

Figure S1. Length and width as proxies for zooid area. While zooid length times width is an excellent proxy for zooid area for both zooids and ovicells (adjusted R-squares 0.9596 and 0.9567, respectively), they separately capture less of the variation in zooid area (0.7688, 0.7454 for length and width, respectively) and for ovicell area (0.865, 0.9263 for length and width, respectively, see Figure S2). Hence, we plot univariate linear models to show that the relationships of length and width to SST, chl-a and absolute latitude are largely similar among zooid area, width and length.

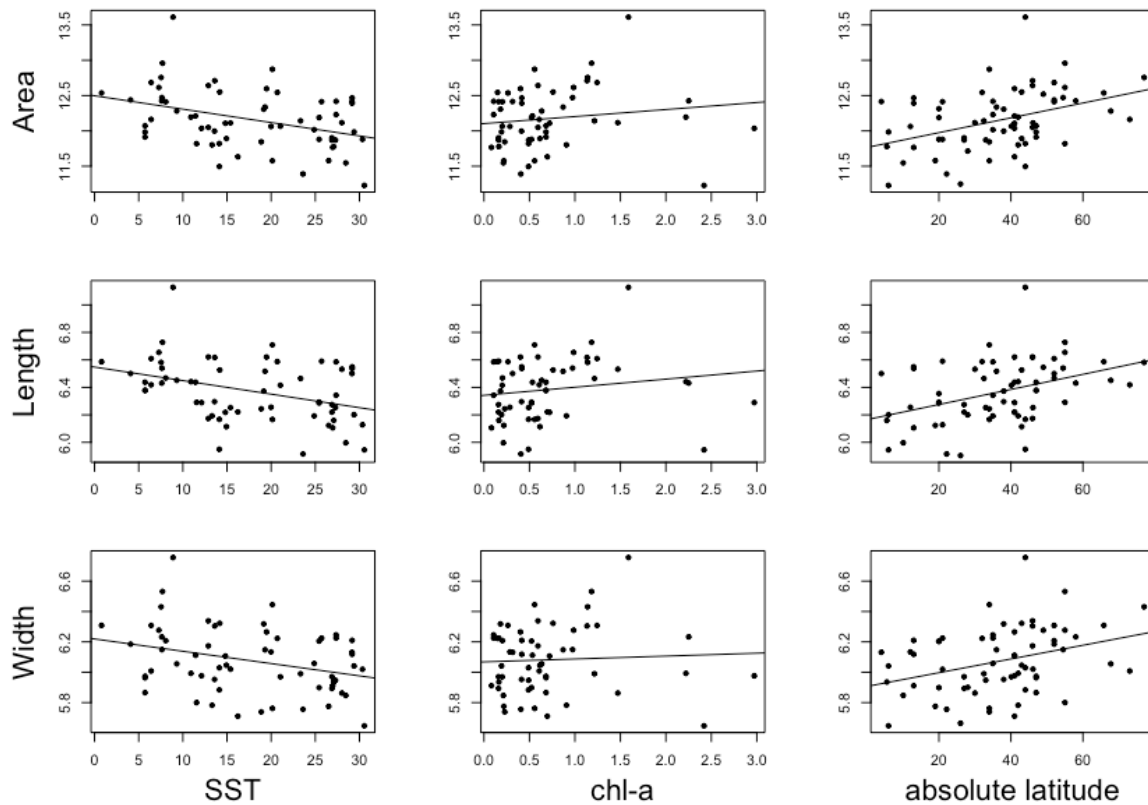


Figure S2. Length and width as proxies for ovicell (larval) area. Univariate linear models as in Figure S1, but for ovicell (larval) area.

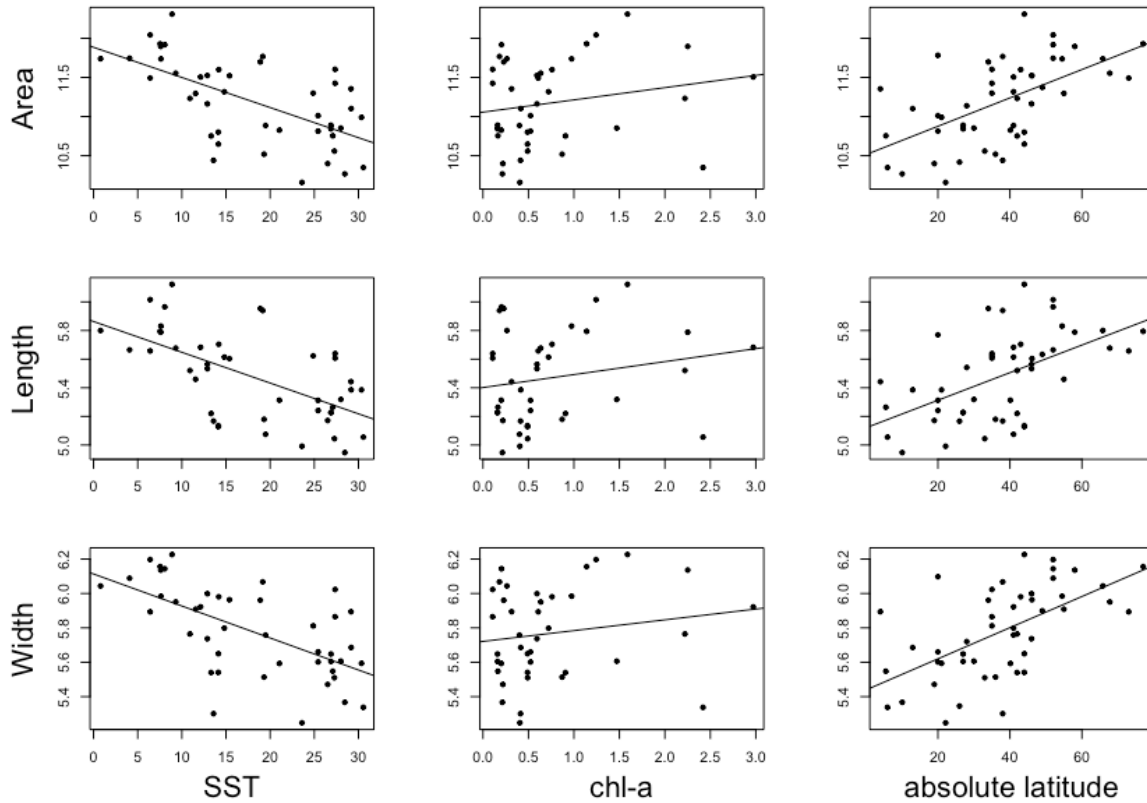
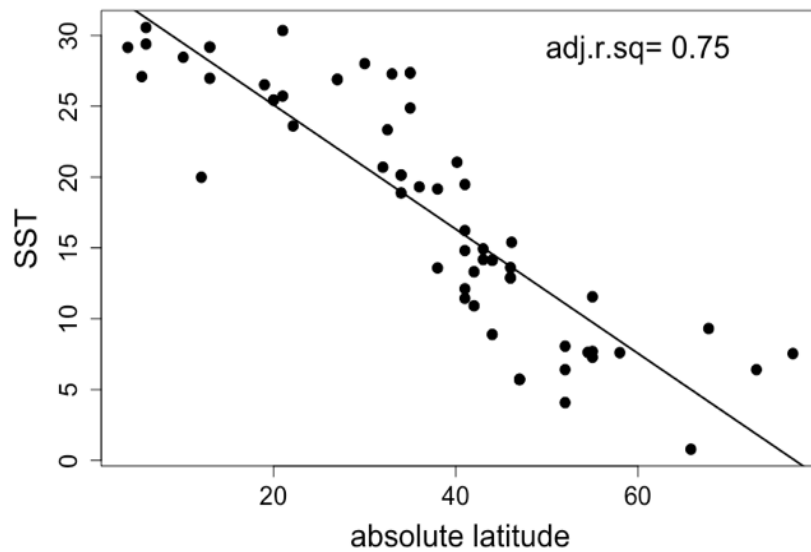


Figure S3. Relationship between latitude and SST for contemporary *Microporella* locations. Absolute latitude explains SST relative well (adjusted R-square = 0.75), where 1 degree latitude move towards the equator gives a 0.43°C increase in SST.



Code S1.

```
---
title: "microporella ovicell ms"
output:
  html_document: default
  pdf_document: default
---

