A basic PLSR example using leaf-level specrta and leaf mass per area (LMA) data from several CONUS NEON sites

Shawn P. Serbin

Overview

his is an R Markdown Notebook to illustrate how to conduct a basic model fit. This example shows you how to retrieve a dataset from the EcoSIS spectral database, choose the "optimal" number of plsr components, and fit a plsr model for leaf-mass area

When you click the **Knit** button in Rstudio a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

Getting Started

Installation

```
list.of.packages <- c("readr","httr","pls","dplyr","reshape2")  # 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

```
# load libraries needed for script
library(pls)
library(readr)
library(dplyr)
library(reshape2)
library(ggplot2)
library(dplyr)
```

Prepare helpers

```
# 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)
}
```

```
# not in
`%notin%` <- Negate(`%in%`)


# Script options
pls.options(plsralg = "oscorespls")
pls.options("plsralg")
```

$plsralg [1] "oscorespls"

```
pls.options()$parallel
```

NULL

```
# NULL
```

Setup temporary folder

```
print(paste0("Output temporary directory: ",outdir))
```

[1] "Output temporary directory: /var/folders/xp/h3k9vf3n2jx181ts786__yjrn9c2gjq/T//RtmpkzZB8m"

```
setwd(outdir) # set working directory
getwd()  # check wd
```

[1] "/private/var/folders/xp/h3k9vf3n2jx181ts786__yjrn9c2gjq/T/RtmpkzZB8m"

Grab data from EcoSIS

URL: https://ecosis.org/package/fresh-leaf-spectra-to-estimate-lma-over-neon-domains-in-eastern-united-states

```
print("**** Downloading Ecosis data ****")
```

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

```
ecosis_id <- "5617da17-c925-49fb-b395-45a51291bd2d"  # NEON dataset
ecosis_file <- sprintf(
  "https://ecosis.org/api/package/%s/export?metadata=true",
  ecosis_id
)
message("Downloading data...")
dat_raw <- read_csv(ecosis_file)
message("Download complete!")
head(dat_raw)
```

A tibble: 6 x 2,162

```
Affiliation Common Name Domain Functional_type LMA Latin Genus
1 University~ black walnut D02 broadleaf 72.9 Juglans
2 University~ black walnut D02 broadleaf 72.9 Juglans
3 University~ black walnut D02 broadleaf 60.8 Juglans
4 University~ black walnut D02 broadleaf 60.8 Juglans
5 University~ black walnut D02 broadleaf 85.9 Juglans
```

6 University~ black walnut D02 broadleaf 85.9 Juglans
# … with 2,156 more variables: `Latin Species` , PI , # Project , Sample_ID , `USDA Symbol` , `350` , # `351` , `352` , `353` , `354` , `355` , # `356` , `357` , `358` , `359` , `360` , # `361` , `362` , `363` , `364` , `365` , # `366` , `367` , `368` , `369` , `370` , # `371` , `372` , `373` , `374` , `375` , # `376` , `377` , `378` , `379` , `380` , # `381` , `382` , `383` , `384` , `385` , # `386` , `387` , `388` , `389` , `390` , # `391` , `392` , `393` , `394` , `395` , # `396` , `397` , `398` , `399` , `400` , # `401` , `402` , `403` , `404` , `405` , # `406` , `407` , `408` , `409` , `410` , # `411` , `412` , `413` , `414` , `415` , # `416` , `417` , `418` , `419` , `420` , # `421` , `422` , `423` , `424` , `425` , # `426` , `427` , `428` , `429` , `430` , # `431` , `432` , `433` , `434` , `435` , # `436` , `437` , `438` , `439` , `440` , # `441` , `442` , `443` , `444` , …

```
names(dat_raw)[1:40]
```

[1] "Affiliation" "Common Name" "Domain" "Functional_type" [5] "LMA" "Latin Genus" "Latin Species" "PI"
[9] "Project" "Sample_ID" "USDA Symbol" "350"
[13] "351" "352" "353" "354"
[17] "355" "356" "357" "358"
[21] "359" "360" "361" "362"
[25] "363" "364" "365" "366"
[29] "367" "368" "369" "370"
[33] "371" "372" "373" "374"
[37] "375" "376" "377" "378"

Create PLSR dataset

```
Start.wave <- 500
End.wave <- 2400
wv <- seq(Start.wave,End.wave,1)

spectra <- data.frame(dat_raw[,names(dat_raw) %in% wv])
names(spectra) <- c(paste0("Wave_",wv))
head(spectra)[1:6,1:10]
```

Wave_500 Wave_501 Wave_502 Wave_503 Wave_504 Wave_505 Wave_506 Wave_507 1 0.044226 0.044605 0.044927 0.045473 0.046241 0.046878 0.047826 0.049090 2 0.046855 0.047601 0.047944 0.048478 0.049381 0.050235 0.051161 0.052191 3 0.043758 0.044171 0.044869 0.045465 0.045984 0.046933 0.047993 0.049090 4 0.041154 0.041603 0.042088 0.042408 0.042639 0.043260 0.044140 0.045058 5 0.037296 0.037944 0.038209 0.038677 0.039388 0.039948 0.040630 0.041501 6 0.043878 0.044257 0.044723 0.045295 0.045949 0.046575 0.047378 0.048357 Wave_508 Wave_509 1 0.050268 0.051525 2 0.053322 0.054357 3 0.050168 0.051441 4 0.045700 0.046476 5 0.042613 0.043731 6 0.049392 0.050387

