Mikeladze G, Gavashelishvili A, Akobia I, Metreveli V (2020). **Estimation of forest cover change using Sentinel-2 multi-spectral imagery in Georgia (the Caucasus)**
iForest – Biogeosciences and Forestry – doi: 10.3832/ifor3386-013

Supplementary Material

Fig. S1 - The density curves of residuals of the canopy closure (FEYE) fitted to Sentinel reflectance values topographically corrected using four different methods: Sen2cor correction, Cosine Correction, Minnaert Correction, Normalization Method.

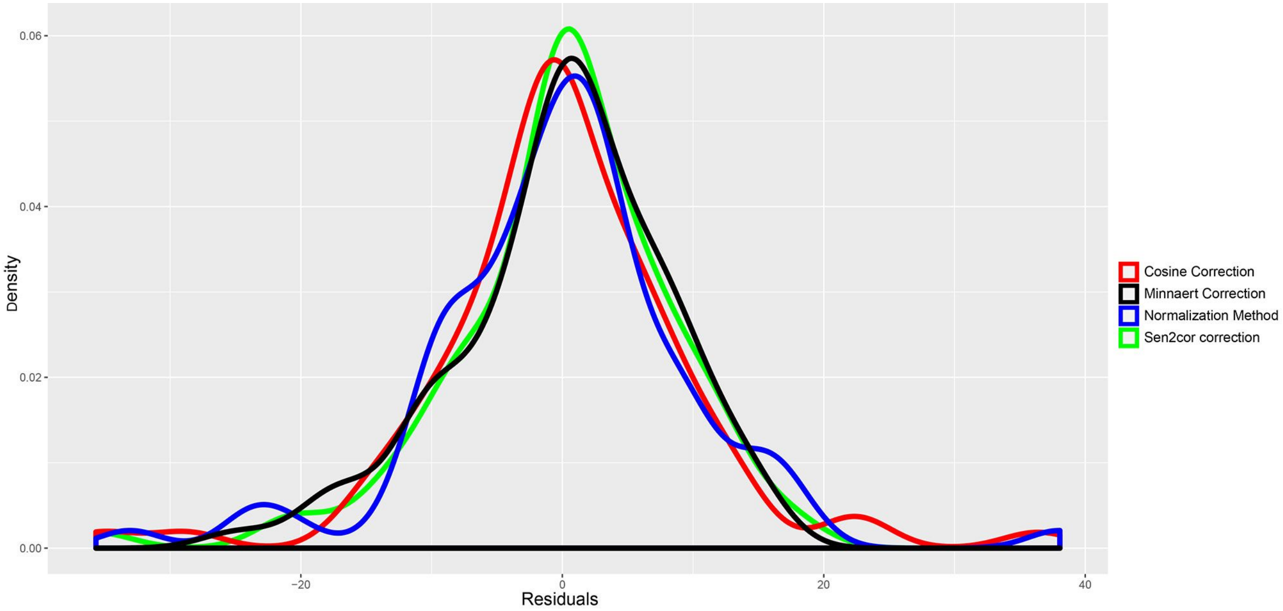


Fig. S2 - The histogram of residuals of the canopy closure (FEYE) fitted to Sentinel reflectance values topographically corrected using Minnaert Correction. The range limits of residuals are 24.894% and 14.952%, while those of the 95% CI range are 16.973% and 14.358%.

