Impact of short- and long- term eposure to elevated seawater pCO2 on metabolic rate and hypoxia tolerance in *Octopus rubescens*

Kirt Onthank

9/15/2019

Loading libraries

```
library(emmeans)
library(car)
library(multcomp)
library(seacarb)
library(knitr)
library(plotKML)
library(rgdal)
library(geoR)
library(plotrix)
library(plotKML)
library(maptools)
library(viridis)
library(nlme)
```

Seawater surface pH spatial anlaysis

Loading GPS coordinates of sampling sites

```
gps=rbind(
readGPX(list.files("./GPS_waypoints",full.names = T)[1],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[2],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[3],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[4],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[5],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[6],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[7],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[8],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[9],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[10],tracks = F,routes = F)$waypoints[,1:6],
readGPX(list.files("./GPS_waypoints",full.names = T)[11],tracks = F,routes = F)$waypoints[,1:6]
)

gps=gps[order(gps$name),]
```

```
gps=gps[1:64,]
gps$name=as.numeric(gps$name)
```

Loading pH data and associating with spatial information

```
ph=read.csv("Seawater_ph_2015.csv")

for (i in 1:length(ph$station)){
  ph$lat[i]=gps$lat[gps$name==ph$station[i]]
  ph$lon[i]=gps$lon[gps$name==ph$station[i]]
  ph$time[i]=gps$time[gps$name==ph$station[i]]
  }
```

estimating pCO_2 values based on alkalinity of 2080 umol/kg

```
ph$pco2=carb(8,ph$ph,0.00208,S=ph$s,T=ph$t)$pCO2
```

Separating shallow and deep samples

```
ph.surface=ph[ph$level=="s",]
ph.deep=ph[ph$level=="d",]
```

Projecting Lat Long coordinates into UTM so that distances can easily be calculated in meters.

```
ph.surface=cbind(ph.surface,
            project(cbind(as.numeric(ph.surface[,c(7)]),as.numeric(ph.surface[,c(6)])),
            "+proj=utm +zone=10 ellps=WGS84"))
colnames(ph.surface)=c(colnames(ph.surface)[1:9],"UTM_E","UTM_N")
ph.deep=cbind(ph.deep,
            project(cbind(as.numeric(ph.deep[,c(7)]),as.numeric(ph.deep[,c(6)])),
            "+proj=utm +zone=10 ellps=WGS84"))
colnames(ph.deep)=c(colnames(ph.deep)[1:9],"UTM_E","UTM_N")
```

Calculating summary statistics of seawater pH and pCO_2

```
seawater.ph=
cbind(
c(range(ph.surface$ph),round(mean(ph.surface$ph),3)),
c(range(ph.deep$ph),round(mean(ph.deep$ph),3)),
c(round(range(ph.surface$pco2)),round(mean(ph.surface$pco2))),
c(round(range(ph.deep$pco2)),round(mean(ph.deep$pco2)))
```

```
)
rownames(seawater.ph)=c("min","max","mean")
colnames(seawater.ph)=c("0m pH","15m pH","0m pCO~2~","15m pCO~2~")

knitr::kable(seawater.ph,align="r",booktabs=TRUE)
```

	0m pH	15m pH	0m pCO_2	15m pCO_2
min	7.658	7.579	792	857
max	7.773	7.741	1048	1257
mean	7.709	7.670	927	1022

Fitting models to surface pH semivariogram data to determine range of spatial autocorrelation

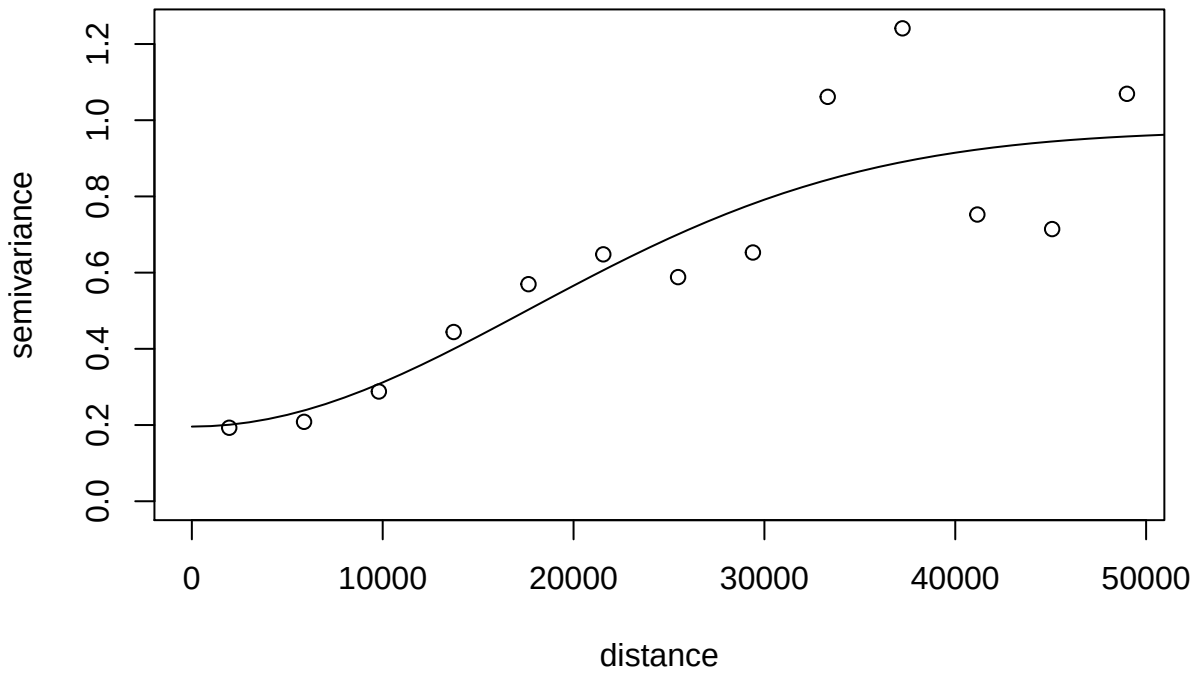
```
ph.pp=as.geodata(ph.surface,coords.col = 10:11)
surface.var=variog(ph.pp,max.dist=100000)
```

```
## variog: computing omnidirectional variogram
```

```
surface.vg=variofit(surface.var, ini.cov.pars=c(0.8, 10000),
               nugget=0.2,cov.model="gaussian",messages=F)
```

Plotting the surface pH semivariogram with the fitted model

```
plot(surface.var)
lines(surface.vg)
```



Summary of model parameters

```
summary(surface.vg)
```

```
## $pmethod
## [1] "WLS (weighted least squares)"
##
## $cov.model
## [1] "gaussian"
##
## $spatial.component
##      sigmasq          phi
## 7.776336e-01 2.489698e+04
##
## $spatial.component.extra
## kappa
##   0.5
##
## $nugget.component
##     tausq
## 0.1959372
##
## $fix.nugget
```

```
## [1] FALSE
##
## $fix.kappa
## [1] TRUE
##
## $practicalRange
## [1] 43092.15
##
## $sum.of.squares
##    value
## 7.260295
##
## $estimated.pars
##        tausq      sigmasq          phi
## 1.959372e-01 7.776336e-01 2.489698e+04
##
## $weights
## [1] "npairs"
##
## $call
## variofit(vario = surface.var, ini.cov.pars = c(0.8, 10000), cov.model = "gaussian",
##     nugget = 0.2, messages = F)
##
## attr(,"class")
## [1] "summary.variomodel"
```