```
sample_info <- dat_raw[,names(dat_raw) %notin% seq(350,2500,1)]
head(sample_info)
```

A tibble: 6 x 11

Affiliation `Common Name` Domain Functional_type LMA `Latin Genus`
1 University~ black walnut D02 broadleaf 72.9 Juglans
2 University~ black walnut D02 broadleaf 72.9 Juglans
3 University~ black walnut D02 broadleaf 60.8 Juglans
4 University~ black walnut D02 broadleaf 60.8 Juglans
5 University~ black walnut D02 broadleaf 85.9 Juglans
6 University~ black walnut D02 broadleaf 85.9 Juglans
# … with 5 more variables: `Latin Species` , PI , Project , # Sample_ID , `USDA Symbol`

```
sample_info2 <- sample_info %>%
  select(Domain,Functional_type,Sample_ID,USDA_Species_Code=`USDA Symbol`,LMA_gDW_m2=LMA)
head(sample_info2)
```

A tibble: 6 x 5

Domain Functional_type Sample_ID USDA_Species_Code LMA_gDW_m2 1 D02 broadleaf P0001 JUNI 72.9 2 D02 broadleaf L0001 JUNI 72.9 3 D02 broadleaf P0002 JUNI 60.8 4 D02 broadleaf L0002 JUNI 60.8 5 D02 broadleaf P0003 JUNI 85.9 6 D02 broadleaf L0003 JUNI 85.9

```
plsr_data <- data.frame(sample_info2,Spectra=I(as.matrix(spectra)))

inVar <- "LMA_gDW_m2"
```

Plot the spectra

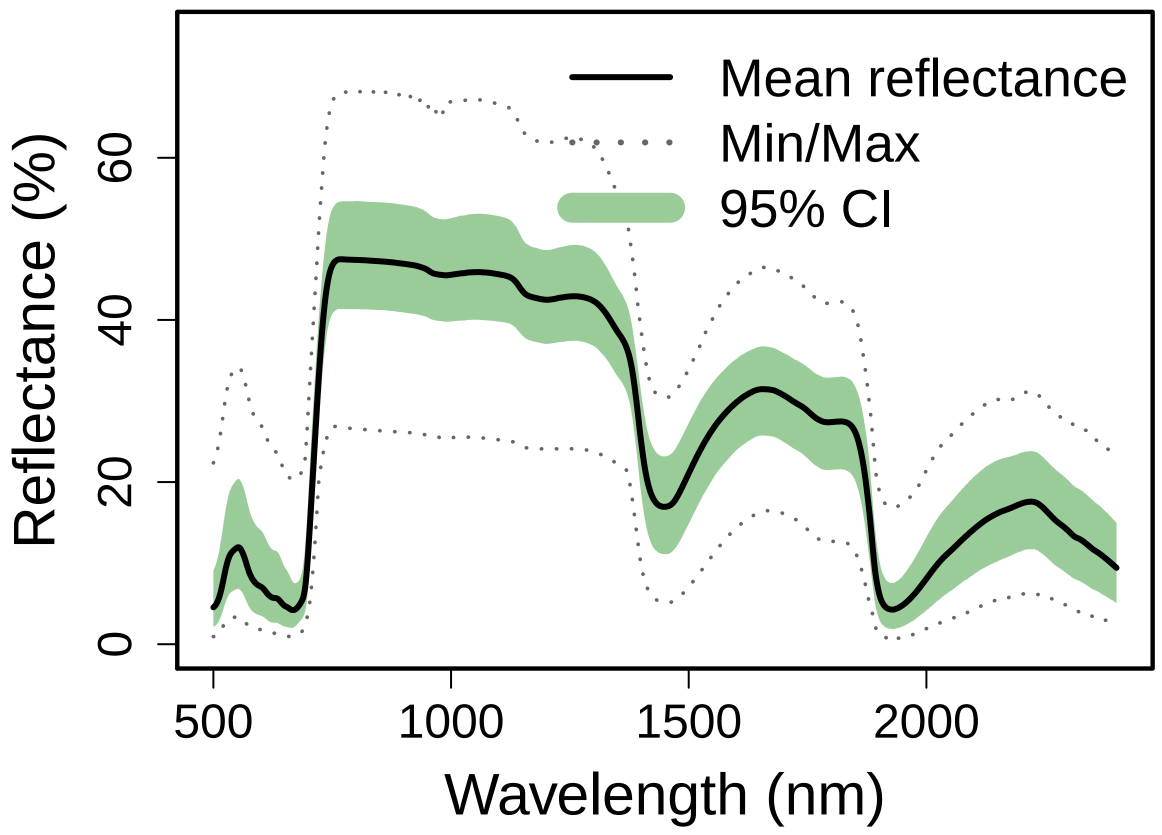
```
cexaxis <- 1.5
cexlab <- 1.8
ylim <- 75

mean_spec <- colMeans(spectra[,which(names(spectra) %in% paste0("Wave_",wv))])
spectra_quantiles <- apply(spectra[,which(names(spectra) %in% paste0("Wave_",wv))],
                           2,quantile,na.rm=T,probs=c(0,0.025,0.05,0.5,0.95,0.975,1))

print("**** Plotting specrtal data ****")
```

[1] "**** Plotting specrtal data ****"

```
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(wv,mean_spec,ylim=c(0,ylim),cex=0.00001, col="white",xlab="Wavelength (nm)",
     ylab="Reflectance (%)",cex.axis=cexaxis, cex.lab=cexlab)
polygon(c(wv ,rev(wv)),c(spectra_quantiles[5,]*100, rev(spectra_quantiles[3,]*100)),
        col="#99CC99",border=NA)
lines(wv,mean_spec*100,lwd=3, lty=1, col="black")
lines(wv,spectra_quantiles[1,]*100,lwd=1.85, lty=3, col="grey40")
lines(wv,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)
```



Run Jackknife test to find number of components - simple example

