

```

# Forest Biomass in Bangladesh: An historical review of forest inventories to
# assist national estimates

# Gael Sola, FAO, gael.sola@fao.org,
# Liam Costello, liam.costello@fao.org
# 04/2016

# Part 01a: reading and merging files, corrections of plot coordinates,
#           add BFI, FAO and LCCS classes

# Contains:
# 0. import and merge data
# 1. Improve the entity_ab column
# 2. Correct incorrect years in the raw data
# 3. Transform NFA plot into land use based plots
# 4. Correct plot coordinates
# 5. Add FAO ecological zones
# 6. Add BFI zones
# 7. Add land cover classification system

#####
# Erase memory
rm(list=ls())

# remove graphs
dev.off()

# Load / install packages
#install.packages("ggthemes")
library(tidyverse)
library(ggthemes)
library(rgdal)
library(rgeos)
library(grid)
library(gridExtra)

# Set working directory
# setwd("//uom-file.unimelb.edu.au/220/Users/lcostello1/Documents/BFI/Harmo/Analysis/data")
setwd("/media/gs/OSDisk/Users/solag/Documents/Work/Bangladesh article harmo/Analysis/data")

# Theme for printing
source("R-corrections/theme_gs_print.R")

#####
#-----#
#           0. import and merge data
#-----#

# Merge
fileName <- list.files(pattern = "csv")
fileList <- lapply(fileName, read.csv, head = T, quote="\\" , stringsAsFactors = F)
names(fileList) <- fileName
table_all <- Reduce(rbind, fileList)

# Remove temp objects
rm(fileName)
rm(fileList)

# Remove columns: GPS with degree symbol
table_all$x_wgs84 <- NULL
table_all$y_wgs84 <- NULL

# Overwrite id_tree_2 with successive integers
table_all$id_tree_2 <- 1:dim(table_all)[1]

t_save <- table_all
table_all <- t_save

```

```

#####
##### Remove rows with tree DBH NA or 0
#####
table_all <- table_all[which(!is.na(table_all$dbh_cm) & table_all$dbh_cm != 0),]

#-----#
#           1. Improve the entity_ab column
#-----#
t_savel <- table_all
table_all <- t_savel

summary(as.factor(table_all$entity_ab))

vec <- c("DDWS", "FKWS", "INP", "TWS", "SKRF", "MKNP")
table_all[which(table_all$entity_ab %in% vec),]$entity_ab <- "CREL-10"

vec <- c("KhNP", "KNP", "LNP", "MNP", "RKWS", "SNP")
table_all[which(table_all$entity_ab %in% vec ),]$entity_ab <- "CREL-14"

table_all[which(table_all$entity_ab == "FD-SAL"),]$entity_ab <- "FD-01-SAL"

table_all[which(table_all$entity_ab == "FD-NFA"),]$entity_ab <- "FD-07-NFA"

table_all[which(table_all$entity_ab == "Sund RF"),]$entity_ab <- "FD-10-SundRF"

vec <- c("FD-Hill", "FD-97-coastal", "FD-97-sund")
table_all[which(table_all$entity_ab %in% vec ),]$entity_ab <- "FRMP-97"

# Check
summary(as.factor(table_all$entity_ab))

# Create unique ID for plots
table_all$id <- paste(table_all$entity_ab,table_all$id_plot,sep="_")

# check NA is IDs
dim(table_all[is.na(table_all$id_plot),])[1]
table_all[is.na(table_all$id), "organisation"]

#-----#
#-----# for table data correction: number of plot and trees per entity - Start
nb_tree <- table_all[, c("id","entity_ab")]
nb_plot <- unique(nb_tree)

# Create a table to store the nb trees/plots
nb_plot_tree <- data.frame(
  names = c(names(table(table_all$entity_ab)), "Total"),
  init = paste(c(table(nb_plot$entity_ab),
    dim(nb_plot)[1]),
    " (",
    c(table(nb_tree$entity_ab),
    dim(nb_tree)[1]),
    ")",
    sep = "" ),
  row.names = NULL)
nb_plot_tree
#-----#
#-----#

#-----#
#           2. Correct incorrect years in the raw data
#-----#
unique(table_all[,c("year", "entity_ab")])

table_all[which(table_all$entity_ab=="FRMP-97" & table_all$year != 1997),"year"] <- 1997
table_all[which(table_all$entity_ab=="FRMP-97" & is.na(table_all$year)),"year"] <- 1997

unique(table_all[,c("year", "entity_ab")])

