Fitting Random Forests for WhoseEgg Shiny App

Katherine Goode

Last Updated: January 28, 2021

This document contains code that fits the three random forest models that will be used in the app: models with invasive carp species grouped into one class and all other species classified into species, genus, and family. The data used to train the model is that used in Goode et al. (2021) to train the augmented models, and the same seed (808) is used, so the models should agree.

Setup

Load packages:

```
library(dplyr)
library(randomForest)
library(purrr)
```

Make a list of the response variables:

```
vars_resp = c(
  "Family_ACGC",
  "Genus_ACGC",
  "Common_Name_ACGC"
)
```

Make a vector of the predictor variables:

```
vars_pred = c(
  "Month",
  "Julian_Day",
  "Temperature",
  "Conductivity",
  "Larval_Length",
  "Membrane_Ave",
  "Membrane_SD",
  "Membrane_CV",
  "Yolk_to_Membrane_Ratio",
  "Yolk_Ave",
  "Yolk_SD",
  "Yolk_CV",
  "Egg_Stage",
  "Compact_Diffuse",
  "Pigment",
  "Sticky_Debris",
  "Deflated"
)
```

Egg Data

Load the data from Goode et al. (2021) to be prepared for the app:

```
eggdata_for_app <-
  # Access the data
  read.csv(
    paste0(
      "https://raw.githubusercontent.com/goodekat/",
      "carp-egg-rf-validation/master/results/eggdata141516.csv"
    )
  ) %>%
  # Convert necessary variables to factors
  mutate_at(
    .vars = c(
      "Egg_Stage",
      "Compact_Diffuse",
      "Pigment",
      "Sticky_Debris",
      "Deflated",
      all_of(vars_resp)
    ),
    .funs = factor
  ) %>%
  # Change the level of ACGC to Invasive Carp for easier terminology in the app
  mutate(
    Family_ACGC = forcats::fct_recode(Family_ACGC, "Invasive Carp" = "ACGC"),
    Genus_ACGC = forcats::fct_recode(Genus_ACGC, "Invasive Carp" = "ACGC"),
    Common_Name_ACGC = forcats::fct_recode(Common_Name_ACGC, "Invasive Carp" = "ACGC")
  ) %>%
  # Make sure the Invasive Carp level is still the first factor level (like ACGC was)
  # Otherwise, the random forest results will change slightly
  mutate(
    Family_ACGC = forcats::fct_relevel(Family_ACGC, "Invasive Carp"),
    Genus_ACGC = forcats::fct_relevel(Genus_ACGC, "Invasive Carp"),
    Common_Name_ACGC = forcats::fct_relevel(Common_Name_ACGC, "Invasive Carp")
  )
```

Save the prepared data (remember that if loaded again to train random forests, the order of the response variable factor levels will have to be the same to produce the exact same random forest):

```
write.csv(
  x = eggdata_for_app,
  file = "../data/eggdata_for_app.csv",
  row.names = FALSE
)
```

Random Forests

Function for fitting a random forest model given a response variable, predictor variables, and a dataset (uses the same seed to fit the random forests as Camacho et al. (2019) and Goode et al. (2021)):

```
fit_rf <- function(resp, preds, data) {

  # Fit the random forest
  set.seed(808)
  rf <- randomForest(
    data %>% pull(resp) ~ .,
    data = data %>% select(all_of(preds)),
    importance = T,
    ntree = 1000
  )

  # Put model in a named list
  rf_list = list(rf)
  names(rf_list) = resp

  # Return the named list
  return(rf_list)

}
```

Fit the random forest models:

```
rfs_for_app <-
  map(
    .x = vars_resp,
    .f = fit_rf,
    preds = vars_pred,
    data = eggdata_for_app
  ) %>%
  flatten()
```

Check to make sure the random forests agree (note that these random forests are available on GitHub: https://github.com/goodekat/carp-egg-rf-validation/blob/master/results/rfs141516.rds)

```
rfs141516 <- readRDS("../../../rf-validation/results/rfs141516.rds")
c(
  identical(rfs141516$Family_ACGC$forest, rfs_for_app$Family_ACGC$forest),
  identical(rfs141516$Genus_ACGC$forest, rfs_for_app$Genus_ACGC$forest),
  identical(rfs141516$Common_Name_ACGC$forest, rfs_for_app$Common_Name_ACGC$forest)
)
```

```
## [1] TRUE TRUE TRUE
```

Save the random forests:

```
saveRDS(
  object = rfs_for_app,
  file = "../data/rfs_for_app.rds"
)
```