```
dims <- dim(plsr_data)
nComps <- 20
iterations <- 20
seg <- 5
prop <- 0.70
jk.out <- matrix(data=NA,nrow=iterations,ncol=nComps)
pls.options(parallel = parallel::detectCores()-1) # Use mclapply
print("*** Running jacknife permutation test.  Please hang tight, this can take awhile ***")
```

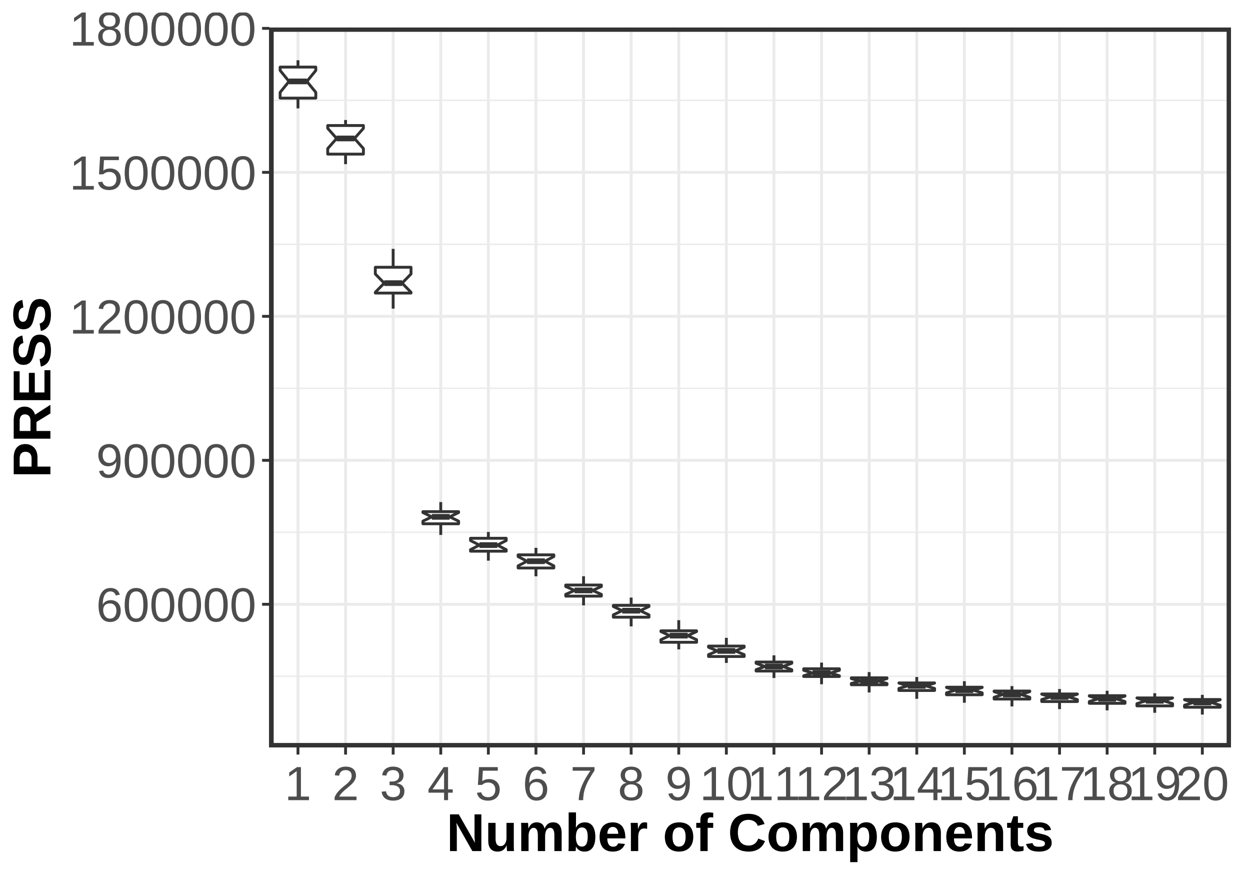
[1] "*** Running jacknife permutation test. Please hang tight, this can take awhile ***"

```
start.time <- Sys.time()
for (i in 1:iterations) {
  rows <- sample(1:nrow(plsr_data),floor(prop*nrow(plsr_data)))
  sub.data <- plsr_data[rows,]
  plsr.out <- plsr(as.formula(paste(inVar,"~","Spectra")), scale=FALSE, center=TRUE, ncomp=nComps,
                  validation="CV", segments = seg, segment.type="interleaved", trace=FALSE, data=sub.da
  resPRESS <- as.vector(plsr.out$validation$PRESS)
  jk.out[i,seq(plsr.out$validation$ncomp)]=resPRESS
}
end.time <- Sys.time()
end.time - start.time
```

Time difference of 1.753883 mins

PRESS plot

```r
pressDF <- as.data.frame(jk.out)
names(pressDF) <- as.character(seq(nComps))
pressDFres <- melt(pressDF)
bp <- ggplot(pressDFres, aes(x=variable, y=value)) + theme_bw() +
  geom_boxplot(notch=TRUE) + labs(x="Number of Components", y="PRESS") +
  theme(axis.text=element_text(size=18), legend.position="none",
        axis.title=element_text(size=20, face="bold"),
        axis.text.x = element_text(angle = 0,vjust = 0.5),
        panel.border = element_rect(linetype = "solid", fill = NA, size=1.5))
bp
```



Calculate optimal number of components