```${r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## to do: add from literature

```${r, echo=F, warning=F,message=FALSE, results='hide'}
rm(list=ls())
setwd("/Users/leehl/Documents/Emanuela di Martino/Microporella")
library(readxl)
library(plyr)
library(magrittr)
library(readr)
library(dplyr)
library(MuMIn)
library(knitr)

`%notin%` <- Negate(`%in%`) #function
#https://stackoverflow.com/questions/24305271/extracting-standardized-coefficients-
from-lm-in-r
lm.beta = function (MOD)
{
 b <- summary(MOD)$coef[-1, 1]
 sx <- sd(MOD$model[-1])
 sy <- sd(MOD$model[1])
 beta <- b * sx/sy
 return(beta)
}
#####

Data= read_excel("Microporella_dataset.final.10.09.2020.xlsx", 1) #1336 30
fec= read_excel("Microporella_dataset.final.10.09.2020.xlsx", 3)
Data=join(Data,fec, by="Image", type="left")
Data=Data[which(Data$Lat!="unknown"),]

Data=subset(Data, select=c("Genus", "Species", "Min age", "Max age", "Image", "Lat",
"Long", "PaleoLat", "Zooid N", "Z_area", "ZL", "ZW", "Ov_area", "OvL","OvW", "N_ non-
ovicellate", "N_ovicellate", "Total"))
```

```
colnames(Data)=c("Genus", "Species", "Min.age", "Max.age", "Image", "Lat", "Long",
"PaleoLat", "Zooid.N", "Z_area", "ZL", "ZW", "Ov_area", "OvL", "OvW", "N_non-ovicellate",
"N_ovicellate", "Total")
Data$taxon=paste(Data$Genus, Data$Species)
Data=Data[which(!is.na(Data$Min.age)),] #remove those with no data at all N=8 species
Recent.edm=Data[which(Data$Min.age==0),] #Data is EDM data Recent.edm is edm Recent
data
#length(unique(Recent.edm$taxon))#58 measured Recent species in EDM
#length(unique((Data[which(Data$Min.age>0),]$Species)))#27 measured Recent species in
EDM
```

```
Old= read_excel("Microporella_dataset.final.10.09.2020.xlsx", 2) #218 19
Old=join(Old,fec, by="Image", type="left")
Old=Old[which(!is.na(Old$Lat)),]
```

```
Old=subset(Old, select=c("Genus", "Species_updated", "Min age", "Max age", "Image",
"Lat", "Long", "PaleoLat", "Zooid N", "Z_area", "ZL", "ZW", "Ov_area", "OvL", "OvW",
"N_non-ovicellate", "N_ovicellate", "Total"))
colnames(Old)=c("Genus", "Species", "Min.age", "Max.age", "Image", "Lat", "Long",
"PaleoLat", "Zooid.N", "Z_area", "ZL", "ZW", "Ov_area", "OvL", "OvW", "N_non-ovicellate",
"N_ovicellate", "Total")
Old$taxon=paste(Old$Genus, Old$Species)
kickout=-which(Old$Image %in% Recent.edm$Image) #repeats
Old=Old[c(kickout),]
```

```
Old=Old[which(!is.na(Old$Min.age)),]
Recent.Old=Old[which(Old$Min.age==0),]
Recent=rbind(Recent.Old, Recent.edm)
#length(unique(Recent$taxon)) N= 66 species with 926 rows this is EDM and LT combined,
so LT added 66-58=8 species;
```

```
#####latitude and longitude fixed####
```

```
#Make latitude readable (remove zero and substitute S and N)
```

```
Lat.new=as.data.frame(c(Recent$Lat))
```

```
colnames(Lat.new)=c("Lat.new")
```

```
Lat.new$no.zero=sub("^0+", "", Lat.new$Lat.new) #remove first zeros
```

```
Lat.new$newS=(as.numeric(gsub("S", " ", Lat.new$no.zero)))
```

```
Lat.new$newN=(as.numeric(gsub("N", " ", Lat.new$no.zero)))
```

```
Lat.new$new = Lat.new$newN # your new merged column start with x
```

```
Lat.new$new [!is.na(Lat.new$newN)] = Lat.new$newN [!is.na(Lat.new$newN)] # merge
with y
```

```
Lat.new$new [!is.na(Lat.new$newS)] = Lat.new$newS [!is.na(Lat.new$newS)] # merge with
y
```

```
#Make longitude readable (remove zero and substitute E and W)
```

```

Long.new=as.data.frame(c(Recent$Long))
colnames(Long.new)=c("Long.new")
Long.new$no.zero=sub("^0+", "", Long.new$Long.new) #remove first zeros

Long.new$newW=-(as.numeric(gsub("W", " ", Long.new$no.zero)))
Long.new$newE=(as.numeric(gsub("E", " ", Long.new$no.zero)))

Long.new$new = Long.new$newW # your new merged column start with x
Long.new$new [!is.na(Long.new$newW)] = Long.new$newW [!is.na(Long.new$newW)] #
merge with y
Long.new$new [!is.na(Long.new$newE)] = Long.new$newE [!is.na(Long.new$newE)] #
merge with y

#####
#####
#Make a new column for age mean and clean latitude and clean longitude
Recent$Lat.clean= Lat.new$new
Recent$Long.clean= Long.new$new
#####

library(dplyr)
setwd("/Users/leehl/Documents/Emanuela di Martino/Microporella/SST/")
lat=seq(89.5,-89.5, -1)
long=seq(-179.5,179.5, 1)
#note that these files have to be down loaded from NASA
files.ss <- list.files(path = "/Users/leehl/Documents/Emanuela di Martino/Microporella/SST",
pattern = "*SS.CSV")
tbl=lapply(files.ss, read_csv, skip=0) %>% bind_rows(.id = "id")#column one is "id" and
column 2 is lat/lon dim(6840 of 38 datasets and 362 columns)
tbl[tbl>1000]<-NA

Recent$SST=NULL
for (i in 1:dim(Recent)[1]){
long.cell=which(long==Recent$Long.clean[i]+.5)+2 #the column that is correct, plus 2
because of the extra 2 columns; long.cell points to which cell to use in the tbl
lat.cell=which(lat==Recent$Lat.clean[i]+.5)
lat.cell.1=which(lat==Recent$Lat.clean[i]+1.5) #go up and down two lats.
lat.cell.2=which(lat==Recent$Lat.clean[i]-1.5)
if (!is.na(Recent$Long.clean[i])){
lat.cells=seq(lat.cell, dim(tbl)[1], 180)
Recent$SST[i]=mean(unlist(tbl[c(lat.cells, lat.cell.1, lat.cell.2),long.cell]),
na.rm=T)}else{Data$SST[i]=NA}
}

Recent$SST[which(Recent$SST>1000)]=NA #convert the "empty" cells

```

```

library(plyr)
###second set of data without lats and longs LH IS HERE
no.files.ss <- list.files(path = "/Users/leehl/Documents/Emanuela di
Martino/Microporella/SST", pattern = "*180.CSV") #with no. SS
dat_csv = ldply(no.files.ss, .fun=read.csv, header=T)
dat_csv=dat_csv[,2:361]
dat_csv[dat_csv>1000]<-NA

Recent$SST.1=NULL
for (i in 1:dim(Recent)[1]){
 long.cell=which(long==Recent$Long.clean[i]+.5) #the column that is correct, plus 1 because
of the one extra columns
 lat.cell=which(lat==Recent$Lat.clean[i]+.5)
 lat.cell.1=which(lat==Recent$Lat.clean[i]+1.5) #go up and down two lats.
 lat.cell.2=which(lat==Recent$Lat.clean[i]-1.5)

 if (!is.na(Recent$Long.clean[i])){
 lat.cells=seq(lat.cell, dim(dat_csv)[1], 180)
 Recent$SST.1[i]=mean(unlist(dat_csv[c(lat.cells,lat.cell.1, lat.cell.2), long.cell]),
na.rm=T)}else{Recent$SST.1[i]=NA}
}
Recent$SST.1[which(Recent$SST.1>1000)]=NA #convert the "empty" cells

Recent$SST.final=(Recent$SST.1+Recent$SST)/2
#####

#####

##CHLOROPHYLL A
#pick out data and make mean
setwd("/Users/leehl/Documents/Emanuela di Martino/Microporella/CHLOR-A.nasa/")
files.ss <- list.files(path = "/Users/leehl/Documents/Emanuela di
Martino/Microporella/CHLOR-A.nasa", pattern = "*SS.CSV") #with SS
#20 files
tbl=lapply(files.ss, read_csv, skip=0) %>% bind_rows(.id = "id")#column one is "id" and
column 2 is lat/lon dim(6840 of 38 datasets and 362 columns)
tbl[tbl>1000]<-NA
lat=seq(89.5,-89.5, -1)
long=seq(-179.5,179.5, 1)
Recent$chl.1=NULL
for (i in 1:dim(Recent)[1]){
 long.cell=which(long==Recent$Long.clean[i]+.5)+2 #the column that is correct, plus 2
because of the extra 2 columns
 lat.cell=which(lat==Recent$Lat.clean[i]+.5)
 lat.cell.1=which(lat==Recent$Lat.clean[i]+1.5) #go up and down two lats.
 lat.cell.2=which(lat==Recent$Lat.clean[i]-1.5)

```

```

#if (!is.na(Data$Long.clean[i])){
 lat.cell=seq(lat.cell, dim(tbl)[1], 180)
 Recent$chl.1[i]=mean(unlist(tbl[c(lat.cell, lat.cell.1, lat.cell.2), long.cell]), na.rm=T)
 #}else{Data$chl.1[i]=NA}

}
Recent$chl.1[which(Recent$chl.1>1000)]=NA #convert the "empty" cells

no.files.ss <- list.files(path = "/Users/leehl/Documents/Emanuela di
Martino/Microporella/CHLOR-A.nasa", pattern = "*180.CSV") #with no. SS
dat_csv = ldply(no.files.ss, read_csv)
dat_csv[dat_csv>1000]<-NA
####second set of data without lats and longs
Recent$chl.2=NULL
for (i in 1:dim(Recent)[1]){
 long.cell=which(long==Recent$Long.clean[i]+.5)+1 #the column that is correct, plus 1
 because of the one extra columns
 lat.cell=which(lat==Recent$Lat.clean[i]+.5)
 lat.cell.1=which(lat==Recent$Lat.clean[i]+1.5) #go up and down two lats.
 lat.cell.2=which(lat==Recent$Lat.clean[i]-1.5)

 if (!is.na(Recent$Long.clean[i])){
 lat.cell=seq(lat.cell, dim(dat_csv)[1], 180)
 Recent$chl.2[i]=mean(unlist(tbl[c(lat.cell, lat.cell.1, lat.cell.2),
long.cell]),na.rm=T)}else{Data$SST[i]=NA}

}
Recent$chl.2[which(Recent$chl.2>1000)]=NA #convert the "empty" cells

Recent$chl=(Recent$chl.1+Recent$chl.2)/2
```



##ovicell as focus  

Using averages of taxa, model ovicell area with different covariates to select best AIC model.  

The full model is  

 $\ln(O \sim Z + fec + SST + chl)$ . Here  $N = 66$ , best model adj R sq is 0.6558 (note that Z alone has  

Adjusted R-squared: 0.6027; SST alone has Adjusted R-squared: 0.3671. chl alone only  

Adjusted R-squared: 0.01281. Fec alone Adjusted R-squared: 0.04768 That means that  

