Disciplina IBE 875 - Modelagem de Distribuição de Espécies

Professores: Rodrigo Tardin, Maria Lucia Lorini, Mariana Mira Vasconcellos

Roteiro – Script 7 – Avaliação dos modelos, consenso e incerteza

Nesse ponto da nossa disciplina, nós já modelamos a distribuição da Araponga usando diversos algoritmos, avaliamos a importância de cada variável preditora e projetamos as predições do modelo no presente e em cenários futuros de mudanças climáticas (2060-2080). Entretanto, uma etapa crucial é avaliar quão bom são nossos modelos e qual o grau de incerteza existente em nossas projeções. Além disso, nesse exercício iremos construir um modelo de consenso, onde iremos combinar todas as predições dos modelos que tiveram bom desempenho em apenas um.

Funções usadas

- get_evaluations

- BIOMOD_EnsembleModeling

- BIOMOD_EnsembleForecasting

Diversas métricas são usadas para avaliar o desempenho dos modelos. Neste exercício iremos focar em duas: ROC e TSS.

- Avaliando o desempenho dos modelos 1 (GLM, GAM e SRE) e 2 (GBM, RF e MaxEnt).

```
# Calculando o ROC e TSS de cada modelo

evalprocnias1 = get_evaluations(procnias1model)

evalprocnias2 = get_evaluations(procnias2model)
```

Nesse momento, criamos um objeto que contém os valores de desempenho dos modelos (Testing.data), o valor de corte para a binarização das projeções (Cutoff) (a ser discutida na etapa final deste exercício) e os valores de sensibilidade (Sensitivity) e especificidade (Specificity). Abaixo vamos rodar um sumário dessas métricas para cada um dos modelos

que rodamos. Você verá que na sua tela de console, cada uma dessas métricas irá ser apresentada para cada um dos 36 modelos.

```
#Testing.data = valor de desempenho, podendo ser ROC, TSS ou Kappa

#Cutoff = Valor de corte para binarizar os modelos

#Sensitivity = % de presenças corretamente preditas

#Especificidade = % de ausências corretamente preditas


# Sumário das avaliações do desempenho de cada modelo (rodada + set PA + algoritmo )

evalprocnias1

evalprocnias2
```

Para facilitar, vamos salvar os valores de desempenho em uma tabela .csv.

```
#Salvando os valores de desempenho (TSS, ROC, Sensibilidade, Especificidade) em uma tabela .csv

write.table(evalprocnias1,paste0("./Outputs/", "_", "evalprocnias1.csv"))

write.table(evalprocnias1,paste0("./Outputs/", "_", "evalprocnias2.csv"))
```

Como em geral rodamos dezenas ou centenas de modelos, uma alternativa boa para entendermos melhor como foi o desempenho dos modelos é gerar um gráfico que nos permite visualizar como foi o desempenho dos diferentes algoritmos.

Ao final da parte do script abaixo, poderemos ver um conjunto de gráficos com os valores de TSS (eixo X) e ROC (eixo Y) dos modelos em função do algoritmo utilizado, do conjunto de pseudoausência e das rodadas. Os gráficos serão gerados através da função ‘grid.arrange’.

- Plotando os valores de desempenho dos modelos gerados

```
#Modelos GLM, GAM, SRE

gg1_procnias1 =  models_scores_graph(procnias1model, metrics = c("TSS", "ROC"), by = "models", plot = FALSE)

gg2_procnias1 =  models_scores_graph(procnias1model, metrics = c("TSS", "ROC"), by = "data_set", plot = FALSE)

gg3_procnias1 =  models_scores_graph(procnias1model, metrics = c("TSS", "ROC"), by = "cv_run", plot = FALSE)
```

```
grid.arrange(gg1_procnias1, gg2_procnias1, gg3_procnias1)


#Modelos GBM, RF, MAXENT

gg1_procnias2 =  models_scores_graph(procnias2model, metrics = c("TSS", "ROC"), by = "models", plot = FALSE)

gg2_procnias2 =  models_scores_graph(procnias2model, metrics = c("TSS", "ROC"), by = "data_set", plot = FALSE)

gg3_procnias2 =  models_scores_graph(procnias2model, metrics = c("TSS", "ROC"), by = "cv_run", plot = FALSE)


grid.arrange(gg1_procnias2, gg2_procnias2, gg3_procnias2)
```

Com o uso da função acima, conseguimos ver que o algoritmo 'SRE' e o 'MaxEnt' foram os modelos que tiveram o pior desempenho, observados pelos menores valores de TSS e ROC, enquanto Random Forest e GBM o melhor desempenho, observado pelos maiores valores de TSS e ROC.