```r
# conduct t.test across components to identify first minimum - just one of the ways to do this
j <-2
results <- as.vector(array(data="NA", dim=c(nComps-1,1)))
for (i in seq_along(1:nComps-1)) {
  comp1 <- i; comp2 <- j
  ttest <- t.test(pressDFres$value[which(pressDFres$variable==comp1)],
                  pressDFres$value[which(pressDFres$variable==comp2)])
  #print(i)
  results[i] <- round(unlist(ttest$p.value),8)
  j <- j+1
  if (j > nComps) {
    break
  }
}
results <- data.frame(seq(2,nComps,1),results)
names(results) <- c("Component", "P.value")
results
```

Component P.value 1 2 0 2 3 0 3 4 0 4 5 0 5 6 5.3e-07 6 7 0 7 8 0 8 9 0 9 10 1.3e-06 10 11 3e-08 11 12 0.0026535 12 13 3.129e-05 13 14 0.00918502 14 15 0.02547395 15 16 0.03095163 16 17 0.12853979 17 18 0.35387817 18 19 0.21422969 19 20 0.41859186

```
# *** based on t.test - optimal components are 16 ***
# NOTE: Becuase the jacknife test above depends on random selection
# the optimal components may change slightly between differe runs of this script
# This is expected given different permutatoins and uncertainty in the data
```

Final PLSR model fit

```
# Simple final model validated with cross-validation.  Segmented cross-validation used
# given the very large sample size.  For models with fewer observations (e.g. <100)
# LOO or leave-one-out cross validation is recommended

#nComps <- 14
first <- min(which(as.numeric(as.character(results$P.value)) > 0.05))
nComps <- results$Component[first]
print(paste0("*** Optimal number of components based on t.test: ", nComps))
```

[1] "*** Optimal number of components based on t.test: 17"

```
segs <- 30
pls.options(parallel = NULL)
plsr.out <- plsr(as.formula(paste(inVar,"~","Spectra")),scale=FALSE,ncomp=nComps,validation="CV",
                 segments=segs, segment.type="interleaved",trace=TRUE,data=plsr_data)
```

Segment: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

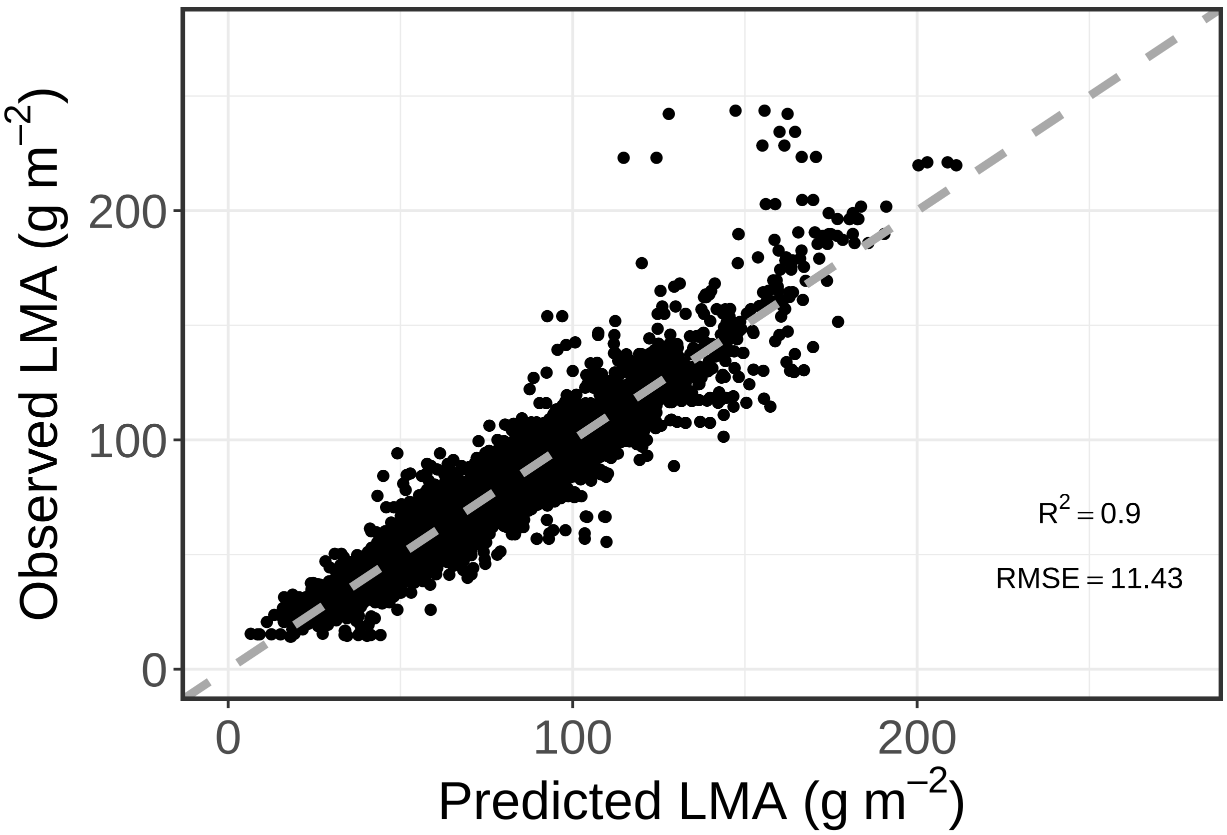
```
fit <- plsr.out$fitted.values[,1,nComps]