Determining the distance of spatial correlation of surfacd pH in kilometers from the phi term

```
surface.vg$cov.pars[2]/1000
```

```
## [1] 24.89698
```

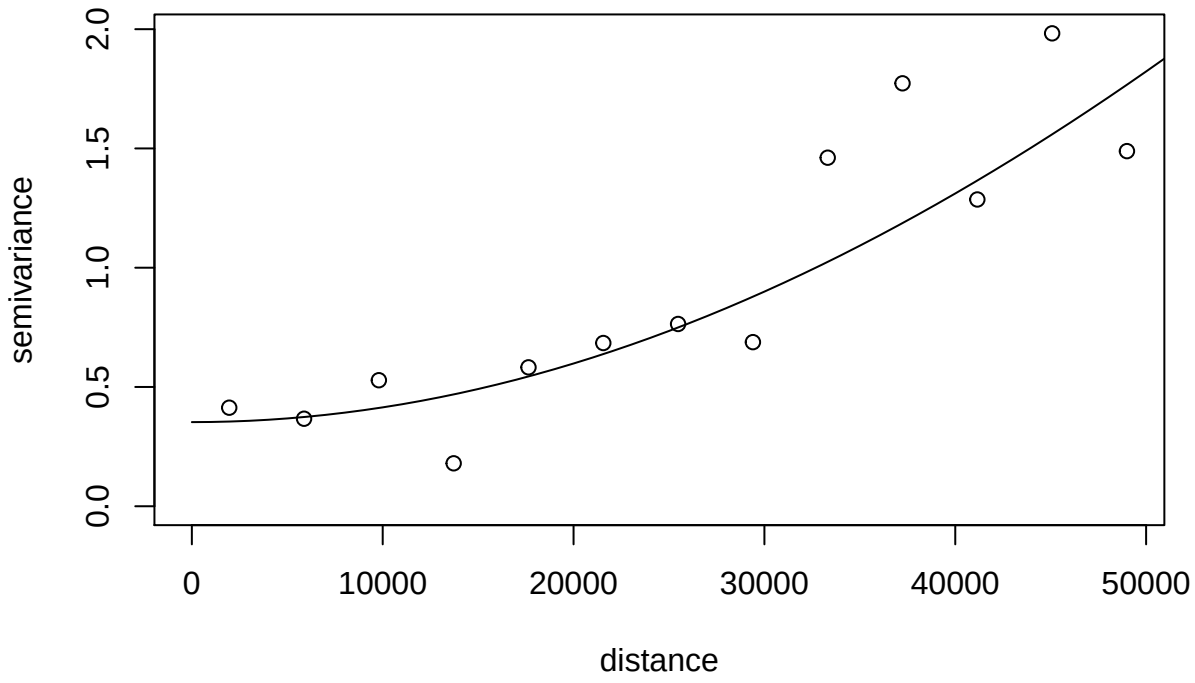
Semivariogram fitting in deep pH

```
ph.deep.pp=as.geodata(ph.deep,coords.col = 10:11)
deep.var=variog(ph.deep.pp,max.dist=100000)
```

```
## variog: computing omnidirectional variogram
```

```
deep.vg=variofit(deep.var, ini.cov.pars=c(0.8, 10000),
                 nugget=0.2,cov.model="gaussian",messages=F)

plot(deep.var)
lines(deep.vg)
```



Semivariogram fit of deep pH summary

```
summary(deep.vg)
```

```
## $pmethod
## [1] "WLS (weighted least squares)"
##
## $cov.model
## [1] "gaussian"
##
## $spatial.component
##      sigmasq          phi
##     14.67641 153906.88841
##
## $spatial.component.extra
## kappa
##   0.5
##
## $nugget.component
##     tausq
## 0.3526234
##
## $fix.nugget
```

```
## [1] FALSE
##
## $fix.kappa
## [1] TRUE
##
## $practicalRange
## [1] 266384.9
##
## $sum.of.squares
##    value
## 5.711114
##
## $estimated.pars
##        tausq      sigmasq          phi
## 3.526234e-01 1.467641e+01 1.539069e+05
##
## $weights
## [1] "npairs"
##
## $call
## variofit(vario = deep.var, ini.cov.pars = c(0.8, 10000), cov.model = "gaussian",
##     nugget = 0.2, messages = F)
##
## attr(,"class")
## [1] "summary.variomodel"
```

Determining the distance of spatial correlation of deep pH in kilometers from the phi term

```
deep.vg$cov.pars[2]/1000
```

```
## [1] 153.9069
```