A partir de agora, vamos construir o modelo de consenso. Os modelos de consenso são boas estratégias porque como vimos as predições dos modelos tendem a diferir muito entre si. Para criar o modelo de consenso precisamos estabelecer um valor de corte para selecionar apenas os modelos que tiveram bom desempenho a partir do argumento 'eval.metric'. Alguns argumentos são importantes, como os modelos que rodamos (modeling.output), a maneira que os modelos vão ser combinados para produzir o consenso (em.by), a métrica de desempenho, o valor de corte e os tipos de modelos de consenso que queremos rodar.

- Construindo o modelo de consenso

```
#Modelo de consenso para os modelos GLM, GAM, SRE que tiveram um desempenho maior do que TSS > 0.7

Ensemble1procnias <- BIOMOD_EnsembleModeling(
  modeling.output = procnias1model, # modelos individuais
  chosen.models = "all" ,
  em.by= "all" , #tipo de combinação para produzir o modelo de consenso (PA + Rodada + Algoritmo)
  eval.metric = c("TSS"), #metrica de desempenho usada para fazer o consenso
  eval.metric.quality.threshold = c(0.7), #valor de corte para os modelos usados para o consenso
  prob.mean = T, #consenso com a média das predições entre os modelos individuais
```

```
  prob.cv = T, #consenso com o coeficiente de variação das predições entre os modelos individuais
  prob.ci = F, #consenso com o intervalo de confiança das predições entre os modelos individuais
  prob.ci.alpha = 0.05,
  prob.median = F, #consenso com a mediana das predições entre os modelos individuais
    committee.averaging = T, #consenso com o comitê de acordo/desacordo das predições entre os modelos individuais
  prob.mean.weight = T, #consenso com a média ponderada das predições entre os modelos individuais
  prob.mean.weight.decay = "proportional" #maneira de calcular a ponderação das médias )

#Modelo de consenso para os modelos GBM, RF, MAXENT que tiveram um desempenho maior do que TSS > 0.7
Ensemble2procnias <- BIOMOD_EnsembleModeling(
  modeling.output = procnias2model, # modelos individuais
  chosen.models = "all" ,
  em.by= "all" , #tipo de combinação para produzir o modelo de consenso (PA + Rodada + Algoritmo)
  eval.metric = c("TSS"), #metrica de desempenho usada para fazer o consenso
  eval.metric.quality.threshold = c(0.7), #valor de corte para os modelos usados para o consenso
  prob.mean = T, #consenso com a média das predições entre os modelos individuais
  prob.cv = T, #consenso com o coeficiente de variação das predições entre os modelos individuais
  prob.ci = F, #consenso com o intervalo de confiança das predições entre os modelos individuais
  prob.ci.alpha = 0.05,
  prob.median = F, #consenso com a mediana das predições entre os modelos individuais
    committee.averaging = T, #consenso com o comitê de acordo/desacordo das predições entre os modelos individuais
  prob.mean.weight = T, #consenso com a média ponderada das predições entre os modelos individuais
  prob.mean.weight.decay = "proportional" #maneira de calcular a ponderação das médias )
```

Quando você rodar os códigos acima você verá na sua tela de console que os modelos de consenso com os parâmetros que você indicou estarão sendo rodados. Os modelos que aparecerão na tela de console são aqueles que foram mantidos e combinados para construir o modelo de consenso. No final da rodada, você verá o sistema de ponderação pelos pesos que serão calculados para se gerar um dos tipos mais robustos de modelo de consenso, o modelo da média ponderada. Observe as informações referentes ao original model scores e final models weights. Ali estão disponíveis as informações dos valores de TSS de cada modelo (original model scores) e o peso que está sendo dado a cada modelo com base nos valores de TSS para construir as predições

de média ponderada (final models weights). Quanto melhor o valor do TSS, maior o peso que será dado a esse modelo individual para o modelo de consenso final. Cada valor desse representa um modelo que foi retido e usado para combinar no modelo de consenso. Se você contar a quantidade de 'scores' você saberá quantos modelos foram retidos para construir o modelo de consenso.

