Sperm size figures

jeremy

Wednesday, January 14, 2015

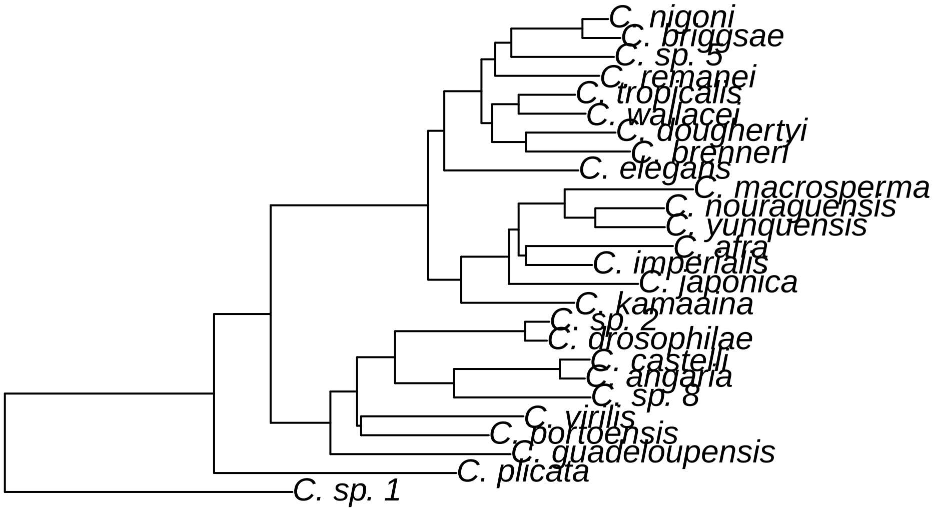
Sperm size figures

This is the output of an R markdown script - a fancy way of embedding r in a page so we can rerun/tune it as needed and then have the commands

I'm embedding, but hiding a bynch of stuff for readability - the actual code is in the .Rmd file.

```
## Warning: package 'survival' was built under R version 3.1.3
```

Here's the tree from Kiontke et al 2011, figure 3 intial values I recieved had angaria and sp12 flipped - I've corrected it here Here's the tree:



Now read in the compiled data - we have oocytesize, spermsize, primaryspermatocytesize, and oocytearea. The species names in the files are inconsistent - they are fixed now. For now, the oocyte area is using the area formulation as made by the initial sperm size formula. All response variables are logged unless otherwise noted.

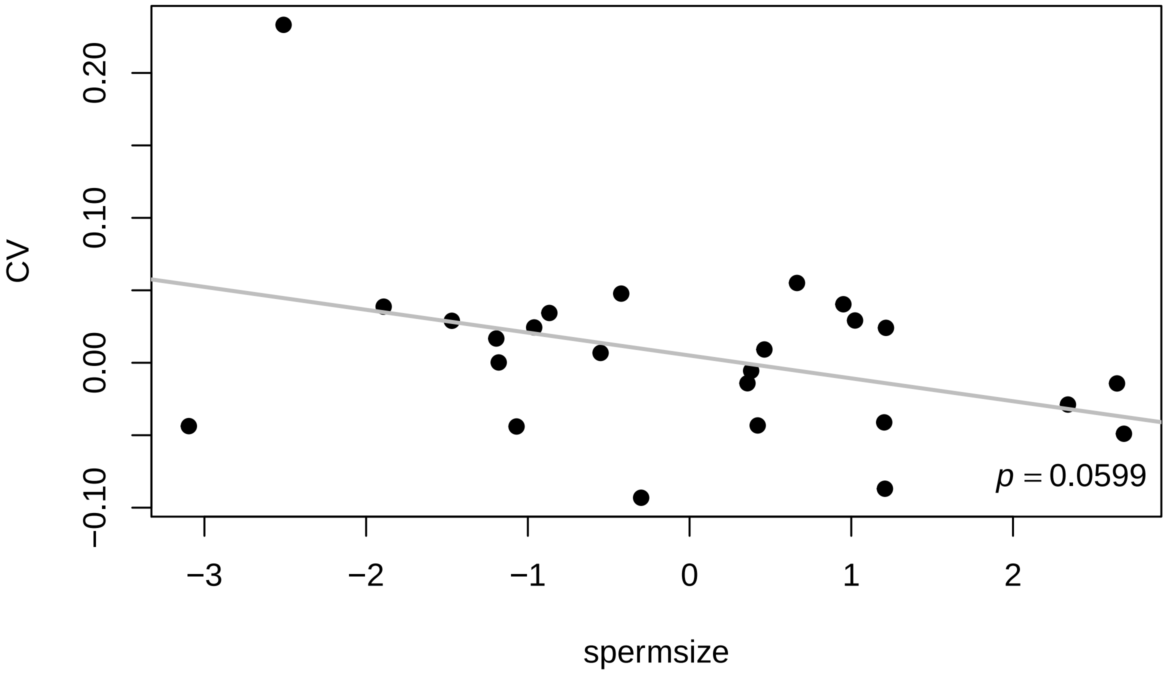
I have filtered out some of the data: Any oocytesize that is not stage -1, and any primaryspermatocyte data that is from hermaphrodites rather than males.

