Example application of Ely et al 2019 LMA PLSR model to existing data in EcoSIS

Shawn Serbin

2019-11-04

Overview

This is an R Markdown Notebook to illustrate how to apply the multi-biome LMA PLSR model to leaf reflectance spectra.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```
#-------------------------------------------------------------------------------------------------#
# get all required libraries
list.of.packages <- c("readr","scales","plotrix","httr","devtools")  # packages needed for script
# check for dependencies and install if needed
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
```

```
# load libraries needed for script
library(readr)    # readr - read_csv function to pull data from EcoSIS
library(plotrix)  # plotCI - to generate obsvered vs predicted plot with CIs
library(scales)   # alpha() - for applying a transparency to data points
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:plotrix':
##
##     rescale
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(httr)
library(knitr)
```

```
# define function to grab PLSR model from GitHub
#devtools::source_gist("gist.github.com/christophergandrud/4466237")
source_GitHubData <-function(url, sep = ",", header = TRUE) {
  require(httr)
  request <- GET(url)
  stop_for_status(request)
  handle <- textConnection(content(request, as = 'text'))
  on.exit(close(handle))
  read.table(handle, sep = sep, header = header)
```

```
}
#-----------------------------------------------------------------------------------------------------#
```

```
#-----------------------------------------------------------------------------------------------------#
### Set working directory (scratch space)
output_dir <- file.path("~",'scratch/')
if (! file.exists(output_dir)) dir.create(file.path(output_dir),recursive=TRUE, showWarnings = FALSE)
#setwd(file.path("~",wd)) # set working directory
opts_knit$set(root.dir = file.path(output_dir))
getwd()  # check wd
```

```
## [1] "/Users/shawnserbin/Data/GitHub/Ely_etal_2019_JournalOfExperimentalBotany/vignettes"
```

```
print(output_dir)
```

```
## [1] "~/scratch/"
```

```
#-----------------------------------------------------------------------------------------------------#
```

```
#-----------------------------------------------------------------------------------------------------#
### PLSR Coefficients - Grab from GitHub
git_repo <- "https://raw.githubusercontent.com/TESTgroup-BNL/Ely_etal_2019_JournalOfExperimentalBotany/m
print("**** Downloading PLSR coefficients ****")
```

```
## [1] "**** Downloading PLSR coefficients ****"
```

```
githubURL <- paste0(git_repo,"PLSR_model_coefficients/LMA/LMA_g_m2_PLSR_Coefficients_12comp.csv")
LeafLMA.plsr.coeffs <- source_GitHubData(githubURL)
rm(githubURL)
githubURL <- paste0(git_repo,"PLSR_model_coefficients/LMA/LMA_g_m2_Jackkife_PLSR_Coefficients.csv")
LeafLMA.plsr.jk.coeffs <- source_GitHubData(githubURL)
#-----------------------------------------------------------------------------------------------------#
```

Example datasets

URL: https://ecosis.org/package/13aef0ce-dd6f-4b35-91d9-28932e506c41 (Lopex)

URL: https://ecosis.org/package/2231d4f6-981e-4408-bf23-1b2b303f475e (Angers)

```
#-----------------------------------------------------------------------------------------------------#
### Grab data
print("**** Downloading Ecosis data ****")
```

```
## [1] "**** Downloading Ecosis data ****"
```

```
ecosis_id <- "13aef0ce-dd6f-4b35-91d9-28932e506c41"  # lopex
ecosis_file <- sprintf(
  "https://ecosis.org/api/package/%s/export?metadata=true",
  ecosis_id
)
message("Downloading data...")
```

```
## Downloading data...
```

```
dat_raw <- read_csv(ecosis_file)
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `English Name` = col_character(),
##   `Latin Name` = col_character(),
##   Measurement_type = col_character(),
##   Refl_file = col_character(),
##   Trans_file = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
message("Download complete!")
```