Plotting seawater pH by geography

```
wash_utm.shp=readShapePoly("./Shapefiles/sample_utm.shp")
salish_utm.shp=readShapePoly("./Shapefiles/salish_utm.shp")
just_wash_utm.shp=readShapePoly("./Shapefiles/just_wash_utm.shp")
```

```
top.top=1
bottom.top=0
top.bottom=.56
bottom.bottom=0



ph=cbind(ph,project(cbind(ph$lon,ph$lat),"+proj=utm +zone=10 ellps=WGS84"))
colnames(ph)=c(colnames(ph)[1:9],"UTM_E","UTM_N")
```

```
xlim.max=545000-10000
xlim.min=505000-20000
ylim.min=5330000
ylim.max=5380000
key.start=mean(c(xlim.max,xlim.min))-23000
key.end=mean(c(xlim.max,xlim.min))+3000
key.top=5334000
key.bottom=5330000

interp=round(seq(from=min(ph$ph),to=max(ph$ph),by=0.001),3)
color.palette=viridis(length(interp))
ph$pHcolor=1
for(i in 1:nrow(ph)){
  ph$pHcolor[i]=color.palette[interp==ph$ph[i]]
}



key.points=seq(from=key.start,to=key.end,length.out=length(interp))
```

```
png("Figure_1.png",pointsize=12,height=9,width=15,units="in",res=300)
#svg("Figure_1.svg",pointsize=12,height=9,width=15)

## Surface pH Plot ####
par(fig=c(0,0.54,bottom.top,top.top))
plot(wash_utm.shp,bg="grey90",col="white",ylim=c(ylim.min,ylim.max),xlim=c(xlim.min,xlim.max),
     lwd=0.3)
box(lwd=2)
par(mgp = c(2, 0.3, 0))
axis(2,cex.axis=1.3,tcl=-0.2
     ,at=c(5338535,5349688,5360881,5372112)
     ,labels=c("48.2","48.3","48.4","48.5")
)
axis(1,cex.axis=1.3,tcl=-0.2
     ,at=c(492612.82,507459.5,522292,537008.5,551608.3)
     ,labels=c("-123.1","-122.9","-122.7","-122.5","-122.3")
)
par(mgp = c(2, 0.5, 0))
mtext("Latitude",side=2,line=2,cex=1.6)
points(ph$UTM_E[ph$level=="s"],ph$UTM_N[ph$level=="s"]
       ,pch=21,bg=ph$pHcolor[ph$level=="s"],cex=1.7,lwd=0.6)
text(mean(c(xlim.max,xlim.min)),5379000,"Surface",cex=4)



# Insert plot ####
par(fig=c(0,0.3,bottom.top,((top.top-bottom.top)*.6)+bottom.top),new=T)
plot(salish_utm.shp,col="white",ylim=c(5150000,5600000),
     xlim=c(440000,510000),lwd=.2,bg="grey20")
par(fig=c(0,0.3,bottom.top,((top.top-bottom.top)*.6)+bottom.top),new=T)
plot(just_wash_utm.shp,col="white",lwd=.2,ylim=c(5150000,5600000),
     xlim=c(440000,510000),bg="transparent")
box(lwd=2)
```

```
segments(507000,5328000,507000,5375000,lwd=3,col="red")
segments(507000,5328000,553000,5328000,lwd=3,col="red")
segments(553000,5328000,553000,5375000,lwd=3,col="red")
segments(507000,5375000,553000,5375000,lwd=3,col="red")
text(580000,5170000,"WASHINGTON",cex=1,font=2)
text(580000,5550000,"BRITISH",cex=1,font=2)
text(580000,5520000,"COLUMBIA",cex=1,font=2)
text(320000,5270000,"PACIFIC",cex=1,font=2,col="white")
text(320000,5235000,"OCEAN",cex=1,font=2,col="white")

## 15 meter pH plot ####
par(fig=c(0.46,1,bottom.top,top.top),new=T)
plot(wash_utm.shp,bg="grey90",col="white",ylim=c(ylim.min,ylim.max),xlim=c(xlim.min,xlim.max),
     lwd=0.3)
box(lwd=2)
par(mgp = c(2, 0.3, 0))
axis(4,cex.axis=1.3,tcl=-0.2
     ,at=c(5338535,5349688,5360881,5372112)
     ,labels=c("48.2","48.3","48.4","48.5")
)
axis(1,cex.axis=1.3,tcl=-0.2
     ,at=c(492612.82,507459.5,522292,537008.5,551608.3)
     ,labels=c("-123.1","-122.9","-122.7","-122.5","-122.3")
)
par(mgp = c(2, 0.5, 0))
points(ph$UTM_E[ph$level=="d"],ph$UTM_N[ph$level=="d"]
       ,pch=21,bg=ph$pHcolor[ph$level=="d"],cex=1.7,lwd=0.6)
points(project(cbind(-122.662895,48.419363),"+proj=utm +zone=10 ellps=WGS84"),
       pch="*",bg='black',cex=3) #RBML
text(project(cbind(-122.662895,48.419363+0.015), "+proj=utm +zone=10 ellps=WGS84"),
     "Rosario Beach Marine Lab",cex=1) #RBNL
points(project(cbind(-122.636824,48.16628),"+proj=utm +zone=10 ellps=WGS84"),
       pch="*",bg='black',cex=3) #Driftwood Park
text(project(cbind(-122.636824,48.16628+0.015), "+proj=utm +zone=10 ellps=WGS84"),
     "Driftwood Park",cex=1) #Driftwood Park
points(project(cbind(-122.623611,48.501944), "+proj=utm +zone=10 ellps=WGS84"),
       pch="*",cex=2) #Anacortes
points(project(cbind(-122.775278,48.116389), "+proj=utm +zone=10 ellps=WGS84"),
       pch="*",cex=2) #Port Townsend
points(project(cbind(-122.658611,48.295), "+proj=utm +zone=10 ellps=WGS84"),
       pch="*",cex=2) #Oak Harbor
text(project(cbind(-122.623611,48.501944+0.015), "+proj=utm +zone=10 ellps=WGS84"),
     "ANACORTES",cex=1) #Anacortes
text(project(cbind(-122.775278+.08,48.116389+.015), "+proj=utm +zone=10 ellps=WGS84"),
     "PORT TOWNSEND",cex=1) #Port Townsend
text(project(cbind(-122.658611,48.295+.015), "+proj=utm +zone=10 ellps=WGS84"),
     "OAK HARBOR",cex=1) #Oak Harbor
text(530000,5366500,"Fidalgo I.",cex=1.3,font=2)
text(530000,5354000,"Whidbey I.",cex=1.3,font=2)
text(507459.5,5371000-1000,"Lopez I.",cex=1.3,font=2)
text(507459.5-14000,5371000+5000,"San Juan I.",cex=1.3,font=2)

##pH Key###
```

```
for(i in 1:length(key.points)){
  lines(rep(key.points[i],2),c(key.bottom,key.top),col=color.palette[i],lwd=3)
}
lines(c(key.start,key.end,key.end,key.start,key.start),c(key.bottom,key.bottom,key.top,key.top,key.bott
text(key.points[c(1,length(interp)/2,length(interp))],rep(key.top+1500,3)
     ,interp[c(1,length(interp)/2,length(interp))],cex=1.2,font=2)
lines(c(key.points[length(interp)/2],key.points[length(interp)/2]),c(key.top+500,key.top),lwd=2)
lines(c(key.points[1],key.points[1]),c(key.top+500,key.top),lwd=2)
lines(c(key.points[length(interp)],key.points[length(interp)]),c(key.top+500,key.top),lwd=2)
text(mean(c(xlim.max,xlim.min)),5379000,"15 m",cex=4)

par(fig=c(0,1,0,1),new=T)
plot(c(0,1),c(0,1),type="n",axes=F,ylab="",xlab="")
mtext("Longitude",side=1,line=2,cex=1.6)
dev.off()
```

