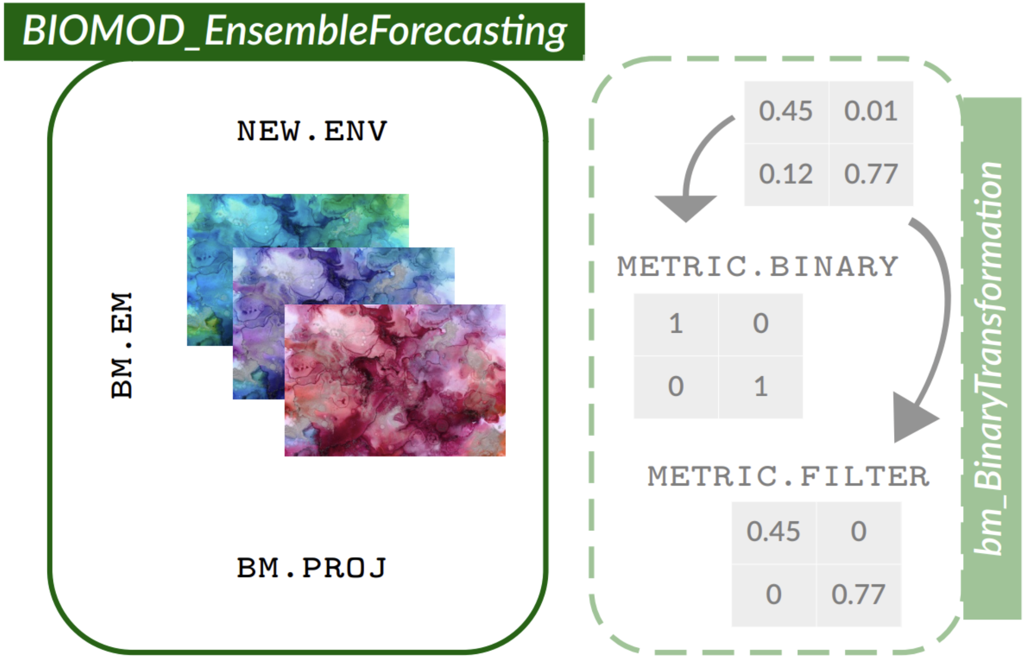


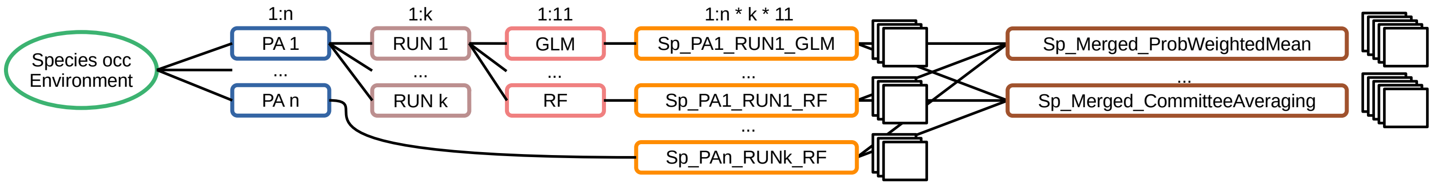
3.b Exploring ensemble models





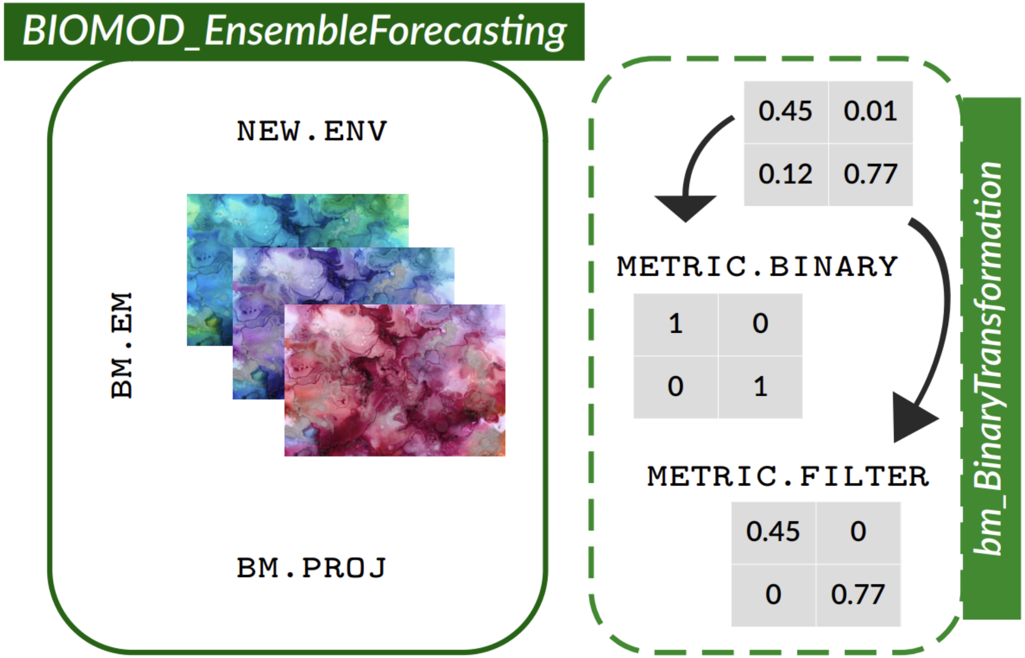
4.b Projecting ensemble models

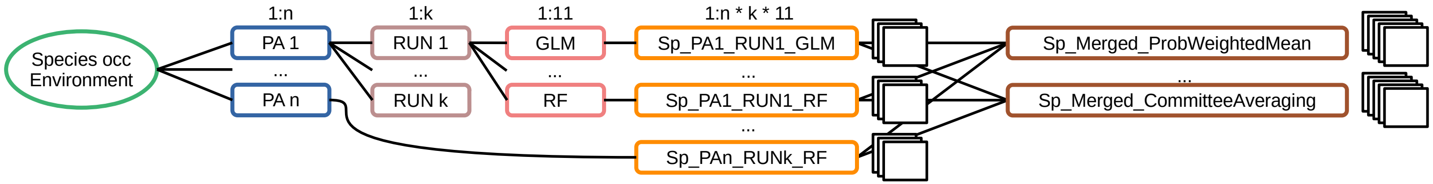