really Z explains a lot of the O)



```

```{r, echo=F, warning=F}

#####
summary(as.numeric(table(Recent$Species)))

Z=tapply(log(Recent$Z_area), Recent$taxon, mean, na.rm=T)

```


```

```
O=tapply(log(Recent$Ov_area), Recent$taxon, mean, na.rm=T)
fec=tapply(Recent$N_ovicellate/(Recent$N_ovicellate+Recent$`N_non-ovicellate`),
Recent$taxon, function(x)x[1])
SST=tapply(Recent$SST.final, Recent$taxon, mean, na.rm=T)
chl=tapply(Recent$chl, Recent$taxon, mean, na.rm=T)
tax=tapply(Recent$taxon, Recent$taxon, function(x)x[1])
latitude=tapply(Recent$Lat.clean, Recent$taxon, mean, na.rm=T)
Rec=as.data.frame(bind_cols(Z,O,fec, SST,chl, latitude))
rownames(Rec)=tax
colnames(Rec)=c("Z", "O", "fec", "SST", "chl", "latitude")
dim(Rec)
```

```
shapiro.test(O)#O is normally distributed (Recent data)
hist(O)
```

```
mod.1.1=lm(O~Z)
mod.1.2=lm(O~fec)
mod.1.3=lm(O~SST)
mod.1.3.latitude=lm(O~abs(latitude))
mod.1.4=lm(O~chl)
```

```
mod.2.1=lm(O~Z+fec)
mod.2.2=lm(O~Z+SST)
mod.2.2.latitude=lm(O~Z+abs(latitude))
```

```
mod.2.3=lm(O~Z+chl)
mod.2.4=lm(O~fec+SST)
mod.2.4.latitude=lm(O~fec+abs(latitude))
```

```
mod.2.5=lm(O~fec+chl)
mod.2.6=lm(O~SST+chl)
mod.2.6.latitude=lm(O~abs(latitude)+chl)
```

```
mod.3.1=lm(O~Z+fec+SST) #best
mod.3.1.latitude=lm(O~Z+fec+abs(latitude)) #best
```

```
mod.3.2=lm(O~Z+fec+chl)
mod.3.3=lm(O~fec+SST+chl)
mod.3.3.latitude=lm(O~fec+abs(latitude)+chl)
mod.3.4=lm(O~Z+SST+chl)
mod.3.4.latitude=lm(O~Z+abs(latitude)+chl)
mod.full=lm(O~Z+fec+SST+chl)
mod.full.latitude=lm(O~Z+fec+abs(latitude)+chl)
```



```
x=model.sel(mod.1.1,mod.1.2, mod.1.3, mod.1.4, mod.2.1,mod.2.2, mod.2.3, mod.2.4,
mod.2.5, mod.2.6, mod.3.1, mod.3.2, mod.3.3, mod.3.4, mod.full)
kable(x)
x.latitude=model.sel(mod.1.1,mod.1.2, mod.1.3.latitude, mod.1.4, mod.2.1,mod.2.2.latitude,
mod.2.3, mod.2.4, mod.2.5, mod.2.6, mod.3.1.latitude, mod.3.2, mod.3.3.latitude,
mod.3.4.latitude, mod.full.latitude)
kable(x.latitude)
summary(mod.3.1)#best model
plot(mod.3.1)
```

```
temp=cbind(tapply(log(Recent$Z_area), Recent$taxon, mean),tapply((Recent$Lat.clean),
Recent$taxon, mean))
```

```
North=which(temp[,2]>0) #45
South=which(temp[,2]<0)# 21
t.test(temp[North],temp[South])
```

```
...
```

Do it for a reduced dataset where no data are missing N =30.

```
``{r, echo=F, warning=F}
Rec.1=na.omit(Rec)
mod.1.1=lm(O~Z, data=Rec.1) #Adjusted R-squared: 0.6446
mod.1.2=lm(O~fec, data=Rec.1)# Adjusted R-squared: 0.1233
mod.1.3=lm(O~SST, data=Rec.1)#Adjusted R-squared: 0.3075
mod.1.4=lm(O~chl, data=Rec.1)

mod.2.1=lm(O~Z+fec, data=Rec.1)
mod.2.2=lm(O~Z+SST, data=Rec.1) #best
mod.2.3=lm(O~Z+chl, data=Rec.1)
mod.2.4=lm(O~fec+SST, data=Rec.1)
mod.2.5=lm(O~fec+chl, data=Rec.1)
mod.2.6=lm(O~SST+chl, data=Rec.1)

mod.3.1=lm(O~Z+fec+SST, data=Rec.1) #best
mod.3.2=lm(O~Z+fec+chl, data=Rec.1)
mod.3.3=lm(O~fec+SST+chl, data=Rec.1)
mod.3.4=lm(O~Z+SST+chl, data=Rec.1)
mod.full=lm(O~Z+fec+SST+chl, data=Rec.1)
```

```
x.less=model.sel(mod.1.1,mod.1.2, mod.1.3, mod.1.4, mod.2.1,mod.2.2, mod.2.3, mod.2.4,
mod.2.5, mod.2.6, mod.3.1, mod.3.2, mod.3.3, mod.3.4, mod.full)
kable(x.less)
```

```
##extra14.12.2020
library(jtools)
summ(mod.2.2, scale=TRUE, n.sd=2, transform.response = TRUE) #standardized
summ(mod.2.2, vifs = TRUE) #VIF less than two
#Standard errors: OLS
#-----
Est. S.E. t val. p VIF
#-----
#(Intercept) 1.17 1.81 0.65 0.52
#Z 0.84 0.14 5.91 0.00 1.27
#SST -0.02 0.01 -2.19 0.04 1.27
#-----
#https://cran.r-project.org/web/packages/jtools/vignettes/summ.html
#library(rqs)
...

```

##estimates from the best ovicell model(reduced data) above  
Adjusted R-squared: 0.6704

```
``{r, echo=F, warning=F}
summary(lm(O~Z+SST, data=Rec.1))#best model
summary(lm(O~Z+SST, data=Rec))#best
mod.best=lm(O~Z+SST, data=Rec)#be
...

```

##best model plot

Fig. 3

Since the best model where all data are used have fecundity not significant with very low effect, we use the full data and the best model from the reduced.

slopes and std given, \* is p value significance, plot best model but with full dataset.

```
``{r, echo=F, warning=F}
par(mfrow=c(1,2), oma=c(1,1,1,1), mar=c(5,5,3,1))
plot(Z,O, main="", cex.axis=1.4, cex.lab=1.6, pch=20, col="darkgrey", cex=1.5,
 ylab="mean log ovicell (larval) size", xlab="mean log zooid size")
Z.new=seq(min(Z, na.rm=T),max(Z, na.rm=T),0.001)
#fec.new=rep(mean(fec, na.rm=T), length(Z.new))
SST.new=rep(mean(SST,na.rm=T), length(Z.new))
newdata=as.data.frame(list(Z=Z.new,SST=SST.new))
pred=predict(mod.best,newdata=newdata,se.fit=TRUE)
lines(newdata$Z, (pred$fit), lwd=3)
lines(newdata$Z, (pred$fit-1.96*pred$se.fit),
 col="darkgrey", lwd=2)
lines(newdata$Z, (pred$fit+1.96*pred$se.fit),

```

```

col="darkgrey", lwd=2)
text(13, 10.3, cex=1.5, paste(round(summary(mod.best)$coefficients[2,1], digits=2), "(",
round(summary(mod.best)$coefficients[2,2], digits=2), ")***"))
title("A.", line = 0.2, adj=0, cex.main=1.8)

#plot(fec,O, main="", ylab="", cex.axis=1.4, cex.lab=1.6, pch=20, col="darkgrey", cex=1.5)
#fec.new=seq(min(fec, na.rm=T),max(fec, na.rm=T),0.01)
#SST.new=rep(mean(fec, na.rm=T), length(fec.new))
#Z.new=rep(mean(Z,na.rm=T), length(fec.new))
#newdata=as.data.frame(list(Z=Z.new,fec=fec.new,SST=SST.new))
#pred=predict(mod.3.1,newdata=newdata,se.fit=TRUE)
#lines(newdata$fec, (pred$fit), lwd=3)
#lines(newdata$fec, (pred$fit-1.96*pred$se.fit),
col="darkgrey", lwd=2)
#lines(newdata$fec, (pred$fit+1.96*pred$se.fit),
col="darkgrey", lwd=2)
#text(0.6, 10.5, cex=1.5, paste(round(summary(mod.3.1)$coefficients[3,1], digits=2), "(",
round(summary(mod.3.1)$coefficients[3,2], digits=2), "#")"))
#title("B.", line = 0.2, adj=0, cex.main=1.8)

plot(SST,O, main="", ylab="", cex.axis=1.4, cex.lab=1.6, pch=20, col="darkgrey", cex=1.5,
, xlab="SST")
SST.new=seq(min(SST, na.rm=T),max(SST, na.rm=T),0.001)
fec.new=rep(mean(fec, na.rm=T), length(SST.new))
Z.new=rep(mean(Z,na.rm=T), length(SST.new))
newdata=as.data.frame(list(Z=Z.new,SST=SST.new))
pred=predict(mod.best,newdata=newdata,se.fit=TRUE)
lines(newdata$SST, (pred$fit), lwd=3)
lines(newdata$SST, (pred$fit-1.96*pred$se.fit),
col="darkgrey", lwd=2)
lines(newdata$SST, (pred$fit+1.96*pred$se.fit),
col="darkgrey", lwd=2)
text(22, 12, cex=1.5, paste(round(summary(mod.best)$coefficients[3,1], digits=2), "(",
round(summary(mod.best)$coefficients[3,2], digits=2), ")***"))
title("B.", line = 0.2, adj=0, cex.main=1.8)
```



##how close does SST reflect latitude for our dataset?  

It's not too bad, so let's go with it. Adjusted R-squared= 0.7502



```

```{r, echo=F, warning=F}
lat=tapply(Recent$Lat.clean, Recent$taxon, mean, na.rm=T)
par(mfrow=c(1,1))

#Fig. S1
par(mfrow=c(1,1))

```


```

```
plot(abs(lat), SST, pch=20, cex=2, xlab="absolute latitude", ylab="SST", cex.lab=1.8,
cex.axis=1.5)
temp=summary(lm(SST~abs(lat)))
abline(temp$coefficients[1,1], temp$coefficients[2,1], lwd=2)
text(60, 29, cex=1.8, paste("adj.r.sq=", round(temp$adj.r.squared, digits=2)))
```

```
par(mfrow=c(2,2))
plot(abs(lat), Z, pch=20,)
temp=summary(lm(Z~abs(lat)))
abline(temp$coefficients[1,1], temp$coefficients[2,1])
text(60, 13.5, paste("adj.r.sq=", round(temp$adj.r.squared, digits=2)))
```

```
plot(SST, Z, pch=20,)
temp=summary(lm(Z~SST))
abline(temp$coefficients[1,1], temp$coefficients[2,1])
text(25, 13.5, paste("adj.r.sq=", round(temp$adj.r.squared, digits=2)))
```

```
plot(abs(lat), O, pch=20,)
temp=summary(lm(O~abs(lat)))
abline(temp$coefficients[1,1], temp$coefficients[2,1])
text(60, 12, paste("adj.r.sq=", round(temp$adj.r.squared, digits=2)))
```

```
plot(SST, O, pch=20,)
temp=summary(lm(O~SST))
abline(temp$coefficients[1,1], temp$coefficients[2,1])
text(25, 12, paste("adj.r.sq=", round(temp$adj.r.squared, digits=2)))
```

...

```
###how close does L*W reflect area?
Really good. Adjusted R-squared: 0.9596 and 0.9567 respectively
```

```
``{r, echo=F, warning=F}
```

```
par(mfrow=c(2,1))
Recent$ LW=Recent$ZL*Recent$ZW
summary(lm(Recent$Z_area ~ Recent$ LW))
plot(Recent$ LW, Recent$Z_area, pch=20)
temp=lm(Recent$Z_area ~ Recent$ LW)
abline(temp$coefficients[1], temp$coefficients[2])
```

```
Recent$OvLW=Recent$OvL*Recent$OvW
summary(lm(Recent$Ov_area ~ Recent$OvLW))
plot(Recent$OvLW, Recent$Ov_area, pch=20)
temp=lm(Recent$Ov_area ~ Recent$OvLW)
abline(temp$coefficients[1], temp$coefficients[2])
...
```

##how well does L and W reflect area separately?

