Appendix 3. Uncertainty Maps

1. Introduction

This appendix is a quick tutorial on how to generate the uncertainty maps related to the carbon stock predictions using the Random Forest (RF) and the Generalized Additive Model (GAM). For that, we rely on the extended Pearson-Tukey approximation [1] to estimate the standard deviation of predictions for each grid cell, which only requires the 5%, 50%, and 95% quantiles of predictions. For the GAM approach, we rely on the `gratia` R package [2], and for the RF approach we rely on the `ranger` R package [3]. After getting the quantiles predictions, we use the equation below to calculate the standard deviation for each grid-cell.

$$\hat{\mu}_i = 0.63\hat{q}_{i,0.5} + 0.185\left(\hat{q}_{i,0.05} + \hat{q}_{i,0.95}\right)$$

$$\hat{\sigma}_i^2 = 0.63\left(\hat{q}_{i,0.5} - \hat{\mu}_i\right)^2 + 0.185\left[\left(\hat{q}_{i,0.05} - \hat{\mu}_i\right)^2 + \left(\hat{q}_{i,0.95} - \hat{\mu}_i\right)^2\right]$$

2. Simulated Data

For this tutorial, we use a small set of simulated data that can be accessed using the code below.

```
#install.packages('readr')
library (readr)

#Loading Simulated Data
urlfile="https://raw.githubusercontent.com/LeoHaneda/Haneda_etal_2024_Simple_methods/main/sim_data"
dat <- data.frame(read_csv(url(urlfile)))
head(dat)

    CH      Lat       Lon    CSD
1 35.0 25.26136 -18.36433 671.42
2 19.5 25.27866 -18.36527  67.45
3 12.0 25.28023 -18.37659  34.11
4 15.8 25.25413 -18.35270  63.87
5 18.5 25.27237 -18.34829  84.34
6 26.0 25.25571 -18.39325 449.46
```

where:

- CSD: Carbon stock density from field plots [$tCO_2e.ha^{-1}$];
- Lat and Lon: Spatial coordinates of the field plots centroid (Latitude and Longitude);

- CH: Canopy height from large-scale map [m];

3. Fitting Models

After having all the data structured, fitting the models is a straightforward step.

3.1. Generalized Additive Model (GAM)

To fit the GAM and get the predictions quantile, we rely on the `mcgv` R package [4], using the standard formula syntax to inform the response and spline functions with the exploratory variables.

```
#install.packages('dplyr')
#install.packages('mgcv')
library(dplyr)
library(mgcv)

library(mgcv) #Wood & Wood (2015)
GAM_mod <- mgcv::gam(CSD ~ s(CH) + s(Lat, Lon, bs="tp"),
                     data = dat, method = "REML")
```

3.2. Quantile Random Forest (Quantile RF)

To fit the RF model, we rely on the `ranger` R package [3], using the standard formula syntax to inform the response and explanatory variable.

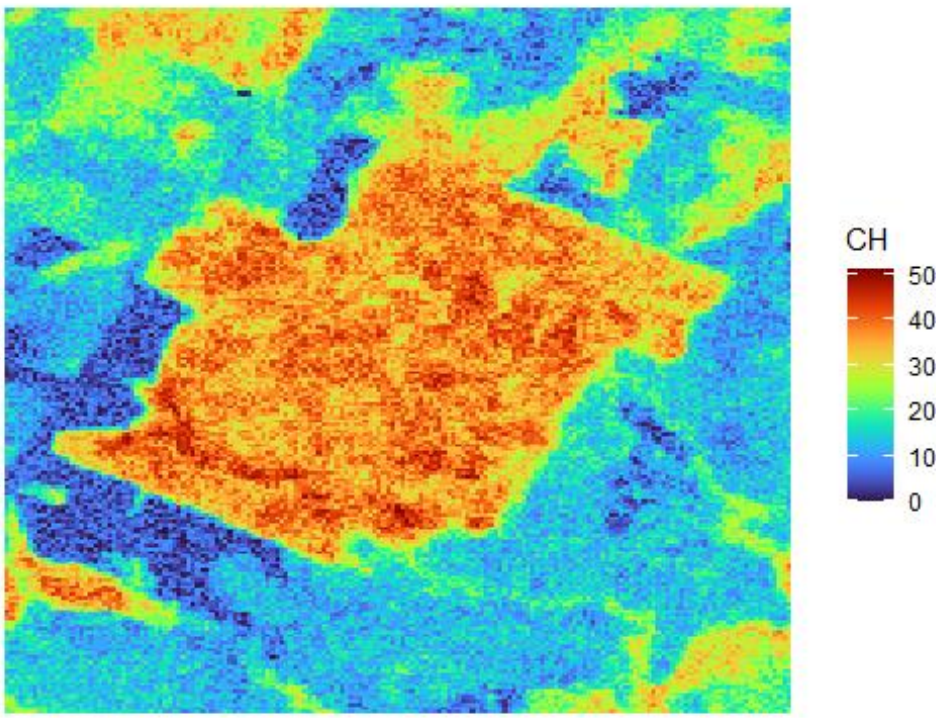
```
#install.packages('ranger')
library(ranger)
RF_mod <- ranger::ranger(CSD ~ Lat + Lon + CH,
                         data = dat, quantreg = T)
```