- Avaliação de desempenho dos modelos de consenso, incluindo as métricas relacionadas à incerteza

```
evalprocnias1ensemble = get_evaluations(Ensemble1procnias)

evalprocnias2ensemble = get_evaluations(Ensemble2procnias)


# Sumário das avaliações do desempenho dos modelos de consenso (rodada + set PA + algoritmo )

evalprocnias1ensemble


evalprocnias2ensemble
```

Usando a sensibilidade (proporção de presenças corretamente preditas) e a especificidade (proporção de ausências corretamente preditas) como medidas de incerteza dos modelos pode-se observar diferenças entre os diferentes algoritmos usados para o consenso? Algum foi menos sensível do que o outro, ou seja, teve menor capacidade de prever as presenças corretamente?

- Salvando os valores de desempenho (TSS, ROC, Sensibilidade, Especificidade) em uma tabela .csv

```
write.table(evalprocnias1ensemble,paste0("./Outputs/", "_", "evalprocnias1ensemble.csv"))


write.table(evalprocnias2ensemble,paste0("./Outputs/", "_", "evalprocnias2ensemble.csv"))
```

Agora que já criamos um modelo de consenso, unindo todos os algoritmos que tiveram bom desempenho, um passo importante é projetar no espaço geográfico as predições dos modelos de consenso. Iremos usar a função BIOMOD_EnsembleForecasting, onde temos que indicar tanto o objeto das projeções dos modelos individuais (projection.output) que fizemos no roteiro 6, como o objeto da modelagem dos modelos de consenso criadas nos passos anteriores (EM.output). Quando você rodar esses códigos você verá na sua tela

de console que os modelos de consenso com os parâmetros que você indicou na etapa BIOMOD_EnsembleModeling estarão sendo projetados no espaço geográfico.

- Projetando no espaço geográfico as predições dos modelos de consenso

```
proj1ensemble=BIOMOD_EnsembleForecasting( projection.output = projec_procnias1, #objeto com as predições dos modelos individuais

                        EM.output = Ensemble1procnias, #objeto com a modelagem consenso

                         binary.meth = "TSS", #método de avaliação de desempenho para transformar as predições contínuas de adequabilidade em predições binárias (presença/ausência)

                        output.format = ".img" #formato par salvar os mapas

                        )


proj2ensemble=BIOMOD_EnsembleForecasting( projection.output = projec_procnias2, #objeto com as predições dos modelos individuais

                        EM.output = Ensemble2procnias, #objeto com a modelagem consenso

                         binary.meth = "TSS", #método de avaliação de desempenho para gerar predições binárias (presença/ausência)

                        output.format = ".img" #formato par salvar os mapas

)
```

Agora vamos plotar os mapas com os modelos de consenso, ou seja, das predições dos melhores modelos combinados em uma só. Como determinamos nos passos anteriores (função BIOMOD_EnsembleModeling), geramos 4 modelos de consenso:

1. Modelo de consenso onde as predições de cada pixel se referem a **média** da adequabilidade entre os modelos que foram combinados para gerar o consenso
2. Modelo de consenso onde as predições de cada pixel se referem a **média ponderada** da adequabilidade entre os modelos que foram combinados para gerar o consenso
3. Modelo de consenso onde as predições de cada pixel se referem ao **coeficiente de variação (CV)** da adequabilidade entre os modelos que foram combinados para gerar o consenso
4. Modelo de consenso onde as predições de cada pixel se referem a um comitê de votos sobre o quanto os modelos concordam ou discordam **(committee averaging)** sobre a adequabilidade entre os modelos que foram combinados para gerar o consenso.

- Plotando as projeções dos modelos de consenso

```
#Média

plot(proj1ensemble, str.grep = "procnias_EMmeanByTSS_mergedAlgo_mergedRun_mergedData")


#Média ponderada

plot(proj1ensemble, str.grep = "procnias_EMwmeanByTSS_mergedAlgo_mergedRun_mergedData")


#Mapas com métricas para avaliar as áreas com maiores ou menores incertezas


#Coeficiente de variação

plot(proj1ensemble, str.grep = "procnias_EMcvByTSS_mergedAlgo_mergedRun_mergedData")


#Committee averaging

plot(proj1ensemble, str.grep = "procnias_EMcaByTSS_mergedAlgo_mergedRun_mergedData")
```