plot_data <- data.frame(plsr_data[, which(names(plsr_data) %notin% "Spectra")], Fitted=fit)
plot_data <- plot_data %>%
  mutate(Residuals = Fitted-LMA_gDW_m2)
head(plot_data)
```

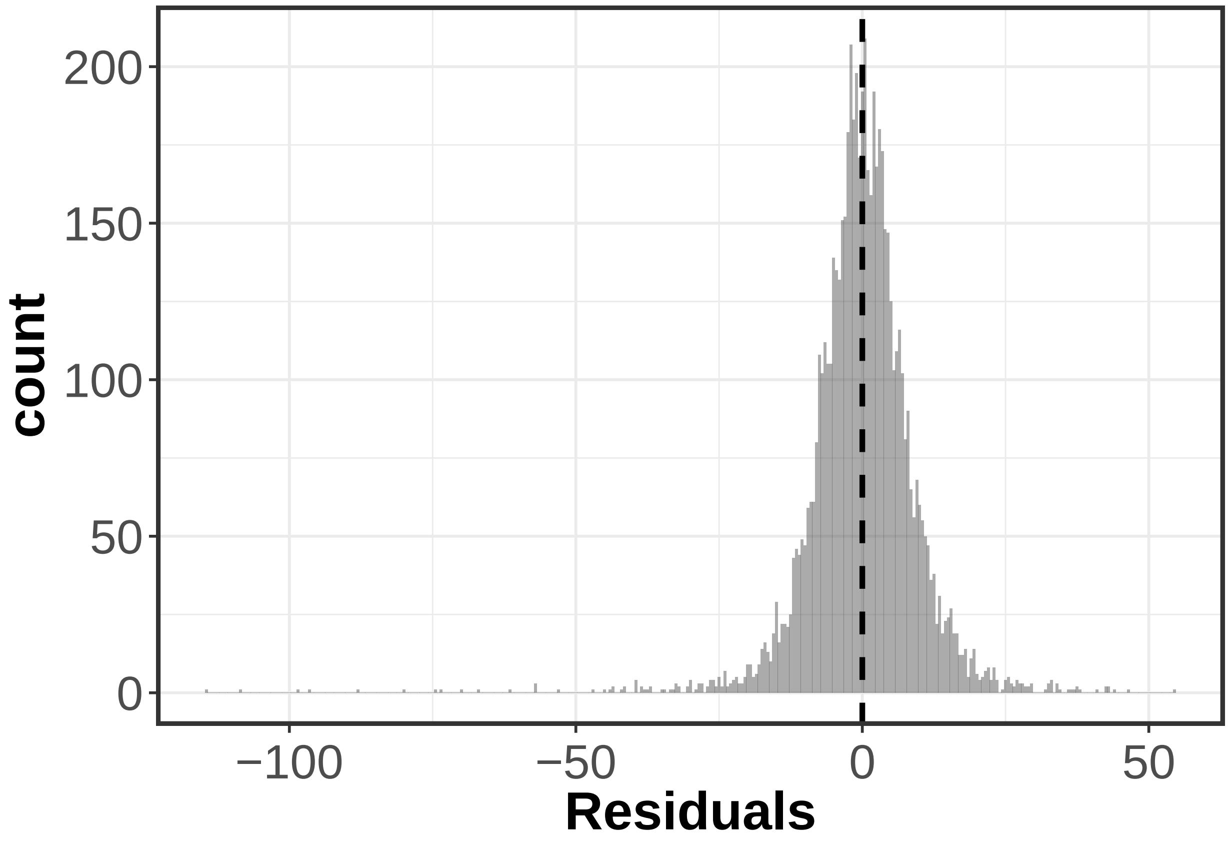
Domain Functional_type Sample_ID USDA_Species_Code LMA_gDW_m2 Fitted 1 D02 broadleaf P0001 JUNI 72.87 74.66709 2 D02 broadleaf L0001 JUNI 72.87 80.85354 3 D02 broadleaf P0002 JUNI 60.77 63.70678 4 D02 broadleaf L0002 JUNI 60.77 53.18834 5 D02 broadleaf P0003 JUNI 85.92 83.66721 6 D02 broadleaf L0003 JUNI 85.92 85.28082 Residuals 1 1.7970935 2 7.9835431 3 2.9367822 4 -7.5816605 5 -2.2527884 6 -0.6391764

Basic scatter plot of results