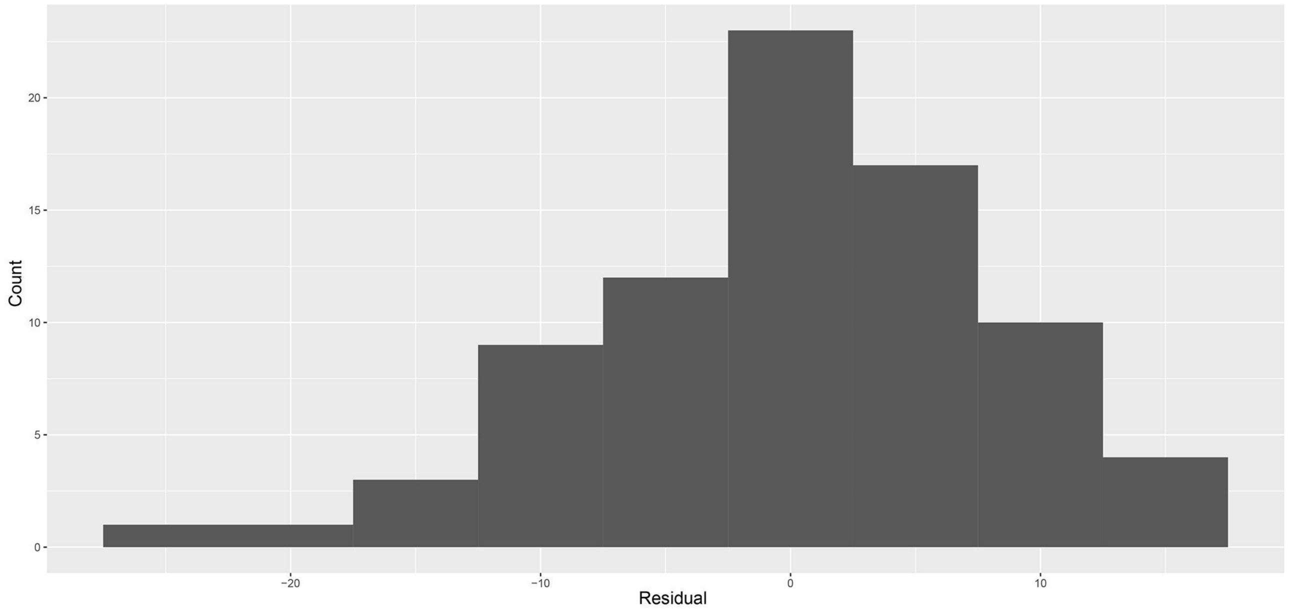


Fig. S3 - Modeled canopy closure (FEYE) in eastern Georgia (at left) and statistically significant differences in the canopy closure in percentage