About 0.75.. So relationships may also be different

```
``{r, echo=F, warning=F}

plot(log(Recent$ZL), log(Recent$Z_area), pch=20)
temp=lm(log(Recent$Z_area) ~ log(Recent$ZL))
summary(temp)
abline(temp$coefficients[1], temp$coefficients[2])
plot(log(Recent$ZW), log(Recent$Z_area), pch=20)
temp=lm(log(Recent$Z_area) ~ log(Recent$ZW))
summary(temp)
abline(temp$coefficients[1], temp$coefficients[2])

plot(log(Recent$OvL), log(Recent$Ov_area), pch=20)
temp=lm(log(Recent$Ov_area) ~ log(Recent$OvL))
summary(temp)
abline(temp$coefficients[1], temp$coefficients[2])
plot(log(Recent$OvW), log(Recent$Ov_area), pch=20)
temp=lm(log(Recent$Ov_area) ~ log(Recent$OvW))
summary(temp)
abline(temp$coefficients[1], temp$coefficients[2])
``
```

#Do Zooid size analyses with Area, length and width (this is SI material?)

Relationships are stronger with Length, when Area is used as the standard. No relationship with chl.

```
``{r, echo=F, warning=F}
#####
#Fig. S2
#Z=tapply(log(Recent$Z_area), Recent$taxon, mean, na.rm=T)
Leng=tapply(log(Recent$ZL), Recent$taxon, mean, na.rm=T)
Width=tapply(log(Recent$ZW), Recent$taxon, mean, na.rm=T)

shapiro.test(Leng)#Length is ok normal p close to 0.5
shapiro.test(Width)#Width is normal
par(mfrow=c(3,3), mar=c(4,5,1,1))

summary(lm(Z~SST))
plot(SST, Z, pch=20, xlab="", ylab="Area", cex.lab=2)
temp=lm(Z~SST)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Z~chl))
```

```

plot(chl, Z, pch=20, xlab="", ylab=" ")
temp=lm(Z~chl)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Z~abs(lat)))
plot(abs(lat), Z, pch=20, xlab="", ylab="")
temp=lm(Z~abs(lat))
abline(temp$coefficients[1], temp$coefficients[2])

summary(lm(Leng~SST))
plot(SST, Leng, pch=20, ylab="Length",xlab="", cex.lab=2)
temp=lm(Leng~SST)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Leng~chl))
plot(chl, Leng, pch=20, ylab="",xlab="")
temp=lm(Leng~chl)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Leng~abs(lat)))
plot(abs(lat), Leng, pch=20, ylab="",xlab="")
temp=lm(Leng~abs(lat))
abline(temp$coefficients[1], temp$coefficients[2])

summary(lm(Width~SST))
plot(SST, Width, pch=20, ylab="Width",xlab="SST", cex.lab=2)
temp=lm(Width~SST)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Width~chl))
plot(chl, Width, pch=20, ylab="",xlab="chl-a", cex.lab=2)
temp=lm(Width~chl)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Width~abs(lat)))
plot(abs(lat), Width, pch=20, ylab="",xlab="absolute latitude", cex.lab=2)
temp=lm(Width~abs(lat))
abline(temp$coefficients[1], temp$coefficients[2])

...

#Do ovicell size analyses with Area, length and width (this is SI material?)

Relationships are stronger with Length, when Area is used as the standard. No relationship
with chl.

```{r, echo=F, warning=F}
#####
#Fig. S3
Leng=tapply(log(Recent$OvL), Recent$taxon, mean, na.rm=T)
Width=tapply(log(Recent$OvW), Recent$taxon, mean, na.rm=T)

```

```
shapiro.test(Leng)#Length is ok normal p close to 0.5
shapiro.test(Width)#Width is normal
par(mfrow=c(3,3), mar=c(4,5,1,1))
```

```
summary(lm(Z~SST))
plot(SST, O, pch=20, xlab="", ylab="Area", cex.lab=2)
temp=lm(O~SST)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(O~chl))
plot(chl, O, pch=20, xlab="", ylab=" ")
temp=lm(O~chl)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(O~abs(lat)))
plot(abs(lat), O, pch=20, xlab="", ylab="")
temp=lm(O~abs(lat))
abline(temp$coefficients[1], temp$coefficients[2])
```

```
summary(lm(Leng~SST))
plot(SST, Leng, pch=20, ylab="Length",xlab="", cex.lab=2)
temp=lm(Leng~SST)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Leng~chl))
plot(chl, Leng, pch=20, ylab="",xlab="")
temp=lm(Leng~chl)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Leng~abs(lat)))
plot(abs(lat), Leng, pch=20, ylab="",xlab="")
temp=lm(Leng~abs(lat))
abline(temp$coefficients[1], temp$coefficients[2])
```

```
summary(lm(Width~SST))
plot(SST, Width, pch=20, ylab="Width",xlab="SST", cex.lab=2)
temp=lm(Width~SST)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Width~chl))
plot(chl, Width, pch=20, ylab="",xlab="chl-a", cex.lab=2)
temp=lm(Width~chl)
abline(temp$coefficients[1], temp$coefficients[2])
summary(lm(Width~abs(lat)))
plot(abs(lat), Width, pch=20, ylab="",xlab="absolute latitude", cex.lab=2)
temp=lm(Width~abs(lat))
abline(temp$coefficients[1], temp$coefficients[2])
```

...

##Forward

The next thing to do is to use paleolatitude and check if fossil only data show the same type of relationship for Z and O. The best model above is

$\text{lm}(O \sim Z + \text{fec} + \text{SST} + \text{chl})$

so we will look at

$\text{lm}(O \sim Z + \text{fec} + \text{lat})$ with three data sets: Recent only, Fossil only and Recent and fossil

This first chunk shows the results for fossil only (total 45 species, but never the full dataset since some have NAs) -

```

`r, echo=F, warning=F}
Fossil.Old=Old[which(Old$Min.age>0),] #dim 30 19
Fossil.edm=Data[which(Data$Min.age>0),] # dim 405 19
Fossil=rbind(Fossil.Old, Fossil.edm)
setdiff(Fossil.Old$Species, Fossil.edm$Species )#19
#Fossil$OvLW=Fossil$OvL *Fossil$OvW
#Fossil$ZLW=Fossil$ZL *Fossil$ZW

#####DO THE PALEO LAT CONVERSION#####
Lat.new=as.data.frame(c(Fossil$PaleoLat))
colnames(Lat.new)=c("Lat.new")
Lat.new$no.zero=sub("^0+", "", Lat.new$Lat.new) #remove first zeros

Lat.new$newS=-(as.numeric(gsub("S", " ", Lat.new$no.zero)))
Lat.new$newN=(as.numeric(gsub("N", " ", Lat.new$no.zero)))

Lat.new$new = Lat.new$newN # your new merged column start with x
Lat.new$new [!is.na(Lat.new$newN)] = Lat.new$newN [!is.na(Lat.new$newN)] # merge
with y
Lat.new$new [!is.na(Lat.new$newS)] = Lat.new$newS [!is.na(Lat.new$newS)] # merge with
y

#Make a new column for age mean and clean latitude and clean longitude
Fossil$Lat.clean= Lat.new$new

#N = 45
lat.foss=tapply(Fossil$Lat.clean, Fossil$taxon, mean, na.rm=T)
tax.foss=tapply(Fossil$taxon, Fossil$taxon, function(x)x[1])

#Z.fos=tapply(log(Fossil$ZLW), Fossil$taxon, mean, na.rm=T)
#O.fos=tapply(log(Fossil$OvLW), Fossil$taxon, mean, na.rm=T)
fec.foss=tapply(Fossil$N_ovicellate/(Fossil$N_ovicellate+Fossil$`N_non-ovicellate`),
Fossil$taxon, function(x)x[1])
Z.fos.1=tapply(log(Fossil$Z_area), Fossil$taxon, mean, na.rm=T)
O.fos.1=tapply(log(Fossil$Ov_area), Fossil$taxon, mean, na.rm=T)

```



```
shapiro.test(O.fos.1)

foss.1.1=lm(O.fos.1~Z.fos.1)
foss.1.2=lm(O.fos.1~fec.foss)
foss.1.3=lm(O.fos.1~abs(lat.foss))

foss.2.1=lm(O.fos.1~Z.fos.1+fec.foss)
foss.2.2=lm(O.fos.1~Z.fos.1+abs(lat.foss))
foss.2.3=lm(O.fos.1~fec.foss+abs(lat.foss))
foss.full=lm(O.fos.1~Z.fos.1+fec.foss+abs(lat.foss))

y=model.sel(foss.1.1, foss.1.2, foss.1.3, foss.2.1, foss.2.2, foss.2.3, foss.full)
kable(y)

summary(foss.2.1)

rock=as.data.frame(bind_cols(Z.fos.1,O.fos.1,fec.foss, lat.foss))
rownames(rock)=tax.foss
colnames(rock)=c("Z.fos.1", "O.fos.1", "fec.foss", "lat.foss")
rock.1=na.omit(rock)
dim(rock.1) #24 left

foss.1.1=lm(O.fos.1~Z.fos.1, data=rock.1)
foss.1.2=lm(O.fos.1~fec.foss, data=rock.1)
foss.1.3=lm(O.fos.1~abs(lat.foss), data=rock.1)

foss.2.1=lm(O.fos.1~Z.fos.1+fec.foss, data=rock.1)
foss.2.2=lm(O.fos.1~Z.fos.1+abs(lat.foss), data=rock.1)
foss.2.3=lm(O.fos.1~fec.foss+abs(lat.foss), data=rock.1)
foss.full=lm(O.fos.1~Z.fos.1+fec.foss+abs(lat.foss), data=rock.1)

y.1=model.sel(foss.1.1, foss.1.2, foss.1.3, foss.2.1, foss.2.2, foss.2.3, foss.full)
#write.csv(y.1, file="y1.csv")
summary(lm(O.fos.1~Z.fos.1+abs(lat.foss), data=rock.1))
summary(lm(O.fos.1~Z.fos.1+abs(lat.foss), data=rock))

summ(foss.2.2, scale=TRUE,n.sd=2, transform.response = TRUE) #standardized
summ(foss.2.2, vifs = TRUE)

##### now combine Recent and fossil data but use lat only, no relationship with latitude
actually

Z=tapply(log(Recent$Z_area), Recent$taxon, mean, na.rm=T)
O=tapply(log(Recent$Ov_area), Recent$taxon, mean, na.rm=T)
```