```
## Download complete!
```

```
# keep just fresh leaf refl obs. remove dried leaves from sample set
remove <- c(176,177,178,179,180,196,197,198,199,200,321,322,323,324,325)
remove <- which(dat_raw$Measurement_type=="transmittance" | dat_raw$`Sample_#` %in% remove)
lopex_dat_clean <- dat_raw[-remove,]
ecosis_id <- "2231d4f6-981e-4408-bf23-1b2b303f475e"  # angers
ecosis_file <- sprintf(
  "https://ecosis.org/api/package/%s/export?metadata=true",
  ecosis_id
)
message("Downloading data...")
```

```
## Downloading data...
```

```
dat_raw <- read_csv(ecosis_file)
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `English Name` = col_character(),
##   `Latin Name` = col_character(),
##   Measurement_type = col_character(),
##   Refl_file = col_character(),
##   Trans_file = col_character()
## )
## See spec(...) for full column specifications.
```

```
message("Download complete!")
```

```
## Download complete!
```

```r
# cleanup and remove dried leaves from dataset
remove <- c(178,179,184,185,196,197,241,250,254,257,258,269)
remove <- which(dat_raw$Measurement_type=="transmittance" | dat_raw$`Sample_#` %in% remove)
angers_dat_clean <- dat_raw[-remove,]
rm(dat_raw)
#-----------------------------------------------------------------------------------------------------#
```

```r
#-----------------------------------------------------------------------------------------------------#
## Concatenate data
Start.wave <- 500
End.wave <- 2400
wv <- seq(Start.wave,End.wave,1)
lopex_spectra_sub <- lopex_dat_clean[,names(lopex_dat_clean)[match(seq(Start.wave,End.wave,1),names(lop
lopex_info <- data.frame(Sample_Num=lopex_dat_clean$`Sample_#`, Common_Species_Name=lopex_dat_clean$`Eng
                         LMA_gDW_m2=(lopex_dat_clean$`Leaf mass per area (g/cm²)`)*10000)
angers_spectra_sub <- angers_dat_clean[,names(angers_dat_clean)[match(seq(Start.wave,End.wave,1),names(a
angers_spectra_sub <- na.omit(angers_spectra_sub)
angers_info <- data.frame(Sample_Num=angers_dat_clean$`Sample_#`, Common_Species_Name=angers_dat_clean$
                          LMA_gDW_m2=(angers_dat_clean$`Leaf mass per area (g/cm )`)*10000)
angers_info <- na.omit(angers_info)
all_data <- rbind(data.frame(Dataset=rep("Lopex",dim(lopex_info)[1]), lopex_info, lopex_spectra_sub),
                  data.frame(Dataset=rep("Angers",dim(angers_info)[1]), angers_info, angers_spectra_sub
## cleanup
rm(angers_dat_clean,lopex_dat_clean, angers_info, angers_spectra_sub, lopex_info, lopex_spectra_sub)
#-----------------------------------------------------------------------------------------------------#
```

