Citrate Synthase (for publication)

Olivia Cattau

2023-01-06

Coding Resources for paper named X authored by Olivia Cattau, Matt George and Steven Roberts University of Washington School of Aquatic and Fisheries Sciences Primary Contact: Olivia Catttau

For Calculating and Analysis of Citrate Sythase Enzyme Activity in Pacific Oysters (C. gigas)

#Load Libraries for the entire script

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggpubr)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.1.2
```

#Load Raw Citrate Synthase Absorbance data and Protein Data from spectrophotometer

```
CS_absorbance<-read.csv(file="/Users/oliviacattau/Documents/GitHub/CS-manuscript/raw-data/Rawdata_Absorb
BSA_absorbance<-read.csv(file="/Users/oliviacattau/Documents/GitHub/CS-manuscript/raw-data/BSA_absorban
```

#Load morphometric data and labels for oyster numbers

```
morph<-read.csv(file="https://raw.githubusercontent.com/mattgeorgephd/NOPP-gigas-ploidy-temp/main/20210
```

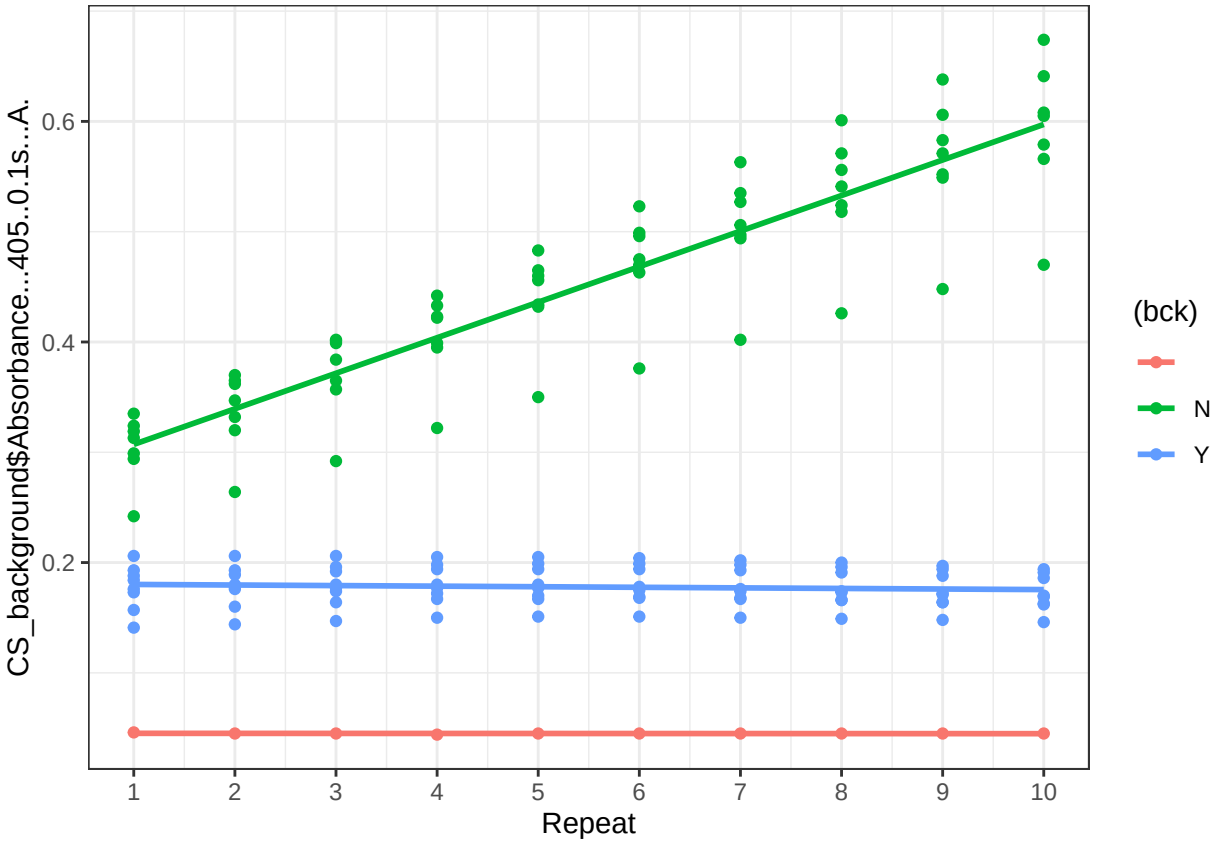
#Look at Background Control for significance