```

fec=tapply(Recent$N_ovicellate/(Recent$N_ovicellate+Recent$`N_non-ovicellate`),
Recent$taxon, function(x)x[1])
lat=tapply(Recent$Lat.clean, Recent$taxon, mean, na.rm=T)
tax=tapply(Recent$taxon, Recent$taxon, function(x)x[1])
Rec.com=as.data.frame(bind_cols(Z,O,fec, lat))
rownames(Rec.com)=tax
colnames(Rec.com)=c("Z.fos.1", "O.fos.1", "fec.foss", "lat.foss")#make names correspond
All=rbind(Rec.com, rock)#112 species
All=na.omit(All) #63 species

All.1.1=lm(O.fos.1~Z.fos.1, data=All)
All.1.2=lm(O.fos.1~fec.foss, data=All)
All.1.3=lm(O.fos.1~abs(lat.foss), data=All)

All.2.1=lm(O.fos.1~Z.fos.1+fec.foss, data=All)
All.2.2=lm(O.fos.1~Z.fos.1+abs(lat.foss), data=All)
All.2.3=lm(O.fos.1~fec.foss+abs(lat.foss), data=All)
All.full=lm(O.fos.1~Z.fos.1+fec.foss+abs(lat.foss), data=All)

all.mod=model.sel(All.1.1, All.1.2, All.1.3, All.2.1, All.2.2, All.2.3, All.full)
#write.csv(all.mod, file="all.mod.csv")

par(mfrow=c(1,2))
plot(abs(All$lat.foss), All$Z.fos.1, pch=20,
      xlab="absolute (paleo)latitude", ylab="mean log zooid area", ylim=c(10, 13.8), cex.lab=1.5,
      cex.axis=1.5)
temp=lm(All$Z.fos.1~abs(All$lat.foss))
abline(temp$coefficients[1], temp$coefficients[2])
summary(temp)
#Coefficients:
#           Estimate Std. Error t value Pr(>|t|)
#(Intercept)  11.720702  0.107053 109.485 < 2e-16 ***
#abs(All$lat.foss) 0.008134  0.002665  3.052 0.00285 **

points(abs(Rec.com$lat.foss), Rec.com$Z.fos.1, col="blue", pch=20)
temp=lm(Rec.com$Z.fos.1~abs(Rec.com$lat.foss))
abline(temp$coefficients[1], temp$coefficients[2], col="blue")
summary(temp)
#           Estimate Std. Error t value Pr(>|t|)
#(Intercept)  11.765438  0.118531  99.261 < 2e-16 ***
#abs(Rec.com$lat.foss) 0.010493  0.002952  3.554 0.000719 ***

points(abs(rock$lat.foss), rock$Z.fos.1, col="red", pch=20)
temp=lm(rock$Z.fos.1~abs(rock$lat.foss))
abline(temp$coefficients[1], temp$coefficients[2], col="red")
summary(temp)

```

```

#oefficients:
#           Estimate Std. Error t value Pr(>|t|)
#(Intercept)  11.657971  0.178534  65.298 <2e-16 ***
#abs(rock$lat.foss) 0.004635  0.004440  1.044  0.302
####
All=rbind(Rec.com, rock)#1
plot(abs(All$lat.foss), All$O.fos.1, pch=20,
      xlab="absolute (paleo)latitude", ylab="mean log ovicell (larval) area", ylim=c(10, 13.8),
      cex.lab=1.5, cex.axis=1.5)
temp=lm(All$O.fos.1~abs(All$lat.foss))
abline(temp$coefficients[1], temp$coefficients[2])
summary(temp)
#
#(Intercept)  10.47786  0.12857  81.492 < 2e-16 ***
#abs(All$lat.foss) 0.01567  0.00317  4.943 4.23e-06 ***

points(abs(Rec.com$lat.foss), Rec.com$O.fos.1, col="blue", pch=20)
temp=lm(Rec.com$O.fos.1~abs(Rec.com$lat.foss))
abline(temp$coefficients[1], temp$coefficients[2], col="blue")
summary(temp)
#(Intercept)  10.513525  0.154522  68.039 < 2e-16 ***
#abs(Rec.com$lat.foss) 0.018116  0.003761  4.817 1.62e-05 ***

points(abs(rock$lat.foss), rock$O.fos.1, col="red", pch=20)
temp=lm(rock$O.fos.1~abs(rock$lat.foss))
abline(temp$coefficients[1], temp$coefficients[2], col="red")
summary(temp)
#(Intercept)  10.454763  0.198568  52.651 <2e-16 ***
#abs(rock$lat.foss) 0.011174  0.004993  2.238  0.0325

...

```

Then we look at how Z and O have changed over time; ask if with this new data, the observation in Liow and Taylor still stand (tendency to give rise to bigger descendents and that it is not because it is "going out of tropics" only). Conclusion maybe that current day distribution is controlled by climate but that we have to take history into account (and despite not having a phylogeny, we can still try to do it). Secondary observation is that larval size does not seem to trade off with fecundity.

Cope's rule figure

```

```{r, echo=F, warning=F}
FR.raw=rbind(subset(Recent, select=c("Genus", "Species", "Lat.clean", "Z_area", "Ov_area",
"Min.age", "Max.age")),
 subset(Fossil, select=c("Genus", "Species", "Lat.clean", "Z_area", "Ov_area", "Min.age",
"Max.age")))

b=tapply(FR.raw$Lat.clean, FR.raw$Species, mean, na.rm=T)

```

```

cc=tapply(FR.raw$Z_area, FR.raw$Species, mean, na.rm=T)
d=tapply(FR.raw$Ov_area, FR.raw$Species, mean, na.rm=T)
ee=tapply((FR.raw$Min.age+FR.raw$Max.age)*0.5, FR.raw$Species, mean, na.rm=T)
FR=cbind(cbind(b,cc), cbind(d,ee))
colnames(FR)=c("lat", "Z.area", "Ov.area", "mid.age")

taxon=rownames(FR)

comb=dim(combn(1:length(taxon),2))[2]#number of combinations
ref=combn(1:length(taxon),2)#Generate all combinations of the elements of x taken m at a
time
y=matrix(nrow=comb, ncol=8, 0)

for (i in 1:comb){
 sp.1=FR[which(rownames(FR)==taxon[ref,i][1]),]#first species
 sp.2=FR[which(rownames(FR)==taxon[ref,i][2]),]#second species
 y[i,1]=log(sp.1[2])-log(sp.2[2])#negative sp.1 is smaller
 y[i,2]=sp.1[4]-sp.2[4] #positive sp.1 is older
 y[i,3]=sp.1[4] #check midpoints
 y[i,4]=sp.2[4] #check midpoints
 y[i,5]=rownames(FR)[which(rownames(FR)==taxon[ref,i][1])]
 y[i,6]=rownames(FR)[which(rownames(FR)==taxon[ref,i][2])]
 y[i,7]=log(sp.1[3])-log(sp.2[3]) #positive sp.1 is older ovicell area
 y[i,8]=abs(sp.1[1])-abs(sp.2[1]) #negative means move from tropics.

}
y=as.data.frame(y)
colnames(y)=c("size.diff", "time.diff", "sp.1.mid", "sp.2.mid", "sp.1", "sp.2", "ovi.diff",
"lat.diff")
y$size.diff=as.numeric(y$size.diff)
y$time.diff=as.numeric(y$time.diff)
y$sp.1.mid=as.numeric(y$sp.1.mid)
y$sp.2.mid=as.numeric(y$sp.2.mid)
y$ovi.diff=as.numeric(y$ovi.diff)
y$lat.diff=as.numeric(y$lat.diff)
y=y[which(y$sp.1.mid!=y$sp.2.mid),]#get rid of those that are same age.

#correct=y[which(y$sp.1.mid>y$sp.2.mid),] #where species 1 is older
#opposite=y[which(y$sp.1.mid<y$sp.2.mid),]

a.larger.1=y[which(y$size.diff>0 & y$time.diff>0),] #sp.1 is older, sp.1 (ancestor) is larger
a.smaller.1=y[which(y$size.diff<0 & y$time.diff>0),]#sp.1 is older, sp.1 (ancestor) is smaller
a.smaller.2=y[which(y$size.diff>0 & y$time.diff<0),] #sp.2 is older, sp.2 (ancestor) is smaller
a.larger.2=y[which(y$size.diff<0 & y$time.diff<0),] #sp.2 is older, sp.2 (ancestor) is larger

a.larger.2.new=a.larger.2[,c(1,2,4,3,6,5,7,8)]

```

```
a.larger.2.new$size.diff=-a.larger.2$size.diff #swtich signs of the size and the time so can be
pasted with d.larger
a.larger.2.new$time.diff=-a.larger.2$time.diff
```

```
d.smaller=rbind(a.larger.1, a.larger.2.new)
rm(a.larger.1, a.larger.2.new)
```

```
a.smaller.2.new=a.smaller.2[,c(1,2,4,3,6,5,7,8)]
a.smaller.2.new$size.diff=-a.smaller.2$size.diff #swtich signs of the size and the time so can
be pasted with d.larger
a.smaller.2.new$time.diff=-a.smaller.2$time.diff
```

```
d.larger=rbind(a.smaller.1, a.smaller.2.new)
rm(a.smaller.1, a.smaller.2.new)
```

```
B=dim(d.larger)[1] #descendent is larger
S=dim(d.smaller)[1]#descendent is smaller
b.d.1=binom.test(B, B+S)
```

```
d=c(d.larger$size.diff,d.smaller$size.diff) #descendent is reference, d>0 means d is larger
d.1=-d#ancestor is reference, d>0 means d is larger
```

```
#b.d.1 and d.1
par(mfrow=c(2,2), mar=c(4,4,3,2))
hist(d.1, main= "All plausible ancestor-descendant pairs ", xlab="difference in log autozoid
area (microns)", col="grey",
 breaks=20, xlim=c(-2,2), ylim=c(0,530))
segments(0,0,0,390, col="red", lwd=4) #the zero
segments(median(d.1),0,median(d.1),490, col="blue", lwd=4) #
#very faint probability of descendent being bigger
text(1.5,450, paste("p=", round(b.d.1$estimate, 2),"\\n(",
 round(b.d.1$conf.int[1],2),"-", round(b.d.1$conf.int[2],2), ")")
 , cex=0.8)
text(1.5,520, paste("N =", B+S), cex=0.8)
```

```
####ovicells
```

```
a.larger.1=y[which(y$ovi.diff>0 & y$time.diff>0),] #sp.1 is older, sp.1 (ancestor) is larger
a.smaller.1=y[which(y$ovi.diff<0 & y$time.diff>0),]#sp.1 is older, sp.1 (ancestor) is smaller
a.smaller.2=y[which(y$ovi.diff>0 & y$time.diff<0),] #sp.2 is older, sp.2 (ancestor) is smaller
a.larger.2=y[which(y$ovi.diff<0 & y$time.diff<0),] #sp.2 is older, sp.2 (ancestor) is larger
```

```
a.larger.2.new=a.larger.2[,c(1,2,4,3,6,5,7,8)]
a.larger.2.new$ovi.diff=-a.larger.2$ovi.diff #swtich signs of the size and the time so can be
pasted with d.larger
a.larger.2.new$time.diff=-a.larger.2$time.diff
```

```

d.smaller=rbind(a.larger.1, a.larger.2.new)
rm(a.larger.1, a.larger.2.new)

a.smaller.2.new=a.smaller.2[,c(1,2,4,3,6,5,7,8)]
a.smaller.2.new$ovi.diff=-a.smaller.2$ovi.diff #swtich signs of the size and the time so can
be pasted with d.larger
a.smaller.2.new$time.diff=-a.smaller.2$time.diff

d.larger=rbind(a.smaller.1, a.smaller.2.new)
rm(a.smaller.1, a.smaller.2.new)

B=dim(d.larger)[1] #descendent is larger
S=dim(d.smaller)[1]#descendent is smaller
b.d=binom.test(B, B+S)
b.d.2=b.d

d=c(d.larger$ovi.diff,d.smaller$ovi.diff) #descendent is reference, d>0 means d is larger
d.2=-d#ancestor is reference, d>0 means d is larger

#par(mfrow=c(1,1), mar=c(5,4,4,2))
hist(d.2, main= "", xlab="difference in log ovicell area (microns)", col="grey",
 breaks=15, xlim=c(-2,2))
segments(0,0,0,190, col="red", lwd=4) #the zero
segments(median(d.2),0,median(d.2),230, col="blue", lwd=4) #
#very faint probability of descendent being bigger
text(-1.5,180, paste("p=", round(b.d$estimate, 2),"\\n(",
 round(b.d.2$conf.int[1],2),"-", round(b.d.2$conf.int[2],2), ")")
 , cex=0.8)
text(-1.5,210, paste("N =", B+S), cex=0.8)

...

#the alternative
``{r, echo=F, warning=F}
N=1000
#####with no re-using of any used species and loop through
bootstrap.b.d=NULL
for (j in 1:N){
 sam=sample(taxon)#sample
 no.pairs=floor(length(taxon)/2)
 y=matrix(nrow=no.pairs, ncol=8, 0)

 for (i in 1:no.pairs){
 sp.1=FR[which(rownames(FR)==sam[i*2-1]),]#first species