```
scatter_plot <- ggplot(plot_data, aes(x=Fitted, y=LMA_gDW_m2)) +
  theme_bw() + geom_point() + geom_abline(intercept = 0, slope = 1, color="dark grey",
            linetype="dashed", size=1.5) + xlim(0, 275) + ylim(0, 275) +
  labs(x=expression(paste("Predicted LMA (",g~m^{-2},")")),
       y=expression(paste("Observed LMA (",g~m^{-2},")"))) +
  annotate("text", x=250, y=70, label = paste0("R^2 == ", round(pls::R2(plsr.out)[[1]][nComps],2)), pars
  annotate("text", x=250, y=40, label = paste0("RMSE == ", round(pls::RMSEP(plsr.out)[[1]][nComps],2)),
  theme(axis.text=element_text(size=18), legend.position="none",
        axis.title=element_text(size=20, face="bold"),
        axis.text.x = element_text(angle = 0,vjust = 0.5),
        panel.border = element_rect(linetype = "solid", fill = NA, size=1.5))
scatter_plot
```

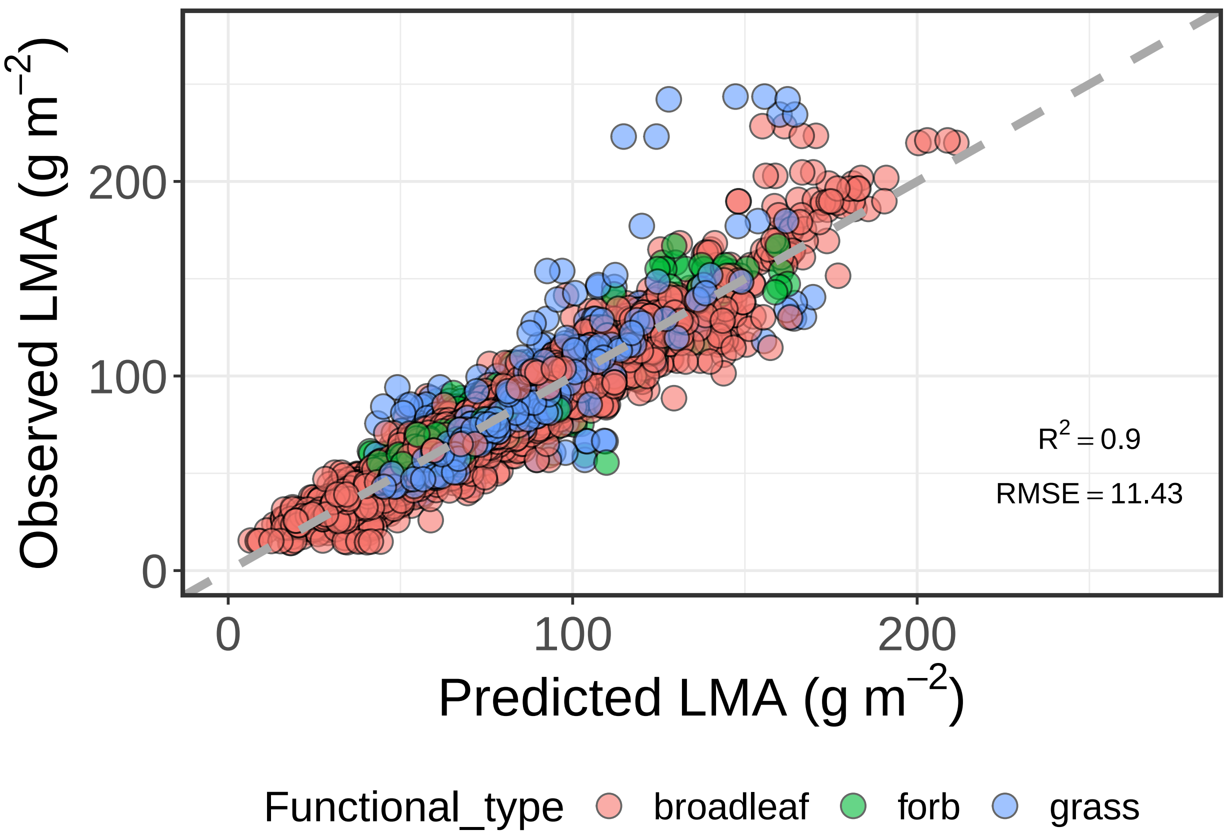


```
resid_histogram <- ggplot(plot_data, aes(x=Residuals)) +
  geom_histogram(binwidth=.5, alpha=.5, position="identity") +
  geom_vline(xintercept = 0, color="black",
             linetype="dashed", size=1) + theme_bw() +
  theme(axis.text=element_text(size=18), legend.position="none",
        axis.title=element_text(size=20, face="bold"),
        axis.text.x = element_text(angle = 0,vjust = 0.5),
        panel.border = element_rect(linetype = "solid", fill = NA, size=1.5))
resid_histogram
```

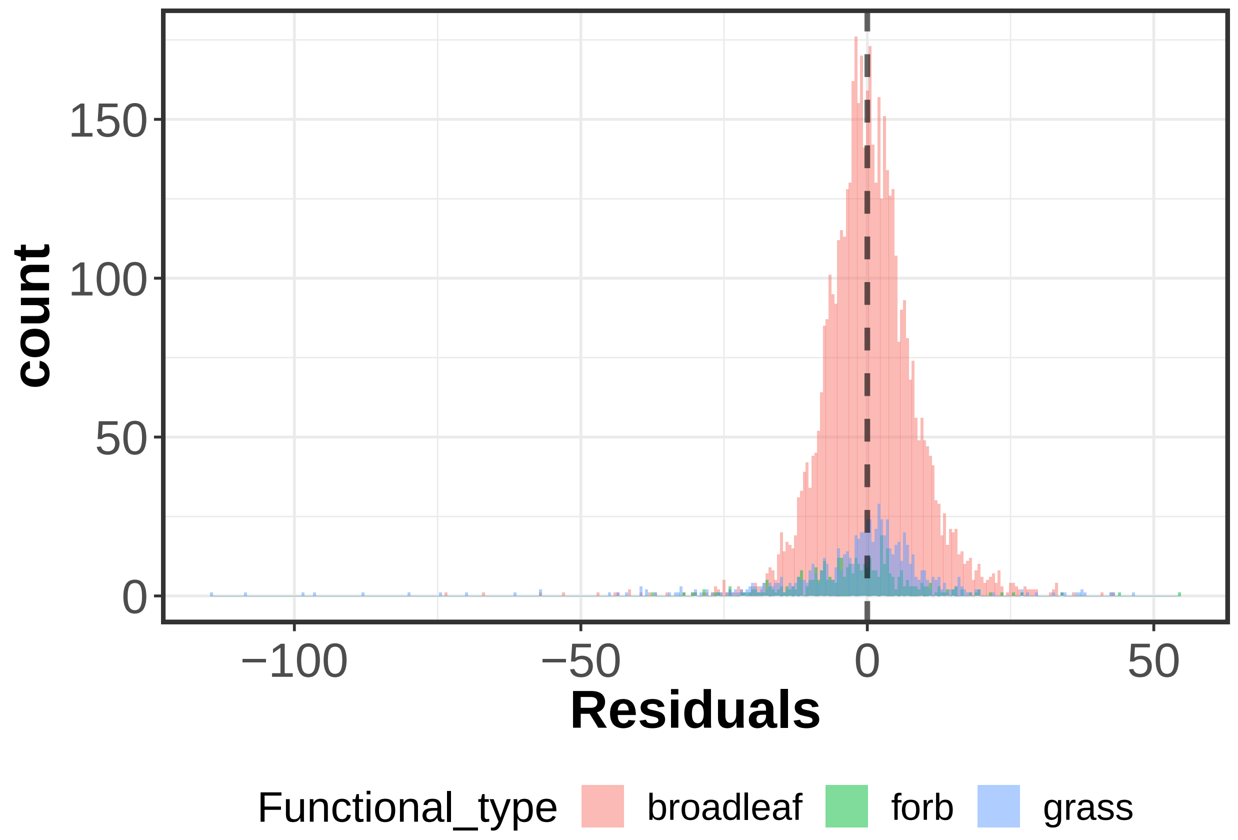


Scatter plot by Functional_type