```
CS_background<-read.csv(file="/Users/oliviacattau/Documents/GitHub/NOPP-gigas-ploidy-temp/202107_EXP2/c

CS_BACKGROUND_ANOVA<-lm(CS_background$Absorbance...405..0.1s...A.~bck, data=CS_background)
car::Anova(CS_BACKGROUND_ANOVA)
```

```
## Anova Table (Type II tests)
##
## Response: CS_background$Absorbance...405..0.1s...A.
##           Sum Sq  Df F value    Pr(>F)
## bck       3.4488   2  355.03 < 2.2e-16 ***
## Residuals 0.7626 157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
background_plot<-ggplot(data=CS_background, aes(x=Repeat, y=CS_background$Absorbance...405..0.1s...A., 
background_plot
```

```
## Warning: Use of `CS_background$Absorbance...405..0.1s...A.` is discouraged.
## i Use `Absorbance...405..0.1s...A.` instead.
## Use of `CS_background$Absorbance...405..0.1s...A.` is discouraged.
## i Use `Absorbance...405..0.1s...A.` instead.
```



background control is significantly different from the standard CS values which means that I do not have

to subtract the background from the sample readings. Also you can see in 'background plot' that the background readings did not increase while the CS readings did.

#Look at plate effects before proceding

```
anova1<-lm(delta.OD~plate*X1*X10*repeat., data=CS_absorbance)
anova(anova1)
```