```

```

sp.2=FR[which(rownames(FR)==sam[i*2]),]#second species
y[i,1]=log(sp.1[2])-log(sp.2[2])#negative sp.1 is smaller
y[i,2]=sp.1[4]-sp.2[4] #positive sp.1 is older
y[i,3]=sp.1[4] #check midpoints
y[i,4]=sp.2[4] #check midpoints
y[i,5]=rownames(FR)[which(rownames(FR)==sam[i*2-1])]
y[i,6]=rownames(FR)[which(rownames(FR)==sam[i*2])]
y[i,7]=sp.1[3]-sp.2[3] #positive sp.1 is older ovicell area
y[i,8]=abs(sp.1[1])-abs(sp.2[1]) #negative means move from tropics.
}
y=as.data.frame(y)
colnames(y)=c("size.diff", "time.diff", "sp.1.mid", "sp.2.mid", "sp.1", "sp.2", "ovi.diff",
"lat.diff")
y$size.diff=as.numeric(y$size.diff)
y$time.diff=as.numeric(y$time.diff)
y$sp.1.mid=as.numeric(y$sp.1.mid)
y$sp.2.mid=as.numeric(y$sp.2.mid)
y$ovi.diff=as.numeric(y$ovi.diff)
y$lat.diff=as.numeric(y$lat.diff)
y=y[which(y$sp.1.mid!=y$sp.2.mid),]#get rid of those that are same age.

a.larger.1=y[which(y$size.diff>0 & y$time.diff>0),] #sp.1 is older, sp.1 (ancestor) is larger
a.smaller.1=y[which(y$size.diff<0 & y$time.diff>0),]#sp.1 is older, sp.1 (ancestor) is smaller
a.smaller.2=y[which(y$size.diff>0 & y$time.diff<0),] #sp.2 is older, sp.2 (ancestor) is smaller
a.larger.2=y[which(y$size.diff<0 & y$time.diff<0),] #sp.2 is older, sp.2 (ancestor) is larger

a.larger.2.new=a.larger.2[,c(1,2,4,3,6,5,7,8)]
a.larger.2.new$size.diff=-a.larger.2$size.diff #swtich signs of the size and the time so can be
pasted with d.larger
a.larger.2.new$time.diff=-a.larger.2$time.diff

d.smaller=rbind(a.larger.1, a.larger.2.new)
rm(a.larger.1, a.larger.2.new)

a.smaller.2.new=a.smaller.2[,c(1,2,4,3,6,5,7,8)]
a.smaller.2.new$size.diff=-a.smaller.2$size.diff #swtich signs of the size and the time so can
be pasted with d.larger
a.smaller.2.new$time.diff=-a.smaller.2$time.diff

d.larger=rbind(a.smaller.1, a.smaller.2.new)
rm(a.smaller.1, a.smaller.2.new)
B=dim(d.larger)[1] #descendent is larger
S=dim(d.smaller)[1]#descendent is smaller
b.d=binom.test(B, B+S)

bootstrap.b.d[j]=b.d$estimate

```

```

}

bootstrap.rand=NULL #this chunk is random not caring about time

for (j in 1:N){
 sam=sample(taxon)#sample
 no.pairs=floor(length(taxon)/2)
 y=matrix(nrow=no.pairs, ncol=8, 0)

 for (i in 1:no.pairs){
 sp.1=FR[which(rownames(FR)==sam[i*2-1]),]#first species
 sp.2=FR[which(rownames(FR)==sam[i*2]),]#second species
 y[i,1]=log(sp.1[2])-log(sp.2[2])#negative sp.1 is smaller
 y[i,2]=sp.1[4]-sp.2[4] #positive sp.1 is older
 y[i,3]=sp.1[4] #check midpoints
 y[i,4]=sp.2[4] #check midpoints
 y[i,5]=rownames(FR)[which(rownames(FR)==taxon[ref[,i][1])]
 y[i,6]=rownames(FR)[which(rownames(FR)==taxon[ref[,i][2])]
 y[i,7]=sp.1[3]-sp.2[3] #positive sp.1 is older ovicell area
 y[i,8]=abs(sp.1[1])-abs(sp.2[1]) #negative means move from tropics.
 }
 y=as.data.frame(y)
 colnames(y)=c("size.diff", "time.diff", "sp.1.mid", "sp.2.mid", "sp.1", "sp.2", "ovi.diff",
"lat.diff")
 y$size.diff=as.numeric(y$size.diff)
 y$time.diff=as.numeric(y$time.diff)
 y$sp.1.mid=as.numeric(y$sp.1.mid)
 y$sp.2.mid=as.numeric(y$sp.2.mid)
 y$ovi.diff=as.numeric(y$ovi.diff)
 y$lat.diff=as.numeric(y$lat.diff)
 y=y[which(y$sp.1.mid!=y$sp.2.mid),]#get rid of those that are same age.

 a.larger.1=y[which(y$size.diff>0),] #sp.1 is older, sp.1 (ancestor) is larger

 a.smaller.2=y[which(y$size.diff<0),] #sp.2 is older, sp.2 (ancestor) is smaller

 B=dim(a.larger.1)[1] #descendent is larger
 S=dim(a.smaller.2)[1]#descendent is smaller
 b.d=binom.test(B, B+S)

 bootstrap.rand[j]=b.d$estimate
}

bootstrap.b.d.1=bootstrap.b.d
bootstrap.rand.1=bootstrap.rand
#par(mfrow=c(2,1))
hist(bootstrap.b.d.1, col=rgb(1,0,0,0.5),xlim=c(0, 1), ylim=c(0,200),

```



```

main=" ", xlab="binomial probability of larger descendant autozoid", breaks=20)

hist(bootstrap.rand.1, col=rgb(0,0,1,0.5), add=T, breaks=20)
abline(v=0.5, col="black", lwd=3)
legend("topleft", legend=c("AD Pairs", "Randomized"),
 col=c(rgb(1,0,0,0.5), rgb(0,0,1,0.5)), lty=1, lwd=10, bty="n")
box()

#ovicell
N=1000
#####with no re-using of any used species and loop through
bootstrap.b.d=NULL
for (j in 1:N){
 sam=sample(taxon)#sample
 no.pairs=floor(length(taxon)/2)
 y=matrix(nrow=no.pairs, ncol=8, 0)

 for (i in 1:no.pairs){
 sp.1=FR[which(rownames(FR)==sam[i*2-1]),]#first species
 sp.2=FR[which(rownames(FR)==sam[i*2]),]#second species
 y[i,1]=log(sp.1[2])-log(sp.2[2])#negative sp.1 is smaller
 y[i,2]=sp.1[4]-sp.2[4] #positive sp.1 is older
 y[i,3]=sp.1[4] #check midpoints
 y[i,4]=sp.2[4] #check midpoints
 y[i,5]=rownames(FR)[which(rownames(FR)==sam[i*2-1])]
 y[i,6]=rownames(FR)[which(rownames(FR)==sam[i*2])]
 y[i,7]=log(sp.1[3])-log(sp.2[3]) #positive sp.1 is older ovicell area
 y[i,8]=abs(sp.1[1])-abs(sp.2[1]) #negative means move from tropics.
 }
 y=as.data.frame(y)
 colnames(y)=c("size.diff", "time.diff", "sp.1.mid", "sp.2.mid", "sp.1", "sp.2", "ovi.diff",
"lat.diff")
 y$size.diff=as.numeric(y$size.diff)
 y$time.diff=as.numeric(y$time.diff)
 y$sp.1.mid=as.numeric(y$sp.1.mid)
 y$sp.2.mid=as.numeric(y$sp.2.mid)
 y$ovi.diff=as.numeric(y$ovi.diff)
 y$lat.diff=as.numeric(y$lat.diff)
 y=y[which(y$sp.1.mid!=y$sp.2.mid),]#get rid of those that are same age.

 a.larger.1=y[which(y$ovi.diff>0 & y$time.diff>0),] #sp.1 is older, sp.1 (ancestor) is larger
 a.smaller.1=y[which(y$ovi.diff<0 & y$time.diff>0),]#sp.1 is older, sp.1 (ancestor) is smaller
 a.smaller.2=y[which(y$ovi.diff>0 & y$time.diff<0),] #sp.2 is older, sp.2 (ancestor) is smaller
 a.larger.2=y[which(y$ovi.diff<0 & y$time.diff<0),] #sp.2 is older, sp.2 (ancestor) is larger

 a.larger.2.new=a.larger.2[,c(1,2,4,3,6,5,7,8)]