```r
#-----------------------------------------------------------------------------------------------------#
## Plot data
waves <- seq(500,2400,1)
cexaxis <- 1.5
cexlab <- 1.8
ylim <- 74
ylim2 <- 80
mean_spec <- colMeans(all_data[,which(names(all_data) %in% paste0("X",seq(Start.wave,End.wave,1)))])
spectra_quantiles <- apply(all_data[,which(names(all_data) %in% paste0("X",seq(Start.wave,End.wave,1)))
                           2,quantile,na.rm=T,probs=c(0,0.025,0.05,0.5,0.95,0.975,1))
print("**** Plotting Ecosis data. Writing to scratch space ****")
```

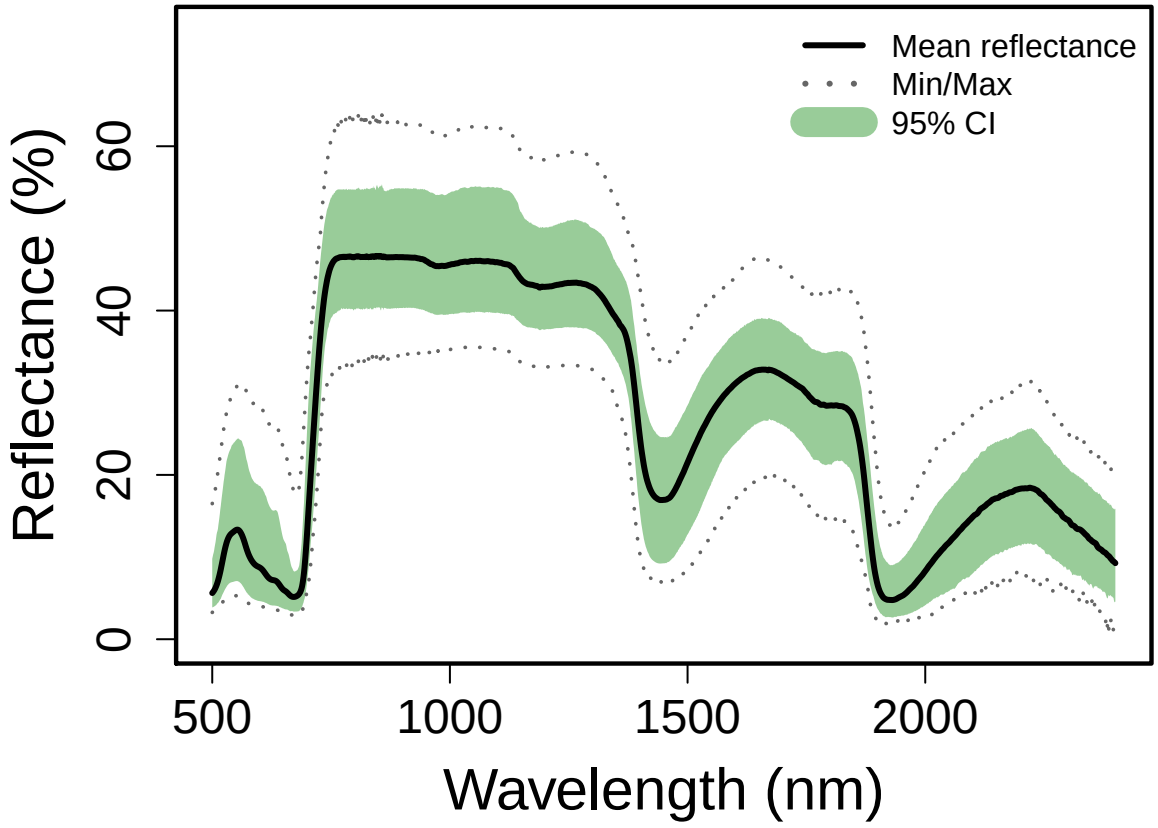
```
## [1] "**** Plotting Ecosis data. Writing to scratch space ****"
```

```r
png(file=file.path(output_dir,'Angers_Lopex_spectra_summary_plot.png'),height=3000,
    width=3900, res=340)
par(mfrow=c(1,1), mar=c(4.5,5.7,0.3,0.4), oma=c(0.3,0.9,0.3,0.1)) # B, L, T, R
plot(waves,mean_spec*100,ylim=c(0,ylim),cex=0.00001, col="white",xlab="Wavelength (nm)",
     ylab="Reflectance (%)",cex.axis=cexaxis, cex.lab=cexlab)
polygon(c(waves ,rev(waves)),c(spectra_quantiles[6,]*100, rev(spectra_quantiles[2,]*100)),
        col="#99CC99",border=NA)
lines(waves,mean_spec*100,lwd=3, lty=1, col="black")
lines(waves,spectra_quantiles[1,]*100,lwd=1.85, lty=3, col="grey40")
lines(waves,spectra_quantiles[7,]*100,lwd=1.85, lty=3, col="grey40")
legend("topright",legend=c("Mean reflectance","Min/Max", "95% CI"),lty=c(1,3,1),
       lwd=c(3,3,15),col=c("black","grey40","#99CC99"),bty="n", cex=1.7)
box(lwd=2.2)
dev.off()
```