```
## Analysis of Variance Table
##
## Response: delta.OD
##                        Df  Sum Sq Mean Sq    F value  Pr(>F)
## plate                   1 0.16792 0.16792 7.7380e+05 < 2e-16 ***
## X1                      1 0.24866 0.24866 1.1459e+06 < 2e-16 ***
## X10                     1 1.46228 1.46228 6.7385e+06 < 2e-16 ***
## repeat.                 1 0.00000 0.00000 8.7000e-03 0.92577
## plate:X1                1 0.00000 0.00000 6.6680e-01 0.41464
## plate:X10               1 0.00000 0.00000 2.6830e-01 0.60477
## X1:X10                  1 0.00000 0.00000 3.3230e-01 0.56466
## plate:repeat.           1 0.00000 0.00000 6.4130e-01 0.42372
## X1:repeat.              1 0.00000 0.00000 4.5740e-01 0.49924
## X10:repeat.             1 0.00000 0.00000 1.5256e+00 0.21749
## plate:X1:X10            1 0.00000 0.00000 2.8394e+00 0.09275 .
## plate:X1:repeat.        1 0.00000 0.00000 1.9600e-02 0.88865
## plate:X10:repeat.       1 0.00000 0.00000 6.3000e-01 0.42782
## X1:X10:repeat.          1 0.00000 0.00000 8.2800e-02 0.77365
## plate:X1:X10:repeat.    1 0.00000 0.00000 1.9219e+00 0.16641
## Residuals             404 0.00009 0.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(anova1) #plate effects not significant
```

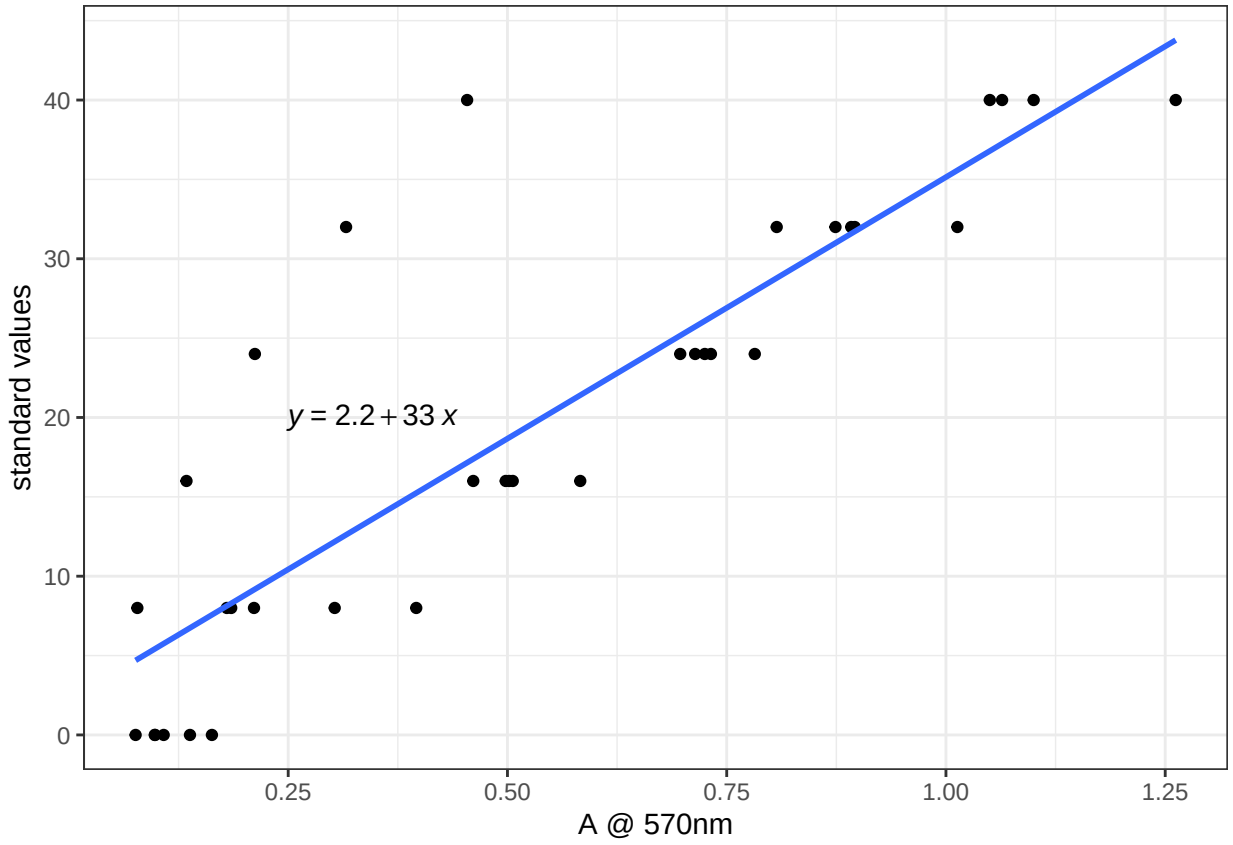
```
##
## Call:
## lm(formula = delta.OD ~ plate * X1 * X10 * repeat., data = CS_absorbance)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.163e-03 -7.047e-05 -2.682e-05  3.029e-05  1.072e-03
##
## Coefficients:
##                       Estimate Std. Error  t value Pr(>|t|)
## (Intercept)          -0.0005721  0.0010343   -0.553   0.5805
## plate                 0.0005177  0.0005329    0.971   0.3319
## X1                   -0.9878681  0.0074546 -132.518   <2e-16 ***
## X10                   0.9930782  0.0059357  167.307   <2e-16 ***
## repeat.               0.0005021  0.0007598    0.661   0.5090
## plate:X1             -0.0040811  0.0037962   -1.075   0.2830
## plate:X10             0.0003381  0.0022781    0.148   0.8821
## X1:X10               -0.0061335  0.0074181   -0.827   0.4088
## plate:repeat.        -0.0003194  0.0002912   -1.097   0.2734
## X1:repeat.           -0.0104784  0.0058275   -1.798   0.0729 .
```