Um procedimento interessante ao gerar modelos de consenso, com forte aplicação para a conservação, é produzir um mapa de presença/ausência ao invés de probabilidades contínuas. Nesse caso, chamamos esse processo de binarização das predições. Como vimos na aula teórica, existem diversas métricas para binarizar, ou seja, transformar probabilidades contínuas em presença/ausência. Para isso tem que ser escolhido um valor de corte (threshold). No BIOMOD 2, o procedimento padrão (‘default’) para calcular o valor de corte é a partir da maximização do TSS, que é equivalente a maximizar a soma da sensibilidade (% de presenças corretamente preditas) e especificidade (% de ausências corretamente preditas). Esse mapa binarizado foi gerado no procedimento anterior, onde indicamos que o TSS seria usado como métrica.

- Primeiro temos que carregar o arquivo que foi salvo no processo de modelagem e projeção

```
procnias1ensemble_bin=raster("./Procnias/proj_GLM_GAM_SRE/proj_GLM_GAM_SRE_procnias_ensemble_TSSbin.img")


#Agora o plot
plot(procnias1ensemble_bin)


#Carregando o mapa binarizado dos modelos com os algoritmos RF, GBM, MAXENT
procnias2ensemble_bin=raster("./Procnias/proj_GBM_RF_MAX/proj_GBM_RF_MAX_procnias_ensemble_TSSbin.img")
```

#Agora o plot

```
plot(procnias2ensemble_bin)
```

Agora que temos o modelo de consenso, usando diferentes métricas no presente (média, media ponderada, coeficiente de variação, comitee averaging, binário), vamos projetá-los em condições futuras. Esse passo só é possível porque seguimos as etapas do roteiro 6 onde preparamos as camadas do futuro e criamos o objeto de projeção para o futuro a partir da modelagem do presente. Vamos fazer um processo muito similar, só que ao invés de usar os modelos individuais e projetá-los no futuro, vamos usar o modelo de consenso realizado acima. Para isso vamos usar a função BIOMOD_EnsembleForecasting

- Projetando no espaço geográfico os modelos de consenso do presente nas condições climáticas futuras

```
proj1ensemble_futuro=BIOMOD_EnsembleForecasting( projection.output = projec_procnias1_futuro, #output gerado da projeção dos modelos individuais no futuro
                                                 EM.output = Ensemble1procnias, #output com o consenso de modelos gerados para o persente
                                                 binary.meth = "TSS", output.format = ".img")

proj2ensemble_futuro=BIOMOD_EnsembleForecasting( projection.output = projec_procnias2_futuro, #output gerado da projeção dos modelos individuais no futuro
                                                 EM.output = Ensemble2procnias, #output com o consenso de modelos gerados para o persente
                                                 binary.meth = "TSS", output.format = ".img")
```

- Plotando as projeções dos modelos de consenso futuros (melhorar com os códigos da Mari

```
#Média

plot(proj1ensemble_futuro, str.grep = "procnias_EMmeanByTSS_mergedAlgo_mergedRun_mergedData")


#Média ponderada

plot(proj1ensemble_futuro, str.grep = "procnias_EMwmeanByTSS_mergedAlgo_mergedRun_mergedData")
```

#Mapas com métricas para avaliar as áreas com maiores ou menores incertezas (melhorar com os cÃ³digos da Mari)

#Coeficiente de variação

plot(proj1ensemble_futuro, str.grep = "procnias_EMcvByTSS_mergedAlgo_mergedRun_mergedData")

#Committee averaging

plot(proj1ensemble_futuro, str.grep = "procnias_EMcaByTSS_mergedAlgo_mergedRun_mergedData")

Assim como fizemos para os modelos de consenso para o tempo presente, agora também podemos plotar o mapa binarizado em cenários futuros.

- Primeiro temos que carregar o arquivo que foi salvo no processo de modelagem e projeção

#Modelos GLM_GAM_SRE

procnias1ensemble_binfuturo=raster("./Procnias/proj_Modelos Futuro_SRE_GLM_GAM/proj_Modelos Futuro_SRE_GLM_GAM_procnias_ensemble_TSSbin.img")

#Agora o plot

plot(procnias1ensemble_binfuturo)

#Modelos RF_GBM_MAXENT

procnias2ensemble_binfuturo=raster("./Procnias/proj_Modelos Futuro_RF_BRT_MAXENT/proj_Modelos Futuro_RF_BRT_MAXENT_procnias_ensemble_TSSbin.img")

Fim do script 7