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

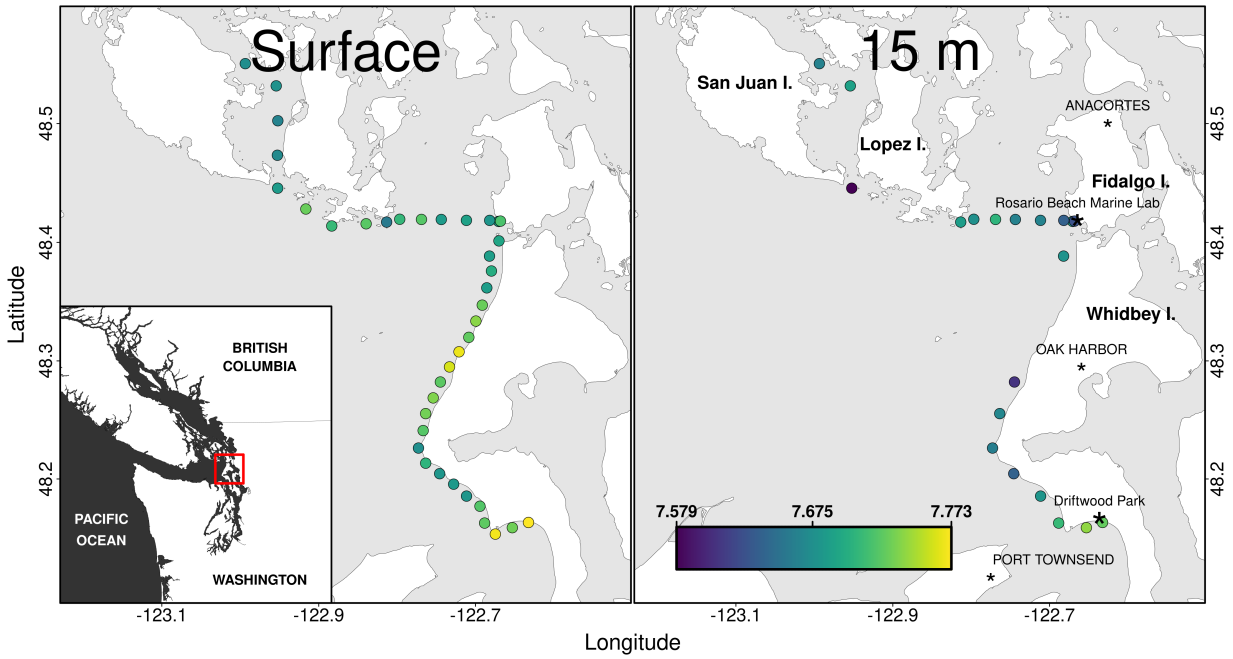


Figure 1: Geographic variation in surface seawater pH near Rosario Beach Marine Lab and Driftwood Park

Treatment carbonate parameters

Reading in carbonate treatment data

```
pco2=read.csv("treatment_carbonate_2014.csv")
```

Calculating pCO_2 values from pH and alkalinity

```
pco2$pco2_low=round(carb(8,pco2$ph_low,pco2$alk_low/1000000,S=30,T=11)$pCO2)
pco2$pco2_high=round(carb(8,pco2$ph_high,pco2$alk_high/1000000,S=30,T=11)$pCO2)
pco2$pco2_sw=round(carb(8,pco2$pH_sw,pco2$alk_sw/1000000,S=30,T=11)$pCO2)
```

Are seawater table pCO_2s sig different from low CO_2 treatment

First testing variances

```
bartlett.test(c(pco2$ph_high,pco2$pH_sw),c(rep("high",36),rep("sw",36)))
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  c(pco2$ph_high, pco2$pH_sw) and c(rep("high", 36), rep("sw", 36))
## Bartlett's K-squared = 20.289, df = 1, p-value = 6.657e-06
```

```
shapiro.test(pco2$pco2_high)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  pco2$pco2_high
## W = 0.98092, p-value = 0.776
```

```
shapiro.test(pco2$pco2_sw)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  pco2$pco2_sw
## W = 0.95822, p-value = 0.1893
```

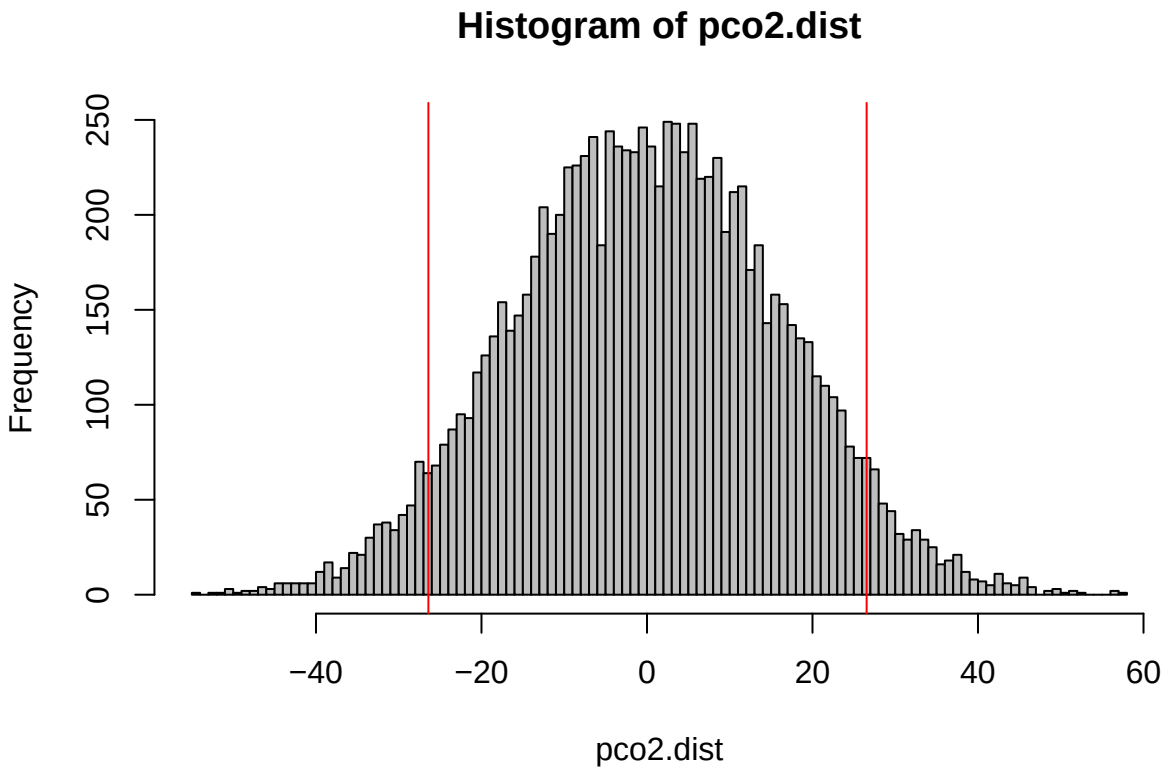
Two-tailed permutation t-test of low CO_2 treatments

Using a permutation test because the assumption of homogeneity of variances is strongly violated

```r
set.seed(56)
mean.diff=mean(pco2$pco2_high)-mean(pco2$pco2_sw)

pco2.dist=0
for (i in 1:10000){
  pco2.samp=sample(c(pco2$pco2_high,pco2$pco2_sw))
  pco2.dist[i]=mean(pco2.samp[1:36]-mean(pco2.samp[37:72]))
}

hist(pco2.dist,breaks=100,col="grey")
abline(v=mean.diff,col="red")
abline(v=mean(pco2.dist)-mean.diff,col="red")
```



Calculating p-value

```r
pco2.p=sum(pco2.dist<mean.diff | pco2.dist>mean(pco2.dist)-mean.diff)/length(pco2.dist)
pco2.p
```

```
## [1] 0.0946
```

p-value is 0.095, so not sufficient evidence to conclude that the low CO_2 treatments are different.