```
scatter_plot <- ggplot(plot_data, aes(x=Fitted, y=LMA_gDW_m2)) +
  theme_bw() + geom_point(aes(fill=Functional_type),alpha=0.6,colour="black", pch=21, size=4) +
  geom_abline(intercept = 0, slope = 1, color="dark grey",
              linetype="dashed", size=1.5) + xlim(0, 275) + ylim(0, 275) +
  labs(x=expression(paste("Predicted LMA (",g~m^{-2},")")),
       y=expression(paste("Observed LMA (",g~m^{-2},")"))) +
  annotate("text", x=250, y=70, label = paste0("R^2 == ", round(pls::R2(plsr.out)[[1]][nComps],2)), par
  annotate("text", x=250, y=40, label = paste0("RMSE == ", round(pls::RMSEP(plsr.out)[[1]][nComps],2)),
  theme(axis.text=element_text(size=18), legend.position="bottom",legend.title=element_text(size=16),
        legend.text=element_text(size=14),
        axis.title=element_text(size=20, face="bold"),
        axis.text.x = element_text(angle = 0,vjust = 0.5),
        panel.border = element_rect(linetype = "solid", fill = NA, size=1.5))
scatter_plot
```

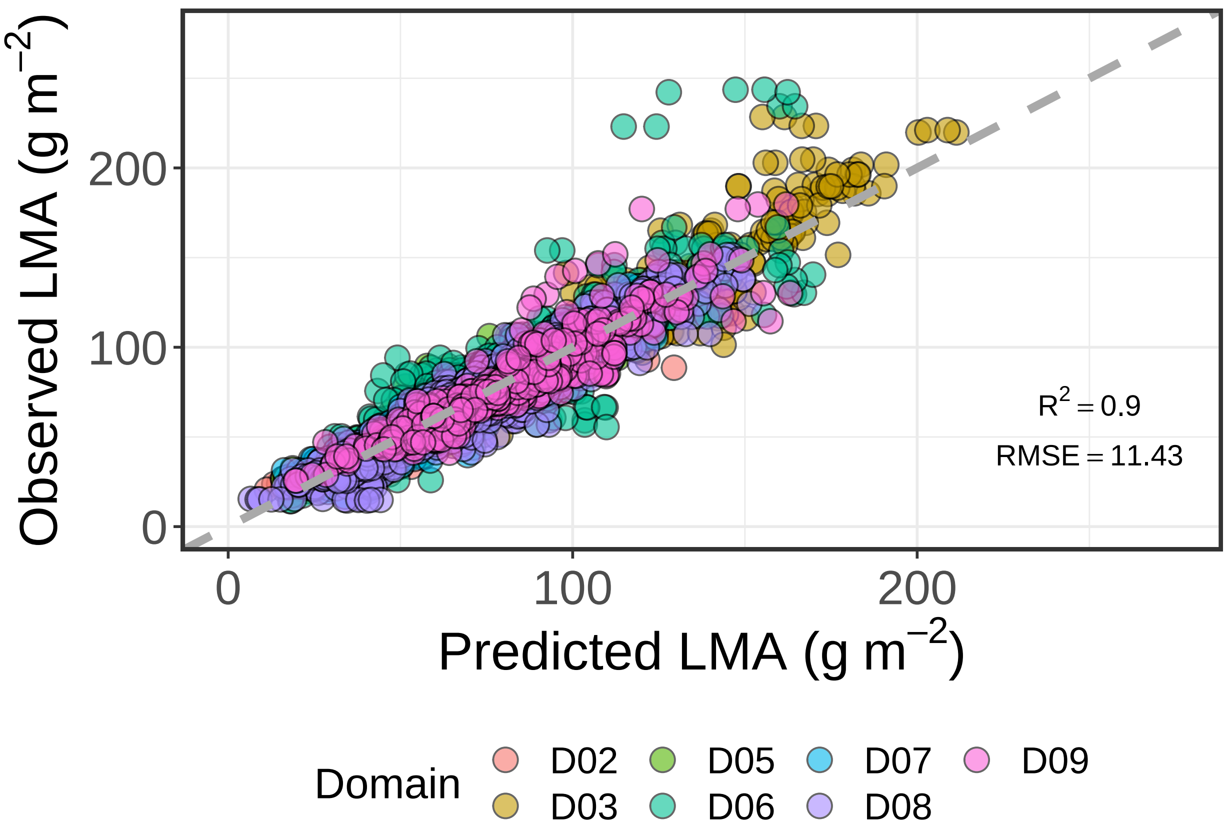