```
## pdf
```

```
##   2
```

```
par(mfrow=c(1,1), mar=c(4.5,5.7,0.3,0.4), oma=c(0.3,0.9,0.3,0.1)) # B, L, T, R
plot(waves,mean_spec*100,ylim=c(0,ylim),cex=0.00001, col="white",xlab="Wavelength (nm)",
     ylab="Reflectance (%)",cex.axis=cexaxis, cex.lab=cexlab)
polygon(c(waves ,rev(waves)),c(spectra_quantiles[6,]*100, rev(spectra_quantiles[2,]*100)),
        col="#99CC99",border=NA)
lines(waves,mean_spec*100,lwd=3, lty=1, col="black")
lines(waves,spectra_quantiles[1,]*100,lwd=1.85, lty=3, col="grey40")
lines(waves,spectra_quantiles[7,]*100,lwd=1.85, lty=3, col="grey40")
legend("topright",legend=c("Mean reflectance","Min/Max", "95% CI"),lty=c(1,3,1),
       lwd=c(3,3,15),col=c("black","grey40","#99CC99"),bty="n", cex=1)
box(lwd=2.2)
```




```
#-------------------------------------------------------------------------------------------------#
```

```
#-------------------------------------------------------------------------------------------------#
print("**** Applying PLSR model to estimate LMA from spectral observations ****")
```

```
## [1] "**** Applying PLSR model to estimate LMA from spectral observations ****"
```

```
# setup model
dims <- dim(LeafLMA.plsr.coeffs)
LeafLMA.plsr.intercept <- LeafLMA.plsr.coeffs[1,]
LeafLMA.plsr.coeffs <- data.frame(LeafLMA.plsr.coeffs[2:dims[1],])
names(LeafLMA.plsr.coeffs) <- c("wavelength","coefs")
LeafLMA.plsr.coeffs.vec <- as.vector(LeafLMA.plsr.coeffs[,2])

# estimate LMA
sub_spec <- as.matrix(droplevels(all_data[,which(names(all_data) %in% paste0("X",seq(Start.wave,End.wave
```

```
temp <- as.matrix(sub_spec) %*% LeafLMA.plsr.coeffs.vec
leafLMA <- data.frame(rowSums(temp))+LeafLMA.plsr.intercept[,2]
leafLMA <- leafLMA[,1]  # convert to standard LMA units from sqrt(LMA)
names(leafLMA) <- "FS_PLSR_LMA_gDW_m2"

# organize output
LeafLMA.PLSR.dataset <- data.frame(all_data[,c(1:4)],FS_PLSR_LMA_gDW_m2=leafLMA)
# Derive LMA estimate uncertainties
print("**** Deriving uncertainty estimates ****")
```

```
## [1] "**** Deriving uncertainty estimates ****"
```

```
dims <- dim(LeafLMA.plsr.jk.coeffs)
intercepts <- LeafLMA.plsr.jk.coeffs[,2]
jk.leaf.lma.est <- array(data=NA,dim=c(dim(sub_spec)[1],dims[1]))
for (i in 1:length(intercepts)){
  coefs <- unlist(as.vector(LeafLMA.plsr.jk.coeffs[i,3:dims[2]]))
  temp <- sub_spec %*% coefs
  values <- data.frame(rowSums(temp))+intercepts[i]
  jk.leaf.lma.est[,i] <- values[,1]
  rm(temp)
}

jk.leaf.lma.est.quant <- apply(jk.leaf.lma.est,1,quantile,probs=c(0.025,0.975))
jk.leaf.lma.est.quant2 <- data.frame(t(jk.leaf.lma.est.quant))
names(jk.leaf.lma.est.quant2) <- c("FS_PLSR_Leaf_LMA_L5","FS_PLSR_Leaf_LMA_U95")
jk.leaf.lma.est.sd <- apply(jk.leaf.lma.est,1,sd)
names(jk.leaf.lma.est.sd) <- "FS_PLSR_Leaf_LMA_Sdev"