Summary of treatment carbonate chemistry parameters

```
carb.summary=
rbind(
c("pH",paste(round(mean(pco2$ph_low),3),"±",round(sd(pco2$ph_low),3)),
  paste(round(mean(pco2$ph_high),3),"±",round(sd(pco2$ph_high),3)),
  paste(round(mean(pco2$pH_sw),3),"±",round(sd(pco2$pH_sw),3))
),
c("Alkalinity (umol/kg)",paste(round(mean(pco2$alk_low)),"±",round(sd(pco2$alk_low))),
  paste(round(mean(pco2$alk_high)),"±",round(sd(pco2$alk_high))),
  paste(round(mean(pco2$alk_sw)),"±",round(sd(pco2$alk_sw)))
),
c("pCO~2~ (uatm)",paste(round(mean(pco2$pco2_low)),"±",round(sd(pco2$pco2_low))),
  paste(round(mean(pco2$pco2_high)),"±",round(sd(pco2$pco2_high))),
  paste(round(mean(pco2$pco2_sw)),"±",round(sd(pco2$pco2_sw)))
)
)
colnames(carb.summary)=c("","High CO~2~","Low CO~2~","Flow-Through")


kable(carb.summary,align=c("l","c","c","c"))
```

	High CO_2	Low CO_2	Flow-Through
pH	7.513 ± 0.035	7.813 ± 0.049	7.778 ± 0.022
Alkalinity (umol/kg)	2091 ± 38	2096 ± 42	2005 ± 11
pCO_2 (uatm)	1422 ± 116	687 ± 84	714 ± 40

Short-term routine metabolic rate linear mixed effects model

Reading in the short-term RMR data and setting pCO2 to factor class:

```
short=read.csv("Short_Term_RMR_2014.csv")
short$pco2=as.factor(short$pco2)
```

Next I set orthogonal contrasts:

```
contrasts(short$pco2)=contr.poly(3)
```

Running the linear mixed effects model and ANOVA using type III sum of squares:

```
short.lme=lme(resp~mass+pco2,random=~1|octopusID,
              correlation=corAR1(form=~run|octopusID),
```

```
              data=short)
Anova(short.lme,type="III")
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: resp
##               Chisq Df Pr(>Chisq)
## (Intercept) 102.906  1  < 2.2e-16 ***
## mass         24.046  1  9.405e-07 ***
## pco2         13.271  2   0.001313 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Finally, I am running Tukey post-hoc test:

```
short.ph=glht(short.lme, linfct=mcp(pco2="Tukey"))
summary(short.ph)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme.formula(fixed = resp ~ mass + pco2, data = short, random = ~1 |
##     octopusID, correlation = corAR1(form = ~run | octopusID))
##
## Linear Hypotheses:
##                  Estimate Std. Error z value Pr(>|z|)
## 700 - 360 == 0    0.3853     0.3131   1.231  0.42810
## 1500 - 360 == 0   1.4557     0.4197   3.469  0.00177 **
## 1500 - 700 == 0   1.0704     0.3377   3.170  0.00413 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Calculating estimated marginal means

```
emmeans(short.lme,"pco2",sigmaAdjust = c(TRUE, FALSE))
```

```
##  pco2 emmean    SE df lower.CL upper.CL
##  360    1.64 0.283 20     1.05     2.23
##  700    2.03 0.209 20     1.59     2.46
##  1500   3.10 0.310 20     2.45     3.75
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
```

Seeing if the two smallest octopuses at 1500 makes a difference

In a previous review of this manuscript, a reviewer raised a "major concern" about this data set explaining:

"The main conclusions related to RMR rely upon data collected at the highest level of CO2 (1500 uatm) in one group of animals (un-acclimated). Moreover, within this group, the trend line is heavily influenced by two individuals (about 20 and 50 g) ... These concerns raise questions about if this treatment actually leads to higher RMR in un-acclimated animals"

We can examine if these two smallest octopuses in the 1500 uatm treatment are the reason we have a signifianct result. First, lets see the masses of the two octopuses he is talking about. Here are the masses of all of the octopuses from the 1500 uatm treatment.

```
sort(short$mass[short$pco2==1500])
```

```
##  [1]  46.0  57.5  87.5 183.0 220.0 227.0 237.2 249.0 270.0 310.0
```

So, if we exlcude octopuses in the 1500 treatment with masses under 60g, we can run the linear mixed model again to see if these two octopuses are essential to the results.

```
test.lme=lme(resp~mass+pco2,random=~1|octopusID,
             correlation=corAR1(form=~run|octopusID),
             data=short[-which(short$pco2==1500&short$mass<60),])
Anova(test.lme,type="III")
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: resp
##               Chisq Df Pr(>Chisq)
## (Intercept) 95.9305  1  < 2.2e-16 ***
## mass        17.0675  1  3.607e-05 ***
## pco2         6.9828  2    0.03046 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The analysis of deviance is still significant with respect to pCO_2 even without these octopuses. Of course, the p-value has risen significantly, but that is to be expected if I eliminate the two most extreme values in the analysis. Nevertheless, this illustrates that those two octopuses are not essential to the result.

Long-term routine metabolic rate ANCOVA

Reading in the short-term RMR data and setting both pCO2 and week to factor class:

```
long=read.csv("Long_Term_RMR_2014.csv")
long$pco2=as.factor(long$pco2)
contrasts(long$pco2)=contr.poly(2)
long$week=as.factor(long$week)
```

Setting orthogonal contrasts

```
contrasts(long$week)=contr.poly(2)
```

Running long LME model and ANOVA with type III sum of squares

```
long.lme=lme(resp_mean~mass+pco2+week,random=~1|octo, data=long)
Anova(long.lme,type="III")
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: resp_mean
##                Chisq Df Pr(>Chisq)
## (Intercept) 159.5367  1  < 2.2e-16 ***
## mass         21.5945  1  3.368e-06 ***
## pco2          0.0017  1     0.9675
## week          1.2966  1     0.2548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
emmeans(long.lme,~pco2+week)
```

```
##  pco2 week emmean    SE df lower.CL upper.CL
##  700  1      1.70 0.164  7     1.31     2.09
##  1500 1      1.71 0.190  7     1.26     2.16
##  700  5      1.49 0.164  7     1.10     1.87
##  1500 5      1.49 0.190  7     1.05     1.94
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
```

Plotting the Routine Metabolic Rate data

Power regressions of short-term data

```
reg.1500=nls(resp~a*mass^b,data=short[short$pco2==1500,],start=list(a=50,b=-0.7))
seq.1500=seq(from=min(short$mass[short$pco2==1500]),to=max(short$mass[short$pco2==1500]),
             length.out=100)
pred.1500=predict(reg.1500,newdata = data.frame(mass=seq.1500))

reg.700=nls(resp~a*mass^b,data=short[short$pco2==700,],start=list(a=50,b=-0.7))
seq.700=seq(from=min(short$mass[short$pco2==700]),to=max(short$mass[short$pco2==700]),
            length.out=100)
pred.700=predict(reg.700,newdata = data.frame(mass=seq.1500))

reg.360=nls(resp~a*mass^b,data=short[short$pco2==360,],start=list(a=50,b=-0.7))
```