We now have clean data, clustered into species, strains or individuals. Variables ending in 1 are in species, 2 are strains and 3 are individuals. The non-clustered data is as is.

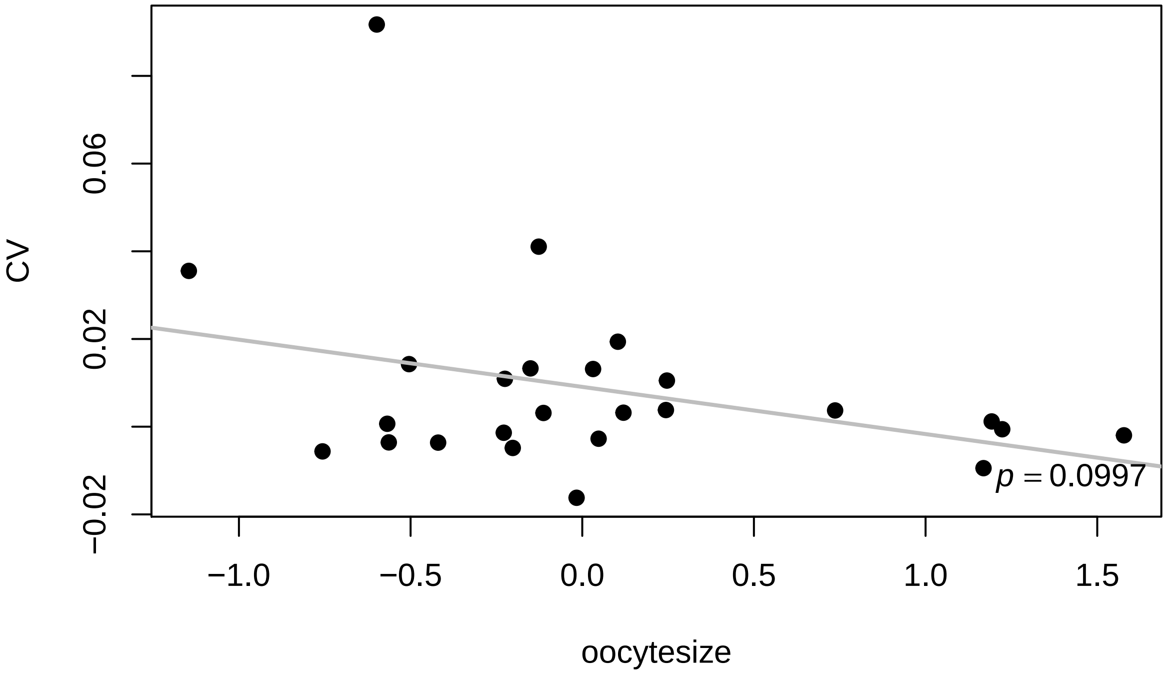
PIC plots - mean vs CV