4. Canopy Height Map for Prediction

After fitting the model, we need to calculate the uncertainty of the predictions and rasterize them to create the uncertainty maps. First, we need to extract the large-scale map information and the spatial coordinates (Latitude and Longitude) for each grid-cell.

```
#install.packages('terra')
#install.packages('ggplot2')
#install.packages('viridis')
library(terra)
library(ggplot2)
library(viridis)

#Load Simulated Canopy Height Map
download.file("https://github.com/LeoHaneda/Haneda_etal_2024_Simple_methods/b
lob/main/sim_rast.zip?raw=TRUE",
  "sim_rast.zip", mode = "wb")
Map <- terra::rast(unzip("./sim_rast.zip"))

#Extract the values and spatial coordinates of each grid-cell from the large-
```

```
scale map
df <- terra::as.data.frame(Map,xy = TRUE, na.rm = FALSE)
names(df) <- c("Lat", "Lon", "CH")

#Plot Canopy Height Map
ggplot(df)+
    geom_tile(aes(x = Lat, y = Lon, fill = CH)) +
    scale_fill_viridis(option = "H") +
    theme_void() +
    theme(legend.position = "right")
```




```
head(df)

        Lat       Lon   CH
1 25.23307 -18.34546 12.2
2 25.23338 -18.34546 13.7
3 25.23370 -18.34546 23.9
4 25.23401 -18.34546 21.8
5 25.23433 -18.34546 28.8
6 25.23464 -18.34546 24.5
```

5. Uncertainty Maps

Then, we make the carbon stock prediction for each grid-cell, calculate the prediction quantiles, and rasterize it using the spatial coordinates to generate the uncertainty map.

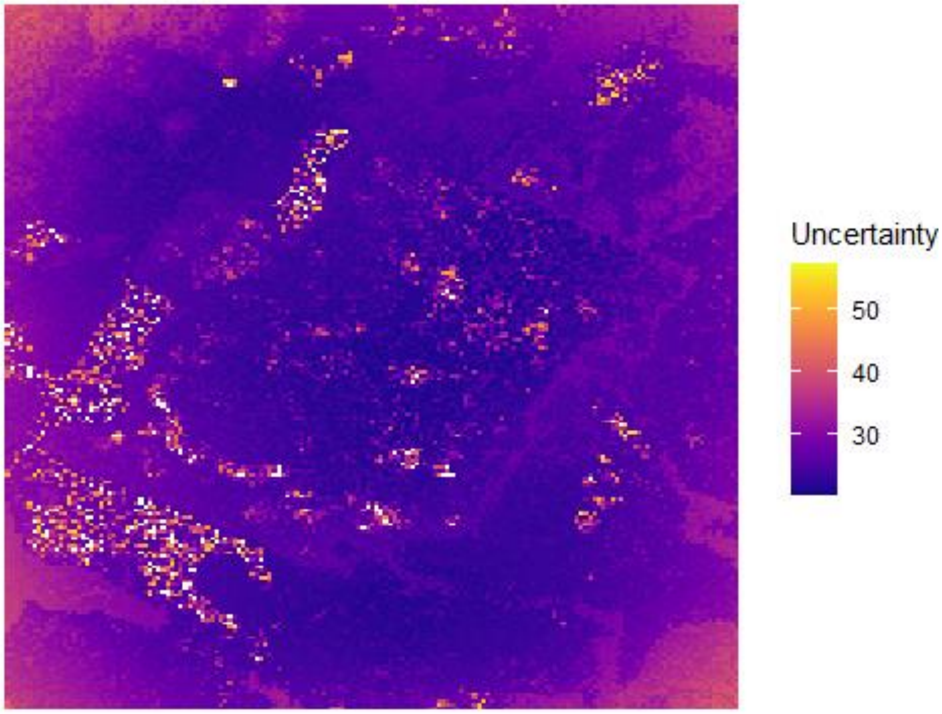
We demonstrate how to generate the uncertainty maps using both approaches: GAM and RF.