```
## X10:repeat.           0.0062535  0.0046125    1.356   0.1759
## plate:X1:X10          0.0049909  0.0039935    1.250   0.2121
## plate:X1:repeat.      0.0030095  0.0020917    1.439   0.1510
## plate:X10:repeat.    -0.0006871  0.0013184   -0.521   0.6025
## X1:X10:repeat.        0.0047070  0.0049979    0.942   0.3469
## plate:X1:X10:repeat. -0.0029104  0.0020994   -1.386   0.1664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0004658 on 404 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.772e+05 on 15 and 404 DF,  p-value: < 2.2e-16
```

no plate effects, can use entire data set

#View Controls plot CS standards to calculate standard curve equation

```
CS_controls<-filter(CS_absorbance, ID == 0 | ID == 8 | ID == 16 | ID == 24 | ID == 32 |ID == 40)

CS_controls2<-(CS_absorbance[c(2:6),]) #y=96x -0.29

#plot CS standards
standard_plot<-ggplot(data=CS_controls,(aes(x=X1, y=as.numeric(ID))))+geom_point()+geom_smooth(method="
standard_plot
```



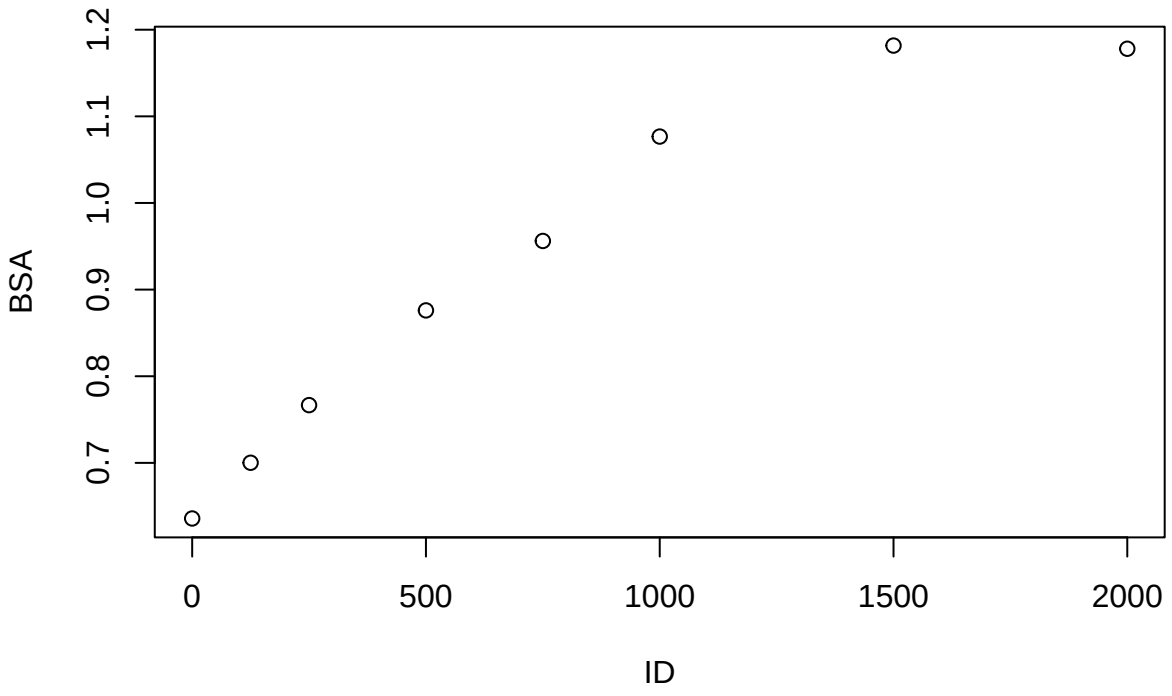
#Extract Equation from standard curve $y = 33x + 2.2$

#Calculate OD and CS values from standard curve $OD = X10 - X1$ $x = OD$ so that $nmolCS = 33 * OD + 2.2$

```
control_table<-CS_absorbance %>%
  filter(plate != 5) %>% #plate 5 had errors
  group_by(ID) %>%
  summarise(avg1=mean(X1), avg2=mean(X10))

CS_absorbance2<-mutate(control_table, OD = avg2-avg1)

CS_absorbance3<-mutate(CS_absorbance2, nmol = 33*OD + 2.2) #nmol of CS enzyme
```

#Calculate protein values from Bovine Serum Assay (BSA) to standardize CS values