at two sites between 01.07.2018 and 05.08.2016. The map is projected to UTM Zone 38N; WGS: 1984.

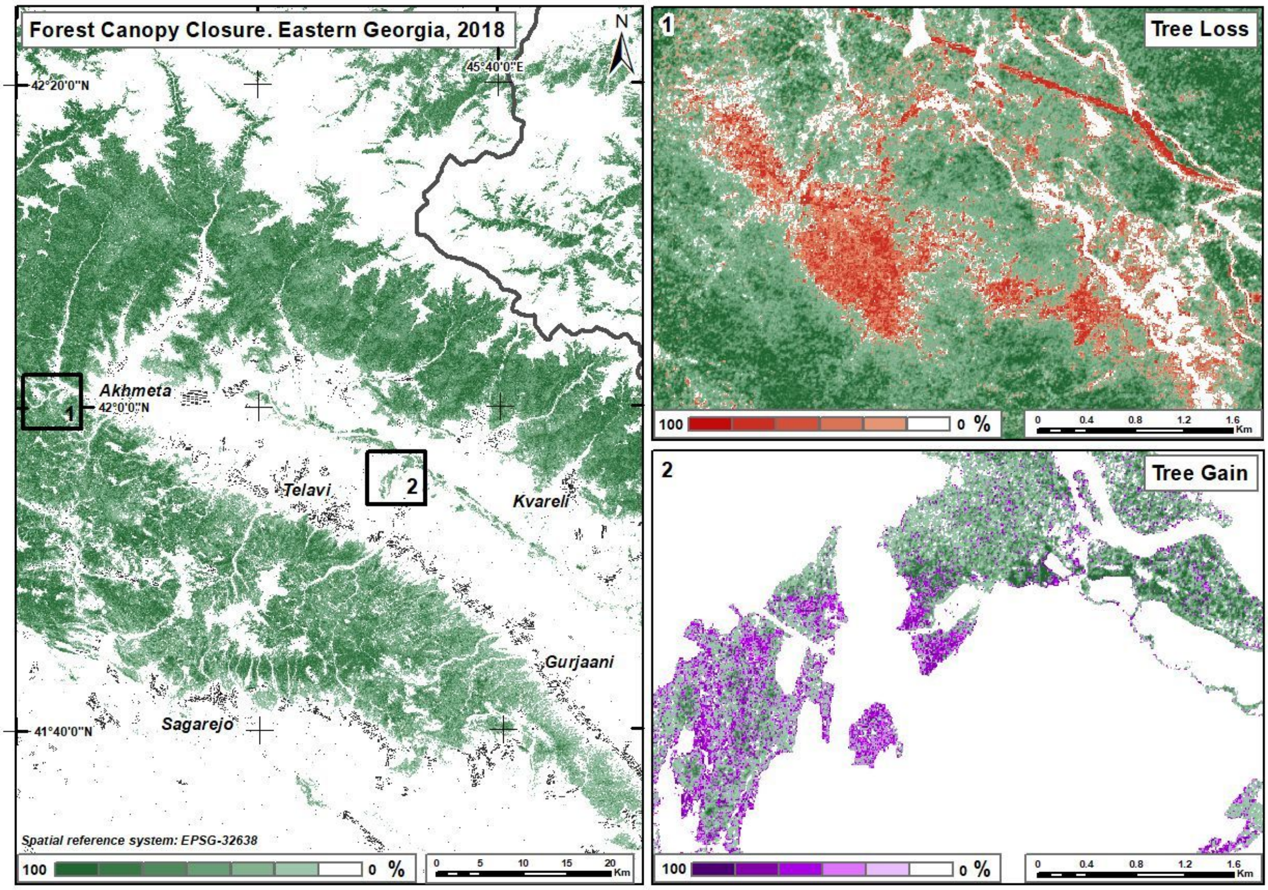
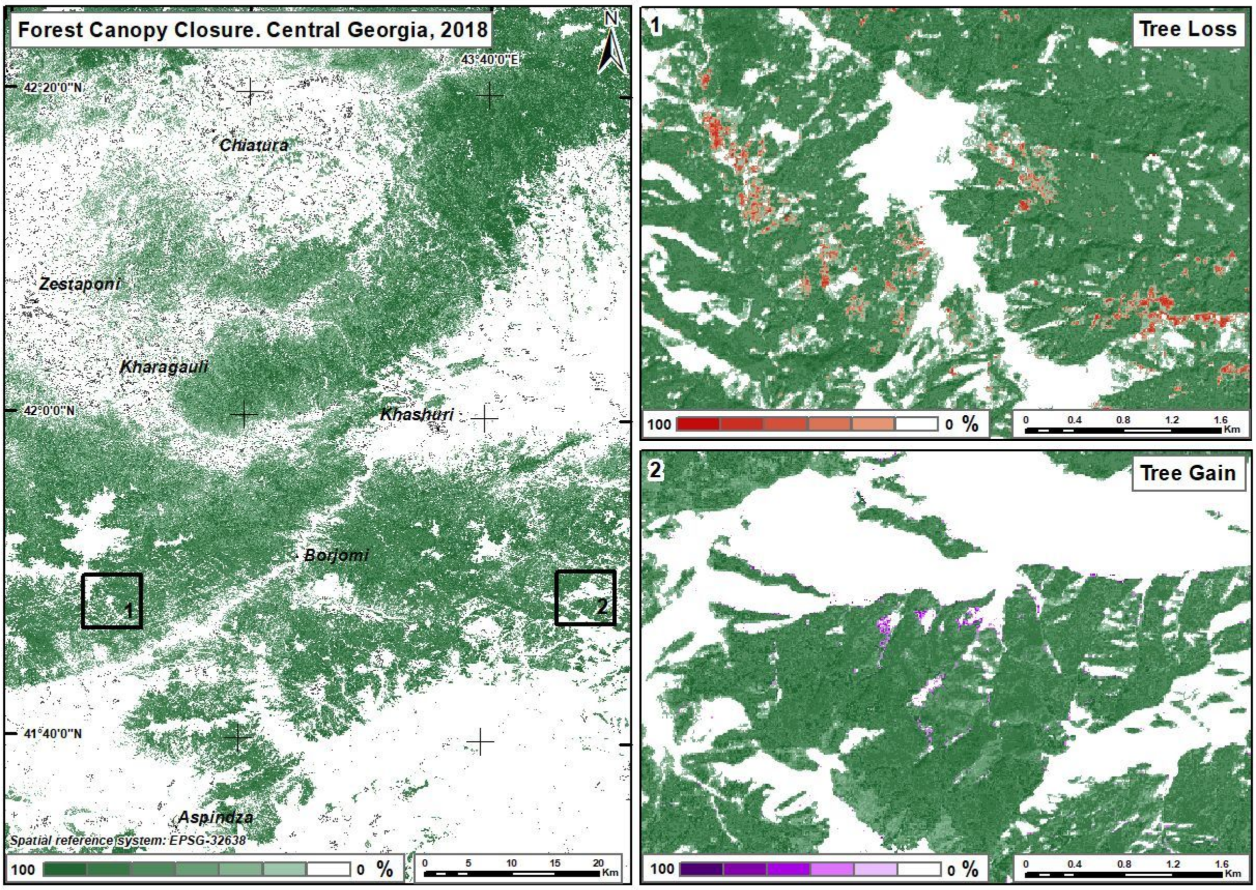