```

```
a.larger.2.new$ovi.diff=-a.larger.2$ovi.diff #switch signs of the size and the time so can be
pasted with d.larger
```

```
a.larger.2.new$time.diff=-a.larger.2$time.diff
```

```
d.smaller=rbind(a.larger.1, a.larger.2.new)
```

```
rm(a.larger.1, a.larger.2.new)
```

```
a.smaller.2.new=a.smaller.2[,c(1,2,4,3,6,5,7,8)]
```

```
a.smaller.2.new$ovi.diff=-a.smaller.2$ovi.diff #switch signs of the size and the time so can
be pasted with d.larger
```

```
a.smaller.2.new$time.diff=-a.smaller.2$time.diff
```

```
d.larger=rbind(a.smaller.1, a.smaller.2.new)
```

```
rm(a.smaller.1, a.smaller.2.new)
```

```
B=dim(d.larger)[1] #descendent is larger
```

```
S=dim(d.smaller)[1]#descendent is smaller
```

```
b.d=binom.test(B, B+S)
```

```
bootstrap.b.d[j]=b.d$estimate
```

```
}
```

```
bootstrap.rand=NULL #this chunk is random not caring about time
```

```
for (j in 1:N){
```

```
 sam=sample(taxon)#sample
```

```
 no.pairs=floor(length(taxon)/2)
```

```
 y=matrix(nrow=no.pairs, ncol=8, 0)
```

```
 for (i in 1:no.pairs){
```

```
 sp.1=FR[which(rownames(FR)==sam[i*2-1]),]#first species
```

```
 sp.2=FR[which(rownames(FR)==sam[i*2]),]#second species
```

```
 y[i,1]=log(sp.1[2])-log(sp.2[2])#negative sp.1 is smaller
```

```
 y[i,2]=sp.1[4]-sp.2[4] #positive sp.1 is older
```

```
 y[i,3]=sp.1[4] #check midpoints
```

```
 y[i,4]=sp.2[4] #check midpoints
```

```
 y[i,5]=rownames(FR)[which(rownames(FR)==taxon[ref,i][1])]#
```

```
 y[i,6]=rownames(FR)[which(rownames(FR)==taxon[ref,i][2])]#
```

```
 y[i,7]=log(sp.1[3])-log(sp.2[3]) #positive sp.1 is older ovicell area
```

```
 y[i,8]=abs(sp.1[1])-abs(sp.2[1]) #negative means move from tropics.
```

```
 }
```

```
 y=as.data.frame(y)
```

```
 colnames(y)=c("size.diff", "time.diff", "sp.1.mid", "sp.2.mid", "sp.1", "sp.2", "ovi.diff",
"lat.diff")
```

```
 y$size.diff=as.numeric(y$size.diff)
```

```
 y$time.diff=as.numeric(y$time.diff)
```

```
 y$sp.1.mid=as.numeric(y$sp.1.mid)
```

```

y$sp.2.mid=as.numeric(y$sp.2.mid)
y$ovi.diff=as.numeric(y$ovi.diff)
y$lat.diff=as.numeric(y$lat.diff)
y=y[which(y$sp.1.mid!=y$sp.2.mid),]#get rid of those that are same age.

a.larger.1=y[which(y$ovi.diff>0),] #sp.1 is older, sp.1 (ancestor) is larger

a.smaller.2=y[which(y$ovi.diff<0),] #sp.2 is older, sp.2 (ancestor) is smaller

B=dim(a.larger.1)[1] #descendent is larger
S=dim(a.smaller.2)[1]#descendent is smaller
b.d=binom.test(B, B+S)

bootstrap.rand[j]=b.d$estimate
}

bootstrap.b.d.2=bootstrap.b.d
bootstrap.rand.2=bootstrap.rand
hist(bootstrap.b.d.2, col=rgb(1,0,0,0.5),xlim=c(0, 1), ylim=c(0,280),
 main=" ", xlab="binomial probability of larger descendant ovicell", breaks=20)

hist(bootstrap.rand.2, col=rgb(0,0,1,0.5), add=T, breaks=20)
abline(v=0.5, col="black", lwd=3)
legend("topleft", legend=c("AD Pairs", "Randomized"),
 col=c(rgb(1,0,0,0.5), rgb(0,0,1,0.5)), lty=1, lwd=10, bty="n")
box()
```



##Fig. 5 (A and B)



```

```{r, echo=F, warning=F}
#1
par(mfrow=c(2,2), mar=c(4,4,3,2), oma=c(2,0,0,0))
hist(d.1, main= "A.", xlab="difference in log zooid area", col="grey",
     breaks=20, xlim=c(-2,2), ylim=c(0,530))
segments(median(d.1),5 ,median(d.1),8, lwd=8) #
text(-1.5,450, paste("p=", round(b.d.1$estimate, 2),"\n(",
      round(b.d.1$conf.int[1],2),"-", round(b.d.1$conf.int[2],2), ")")
     , cex=0.8)
#2
hist(d.2, main= "B.", xlab="difference in log ovicell (larval) size", col="grey",ylab="",
     breaks=15, xlim=c(-2,2))
segments(median(d.2),5,median(d.2),8, lwd=8) #
text(-1.5,180, paste("p=", round(b.d.2$estimate, 2),"\n(",
      round(b.d.2$conf.int[1],2),"-", round(b.d.2$conf.int[2],2), ")")
     , cex=0.8)
#3

```


```

```

hist(bootstrap.b.d.1, col=rgb(1,0,0,0.5),xlim=c(0, 1), ylim=c(0,280),
 main="C.", xlab="binomial probability of larger descendant zoid", breaks=20)

hist(bootstrap.rand.1, col=rgb(0,0,1,0.5), add=T, breaks=20)
legend("topleft", legend=c("AD Pairs", "Randomized"),
 col=c(rgb(1,0,0,0.5), rgb(0,0,1,0.5)), lty=1, lwd=10, bty="n")

#4
hist(bootstrap.b.d.2, col=rgb(1,0,0,0.5),xlim=c(0, 1), ylim=c(0,280),
 main="D.", ylab="", xlab="binomial probability of larger descendant ovicell (larva)",
 breaks=20)

hist(bootstrap.rand.2, col=rgb(0,0,1,0.5), add=T, breaks=20)
legend("topleft", legend=c("AD Pairs", "Randomized"),
 col=c(rgb(1,0,0,0.5), rgb(0,0,1,0.5)), lty=1, lwd=10, bty="n")

...

#The data are presented (and Ecol Letters 2017 data read in)

```{r, echo=F, warning=F}
gosia=read_excel("/Users/leehl/Documents/Emanuela di
Martino/Microporella/Microporella_dataset.final.10.09.2020.xlsx", 4) #

other=tapply(gosia$length*gosia$width, gosia$colony_id, mean)

Pooled= rbind(subset(Recent, select=c("taxon", "Min.age", "Max.age", "Lat.clean", "Z_area",
"Ov_area")),
subset(Fossil, select=c("taxon", "Min.age", "Max.age", "Lat.clean", "Z_area", "Ov_area") ))
Data.tax=unique(Pooled$taxon)

#Data summary plot
par(mfrow=c(2,3), mar=c(4,4,2,2), oma=c(2,1,1,0))
plt=Pooled[which(Pooled$taxon==Data.tax[1] & !is.na(Pooled$Z_area)),]#Zoid area

plot(jitter(rep(1, dim(plt)[1]), factor=0),log(plt$Z_area), ylim=c(10.8,
14),xlim=c(0,length(Data.tax)+1), pch=20, xlab="", ylab="log zoid area", cex.lab=1.5,
cex.axis=1.5)
points(rep(1, dim(plt)[1]),log(plt$Z_area), type="l")
for (i in 2:length(Data.tax)){
  plt=Pooled[which(Pooled$taxon==Data.tax[i] & !is.na(Pooled$Z_area)),]#Zoid area
  points(jitter(rep(i, dim(plt)[1]), factor=0),log(plt$Z_area), pch=20)
  points(rep(i, dim(plt)[1]),log(plt$Z_area), type="l")
}
t.test(log(other), conf.level = 0.95)
#points(rep(107, 3), c(12.00576, mean(log(other)),12.24448) , col="red", pch=20)
points(rep(107, 3), c(11.98064, mean(log(other)),12.26960) , col="red", pch=20)
points(rep(107, 3), c(11.98064, mean(log(other)),12.26960) , type="l",col="red", pch=20)

```

```

#points(rep(107, 3), c(12.00576, mean(log(other)),12.24448) , type="l",col="red", pch=20)
title("A.", line = 0.3, adj=0, cex.main=1.5)

#plot of fossil taxa (according to age, minimum plotted)
plt=Pooled[which(Pooled$taxon==Data.tax[1] & !is.na(Pooled$Z_area)),]#Zooid area

plot(jitter(rep(-plt$Min.age[1], dim(plt)[1]), factor=0),log(plt$Z_area), ylim=c(10.8,
14),xlim=c(-16,0), pch=20, xlab="", ylab="", cex.lab=1.5, cex.axis=1.5, axes=F)
axis(2, cex.axis=1.5)
axis(1, at= seq(-15,0,5), labels=seq(15,0,-5), cex.axis=1.5)
box()
points(rep(-plt$Min.age[1], dim(plt)[1]),log(plt$Z_area), type="l")

for (i in 2:length(Data.tax)){
  plt=Pooled[which(Pooled$taxon==Data.tax[i] & !is.na(Pooled$Z_area)),]#Zooid area
  points(jitter(rep(-plt$Min.age[1], dim(plt)[1]), factor=0),log(plt$Z_area), pch=20)
  points(rep(-plt$Min.age[1], dim(plt)[1]),log(plt$Z_area), type="l")
}
title("B.", line = 0.3, adj=0, cex.main=1.5)

#plot of fossil taxa (according to lat, minimum plotted)
plt=Pooled[which(Pooled$taxon==Data.tax[1] & !is.na(Pooled$Z_area)),]#Zooid area
plot(jitter(rep(plt$Lat.clean[1], dim(plt)[1]), factor=0),log(plt$Z_area), ylim=c(10.8,
14),xlim=c(-90,90), pch=20, xlab="", ylab="", cex.lab=1.5, cex.axis=1.5)
points(rep(plt$Lat.clean[1], dim(plt)[1]),log(plt$Z_area), type="l")
title("C.", line = 0.3, adj=0, cex.main=1.5)

for (i in 2:length(Data.tax)){

  plt=Pooled[which(Pooled$taxon==Data.tax[i] & !is.na(Pooled$Z_area)),]#Zooid area
  if (!is.na(plt$Lat.clean[1])){
    points(jitter(rep(plt$Lat.clean[1], dim(plt)[1]), factor=0),log(plt$Z_area), pch=20)
    points(rep(plt$Lat.clean[1], dim(plt)[1]),log(plt$Z_area), type="l") }
}

plt=Pooled[which(Pooled$taxon==Data.tax[1] & !is.na(Pooled$Ov_area)),]#Ovicell area
plot(jitter(rep(1, dim(plt)[1]), factor=0),log(plt$Ov_area), ylim=c(9,
13),xlim=c(0,length(Data.tax)), pch=20, xlab="species index", ylab="log ovicell (larval) area",
cex.lab=1.5, cex.axis=1.5)
points(rep(1, dim(plt)[1]),log(plt$Ov_area), type="l")
title("D.", line = 0.3, adj=0, cex.main=1.5)

for (i in 2:length(Data.tax)){
  plt=Pooled[which(Pooled$taxon==Data.tax[i] & !is.na(Pooled$Ov_area)),]
  points(jitter(rep(i, dim(plt)[1]), factor=0),log(plt$Ov_area), pch=20)
  points(rep(i, dim(plt)[1]),log(plt$Ov_area), type="l")
}

```