```
protein_data<-BSA_absorbance %>%
  group_by(ID) %>%
  summarise(BSA = mean(A))

bsa_standards<-filter(protein_data, ID == 0 | ID == 125 |ID == 250| ID == 500 | ID == 750 | ID == 1000|
plot(bsa_standards) #linear portion only from 0 to 1000
```




```
bsa_standards<-(protein_data[c(1, 2, 3, 6, 7, 8),]) #select correct values [0:1000]

bsa_standards$ID <- as.numeric(as.character(factor(bsa_standards$ID, levels=c("0", "125", "250", "500",

BSA_standard_curve<-ggplot(data=bsa_standards,(aes(x=BSA, y=(ID))))+geom_point()+geom_smooth(method="lm
```

Equation from BSA protein curve produces $y = 2300x - 1500$ where y=Protein in ug/mL and x=Absorbance from spectrophotometer from BSA assay

#Make final table with combined CS and BSA values $nmolCS = 33 * OD + 2.2$

```
protein_data$protein<-(2300*protein_data$BSA)-1500 #ug/mL

protein_data$P<-(protein_data$protein)/1000/1000 #ug/ml-> ug/uL-> mg/uL  also equal to total protein ex

t<-45 #min
m<-33 #slope from CS standard curve
b<-2.2 #intercept from CS standard curve
V<-50 #uL from CS procedure
D<-1 #dilution coefficient, should be 1 since we did not dilute

Full_dataset<-full_join(CS_absorbance3, protein_data, by='ID')

Full_dataset<-Full_dataset[complete.cases(Full_dataset),] #remove NAs

Full_dataset$CS_activity<-(Full_dataset$nmol/(t*V))*D/Full_dataset$P
```

#Attach Morphometric Data Join the results (CS and BSA data) to morphometric data gathered during the experiment, n=68

```
treatments<-read.csv("/Users/oliviacattau/Documents/GitHub/NOPP-gigas-ploidy-temp/202107_EXP2/citrate_s

filter_my_data<-morph %>%
  filter(morph$ID %in% Full_dataset$ID)

filter_my_data2<-filter_my_data[c(1,2,5,6,10,11,12,14,16,17)] #remove columns without data
#keep: ploidy, trt, shell_length, shell_width, shell_height, mortality, shell volume, cal_dry_weight, p

Full_data<-full_join(filter_my_data2, treatments, by="ID")

Full_data2<-full_join(Full_data, Full_dataset, by="ID")

Full_data3<-Full_data2[complete.cases(Full_data2),] #remove NAs


#add mortality data
#mortality
mortality<-read.csv("/Users/oliviacattau/Documents/GitHub/NOPP-gigas-ploidy-temp/202107_EXP2/citrate_sy
final_mort<-mortality$X..survival
final_list<-mortality$trt
mortality2<-data.frame(final_mort, final_list)
```

#Data Visualization

```
ploidy_linear_plot<-ggplot(data=Full_data3,(aes(x=protein, y=CS_activity, color=ploidy)))+geom_point()+

boxplot<-ggplot(data=Full_data3, aes(x=factor(trt), y=CS_activity))+
  geom_boxplot(aes(x=factor(trt), y=CS_activity, color=ploidy))+
  theme_bw()+
  ylab(expression('CS activity nmol' (min^-1) (mg^-1)))+
  xlab("Treatment Group")+
  stat_compare_means(comparisons=list(c("T-heat", "T-desiccation")), method = "wilcox.test", aes(label=
  stat_compare_means(comparisons=list(c("D-heat", "D-desiccation")), method = "wilcox.test", aes(label=
  stat_compare_means(comparisons=list(c("T-control", "T-desiccation")), method = "wilcox.test", aes(lab
```

```
  stat_compare_means(comparisons=list(c("D-control", "D-desiccation")), method = "wilcox.test", aes(la
 stat_compare_means(comparisons=list(c("D-desiccation", "T-desiccation")), method = "wilcox.test", aes
   # Add a second axis and specify its features
   sec.axis = sec_axis(~.*7, name="% survival")
 ) +theme(axis.title.y.right = element_text(color="red"))

boxplot
```