Fig. S4 - Modeled canopy closure (FEYE) in central Georgia (at left) and statistically significant differences in the canopy closure in percentage at two sites between 28.08.2018 and 14.08.2015. The map is projected to UTM Zone 38N; WGS: 1984.



Tab. S1 - List of the variables used for modeling forest cover. Response variables or proxies for forest cover are as follows: FEYE, TCOV, TBA, TAH.

Variable	**Description**
FEYE	Canopy closure (%) based on the ratio of the area of canopy to sky, calculated from upward-looking fisheye photographs taken beneath forest canopy.
TCOV	Tree cover percentage defined by foresters on the field
TBA	Total Basal Area (m^2)
TAH	Tree Average height (m)
DT	Difference in time (=days) between sampling and Sentinel-2A image acquisition in 2018
SLP	Slope (degree)
EXP	Aspect (degree)
B02	Sentinel-2A Band 2 – Blue
B03	Sentinel-2A Band 3 – Green
B04	Sentinel-2A Band 4 – Red
B05	Sentinel-2A Band 5 – Vegetation red edge
B06	Sentinel-2A Band 6 – Vegetation red edge
B07	Sentinel-2A Band 7 – Vegetation red edge
B08	Sentinel-2A Band 8 – NIR
B8A	Sentinel-2A Band 8A – Narrow NIR
B11	Sentinel-2A Band 11 – SWIR
B12	Sentinel-2A Band 12 – SWIR
NBR1	Normalized Difference NIR/SWIR: (B08 - B11) / (B08 + B11)
NBR2	Normalized Difference NIR/SWIR: (B8A - B11) / (B8A + B11)
NBR3	Normalized Difference NIR/SWIR: (B08 - B12) / (B08 + B12)
NBR4	Normalized Difference NIR/SWIR: (B8A - B12) / (B8A + B12)
AFRI1	Aerosol free vegetation index: (B08 - (0.66 * B11)) / (B08 + (0.66 * B11))
AFRI2	Aerosol free vegetation index: (B8A - (0.66 * B11)) / (B8A + (0.66 * B11))
AFRI3	Aerosol free vegetation index: (B08 - (0.66 * B12)) / (B08 + (0.66 * B12))
AFRI4	Aerosol free vegetation index: (B8A - (0.66 * B12)) / (B8A + (0.66 * B12))
BNDVI1	Normalized Difference NIR/Blue: (B08 - B02) / (B08 + B02)
BNDVI2	Normalized Difference NIR/Blue: (B8A - B02) / (B8A + B02)
BWDRVI1	Blue-wide dynamic range vegetation index: ((0.1 * B08) - B02) / ((0.1 * B08) + B02)
BWDRVI2	Blue-wide dynamic range vegetation index: ((0.1 * B8A) - B02) / ((0.1 * B8A) + B02)
NDVI1	Normalized Difference NIR/Red: (B08 - B04) / (B08 + B04)
NDVI2	Normalized Difference NIR/Red: (B8A - B04) / (B8A + B04)
WDRVI1	Wide Dynamic Range Vegetation Index: ((0.1 * B08) - B04) / ((0.1 * B08) + B04)
WDRVI2	Wide Dynamic Range Vegetation Index: ((0.1 * B8A) - B04) / ((0.1 * B8A) + B04)
SAVI1	Soil Adjusted Vegetation Index: ((1 + 0.5) * (B08 - B04)) / (B08 + B04 + 0.5)
SAVI2	Soil Adjusted Vegetation Index: ((1 + 0.5) * (B8A - B04)) / (B8A + B04 + 0.5)
GNDVI1	Normalized Difference NIR/Green: (B08 - B03) / (B08 + B03)
GNDVI2	Normalized Difference NIR/Green: (B8A - B03) / (B8A + B03)

Variable	**Description**
NDRE1	Normalized Difference NIR/Red-edge: (B08 - B05) / (B08 + B05)
NDRE2	Normalized Difference NIR/Red-edge: (B08 - B06) / (B08 + B06)
NDRE3	Normalized Difference NIR/Red-edge: (B08 - B07) / (B08 + B07)
NDRE4	Normalized Difference NIR/Red-edge: (B8A - B05) / (B8A + B05)
NDRE5	Normalized Difference NIR/Red-edge: (B8A - B06) / (B8A + B06)
NDRE6	Normalized Difference NIR/Red-edge: (B8A - B07) / (B8A + B07)
VIgreen	Normalized Difference Green/Red index: (B03 - B04) / (B03 + B04)
CIgreen1	Chlorophyll Index Green: (B08 / B03) - 1
CIgreen2	Chlorophyll Index Green: (B8A / B03) - 1
CIrededge1	Chlorophyll IndexRed-Edge: (B08 / B05) - 1
CIrededge2	Chlorophyll IndexRed-Edge: (B08 / B06) - 1
CIrededge3	Chlorophyll IndexRed-Edge: (B08 / B07) - 1
CIrededge4	Chlorophyll IndexRed-Edge: (B8A / B05) - 1
CIrededge5	Chlorophyll IndexRed-Edge: (B8A / B06) - 1
CIrededge6	Chlorophyll IndexRed-Edge: (B8A / B07) - 1
CI	Coloration Index: (B04 - B02) / B04
CVI1	Chlorophyll vegetation index: B08 * (B04 / (B03^2))
CVI2	Chlorophyll vegetation index: B8A * (B04 / (B03^2))
CCCI1	Canopy Chlorophyll Content Index: ((B08 - B05) / (B08 + B05)) / ((B08 - B04) / (B08 + B04))
CCCI2	Canopy Chlorophyll Content Index: ((B08 - B06) / (B08 + B06)) / ((B08 - B04) / (B08 + B04))
CCCI3	Canopy Chlorophyll Content Index: ((B08 - B07) / (B08 + B07)) / ((B08 - B04) / (B08 + B04))
CCCI4	Canopy Chlorophyll Content Index: ((B8A - B05) / (B8A + B05)) / ((B8A - B04) / (B8A + B04))
CCCI5	Canopy Chlorophyll Content Index: ((B8A - B06) / (B8A + B06)) / ((B8A - B04) / (B8A + B04))
CCCI6	Canopy Chlorophyll Content Index: ((B8A - B07) / (B8A + B07)) / ((B8A - B04) / (B8A + B04))
EVI1	Enhanced Vegetation Index: 2.5 * ((B08 - B04) / (B08 + (6 * B04) - (7.5 * B02) + 1))
EVI2	Enhanced Vegetation Index: 2.5 * ((B8A - B04) / (B8A + (6 * B04) - (7.5 * B02) + 1))
GARI1	Green atmospherically resistant vegetation index: (B08 - (B03 - (B02 - B04))) / (B08 - (B03 + (B02 - B04)))
GARI2	Green atmospherically resistant vegetation index: (B8A - (B03 - (B02 - B04))) / (B8A - (B03 + (B02 - B04)))
GLI	Green leaf index: ((2 * B03) - (B04 + B02)) / ((2 * B03) + (B04 + B02))
GBNDVI1	Green-Blue NDVI: (B08 - (B03 + B02)) / (B08 + (B03 + B02))
GBNDVI2	Green-Blue NDVI: (B8A - (B03 + B02)) / (B8A + (B03 + B02))
GRNDVI1	Green-RED NDVI: (B08 - (B03 + B04)) / (B08 + (B03 + B04))
GRNDVI2	Green-RED NDVI: (B8A - (B03 + B04)) / (B8A + (B03 + B04))
SLAVI1	Specific Leaf Area Vegetation Index: B08 / (B04 + B11)