## Combine into final dataset
stats <- data.frame(jk.leaf.lma.est.sd,jk.leaf.lma.est.quant2)
names(stats) <- c("FS_PLSR_Leaf_LMA_Sdev","FS_PLSR_Leaf_LMA_L5","FS_PLSR_Leaf_LMA_U95")
LeafLMA.PLSR.dataset.out <- data.frame(LeafLMA.PLSR.dataset,stats,
                                       residual=(LeafLMA.PLSR.dataset$FS_PLSR_LMA_gDW_m2-LeafLMA.PLSR.da
# output results
write.csv(x = LeafLMA.PLSR.dataset.out, file = file.path(output_dir,"Angers_Lopex_PLSR_estimated_LMA_da
          row.names = F)
# calculate error stats
rmse <- sqrt(mean(LeafLMA.PLSR.dataset.out$residual^2))
# calculate fit stats
reg <- lm(LeafLMA.PLSR.dataset.out$FS_PLSR_LMA_gDW_m2~LeafLMA.PLSR.dataset.out$LMA_gDW_m2)
#-------------------------------------------------------------------------------------------------#
```

```
#-------------------------------------------------------------------------------------------------#
## Plot up results
ptcex <- 1.8
cexaxis <- 1.3
cexlab <- 1.8
print("**** Plotting Lopex/Angers LMA validation plot. Writing to scratch space ****")
```

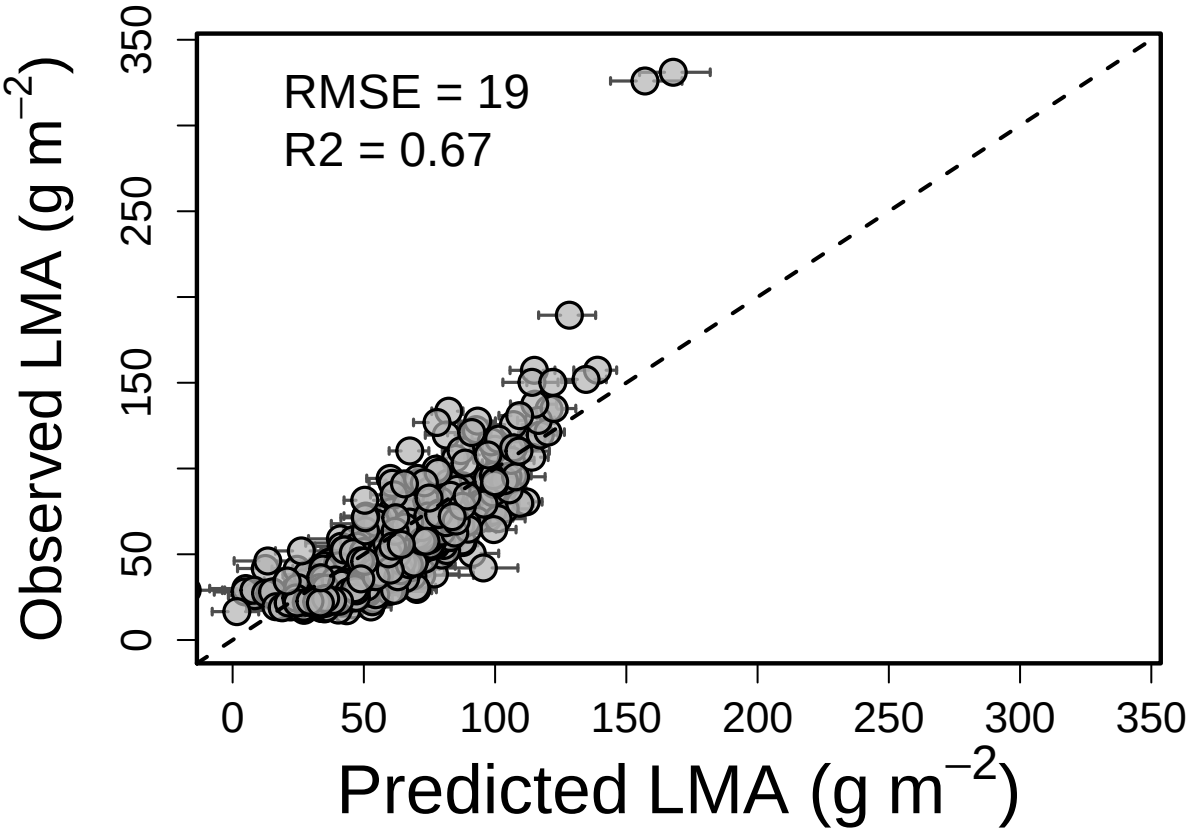
```
## [1] "**** Plotting Lopex/Angers LMA validation plot. Writing to scratch space ****"
```