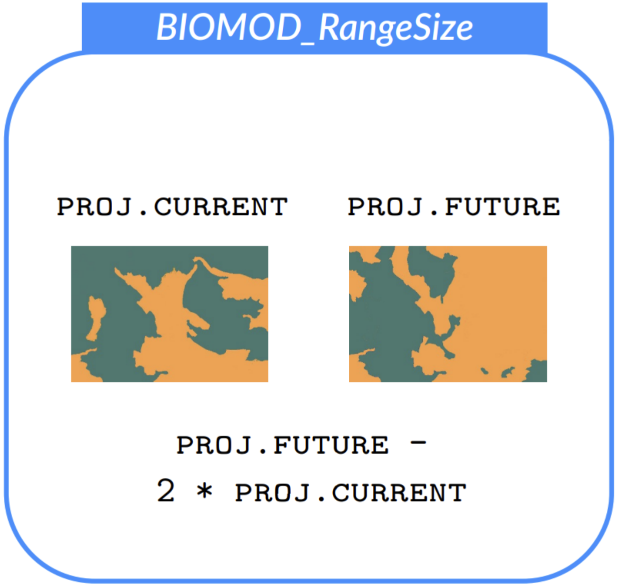
4.b Projecting ensemble models

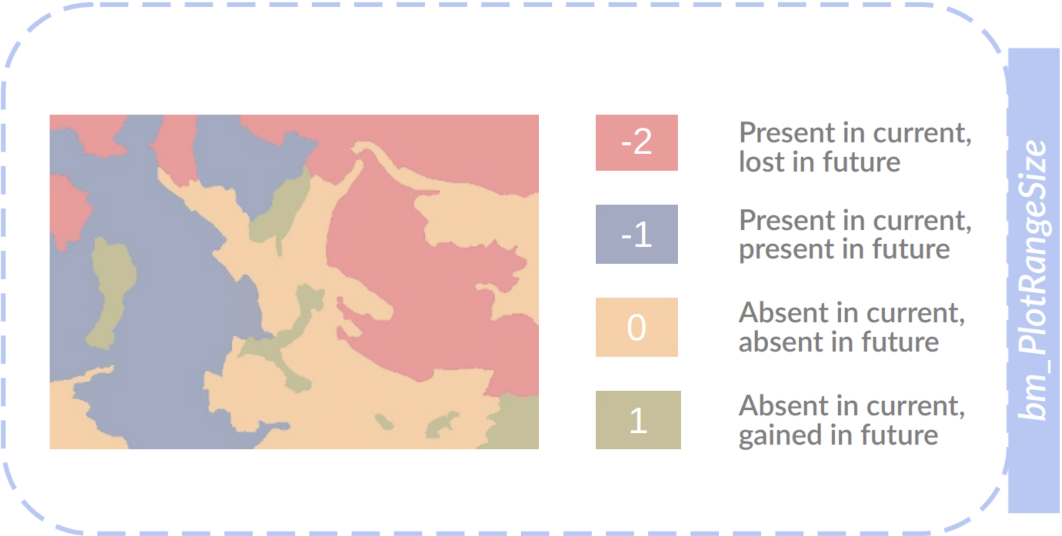
- transformation associated to one evaluation metric (*one map created for each metric selected*)
- use the threshold maximising the chosen metric