```
seq.360=seq(from=min(short$mass[short$pco2==360]),to=max(short$mass[short$pco2==360]),
            length.out=100)
pred.360=predict(reg.360,newdata = data.frame(mass=seq.360))
```

Power regressions of long-term data

```
long.1=long[long$week==1,]
long.3=long[long$week==5,]
reg3.1500=nls(resp_mean~a*mass^b,data=long.3[long.3$pco2==1500,],start=list(a=50,b=-0.7))
seq3.1500=seq(from=min(long.3$mass[long.3$pco2==1500]),to=max(long.3$mass[long.3$pco2==1500]),
              length.out=100)
pred3.1500=predict(reg3.1500,newdata = data.frame(mass=seq.1500))

reg1.1500=nls(resp_mean~a*mass^b,data=long.1[long.1$pco2==1500,],start=list(a=50,b=-0.7))
seq1.1500=seq(from=min(long.1$mass[long.1$pco2==1500]),to=max(long.1$mass[long.1$pco2==1500]),
              length.out=100)
pred1.1500=predict(reg1.1500,newdata = data.frame(mass=seq1.1500))

reg3.700=nls(resp_mean~a*mass^b,data=long.3[long.3$pco2==700,],start=list(a=5,b=-0.7))
seq3.700=seq(from=min(long.3$mass[long.3$pco2==700]),to=max(long.3$mass[long.3$pco2==700]),
             length.out=100)
pred3.700=predict(reg3.700,newdata = data.frame(mass=seq3.1500))

reg1.700=nls(resp_mean~a*mass^b,data=long.1[long.1$pco2==700,],start=list(a=5,b=-0.7))
seq1.700=seq(from=min(long.1$mass[long.1$pco2==700]),to=max(long.1$mass[long.1$pco2==700]),
             length.out=100)
pred1.700=predict(reg1.700,newdata = data.frame(mass=seq1.1500))
```

Estimated Marginal Means

I had to use a work-around to get the estimated marginal means from short-term and long-term ANCOVAs into a form I could pull the values for plotting. This included writing the data out to a csv, then reading it back in.

```
write.csv(emmeans(long.lme,~pco2+week),"long_emmeans.csv")
long.em=read.csv("long_emmeans.csv")
write.csv(emmeans(short.lme,"pco2"),"short_emmeans.csv")
short.em=read.csv("short_emmeans.csv")
short.em$week=0
short.em=short.em[,c(1,2,8,3:7)]
adj=rbind(long.em,short.em)
```

Producing the plot

```
graph.width.left=0.7
graph.width.right=0.425
graph.height=0.421
```

```
png("Figure_2.png",width=5,height=7,units="in",pointsize=6,res=300)
#svg("Figure_2.svg",width=5,height=7,pointsize=6)
par(fig=c(0.05,graph.width.left,1-graph.height,1))
plot(resp~mass,data=short[short$pco2==1500,],log="xy",pch=21,bg="black",ylim=c(0.5,10),
     xlim=c(15,400),axes=F,xlab="",ylab="",cex=1.4,type="n")
lines(seq.700,pred.700,col="grey60",lwd=3,lty=1)
lines(seq.360,pred.360,col="black",lwd=2,lty=3)
lines(seq.1500,pred.1500,col="black",lwd=3,lty=1)
points(resp~mass,data=short[short$pco2==360,],pch=21,bg="white",cex=1.4)
points(resp~mass,data=short[short$pco2==700,],pch=21,bg="grey",cex=1.4)
points(resp~mass,data=short[short$pco2==1500,],pch=21,bg="black",cex=1.4)
box(lwd=2)
axis(2,cex.axis=1.5,at=c(0.5,1,2,4,8),labels=c("",1,2,4,8))
text(25,8,"0 wk",cex=2)

par(fig=c(0.05,graph.width.left,0.5-(graph.height/2),0.5+(graph.height/2)),new=T)
plot(resp_mean~mass,data=long.3[long.3$pco2==700,],log="xy",pch=21,bg="green",xlim=c(15,400),
     ylim=c(0.5,10),axes=F,type="n",ylab="",xlab="")
lines(seq1.1500,pred1.1500,col="black",lwd=3,lty=1)
lines(seq1.700,pred1.700,col="grey60",lwd=3,lty=1)
points(resp_mean~mass,data=long.1[long.1$pco2==700,],pch=21,bg="grey",cex=1.4)
points(resp_mean~mass,data=long.1[long.1$pco2==1500,],pch=21,bg="black",cex=1.4)
box(lwd=2)
axis(2,cex.axis=1.5,at=c(0.5,1,2,4,8),labels=c("",1,2,4,8))
text(25,8,"1 wk",cex=2)


par(fig=c(0.05,graph.width.left,0,graph.height),new=T)
plot(resp_mean~mass,data=long.3[long.3$pco2==700,],log="xy",pch=21,bg="green",xlim=c(15,400),
     ylim=c(0.5,10),axes=F,type="n",ylab="",xlab="")
lines(seq3.1500,pred3.1500,col="black",lwd=3,lty=1)
lines(seq3.700,pred3.700,col="grey60",lwd=3,lty=1)
points(resp_mean~mass,data=long.3[long.3$pco2==1500,],pch=21,bg="black",cex=1.4)
points(resp_mean~mass,data=long.3[long.3$pco2==700,],pch=21,bg="grey",cex=1.4)
box(lwd=2)
axis(1,cex.axis=1.5)
axis(2,cex.axis=1.5,at=c(0.5,1,2,4,8),labels=c(0.5,1,2,4,8))
mtext("Mass (g)",side=1,cex=1.6,line=2.5)
text(25,8,"5 wk",cex=2)


par(fig=c(1-graph.width.right,1,1-graph.height,1),new=T)
plotCI(1:3,adj$emmean[adj$week==0],ui=adj$upper.CL[adj$week==0],li=adj$lower.CL[adj$week==0],
       xlab="",ylab="",pch=21,pt.bg=c("white","grey","black"),lwd=1,cex=1.8,axes=F,
       ylim=c(0.5,10),xlim=c(0,4),log="y")
box(lwd=2)
axis(4,cex.axis=1.5,at=c(0.5,1,2,4,8),labels=c("","","","",""))
points(3,adj$upper.CL[adj$week==0][3]+1,pch="*",cex=4)

par(fig=c(1-graph.width.right,1,0.5-(graph.height/2),0.5+(graph.height/2)),new=T)
plotCI(2:3,adj$emmean[adj$week==1],ui=adj$upper.CL[adj$week==1],li=adj$lower.CL[adj$week==1],
       xlab="",ylab="",pch=21,pt.bg=c("grey","black"),lwd=1,cex=1.8,axes=F,ylim=c(0.5,10),
       xlim=c(0,4),log="y")
```