Variable	**Description**
SLAVI2	Specific Leaf Area Vegetation Index: B8A / (B04 + B11)
SLAVI3	Specific Leaf Area Vegetation Index: B08 / (B04 + B12)
SLAVI4	Specific Leaf Area Vegetation Index: B8A / (B04 + B12)
IMG	Two scenes taken on 01.07.2018 in Eastern Georgia and on 28.08.2018 in Central Georgia

Tab. S2 - The accuracy of the best fit GAM models for forest cover measures based on Minnaert-corrected Sentinel bands and family = Gamma(link = "log").

Forest Density measure	R^2_{adj}	**Deviance explained (%)**
FEYE	0.839	69.7
TCOV	0.699	67.9
TAH	0.651	67
TBA	0.571	68.9

Appendix 1 - The main R codes used in the study.

```
setwd("C:\\...")

df <- read.csv("forest stand data.csv", stringsAsFactors=FALSE)

df$Sample.Date <- as.POSIXct(df$FT, format="%d-%m-%y", tz="Asia/Tbilisi") # converts FT to a date
format variable at a time zone (tz) in which the data are collected.

df$Jul.Day <- as.numeric(format(df$Sample.Date, "%j")) # returns Julian Day Numbers of the year.

df$BA <- with(df, TBA / cos(SLP*(pi/180))) # adjusts TBA to horizontal.

df$IMG_18[df$IMG_18=="Image3"] <- "Kakheti 01.07.2018" # specifies the place and data of Image3.

df$IMG_18[df$IMG_18=="Image4"] <- "Meskheti 28.08.2018" # specifies the place and data of Image4.

df$IMG_15_16[df$IMG_15_16=="Image1"] <- "Meskheti 14.08.2015" # specifies the place and data of
Image1.

df$IMG_15_16[df$IMG_15_16=="Image2"] <- "Kakheti 05.08.2016" # specifies the place and data of
Image2.

df$IMG_18 <- factor(df$IMG_18) # factorizes IMG_18.


df$closed <- round(df$FEYE*10, digits = 0) # returns canopy closure per mille.

df$open <- round(1000 - (df$FEYE*10), digits = 0) # returns canopy opening per mille.


df$DOM_COV1[df$DOM_SPEC=="Abies nordmaniana" | df$DOM_SPEC=="Picea orientalis" | df$DOM_SPEC=="Pinus
kochiana"] <- "needle-leaf" # groups dominant coniferous trees.

df$DOM_COV1[df$DOM_SPEC=="Carpinus betulus" | df$DOM_SPEC=="Fagus orientalis" |
df$DOM_SPEC=="Quercus iberica"] <- "broad-leaf" # groups dominant deciduous trees.

df$DOM_COV1[df$DOM_SPEC=="Dryopteris filix mas" | df$DOM_SPEC=="Festuca sp." | df$DOM_SPEC=="Rubus
caucasicus" | df$DOM_SPEC=="Subalpine meadow"] <- "herbaceous" # groups dominant non-tree plants.

df$DOM_COV1 <- factor(df$DOM_COV1)


df$DOM_COV2[df$DOM_SPEC=="Abies nordmaniana" | df$DOM_SPEC=="Picea orientalis"] <- "Abies-Picea"

df$DOM_COV2[df$DOM_SPEC=="Pinus kochiana"] <- "Pinus"

df$DOM_COV2[df$DOM_SPEC=="Carpinus betulus" | df$DOM_SPEC=="Fagus orientalis" |
df$DOM_SPEC=="Quercus iberica"] <- "broad-leaf"

df$DOM_COV2[df$DOM_SPEC=="Rubus caucasicus"] <- "Rubus"

df$DOM_COV2[df$DOM_SPEC=="Subalpine meadow"] <- "Subalpine"

df$DOM_COV2[df$DOM_SPEC=="Dryopteris filix mas" | df$DOM_SPEC=="Festuca sp."] <- "herbaceous2"

df$DOM_COV2 <- factor(df$DOM_COV2)


df$DOM_COV3[df$DOM_SPEC=="Abies nordmaniana" | df$DOM_SPEC=="Picea orientalis" | df$DOM_SPEC=="Pinus
kochiana" | df$DOM_SPEC=="Carpinus betulus" | df$DOM_SPEC=="Fagus orientalis" |
df$DOM_SPEC=="Quercus iberica"] <- "Forest"

df$DOM_COV3[df$DOM_SPEC=="Rubus caucasicus"] <- "Rubus"

df$DOM_COV3[df$DOM_SPEC=="Subalpine meadow"] <- "Subalpine"

df$DOM_COV3[df$DOM_SPEC=="Dryopteris filix mas" | df$DOM_SPEC=="Festuca sp."] <- "herbaceous2"

df$DOM_COV3 <- factor(df$DOM_COV3)
```

```
df$DOM_COV4[df$DOM_SPEC=="Abies nordmaniana" | df$DOM_SPEC=="Picea orientalis" | df$DOM_SPEC=="Pinus
kochiana" | df$DOM_SPEC=="Carpinus betulus" | df$DOM_SPEC=="Fagus orientalis" |
df$DOM_SPEC=="Quercus iberica"] <- "Forest"

df$DOM_COV4[df$DOM_SPEC=="Rubus caucasicus"] <- "Rubus"

df$DOM_COV4[df$DOM_SPEC=="Subalpine meadow" | df$DOM_SPEC=="Dryopteris filix mas" |
df$DOM_SPEC=="Festuca sp."] <- "herbaceous3"

