

1. Dynamic A vs ci curve fitting and parameter extraction

```
#Open necessary libraries
library(plantecophys)
library(plyr)
library(dplyr)
library(nlstools)

#Read input data file into data frame
ACis <- read.csv(file= "ACis.csv", header=TRUE, sep=",")

#Fit curves and output to dataframe "fits"
fits <- fitacis(ACis, group = "Curve", fitmethod="bilinear",id= NULL,fitTPU=FALSE,useRd=FALSE,
               theta=0.67,Km=712.5787,GammaStar=42.84107,
               EaV=58550,EdVC=20000,delsC=629.26,
               EaJ=29680,EdVJ=200000,delsJ=631.88)

#Output coefficients to dataframe "coefdf"
coefdf<-as.data.frame(coef(fits))

#Extract Ac/Aj transition points into data frame "transitionpointsdf"
transitionpoints<-lapply(fits,"[["Ci_transition")
transitionpointsdf <- ldply(transitionpoints,data.frame)
names(transitionpointsdf)[names(transitionpointsdf)=='X..i..']<-'Transition_point'

#Calculate photosynthesis at transition point, extract into dataframe "Aattransition"
range<-as.data.frame(1:4500) ;colnames(range)<-c("Fitnumber")
Aattransition = NULL
for (k in 1:length(range$Fitnumber)){
  need<-as.integer(head(range, n=1))
  ACa <- fits[[need]]$Photosyn(Ci=fits[[need]]$Ci_transition) ;
  thecurvewefiitfortransition<-need
  data<-cbind(ACa,thecurvewefiitfortransition)
  Aattransition<-rbind(Aattransition , data)
  range <- range[-1,];range<-as.data.frame(range)
}
names(Aattransition)[names(Aattransition)=='ALEAF']<-'A_transition'

#Combine dataframes into one output dataframe "outputdf" and save as CSV file
```

```
outputdf<-cbind(coefdf,transitionpointsdf,Aattrtransition["A_transition"])
write.csv(outputdf,"output.csv")
```

```
# Create dataframe containing modelled curves "modelledcurvesdf"
modelledcurves<-lapply( fits , "[[" , "df" )
modelledcurvesdf <- ldply(modelledcurves, data.frame)
write.csv(modelledcurvesdf, file = "modelledcurves.csv", row.names=FALSE)
```

2. Analysis of kinetic parameters

```
#load necessary libraries
library(doBy) #summaryBy()
library(dplyr)
library(ggplot2)
library(sjstats)
library(stats)
library(gridExtra)
library(outliers)

options("scipen"=100, "digits"=4)

#read in data file
x <- read.table("alldata.csv", header=TRUE, sep=",", na.strings="NA", dec=".", strip.white=TRUE)

x$gt <- as.factor(x$gt)

#split into narrabri and camden
xn <- subset(x, site=="n")
xc <- subset(x, site=="c")

#convert camden data to minutes
xc$t95 =xc$t95/60
xc$t7525 =xc$t7525/60
xc$t25 =xc$t25/60

#remove outliers
for(i in unique(xc$gt)) {
  xcsub=subset(xc, gt==i) #subset df for each genotype
  t=grubbs.test(xcsub$t95); s=t$alternative #grubbs test , get output string from test
  if(t$p.value < 0.05) {
    if(substring(s,1,regexpr(" ",s)-1)=="highest") {xc$t95[[which(xc$t95==max(xcsub$t95))]]<-NA}
    if(substring(s,1,regexpr(" ",s)-1)=="lowest") {xc$t95[[which(xc$t95==min(xcsub$t95))]]<-NA}
  }
}
```

```

}
t=grubbs.test(xcsub$t7525); s=t$alternative
if(t$p.value < 0.05) {
  if(substring(s,1,regexpr(" ",s)-1)=="highest") {xc$t7525[[which(xc$t7525==max(xcsub$t7525))]]<-NA}
  if(substring(s,1,regexpr(" ",s)-1)=="lowest") {xc$t7525[[which(xc$t7525==min(xcsub$t7525))]]<-NA}
}
t=grubbs.test(xcsub$t25); s=t$alternative
if(t$p.value < 0.05) {
  if(substring(s,1,regexpr(" ",s)-1)=="highest") {xc$t25[[which(xc$t25==max(xcsub$t25))]]<-NA}
  if(substring(s,1,regexpr(" ",s)-1)=="lowest") {xc$t25[[which(xc$t25==min(xcsub$t25))]]<-NA}
}
}

#test normality for camden
shapiro.test((xc$t7525))
shapiro.test((xc$t95))
shapiro.test((xc$t25))

shapiro.test(1/xc$t7525)
shapiro.test(1/xc$t95)
shapiro.test(1/xc$t25)

#camden: use inverses for 7525, 95; use untransformed for 25
xc$it95 <- 1/xc$t95
xc$it7525 <- 1/xc$t7525

#test normality for narrabri
shapiro.test((xn$t7525))
shapiro.test((xn$t95))
shapiro.test((xn$t25))

shapiro.test(log(xn$t7525))
shapiro.test(log(xn$t95))
shapiro.test(log(xn$t25))

#narr: use log-tx for 7525, 95; use untx for 25

#test for differences among genotypes
summary(aov(it95 ~ gt, data=xc))
summary(aov(it7525 ~ gt, data=xc))
summary(aov(t25 ~ gt, data=xc))

summary(aov(log(t95) ~ gt, data=xn))
summary(aov(log(t7525) ~ gt, data=xn))
summary(aov(t25 ~ gt, data=xn))

```