```
box(lwd=2)
axis(4,cex.axis=1.5,at=c(0.5,1,2,4,8),labels=c("","","","",""))

par(fig=c(1-graph.width.right,1,0,graph.height),new=T)
plotCI(2:3,adj$emmean[adj$week==5],ui=adj$upper.CL[adj$week==5],li=adj$lower.CL[adj$week==5],
       xlab="",ylab="",pch=21,pt.bg=c("grey","black"),lwd=1,cex=1.8,axes=F,ylim=c(0.5,10),
       xlim=c(0.5,3.5),log="y")
box(lwd=2)
axis(4,cex.axis=1.5,at=c(0.5,1,2,4,8),labels=c("","","","",""))
axis(1,cex.axis=1.5,at=c(1,2,3),labels=c("360","700","1500"))
mtext(expression("pCO"[2]*" ("*mu*"atm)"),side=1,cex=1.6,line=3)

par(fig=c(0.05,1,0,1),new=T)
plot(c(0,1,0,1),type="n",axes=F,ylab="",xlab="")
mtext(expression("Routine Metabolic Rate ("*mu*"molO"[2]*" g"^-1*"hr"^-1*")"),side=2,cex=1.5,line=3)

dev.off()
```

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

Plotting a summary of estimated marginal means

```
stag=.06

png(file="Figure_3.png",height=3.5,width=3.5,units="in",pointsize=6,res=300)
#svg(file="Figure_3.svg",height=3.5,width=3.5,pointsize=6)

par(fig=c(0.04,1,0,1))
plotCI(adj$week+c(-1*stag,stag,-1*stag,stag,-2*stag,0,2*stag),adj$emmean,
       ui=adj$upper.CL,li=adj$lower.CL,xlab="",ylab="",pch=21,
       pt.bg=c("grey","black","grey","black","white","grey","black"),lwd=0,
       cex=1,axes=F)
box(lwd=2)
axis(1,at=c(0,1,5),lwd=2,cex.axis=1.5)
axis(2,lwd=2,cex.axis=1.5,at=c(1,2,3,4))
mtext(expression("Routine Metabolic Rate ("*mu*"molO"[2]*" g"^-1*"hr"^-1*")"),
      side=2,cex=1.8,line=2.5)
mtext("Weeks in Treatment",side=1,cex=1.8,line=2.5)
lines(c(1,5),adj$emmean[c(1,3)]-.01,lwd=2.5,col="black",lty=2)
lines(c(1,5),adj$emmean[c(2,4)]+.01,lwd=2.5,col="black")
plotCI(adj$week+c(-1*stag,stag,-1*stag,stag,-2*stag,0,2*stag),adj$emmean,
       ui=adj$upper.CL,li=adj$lower.CL,xlab="",ylab="",pch=21,
       pt.bg=c("grey","black","grey","black","white","grey","black"),lwd=1,
       cex=1.8,axes=F,add=T)
legend("topright", c(expression("1500 "*mu*"atm"),expression("700 "*mu*"atm"),
                     expression("360 "*mu*"atm")), pch = 21,bty="n",
       title = expression("Treatment pCO"["2"]),pt.bg=c("black","grey","white"),
       inset = .02,cex=1.8,box.lwd=2,pt.lwd=1)
dev.off()
```

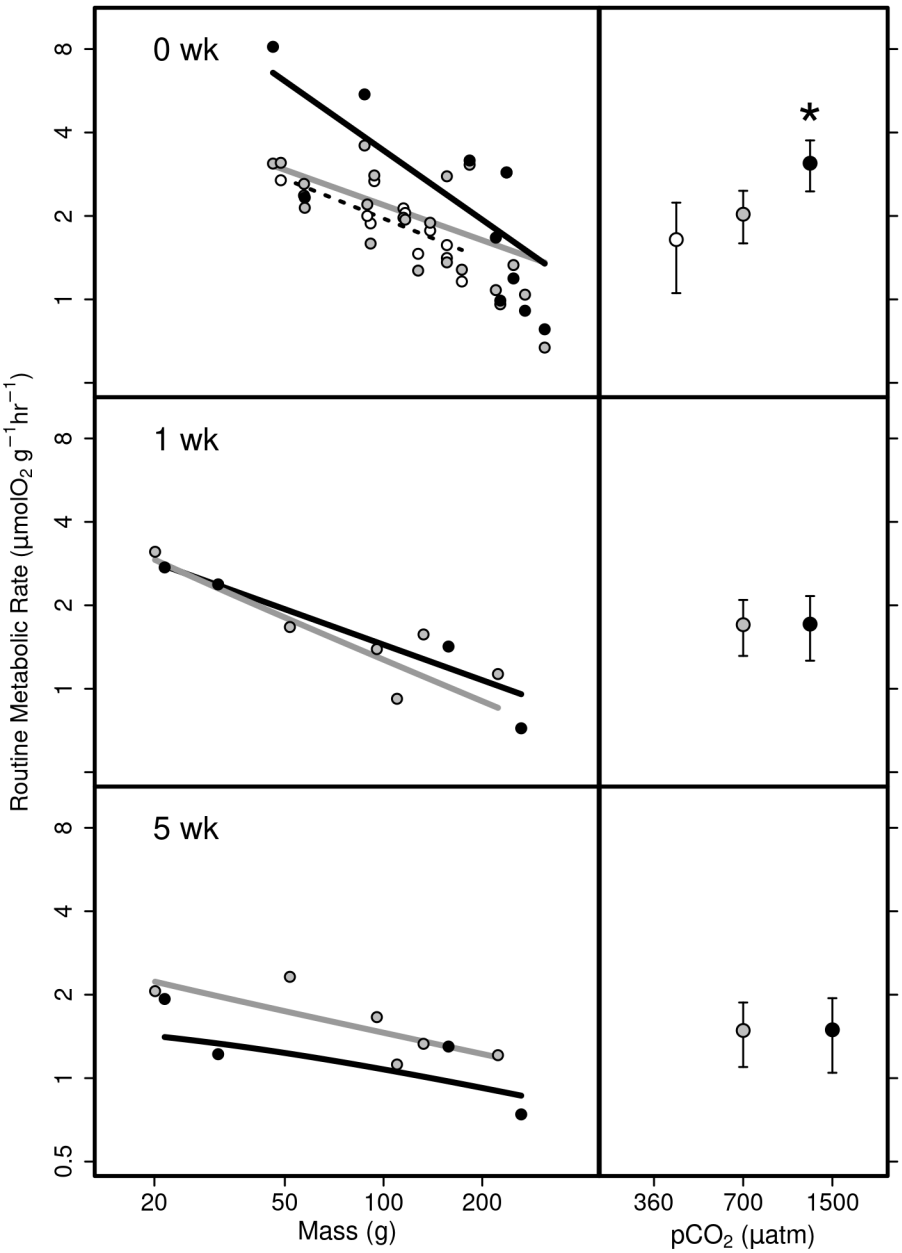


Figure 2: Routine metabolic rates (RMR) of Octopus rubescens at three CO2 partial pressures...

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

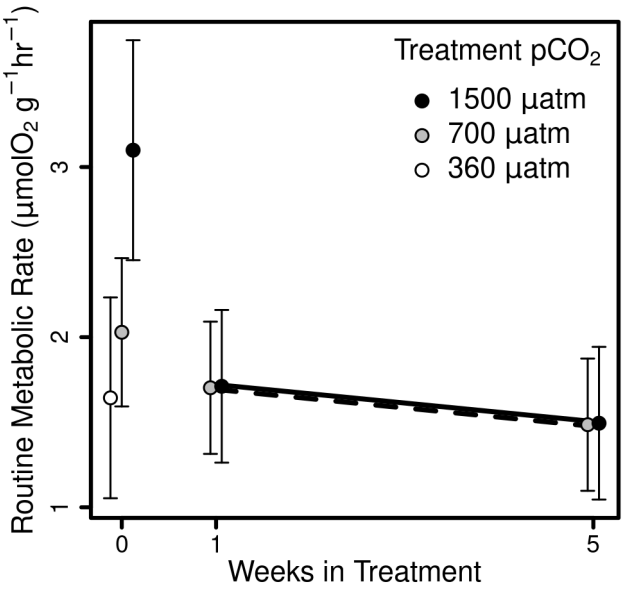


Figure 3: Estimated marginal means of routine metabolic rates of Octopus rubescens...

Critical oxygen pressure

One-tailed permutation t-test of P_{CRIT} values

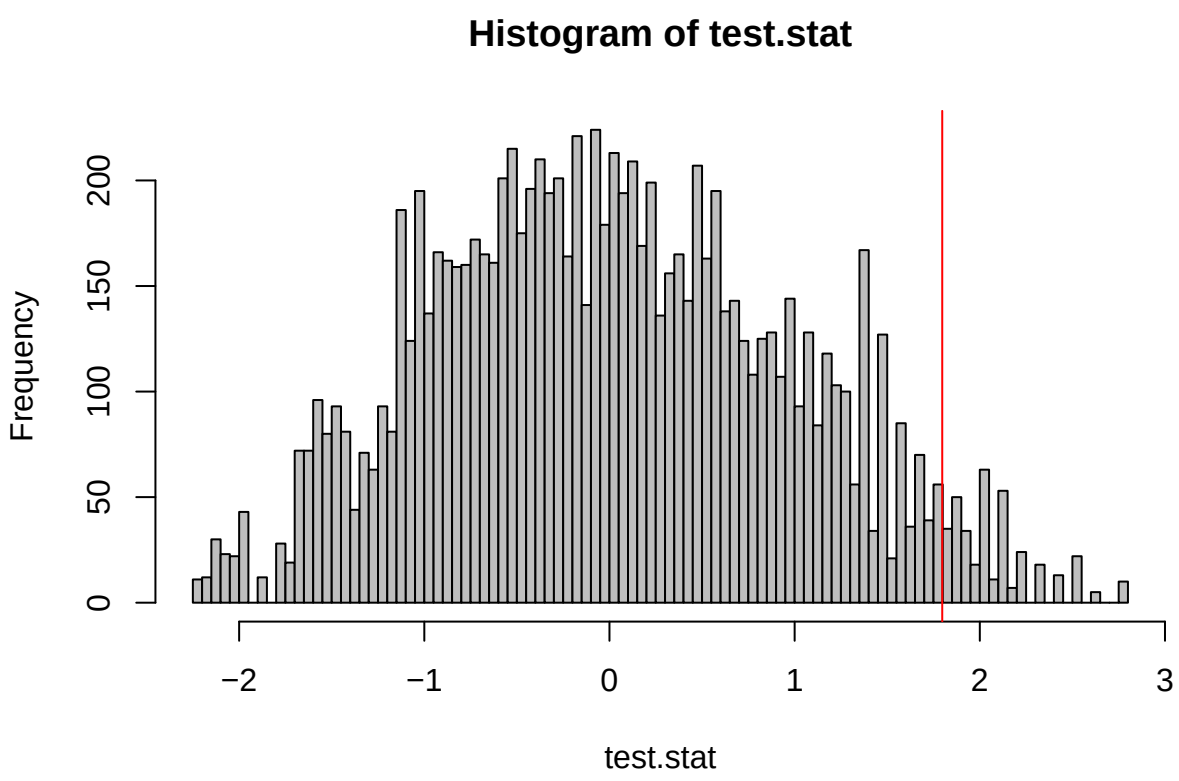
I am using a permutation test here because there are only 4 values at 1500 uatm

