

# Optimisation\_RF&XGboost

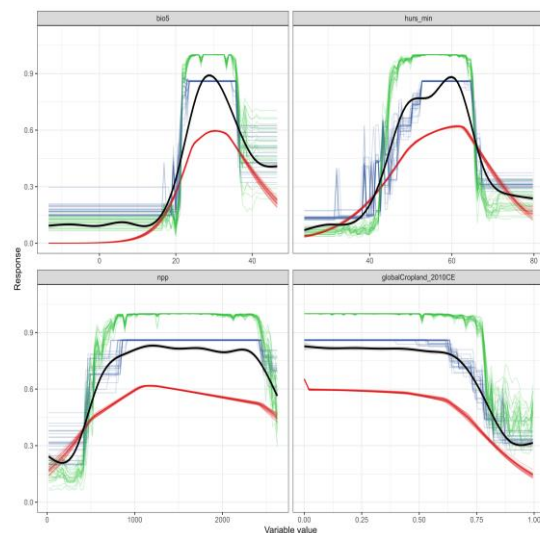
Eléna Manfrini

2025-01-24

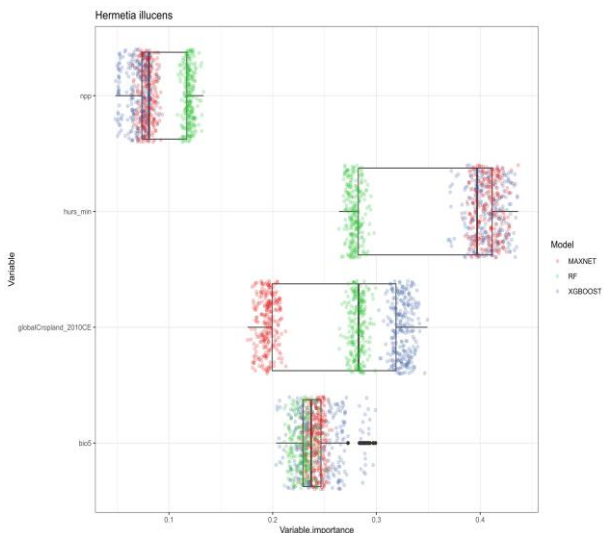
Ce document a pour objectif d'examiner l'impact de l'optimisation des paramètres des modèles XGBoost et Random Forest (RF) sur la performance de nos modèles en réponse aux variables étudiées.

## Paramétrisation par défaut de biomod 2

```
===== BIOMOD.models.options =====  
  
> RF options (datatype: binary , package: randomForest , function: randomForest ) :  
  ( dataset_allData_allRun )  
  - mtry = 1  
  - type = "classification"  
  
> XGBOOST options (datatype: binary , package: xgboost , function: xgboost ) :  
  ( dataset_allData_allRun )  
  - data = NULL  
  - label = NULL  
  - missing = NA  
  - weight = NULL  
  - params = list()  
  - nrounds = 4  
  - verbose = 1  
  - print_every_n = 1  
  - early_stopping_rounds = NULL  
  - maximize = NULL  
  - save_period = NULL  
  - save_name = "xgboost.model"  
  - xgb_model = NULL  
  - callbacks = list()  
  
> MAXNET options (datatype: binary , package: maxnet , function: maxnet ) :  
  ( dataset_allData_allRun )  
  - p =  
  - data =  
  - regmult = 1  
  - regfun = maxnet.default.regularization  
  - addsamplestobackground = T
```



Courbes de réponse



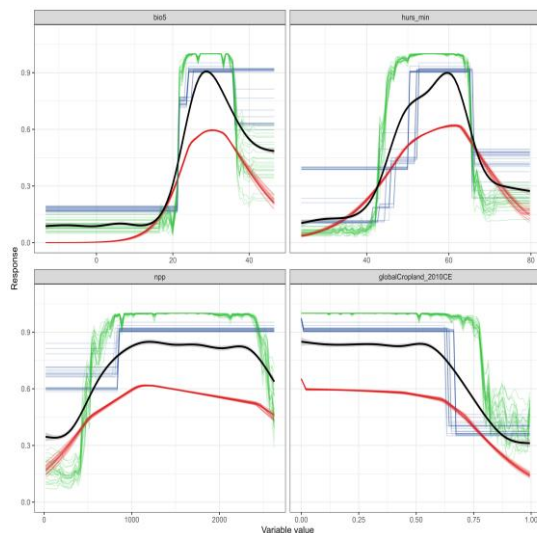
Importance des variables

# Optimisation des paramètres XGBOOST & RF

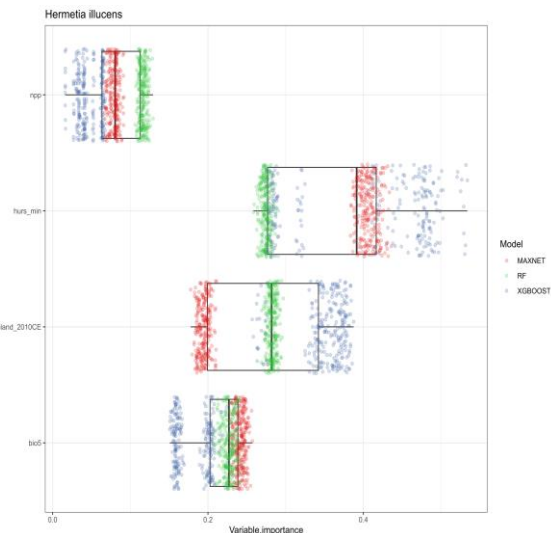
Script XGBoost\_param : - nrounds = 500 - max\_depth = 4 - eta = 0.1 - gamma = 1

Script RF\_param : - ntrees = 1000 - mtry = 4

```
----- BIOMOD.models.options -----  
  
> RF options (datatype: binary , package: randomForest , function: randomForest ) :  
  ( dataset _allData_allRun )  
  - mtry = 2 (default: 1 )  
  - type = "classification"  
  - ntree = 500 (default: NULL )  
  - strata = 0 1 Levels: 0 1 (default: NULL )  
  - nodesize = 5 (default: NULL )  
  - ntrees = 1000 (default: NULL )  
  
> XGBOOST options (datatype: binary , package: xgboost , function: xgboost ) :  
  ( dataset _allData_allRun )  
  - missing = NA  
  - params = list()  
  - nrounds = 500 (default: 4 )  
  - verbose = 1  
  - print_every_n = 1  
  - save_name = "xgboost.model"  
  - callbacks = list()  
  - max.depth = 4 (default: NULL )  
  - eta = 1 (default: NULL )  
  - nthread = 2 (default: NULL )  
  - objective = "binary:logistic" (default: NULL )  
  
> MAXNET options (datatype: binary , package: maxnet , function: maxnet ) :  
  ( dataset _allData_allRun )  
  - p =  
  - data =  
  - regmult = 1  
  - regfun = maxnet.default.regularization  
  - addsamplestobackground = T  
-----
```



Courbe de réponse



Importance des variables

Les courbes de réponse des modèles, en particulier celles de XGBoost, semblent mieux converger après optimisation des paramètres. J'ai également effectué un test dans le script XGBoost\_param pour vérifier l'éventualité d'un surapprentissage, et les résultats montrent qu'il n'y en a pas. Nous pouvons donc conserver ces paramètres pour nos modèles finaux.

En revanche, qu'il y ait une optimisation des paramètres ou pas, l'importance des variables ne varie pas.