df$DOM_COV4 <- factor(df$DOM_COV4)


df$DOM_COV5[df$DOM_SPEC=="Abies nordmaniana" | df$DOM_SPEC=="Picea orientalis" | df$DOM_SPEC=="Pinus
kochiana" | df$DOM_SPEC=="Carpinus betulus" | df$DOM_SPEC=="Fagus orientalis" |
df$DOM_SPEC=="Quercus iberica"] <- "Forest"

df$DOM_COV5[df$DOM_SPEC=="Subalpine meadow" | df$DOM_SPEC=="Dryopteris filix mas" |
df$DOM_SPEC=="Festuca sp." | df$DOM_SPEC=="Rubus caucasicus"] <- "herbaceous"

df$DOM_COV5 <- factor(df$DOM_COV5)


# Possible cover in 2015-16:

df$DOM_COV_15_16[df$DOM_COV5=="Forest"] <- "Forest"

df$DOM_COV_15_16[df$DOM_COV5=="herbaceous"] <- "Forest"

df$DOM_COV_15_16[df$DOM_SPEC=="Subalpine meadow"] <- "herbaceous"


library(ggplot2)

library(ggrepel)

library(mgcv)

#-----------------------------------------------------

A <- gam(cbind(closed, open) ~ s(B03_A18) + s(B08_A18) + s(B12_A18), data = df, method = "REML",
family = binomial(link = "logit")) # runs a GAM on a proportion dependent variable and the Sen2cor-
corrected Sentinel bands.

C <- gam(cbind(closed, open) ~ s(B03_C18) + s(B08_C18) + s(B12_C18), data = df, method = "REML",
family = binomial(link = "logit")) # runs a GAM on a proportion dependent variable and the Cosine-
corrected Sentinel bands.

M <- gam(cbind(closed, open) ~ s(B03_M18) + s(B08_M18) + s(B12_M18), data = df, method = "REML",
family = binomial(link = "logit")) # runs a GAM on a proportion dependent variable and the Minnaert-
corrected Sentinel bands.

N <- gam(cbind(closed, open) ~ s(B03_N18) + s(B08_N18) + s(B12_N18), data = df, method = "REML",
family = binomial(link = "logit")) # runs a GAM on a proportion dependent variable and the
Normalization-corrected Sentinel bands.


df$pred.A <- predict(A, type="response") # adds A-fitted values to the data frame.

df$pred.C <- predict(C, type="response") # adds C-fitted values to the data frame.

df$pred.M <- predict(M, type="response") # adds M-fitted values to the data frame.

df$pred.N <- predict(N, type="response") # adds N-fitted values to the data frame.


df$resid.A <- with(df, FEYE - (pred.A * 100)) # adds A-residuals to the data frame.

df$resid.C <- with(df, FEYE - (pred.C * 100)) # adds C-residuals to the data frame.
```

```
df$resid.M <- with(df, FEYE - (pred.M * 100)) # adds M-residuals to the data frame.

df$resid.N <- with(df, FEYE - (pred.N * 100)) # adds N-residuals to the data frame.


df$resid2.A <- with(df, (FEYE - (pred.A * 100))^2) # adds squared A-residuals to the data frame.

df$resid2.C <- with(df, (FEYE - (pred.C * 100))^2) # adds squared C-residuals to the data frame.

df$resid2.M <- with(df, (FEYE - (pred.M * 100))^2) # adds squared M-residuals to the data frame.

df$resid2.N <- with(df, (FEYE - (pred.N * 100))^2) # adds squared N-residuals to the data frame.


# Fig. S1

library(ggplot2)
ggplot(df) + geom_density(aes(x = resid.A, color="Sen2cor correction"), size=2) + geom_density(aes(x
= resid.C, color="Cosine Correction"), size=2) + geom_density(aes(x = resid.N, color="Normalization
Method"), size=2) + geom_density(aes(x = resid.M, color="Minnaert Correction"), size=2) +
labs(x="Residuals", y="Density") + scale_color_manual(name = "", values = c("Sen2cor correction" =
"green", "Cosine Correction" = "red", "Minnaert Correction" = "black", "Normalization Method" =
"blue"))


# Calculating the 100-95% range limits of residuals of models derived through different topographic
corrections: A, C, M, N.

round(quantile(df$resid.A, probs = c(0, 0.025, 0.975, 1)), 3)

round(quantile(df$resid.C, probs = c(0, 0.025, 0.975, 1)), 3)

round(quantile(df$resid.M, probs = c(0, 0.025, 0.975, 1)), 3)

round(quantile(df$resid.N, probs = c(0, 0.025, 0.975, 1)), 3)