```
pcrit=read.csv("Pcrit_2014.csv")
pcrit$pco2=as.numeric(pcrit$pco2)
pcrit$set=1
pcrit$set[pcrit$pco2==1500]=2
set.seed(56)
obs.stat=
  mean(pcrit$Pc[pcrit$pco2==1500])-
  mean(pcrit$Pc[pcrit$pco2==700])

test.stat=0

for (i in 1:10000) {
  temp=pcrit
  temp$pco2=sample(temp$pco2)
  test.stat[i]=mean(temp$Pc[temp$pco2==1500])-mean(temp$Pc[temp$pco2==700])
}
```

```r
hist(test.stat,breaks=100,col="grey")
abline(v=obs.stat,col="red")
```




```r
test.p=sum(test.stat>=obs.stat)/length(test.stat)
test.p
```

```
## [1] 0.0386
```

```r
png(file="Figure_4.png",height=3,width=3.3,units="in",pointsize=3,res=300)
#svg(file="Figure_4.svg",height=3,width=3.3,pointsize=3)
par(fig=c(0.05,1,0.05,1))
boxplot(Pc~pco2,data=pcrit,col="grey",axes=F,ylim=c(2.5,9.5),ylab="",xlab="")
axis(1,lwd=2,cex.axis=2.5,padj=1,at=c(1,2),labels=c(700,1500))
axis(2,lwd=2,cex.axis=2.5)
box(lwd=2)
mtext(expression("P"["crit"]*" (kPa)"),side=2,cex=3,line=3.5)
mtext(expression("pCO"[2]*" ("*mu*"atm)"),side=1,cex=3,line=6)
points(Pc~jitter(set,factor=0.1),data=pcrit,pch=21,bg="gray25",lwd=.5,cex=2.5)
dev.off()
```

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

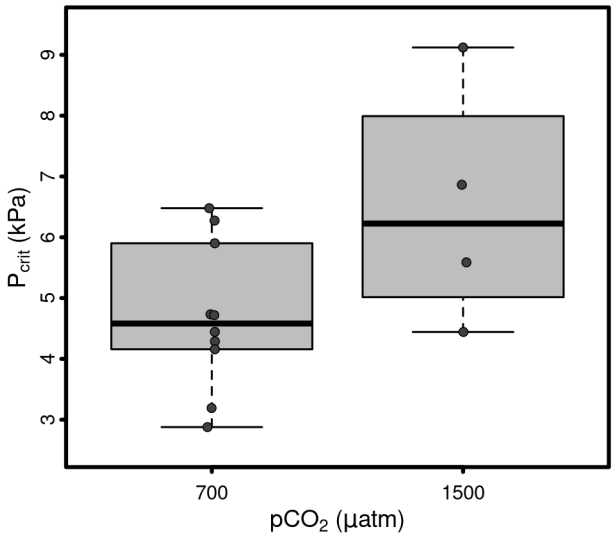


Figure 4: Critical oxygen pressure of Octopus rubescens...