```

```

#-----#
#           3. Transform NFA plot into land use based plots
#-----#

# for NFA, the column soil contain the land use code
str(table_all[table_all$entity_ab == "FD-07-NFA", "soil"])

# id_plot is modified to take land use into account
table_all$id_plot <- ifelse(table_all$entity_ab == "FD-07-NFA", table_all$soil, table_all$id_plot)

# update unique ID for plots
table_all$id <- paste(table_all$entity_ab, table_all$id_plot, sep="_")

# check NA is IDs
dim(table_all[is.na(table_all$id_plot), ])[1]
table_all[is.na(table_all$id), "organisation"]

#-----#
#-----# for table data correction: update nb plots for NFA
nb_tree <- table_all[, c("id", "entity_ab")]
nb_plot <- unique(nb_tree)

# Create a table to store the nb trees/plots
nb_plot_tree <- cbind(nb_plot_tree,
                     init_NFA_lu = paste(c(table(nb_plot$entity_ab),
                                             dim(nb_plot)[1]),
                                          " (",
                                          c(table(table_all$entity_ab),
                                             dim(table_all)[1]),
                                          ")",
                                          sep = ""))
)
nb_plot_tree
#-----#
#-----#

#-----#
#           4. Correct plot coordinates
#-----#

t_save4 <- table_all
table_all <- t_save4

# Check number of trees with GPS coordinate NA or 0
test <- table_all[is.na(table_all$x_dec_wgs84) | is.na(table_all$y_dec_wgs84) |
                 table_all$x_dec_wgs84==0 | table_all$y_dec_wgs84==0, ]
test <- unique(test[, c("id", "entity_ab", "x_dec_wgs84", "y_dec_wgs84")])
table(test$entity_ab)

# Check on map:
test <- unique(table_all[!is.na(table_all$x_dec_wgs84), c("id", "x_dec_wgs84", "y_dec_wgs84")])
plot1 <- ggplot(test) +
  geom_point(aes(x=x_dec_wgs84, y=y_dec_wgs84), cex = 0.5) +
  coord_fixed(ratio=1) +
  scale_x_continuous(limits = c(88, 93), breaks = seq(88, 93, 1)) +
  scale_y_continuous(limits = c(20, 27), breaks = seq(20, 27, 1)) +
  labs(x = NULL, y = NULL) +
  annotate("text", x = 92, y = 27, label = "346 plots outside \nmap boundaries", size = 3)

plot1
# Apply the manual corrections
source("R-corrections/corr01_plot_coord_10nov.R")

#####
##### Remove the plots with GPS values 0 or NA
#####
table_all <- table_all[table_all$x_corr != 0 & table_all$y_corr != 0 &
                      !is.na(table_all$x_corr) & !is.na(table_all$y_corr), ]

```

```

# Check map:
test <- unique(table_all[,c("id", "id_plot", "x_corr", "y_corr", "entity_ab")])
plot2 <- test %>%
  arrange(desc(entity_ab)) %>%
  ggplot() +
  geom_point(aes(x=x_corr, y=y_corr, colour=entity_ab, pch=entity_ab)) +
  # geom_text(aes(x=x_corr, y=y_corr, colour=entity_ab, pch=entity_ab, label = id)) +
  coord_fixed(ratio=1) +
  scale_x_continuous(limits = c(88, 93), breaks = seq(88, 93, 1)) +
  scale_y_continuous(limits = c(20, 27), breaks = seq(20, 27, 1)) +
  labs(x = NULL, y = NULL, color = "Project code", pch = "Project code")
plot2
# export the plot
ggsave("results/gr_map_plot_location_corr_entity.png", width = 15, height = 12, units = "cm", dpi =

# Compa before after corrections
plot_mix <- arrangeGrob(plot1, plot2 +theme(legend.position = "none"),
  ncol = 2
)

grid.arrange(plot_mix)
# save plot
ggsave(filename = "results/gr_map_plot_location_before_after_corr.png",
  plot = plot_mix,
  width = 15, height = 12, units = "cm", dpi = 300
)

### NOTE: CREL-14 are left uncorrected, the location of plots are
### outside the sanctuaries boundaries but not too far off.

#-----#
#-----# for table data correction: correction of gps coordinates
nb_tree <- table_all[,c("id", "entity_ab", "x_corr", "y_corr")]
nb_plot <- unique(nb_tree)

# Update the table nb tree/plots
nb_plot_tree <- cbind(nb_plot_tree,
  plot_coord_cleaning = paste(c(table(nb_plot$entity_ab),
    dim(nb_plot)[1]),
    " (",
    c(table(nb_tree$entity_ab),
    dim(nb_tree)[1]),
    ")",
    sep = ""))
)

nb_plot_tree
#-----#
#-----#

#-----#
#                               5. Add FAO ecological zone
#-----#

t_save5 <- table_all
table_all <- t_save5

# Run the addition of FAO Biome
# !!! Require rgdal package and the FAO biome shapefile
# !!! The directory to FAO biome shp needs to be manually specified in the source code)
source("R-corrections/corr02_fao_biome_10nov.R")

# Check
summary(as.factor(table_all$fao_biome))
# Check plot nb
length(unique(table_all$id))
dim(unique(table_all[,c("id", "x_corr", "y_corr", "fao_biome")]))