#-----------------------------------------------------

        # Leave-one-out cross validation for different topographic corrections: A, C, M, N.

df <- df[sample(nrow(df)), ] # randomly shuffles the data.

k <- nrow(df)

folds <- cut(seq(from=1, to=nrow(df)), breaks=k, labels=FALSE) # creates unique numbers for k
equally size folds.

df$ID <- folds # adds fold IDs.


library(mgcv)
library(parallel)
library(foreach)
library(doParallel)
no_cores <- detectCores() # finds the number of cores in system.
registerDoParallel(makeCluster(no_cores))


# Parallelizing loops for predictions via different models:

pred.M.cv      <- foreach(i=1:k, .combine = c, .packages="mgcv")  %dopar% predict(gam(cbind(closed,
open) ~ s(B03_M18) + s(B08_M18) + s(B12_M18), data = df, subset=(ID != i), method = "REML", family =
binomial(link = "logit")), df[df$ID==i, ], type="response")

pred.A.cv <- foreach(i=1:k, .combine = c, .packages="mgcv")  %dopar% predict(gam(cbind(closed, open)
~ s(B03_A18) + s(B08_A18) + s(B12_A18), data = df, subset=(ID != i), method = "REML", family =
binomial(link = "logit")), df[df$ID==i, ], type="response")

pred.C.cv <- foreach(i=1:k, .combine = c, .packages="mgcv")  %dopar% predict(gam(cbind(closed, open)
~ s(B03_C18) + s(B08_C18) + s(B12_C18), data = df, subset=(ID != i), method = "REML", family =
binomial(link = "logit")), df[df$ID==i, ], type="response")
```

```
pred.N.cv <- foreach(i=1:k, .combine = c, .packages="mgcv")  %dopar% predict(gam(cbind(closed, open)
~ s(B03_N18) + s(B08_N18) + s(B12_N18), data = df, subset=(ID != i), method = "REML", family =
binomial(link = "logit")), df[df$ID==i, ], type="response")


# MSE (Mean squared error of Cross-validation)

mean((df$FEYE - (pred.M.cv * 100))^2)

mean((df$FEYE - (pred.A.cv * 100))^2)

mean((df$FEYE - (pred.C.cv * 100))^2)

mean((df$FEYE - (pred.N.cv * 100))^2)

#------------------------------------------------------

# Best GAM model (i.e. M)

summary(M) # summarizes the GAM named M.

# Fig. 3
ggplot(df, aes(x = (pred.M * 100), y = FEYE)) + labs(x = "Predicted FEYE", y = "Observed FEYE",
shape="Sentinel Image\nPlace/Time", size="Absolute\nResiduals", color="Dominant\nVegetation") +
geom_point(aes(color = DOM_COV1, shape=IMG_18, size=abs(resid.M))) +
geom_text_repel(aes(label=DOM_SPEC), size = 2) # returns a Scatter plot of fitted and observed FEYE
values.

# Fig. S2
ggplot(df, aes(x = resid.M)) + geom_histogram(binwidth=5) + labs(x="Residual", y="Count") # plots a
histogram of model residuals.


plot(resid.M ~ DT_18, data=df) # plots model residuals against difference in time between sampling
forest density parameters and Sentinel image acquisition.

plot(abs(resid.M) ~ abs(DT_18), data=df) # plots absolute model residuals against absolute
difference in time between sampling forest density parameters and Sentinel image acquisition.

summary(lm(abs(resid.M) ~ abs(DT_18), data=df)) # summarizes relationship between absolute model
residuals and absolute difference in time between sampling forest density parameters and Sentinel
image acquisition.


# Fig. 2. Plotting 3D surfaces of FEYE vs Sentinel-2A spectral bands:

par(mfrow=c(1,3))

vis.gam(M, type="response", view=c("B03_M18", "B08_M18"), theta=30, ticktype="detailed")

vis.gam(M, type="response", view=c("B03_M18", "B12_M18"), theta=30, ticktype="detailed")

vis.gam(M, type="response", view=c("B08_M18", "B12_M18"), theta=-45, ticktype="detailed")


#------------------------------------------------------

        # Statistically significant forest closure change between 2015/2016 and 2018

data15_16 <- df[, c("B02_M15_16", "B03_M15_16", "B04_M15_16", "B05_M15_16", "B06_M15_16",