```
png(file=file.path(output_dir,'Angers_Lopex_LMA_validation_plot.png'),height=3000,
    width=3900, res=340)
par(mfrow=c(1,1), mar=c(4.5,5.4,1,1), oma=c(0.3,0.9,0.3,0.1)) # B, L, T, R
```

```
plotCI(LeafLMA.PLSR.dataset.out$FS_PLSR_LMA_gDW_m2,LeafLMA.PLSR.dataset.out$LMA_gDW_m2,
       li=LeafLMA.PLSR.dataset.out$FS_PLSR_Leaf_LMA_L5,gap=0.009,sfrac=0.004,lwd=1.6,
       ui=LeafLMA.PLSR.dataset.out$FS_PLSR_Leaf_LMA_U95,err="x",pch=21,col="black",
       pt.bg=alpha("grey70",0.7),scol="grey30",xlim=c(0,340),cex=ptcex,
       ylim=c(0,340),xlab="",
       ylab=expression(paste("Observed LMA (",g~m^{-2},")")),main="",
       cex.axis=cexaxis,cex.lab=cexlab)
mtext(side = 1, text = expression(paste(Predicted~LMA," (",g~m^{-2},")")), line = 3.5,
      cex=2.2)
abline(0,1,lty=2,lw=2)
legend("topleft",legend = c(paste0("RMSE = ",round(rmse)),
                            paste0("R2 = ",round(summary(reg)$r.squared,2))), bty="n", cex=1.5)
box(lwd=2.2)
dev.off()
```

```
## pdf
##   2
```

```
par(mfrow=c(1,1), mar=c(4.5,5.4,1,1), oma=c(0.3,0.9,0.3,0.1)) # B, L, T, R
plotCI(LeafLMA.PLSR.dataset.out$FS_PLSR_LMA_gDW_m2,LeafLMA.PLSR.dataset.out$LMA_gDW_m2,
       li=LeafLMA.PLSR.dataset.out$FS_PLSR_Leaf_LMA_L5,gap=0.009,sfrac=0.004,lwd=1.6,
       ui=LeafLMA.PLSR.dataset.out$FS_PLSR_Leaf_LMA_U95,err="x",pch=21,col="black",
       pt.bg=alpha("grey70",0.7),scol="grey30",xlim=c(0,340),cex=ptcex,
       ylim=c(0,340),xlab="",
       ylab=expression(paste("Observed LMA (",g~m^{-2},")")),main="",
       cex.axis=cexaxis,cex.lab=cexlab)
mtext(side = 1, text = expression(paste(Predicted~LMA," (",g~m^{-2},")")), line = 3.5,
      cex=2.2)
abline(0,1,lty=2,lw=2)
legend("topleft",legend = c(paste0("RMSE = ",round(rmse)),
                            paste0("R2 = ",round(summary(reg)$r.squared,2))), bty="n", cex=1.5)
box(lwd=2.2)
```




```
#-------------------------------------------------------------------------------------------------#
```

```
#-------------------------------------------------------------------------------------------------#
rm(list=ls(all=TRUE))    # clear workspace
### EOF
```