```

```

# Control
test <- unique(table_all[,c("id", "x_corr", "y_corr", "entity_ab", "fao_biome")])
ggplot(test) +
  geom_point(aes(x=x_corr, y=y_corr, colour=fao_biome), cex=0.5) +
  coord_fixed(ratio=1) +
  scale_x_continuous(limits = c(88, 93), breaks = seq(88, 93, 1)) +
  scale_y_continuous(limits = c(20, 27), breaks = seq(20, 27, 1)) +
  theme(legend.position = "bottom", legend.title = element_blank()) +
  labs(x = NULL, y = NULL, color = "FAO ecological zone")

# save plot
ggsave("results/gr_map fao biomes.png", width = 15, height = 12, units = "cm", dpi = 300)

#-----#
#                               6. Add BFI Zones
#-----#

t_save6 <- table_all
table_all <- t_save6

# Add BFI zone based on shapefile and plot coordinates
source("R-corrections/corr03_bfi_zone_10nov.R")

# Check summary
summary(table_all$bfi_zone)

# Check plot nb
length(unique(table_all$id))
dim(unique(table_all[,c("id", "x_corr", "y_corr", "bfi_zone")]))

# Check on the map
test <- unique(table_all[,c("id", "x_corr", "y_corr", "bfi_zone")])
test %>%
  arrange(desc(bfi_zone)) %>%
  ggplot() +
  geom_point(aes(x=x_corr, y=y_corr, colour=bfi_zone, pch=bfi_zone), cex=1) +
  coord_fixed(ratio=1) +
  scale_x_continuous(limits = c(88, 93), breaks = seq(88, 93, 1)) +
  scale_y_continuous(limits = c(20, 27), breaks = seq(20, 27, 1)) +
  labs(x = NULL, y = NULL, color = "BFI zone", pch = "BFI zone")

# save plot
ggsave("results/gr_map plots with bfi zones.png", width = 15, height = 12, units = "cm", dpi = 300)

#-----#
#-----# for table data correction: nb trees bfi zone
nb_tree <- table_all[,c("id", "entity_ab", "fao_biome", "bfi_zone")]
nb_plot <- unique(nb_tree)

# Update the table nb tree/plots
nb_plot_tree <- cbind(nb_plot_tree,
  fao_bfi = paste(c(table(nb_plot$entity_ab), dim(nb_plot)[1]),
    " (",
    c(table(nb_tree$entity_ab), dim(nb_tree)[1]),
    ")",
    sep = ""))
)

nb_plot_tree
#-----#
#-----#

#-----#
#                               7. Add land cover classification system
#-----#

t_save7 <- table_all
table_all <- t_save7

```

```

# Add Lccs classes based on each project characteristics (no gps coordinates)
source("R-corrections/corr04_lccs1-6_10n0v.R")

# # Control
# summary(as.factor(table_all$lccs6))
# unique(table_all[which(is.na(table_all$lccs6)), "lu_proj"])
# unique(table_all[which(is.na(table_all$lccs1)), "id"])

# Control plots with 2 lccs classes:
test <- unique(table_all[,c("id","entity_ab", "lccs6")])
test$count <- 1
test <- aggregate(count~id, data=test, FUN=sum)
summary(test)
list <- test[test$count>1, "id"]
list

# Give NA to multi lccs class plots
table_all[which(table_all$id %in% list),c("lccs1","lccs2","lccs3","lccs4","lccs5","lccs6")] <- NA

# Plot lccs2 with non vegetated classes regrouped (factors reversed for better display)
test <- unique(table_all[!is.na(table_all$lccs6),c("id","x_corr","y_corr", "entity_ab","lccs1", "lccs2")])
test$lccs2_modif <- factor(ifelse(test$lccs1 == "A", "non_veg", test$lccs2))
levels(test$lccs2_modif) <- c("Non vegetated", "Rural settlements",
                             "(Partially) flooded vegetation", "Terrestrial vegetation")
test$rev_lccs2_modif <- with(test, factor(lccs2_modif, levels=rev(levels(lccs2_modif))))
test %>%
  arrange(desc(rev_lccs2_modif)) %>%
  ggplot() +
  geom_point(aes(x=x_corr, y=y_corr, color=rev_lccs2_modif, pch=rev_lccs2_modif)) +
  coord_fixed(ratio=1) +
  scale_x_continuous(limits = c(88, 93), breaks = seq(88, 93, 1)) +
  scale_y_continuous(limits = c(20, 27), breaks = seq(20, 27, 1)) +
  labs(x = "", y = "", color = "Land cover classification \nsystem level 2", pch = "Land cover class")

# save plot
ggsave("results/gr_map plots with lccs2.png", width = 15, height = 12, units = "cm", dpi = 300)

#####
##### Removing all plots with LCCS6 is NA
#####
table_all <- table_all[!is.na(table_all$lccs6),]

#-----#
#-----# for table data correction: nb trees reference system
nb_plot <- test
nb_tree <- table_all[which(!is.na(table_all$lccs6)),]
# Update the table nb tree/plots
nb_plot_tree <- cbind(nb_plot_tree,
                     lccs6 = paste(c(table(nb_plot$entity_ab),
                                       dim(nb_plot)[1]),
                                   " (",
                                   c(table(nb_tree$entity_ab),
                                       dim(nb_tree)[1]),
                                   ")",
                                   sep = ""))

nb_plot_tree
#-----#
#-----#

#-----#
#               4. write tables
#-----#

write.csv(table_all, "results/table_all_01a_plot_corr.csv", row.names = F)
write.csv(nb_plot_tree, "results/nb_plot_tree01a.csv", row.names = F)

```