```

#plot of fossil taxa (according to age, minimum plotted)
plt=Pooled[which(Pooled$taxon==Data.tax[1] & !is.na(Pooled$Ov_area)),]#Ovicell area
plot(jitter(rep(-plt$Min.age[1], dim(plt)[1]), factor=0),log(plt$Ov_area), ylim=c(9,
13),xlim=c(-16,0), pch=20, xlab="Mya", ylab="", cex.lab=1.5, cex.axis=1.5, axes=F)
points(rep(-plt$Min.age[1], dim(plt)[1]),log(plt$Ov_area), type="l")
axis(2, cex.axis=1.5)
axis(1, at=seq(-15,0,5), labels=seq(15,0,-5),cex.axis=1.5)
box()
title("E.", line = 0.3, adj=0, cex.main=1.5)

for (i in 2:length(Data.tax)){
  plt=Pooled[which(Pooled$taxon==Data.tax[i] & !is.na(Pooled$Ov_area)),]#
  points(jitter(rep(-plt$Min.age[1], dim(plt)[1]), factor=0),log(plt$Ov_area), pch=20)
  points(rep(-plt$Min.age[1], dim(plt)[1]),log(plt$Ov_area), type="l")
}

#plot of fossil taxa (according to lat, minimum plotted)
plt=Pooled[which(Pooled$taxon==Data.tax[1] & !is.na(Pooled$Ov_area)),]#Zooid area
plot(jitter(rep(plt$Lat.clean[1], dim(plt)[1]), factor=0),log(plt$Ov_area), ylim=c(9,
13),xlim=c(-90,90), pch=20, xlab="(paleo)latitude", ylab="", cex.lab=1.5, cex.axis=1.5)
points(rep(plt$Lat.clean[1], dim(plt)[1]),log(plt$Ov_area), type="l")
title("F.", line = 0.3, adj=0, cex.main=1.5)

for (i in 2:length(Data.tax)){

  plt=Pooled[which(Pooled$taxon==Data.tax[i] & !is.na(Pooled$Ov_area)),]#Zooid area
  if (!is.na(plt$Lat.clean[1])){
    points(jitter(rep(plt$Lat.clean[1], dim(plt)[1]), factor=0),log(plt$Ov_area), pch=20)
    points(rep(plt$Lat.clean[1], dim(plt)[1]),log(plt$Ov_area), type="l") }
}

...

## out of tropics
Code modified from 2019 paper

```{r, echo=TRUE}

#Pooled$taxon.time=paste(Pooled$taxon,Pooled$Min.age)
Pooled$mid=(Pooled$Min.age+Pooled$Max.age)*0.5
#taxon.time=unique(Pooled$taxon.time)
taxon=unique(Pooled$taxon)
N=length(taxon)
comb=dim(combn(1:length(taxon),2))[2]#number of combinations
ref=combn(1:length(taxon),2)#Generate all combinations of the elements of x taken m at a
time

```

```

y=matrix(nrow=comb, ncol=13, 0)

for (i in 1:comb){
 sp.1=Pooled[which(Pooled$taxon==taxon[ref,i][1])],#first species
 sp.2=Pooled[which(Pooled$taxon==taxon[ref,i][2])],#second species
 y[i,1]=as.numeric(mean(log(sp.1$Z_area),na.rm=T)-mean(log(sp.2$Z_area),
na.rm=T))#negative sp.1 is smaller
 y[i,2]=as.numeric(mean(sp.1$mid)-mean(sp.2$mid))#positive sp.1 is older
 y[i,3]=mean(sp.1$mid, na.rm=T)
 y[i,4]=mean(sp.2$mid, na.rm=T)
 y[i,5]=sp.1$taxon[1]
 y[i,6]=sp.2$taxon[1]
 y[i,7]=paste("genus")
 y[i,8]=mean(log(sp.1$Z_area),na.rm=T)
 y[i,9]=mean(log(sp.2$Z_area),na.rm=T)
 y[i,10]=abs(mean(sp.1$Lat.clean))-abs(mean(sp.2$Lat.clean))#positive means goes
warmer unless the next column is FALSE
 y[i,11]=
 (mean(sp.1$Lat.clean) >0 & mean(sp.2$Lat.clean)>0)|(mean(sp.1$Lat.clean) <0 &
mean(sp.2$Lat.clean) >0)
 y[i,12]=mean(sp.1$Lat.clean)
 y[i,13]=mean(sp.2$Lat.clean)
}
y=as.data.frame(y)
colnames(y)=c("size.diff", "time.diff", "sp.1.start", "sp.2.start", "sp.1", "sp.2", "genus",
"sp.1.area", "sp.2.area", "shift", "same.hemis", "sp.1.lat", "sp.2.lat")

y=y[!is.na(y$shift),] #we don't have data for some paleolats.

y$size.diff=as.numeric(paste(y$size.diff))
y$time.diff=as.numeric(paste(y$time.diff))
y$sp.1.start=as.numeric(paste(y$sp.1.start))
y$sp.2.start=as.numeric(paste(y$sp.2.start))
y$sp.1.area=as.numeric(paste(y$sp.1.area))
y$sp.2.area=as.numeric(paste(y$sp.2.area))
y$shift=as.numeric(paste(y$shift))
y$sp.1.lat=as.numeric(paste(y$sp.1.lat))
y$sp.2.lat=as.numeric(paste(y$sp.2.lat))
y$sp.1=as.character(paste(y$sp.1))
y$sp.2=as.character(paste(y$sp.2))

turn=which(y$time.diff<0)#make sure all the sp.1 are older by this chunk
yy=y
for (i in 1:length(turn)){
 yy$size.diff[turn[i]]=-y$size.diff[turn[i]]
}

```

```
yy$time.diff[turn[i]]=-y$time.diff[turn[i]]
yy$sp.1.start[turn[i]]=y$sp.2.start[turn[i]]
yy$sp.2.start[turn[i]]=y$sp.1.start[turn[i]]
yy$sp.1[turn[i]]=y$sp.2[turn[i]]
yy$sp.2[turn[i]]=y$sp.1[turn[i]]
yy$sp.1.area[turn[i]]=y$sp.2.area[turn[i]]
yy$sp.2.area[turn[i]]=y$sp.1.area[turn[i]]
yy$shift[turn[i]]=y$shift[turn[i]]
yy$sp.1.lat[turn[i]]=y$sp.1.lat[turn[i]]
yy$sp.2.lat[turn[i]]=y$sp.2.lat[turn[i]]
}
yy=yy[which(yy$time.diff!=0),]
```

...

Now in the last section, let's try to model the differences between AD as a function of latitudinal shifts as well and see if it does better than a null model in which we use genus random effects.

```
``{r, echo=TRUE}
best.model=matrix(nrow=1000,ncol=2, NA)
model.one=list()
for (k in 1:1000){
 pick=NULL

 for (i in 1:length(unique(c(yy$sp.1)))){
 temp=which(yy$sp.1==unique(yy$sp.1)[i])
 pick[i]=sample(temp,1)}

 picked=yy[pick,]
 station=picked[which(picked$same.hemis=="TRUE"),]
 stat.mobil=station

 stat.mobil$cat=NA
 stat.mobil$cat[which(stat.mobil$size.diff<0)]=1 #grew bigger
 stat.mobil$cat[which(stat.mobil$size.diff>0)]=0

 #mod1=lm(size.diff~shift, data=stat.mobil)#lat shift only + is to tropics
 #mod2=lm(size.diff~shift+time.diff, data=stat.mobil) #best
 #mod3=lm(size.diff~time.diff, data=stat.mobil) #how much time
 #mod4=lm(size.diff~shift*time.diff, data=stat.mobil) #how much time

 mod1=glm(cat~shift, data=stat.mobil, family=binomial)#lat shift only + is to tropics
 mod2=glm(cat~shift+time.diff, data=stat.mobil, family=binomial) #best
 mod3=glm(cat~time.diff, data=stat.mobil, family=binomial) #how much time
```



```
mod4=glm(cat~shift*time.diff, data=stat.mobil, family=binomial) #how much time
```

```
models=model.sel(mod1,mod2, mod3, mod4)
best.model[k,1]=rownames(models)[1]
best.model[k,2]=models$weight[1]
model.one[[k]]=(summary(mod1))$coefficients
}
```

```
one=best.model[which(best.model[,1]=="mod1"),]
one.n=dim(one)[1] #440
one.ave=median(as.numeric(one[,2]))#0.4593031
one.sum=one.n*one.ave #202
```

```
Reduce('+', model.one) #not significant!
par(mfrow=c(1,1))
plot(stat.mobil$shift, stat.mobil$size.diff) #mod1
```

```
two=best.model[which(best.model[,1]=="mod2"),]
two.n=dim(two)[1] #124
two.ave=median(as.numeric(two[,2])) #0.443
two.sum=two.n*two.ave#54.9431
```

```
three=best.model[which(best.model[,1]=="mod3"),]
three.n=dim(three)[1] #301
three.ave=median(as.numeric(three[,2]))#0.4489228
three.sum=three.n*three.ave #135
```

```
four=best.model[which(best.model[,1]=="mod4"),]
four.n=dim(four)[1] #135
four.ave=median(as.numeric(four[,2]))#0.489838
four.sum=four.n*four.ave #66
```

```
shift.dis=NA
p.dis=NA
for (k in 1:1000){
 shift.dis[k]=model.one[[k]][2,1]#shift
 p.dis[k]=model.one[[k]][2,4]#shift
}
```

```
#Fig. 6
par(mfrow=c(2,1), oma=c(3,1,3,1))
hist(shift.dis, main="",xlab="distribution of latitudinal shift estimated")
points(median(shift.dis), 0, cex=2, pch=20)
hist(p.dis, main="",xlab="distribution of p-value for latitudinal shift")
points(median(p.dis), 0, cex=2, pch=20)
```

```
...
```