5.1. Generalized Additive Model (GAM)

We use the `fitted_samples()` function to get the prediction quantiles and calculate the standard deviation for each grid-cell and rasterize them using their spatial coordinates. To avoid extrapolation, we remove all grid-cell that are outside the range of values used in the training dataset, which are represented on the map as empty pixels.

```
#install.packages('dplyr')
#install.packages('gratia')
#install.packages('ggplot2')
#install.packages("viridis")
library(dplyr)
library(gratia)
library(ggplot2)
library(viridis)

#Removing grid-cells outside the range of the trainnig dataset to avoid
extrapolation
range.lat=range(dat$Lat)
cond.lat=df$Lat < range.lat[1] | df$Lat > range.lat[2]
df$Lat[cond.lat] <- NA

range.lon=range(dat$Lon)
cond.lon=df$Lon < range.lon[1] | df$Lon > range.lon[2]
df$Lon[cond.lon] <- NA

range.ch=range(dat$CH)
cond.ch=df$CH < range.ch[1] | df$CH > range.ch[2]
df$CH[cond.ch] <- NA

df <- na.omit(df)

#Get simulated values from posterior distribution
sim <- as.data.frame(gratia::fitted_samples(GAM_mod, df, n = 1000,
                              method = "gaussian",seed = 123))

# Compute quantiles for each group
quantiles <- tapply(sim[, 4], sim$.row, function(x) quantile(x, probs =
c(0.05, 0.5, 0.95)))

# Convert the list of quantiles to a data frame
quant <- as.data.frame(do.call(rbind, quantiles))
names(quant) <- c("q_0.05", "q_0.5", "q_0.95")
rownames(quant) <- names(quantiles)

#Calculating standard deviation
```

```r
mu <- (0.63*quant[,2]) + (0.185*(quant[,1] + quant[,3]))
var <- (0.63*(quant[,2] - mu)^2) + 0.185*(((quant[,1] - mu)^2) + ((quant[,3]
- mu)^2))
sd <- sqrt(var)

#Matching the grid-cell standard deviation with respective spatial
coordinates
df_pred <- data.frame(X = df$Lat,
                      Y = df$Lon,
                      Uncertainty = sd)
head(df_pred)

         X         Y Uncertainty
1 25.23338 -18.34641    42.80167
2 25.23370 -18.34641    42.44374
3 25.23401 -18.34641    41.77965
4 25.23433 -18.34641    40.03648
5 25.23464 -18.34641    39.21395
6 25.23495 -18.34641    40.63537

#Raterize
Unc_pred_map <- terra::rast(df_pred, type = "xyz")

#Setting Coordinate System
terra::crs(Unc_pred_map) <- terra::crs('+init=EPSG:4326') #WGS 84

#Save raster
writeRaster(Unc_pred_map, "./Uncertainty_map.tif", overwrite = T)

#Plotting Uncertainty Map
ggplot(df_pred)+
    geom_raster(aes(x = X, y = Y, fill = Uncertainty))+
    scale_fill_viridis(option = "C") +
    theme_void() +
    theme(legend.position = "right")
```