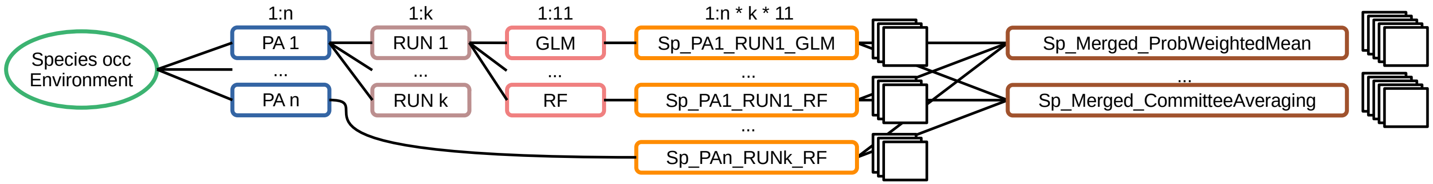




5. Species range change

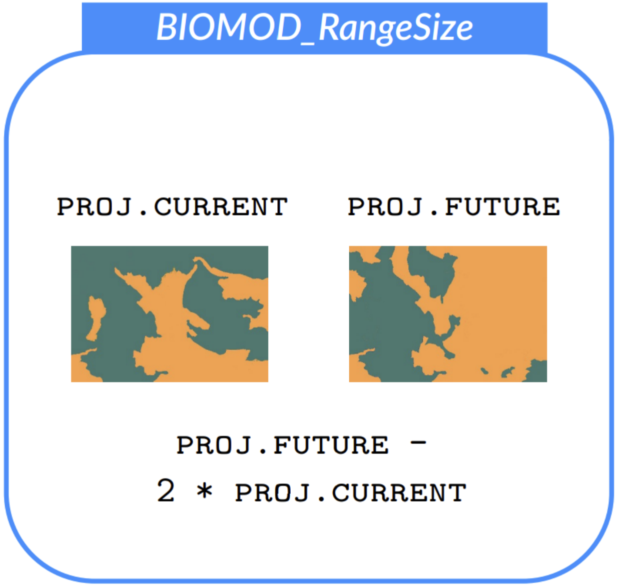


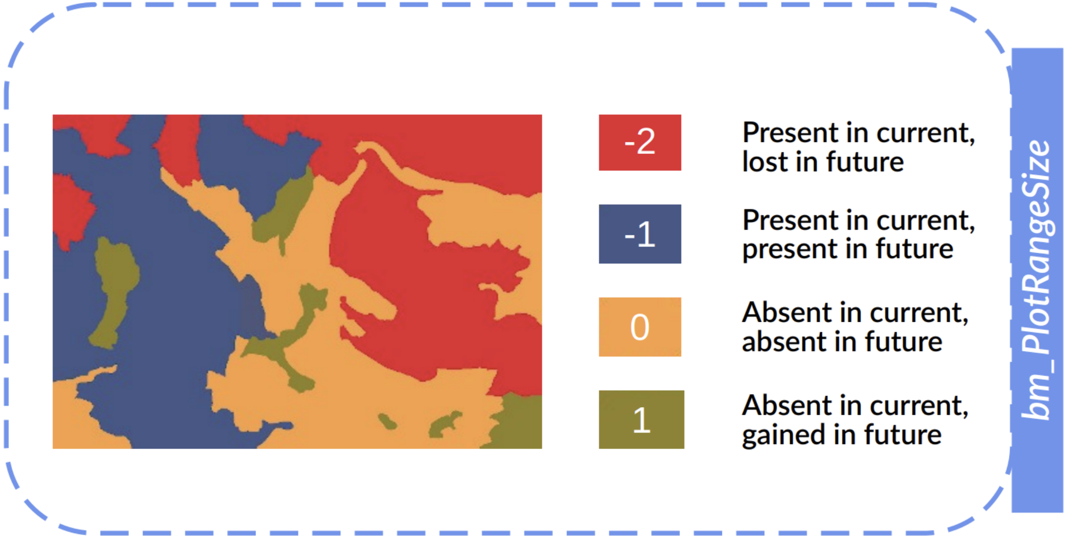


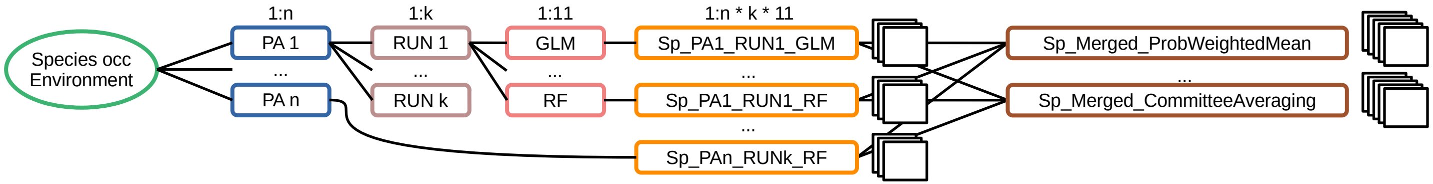


5. Species range change

- 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

BIOMOD_RangeSize

PROJ.CURRENT

PROJ.FUTURE

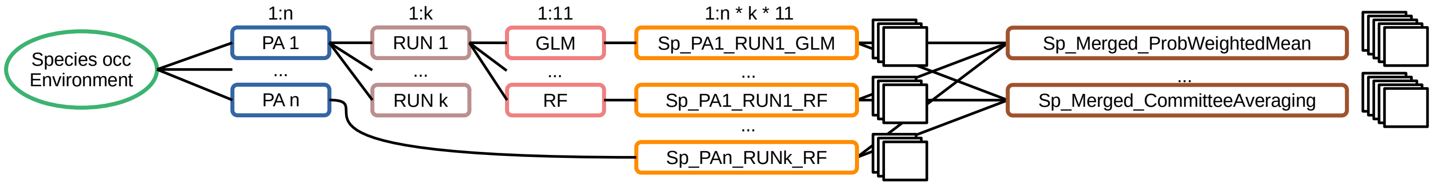
PROJ.FUTURE - 2 * PROJ.CURRENT

-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

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/