"B07_M15_16", "B08_M15_16", "B11_M15_16", "B12_M15_16", "B8A_M15_16")] # extracts Sentinel bands
acquired in 2015-2016.

colnames(data15_16) <- c("B02_M18", "B03_M18", "B04_M18", "B05_M18", "B06_M18", "B07_M18",
"B08_M18", "B11_M18", "B12_M18", "B8A_M18") # renames the vectors of Sentinel bands to make them
visible to M (see the next code).

pred15_16 <- as.data.frame(predict(M, newdata = data15_16, type = "response", se.fit=TRUE)) # makes
predictions on data15_16.
```

```
colnames(pred15_16) <- c("pred15_16", "se.pred15_16") # renames prediction estimates and their
standard errors.

pred18 <- as.data.frame(predict(M, type = "response", se.fit=TRUE)) # makes predictions on training
data.

colnames(pred18) <- c("pred18", "se.pred18") # renames prediction estimates and their standard
errors.


df <- cbind(df, pred15_16, pred18) # binds the base data frame, prediction estimates and their
standard errors.


df$loss.gain <- NA # adds a new empty variable (vector) "loss.gain" to a data frame "df".


for (i in 1:nrow(df)) {


if((df$pred18[i] - (2 * df$se.pred18[i])) > (df$pred15_16[i] + (2 * df$se.pred15_16[i]))) {


df$loss.gain[i] <- "forest gain"


} else if((df$pred18[i] + (2 * df$se.pred18[i])) < (df$pred15_16[i] - (2 * df$se.pred15_16[i]))) {


df$loss.gain[i] <- "forest loss"


} else {


df$loss.gain[i] <- "no change"

}
}

tab <- table(df$loss.gain)

prop.table(tab) * 100

#-----------------------------------------------------

        # Mapping FEYE and its standard error (i.e. Predicting a model to a set of predictor
rasters)

library(raster)


predfun <- function(model, data) {

v <- predict(model, data, type="response", se.fit=TRUE)

cbind(p=as.vector(v$fit), se=as.vector(v$se.fit))

} # predfun returns a matrix of two variables: p and se.


        # (1) FEYE and its standard error mapped for the Meskheti 14.08.2015 Sentinel image

var1.03 <- raster("B03_M1.asc")

var1.08 <- raster("B08_M1.asc")

var1.12 <- raster("B12_M1.asc")

stack.1 <- stack(var1.03, var1.08, var1.12)
```

```
names(stack.1) <- c("B03_M18", "B08_M18", "B12_M18")
gam1p <- predict(stack.1, M, fun=predfun, index=1, progress="text", filename="FEYE1.asc")
gam1se <- predict(stack.1, M, fun=predfun, index=2, progress="text", filename="FEYE1_se.asc")

        # (2) FEYE and its standard error mapped for the Kakheti 05.08.2016 Sentinel image
var2.03 <- raster("B03_M2.asc")
var2.08 <- raster("B08_M2.asc")
var2.12 <- raster("B12_M2.asc")
stack.2 <- stack(var2.03, var2.08, var2.12)
names(stack.2) <- c("B03_M18", "B08_M18", "B12_M18")
gam2p <- predict(stack.2, M, fun=predfun, index=1, progress="text", filename="FEYE2.asc")
gam2se <- predict(stack.2, M, fun=predfun, index=2, progress="text", filename="FEYE2_se.asc")

        # (3) FEYE and its standard error mapped for the Kakheti 01.07.2018 Sentinel image
var3.03 <- raster("B03_M3.asc")
var3.08 <- raster("B08_M3.asc")
var3.12 <- raster("B12_M3.asc")
stack.3 <- stack(var3.03, var3.08, var3.12)
names(stack.3) <- c("B03_M18", "B08_M18", "B12_M18")
gam3p <- predict(stack.3, M, fun=predfun, index=1, progress="text", filename="FEYE3.asc")
gam3se <- predict(stack.3, M, fun=predfun, index=2, progress="text", filename="FEYE3_se.asc")

        # (4) FEYE and its standard error mapped for the Meskheti 28.08.2018 Sentinel image
var4.03 <- raster("B03_M4.asc")
var4.08 <- raster("B08_M4.asc")
var4.12 <- raster("B12_M4.asc")
stack.4 <- stack(var4.03, var4.08, var4.12)
names(stack.4) <- c("B03_M18", "B08_M18", "B12_M18")
gam4p <- predict(stack.4, M, fun=predfun, index=1, progress="text", filename="FEYE4.asc")
gam4se <- predict(stack.4, M, fun=predfun, index=2, progress="text", filename="FEYE4_se.asc")
#----------------------------------------------------
```