```
resid_histogram <- ggplot(plot_data, aes(x=Residuals, fill=Functional_type)) +
  geom_histogram(binwidth=.5, alpha=.5, position="identity") +
  geom_vline(xintercept = 0, color="black", alpha=0.6,
             linetype="dashed", size=1) + theme_bw() +
  theme(axis.text=element_text(size=18), legend.position="bottom",legend.title=element_text(size=16),
        legend.text=element_text(size=14),
        axis.title=element_text(size=20, face="bold"),
        axis.text.x = element_text(angle = 0,vjust = 0.5),
        panel.border = element_rect(linetype = "solid", fill = NA, size=1.5))
resid_histogram
```

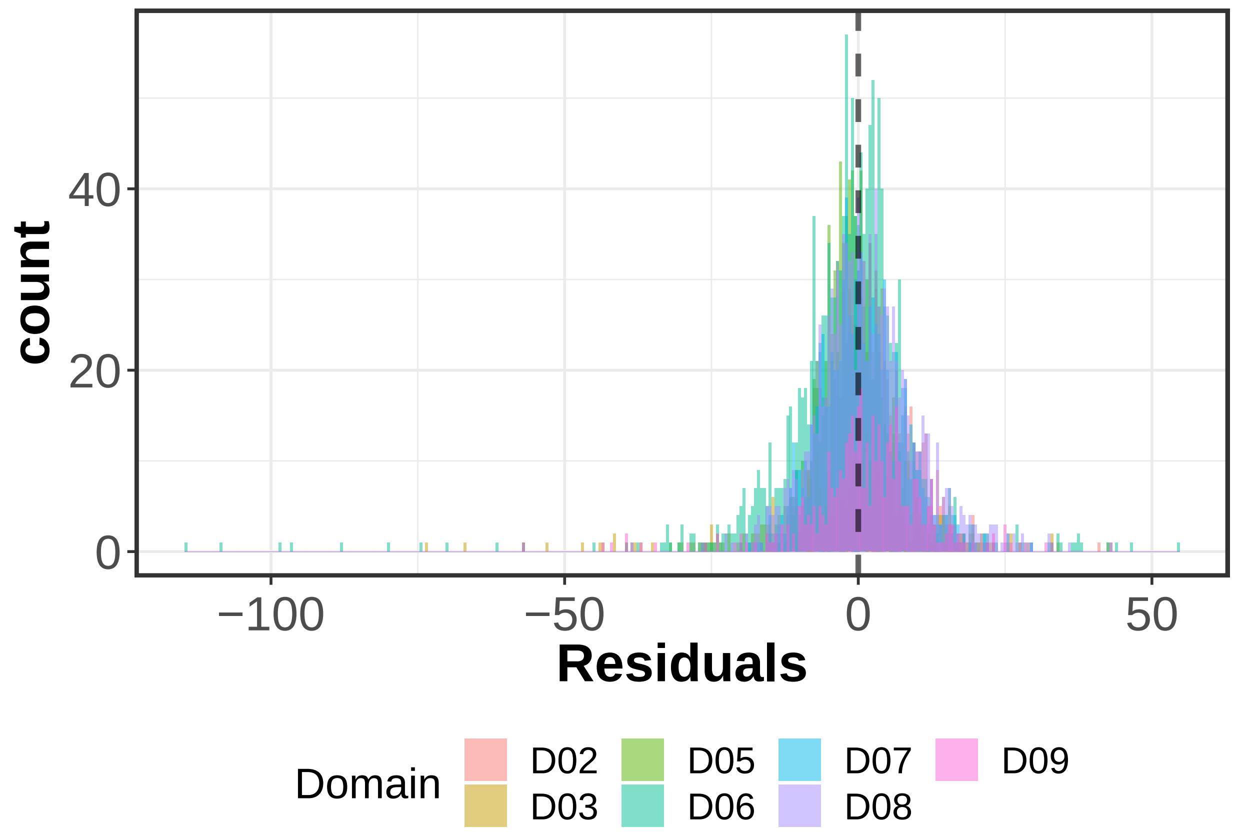


Scatter plot by Domain

```
scatter_plot <- ggplot(plot_data, aes(x=Fitted, y=LMA_gDW_m2)) +
  theme_bw() + geom_point(aes(fill=Domain),alpha=0.6,colour="black", pch=21, size=4) +
  geom_abline(intercept = 0, slope = 1, color="dark grey",
              linetype="dashed", size=1.5) + xlim(0, 275) + ylim(0, 275) +
  labs(x=expression(paste("Predicted LMA (",g~m^{-2},")")),
       y=expression(paste("Observed LMA (",g~m^{-2},")"))) +
  annotate("text", x=250, y=70, label = paste0("R^2 == ", round(pls::R2(plsr.out)[[1]][nComps],2)), par
  annotate("text", x=250, y=40, label = paste0("RMSE == ", round(pls::RMSEP(plsr.out)[[1]][nComps],2)),
  theme(axis.text=element_text(size=18), legend.position="bottom",legend.title=element_text(size=16),
        legend.text=element_text(size=14),
        axis.title=element_text(size=20, face="bold"),
        axis.text.x = element_text(angle = 0,vjust = 0.5),
        panel.border = element_rect(linetype = "solid", fill = NA, size=1.5))
scatter_plot
```



```
resid_histogram <- ggplot(plot_data, aes(x=Residuals, fill=Domain)) +
  geom_histogram(binwidth=.5, alpha=.5, position="identity") +
  geom_vline(xintercept = 0, color="black", alpha=0.6,
             linetype="dashed", size=1) + theme_bw() +
  theme(axis.text=element_text(size=18), legend.position="bottom",legend.title=element_text(size=16),
        legend.text=element_text(size=14),
        axis.title=element_text(size=20, face="bold"),
        axis.text.x = element_text(angle = 0,vjust = 0.5),
        panel.border = element_rect(linetype = "solid", fill = NA, size=1.5))
resid_histogram
```


count
40
20
0
−100
−50
0
50
Residuals
Domain
D02
D03
D05
D06
D07
D08
D09

eof