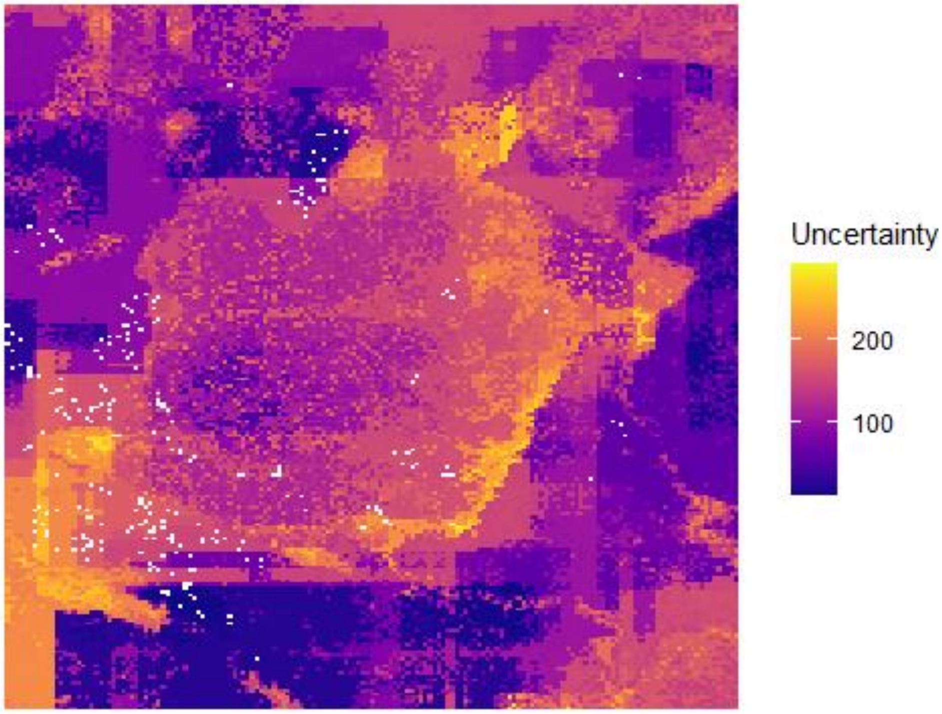


5.2. Quantile Random Forest (Quantile RF)

We use the `predict()` function to get the prediction quantiles and calculate the standard deviation for each grid-cell and rasterize them using their spatial coordinates.

```
#install.packages('ranger')
#install.packages('terra')
#install.packages('ggplot2')
#install.packages("viridis")
library(ranger)
library(terra)
library(ggplot2)
library(viridis)

#Removing grid-cells outside the range of the trainnig dataset to avoid
extrapolation
range.lat=range(dat$Lat)
cond.lat=df$Lat < range.lat[1] | df$Lat > range.lat[2]
df$Lat[cond.lat] <- NA

range.lon=range(dat$Lon)
cond.lon=df$Lon < range.lon[1] | df$Lon > range.lon[2]
df$Lon[cond.lon] <- NA

range.ch=range(dat$CH)
cond.ch=df$CH < range.ch[1] | df$CH > range.ch[2]
```

```r
df$CH[cond.ch] <- NA

df <- na.omit(df)

#Get predictions quantiles
quant <- predict(RF_mod, data = df,
                type='quantiles',
                quantiles=c(0.05,0.5,0.95))$predictions

#Calculating standard deviation
mu <- (0.63*quant[,2]) + (0.185*(quant[,1] + quant[,3]))
var <- (0.63*(quant[,2] - mu)^2) + 0.185*(((quant[,1] - mu)^2) + ((quant[,3]
- mu)^2))
sd <- sqrt(var)

#Matching the grid-cell standard deviation with respective spatial
coordinates
df_pred <- data.frame(X = df$Lat,
                      Y = df$Lon,
                      Uncertainty = sd)
head(df_pred)

         X         Y Uncertainty
1 25.23338 -18.34641    173.0836
2 25.23370 -18.34641    171.8212
3 25.23401 -18.34641    171.7547
4 25.23433 -18.34641    121.4031
5 25.23464 -18.34641    138.2444
6 25.23495 -18.34641    146.4523

#Raterize
Unc_pred_map <- terra::rast(df_pred, type = "xyz")

#Setting Coordinate System
terra::crs(Unc_pred_map) <- terra::crs('+init=EPSG:4326')

#Save raster
writeRaster(Unc_pred_map, "./Uncertainty_map.tif", overwrite = T)

#Plotting Uncertainty Map
ggplot(df_pred)+
    geom_raster(aes(x = X, y = Y, fill = Uncertainty))+
    scale_fill_viridis(option = "C") +
    theme_void() +
    theme(legend.position = "right")
```



Reference

1. Pfeifer, P. E., Bodily, S. E. & Frey, S. C. Pearson-tukey three-point approximations versus monte carlo simulation. *Decision Sciences* **22**, 74–90 (1991).

2. Simpson, G. L. gratia: Graceful 'ggplot'-Based Graphics and Other Functions for GAMs Fitted Using "mgcv." *The R Foundation* (2024) doi:10.32614/cran.package.gratia.

3. Wright, M. N. & Ziegler, A. ranger : A Fast Implementation of Random Forests for High Dimensional Data in*C++* and*R*. *J. Stat. Softw.* **77**, 1–17 (2017).

4. Wood, S. N. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *J. Royal Statistical Soc. B* **73**, 3–36 (2011).