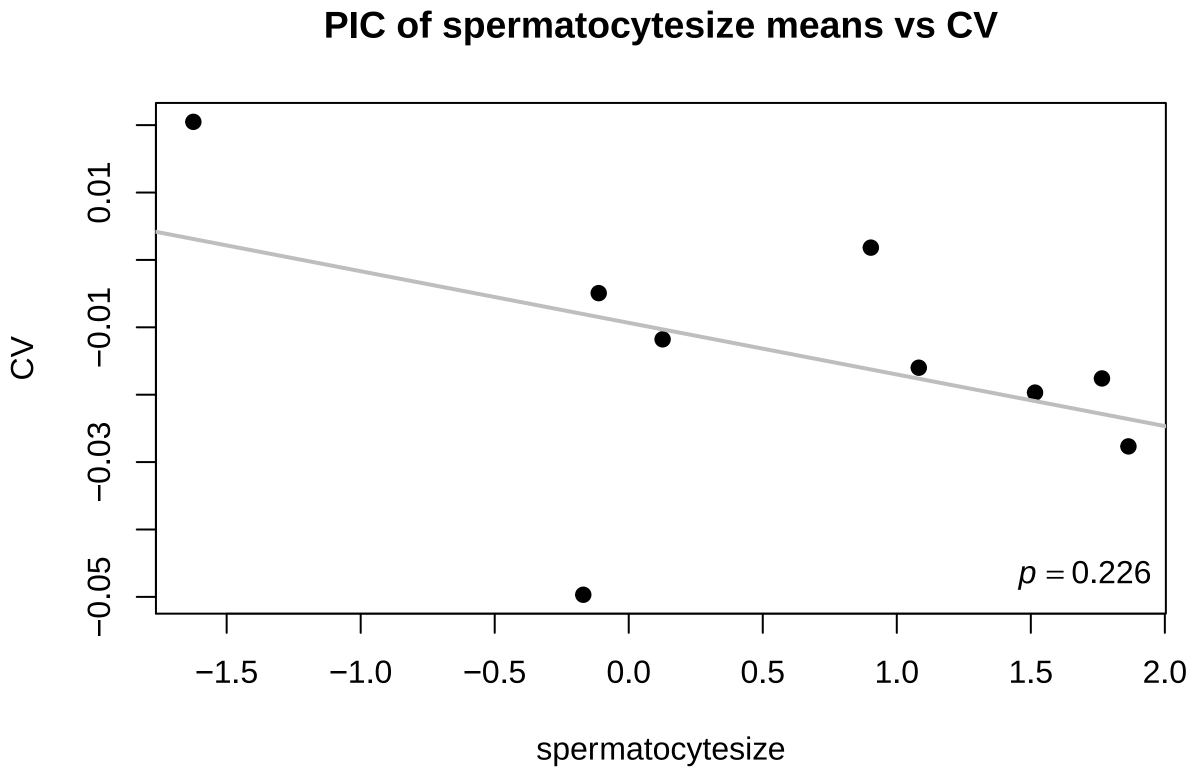
Plotting the phylogenetic independent contrasts (PIC) of means of traits vs CV of traits. All means and CVs are grouped at strain level.

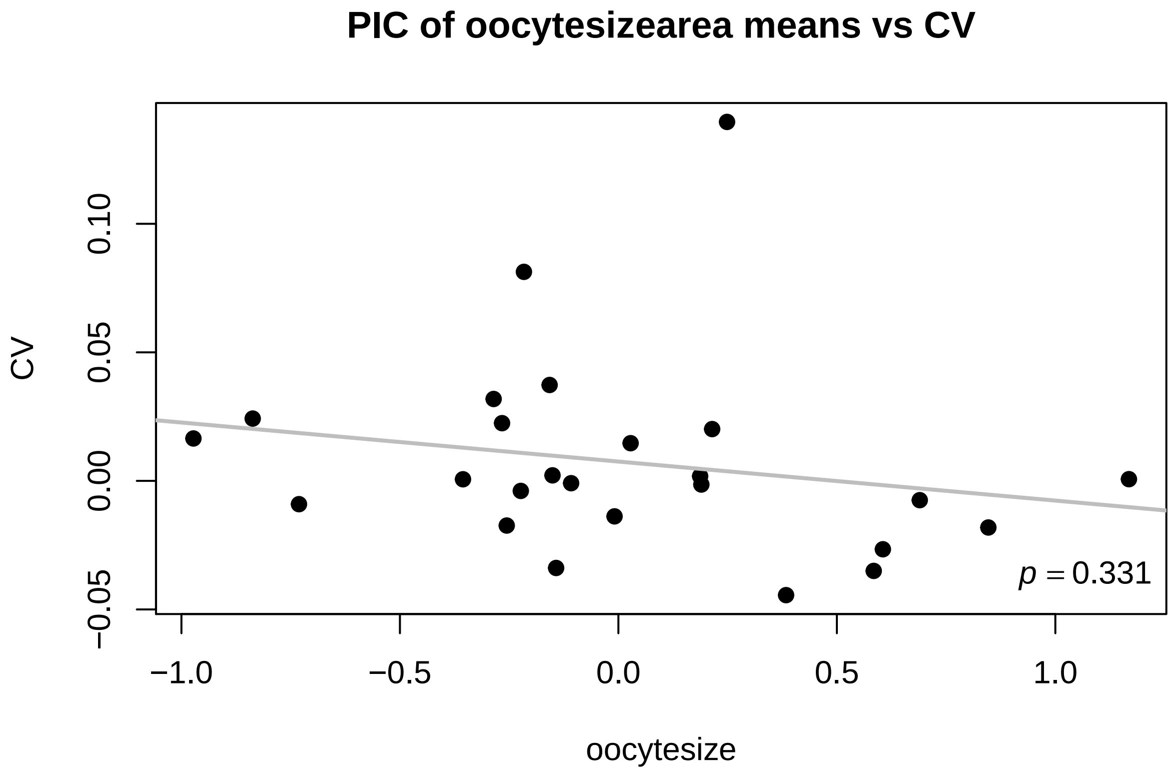
PIC of spermsize means vs CV

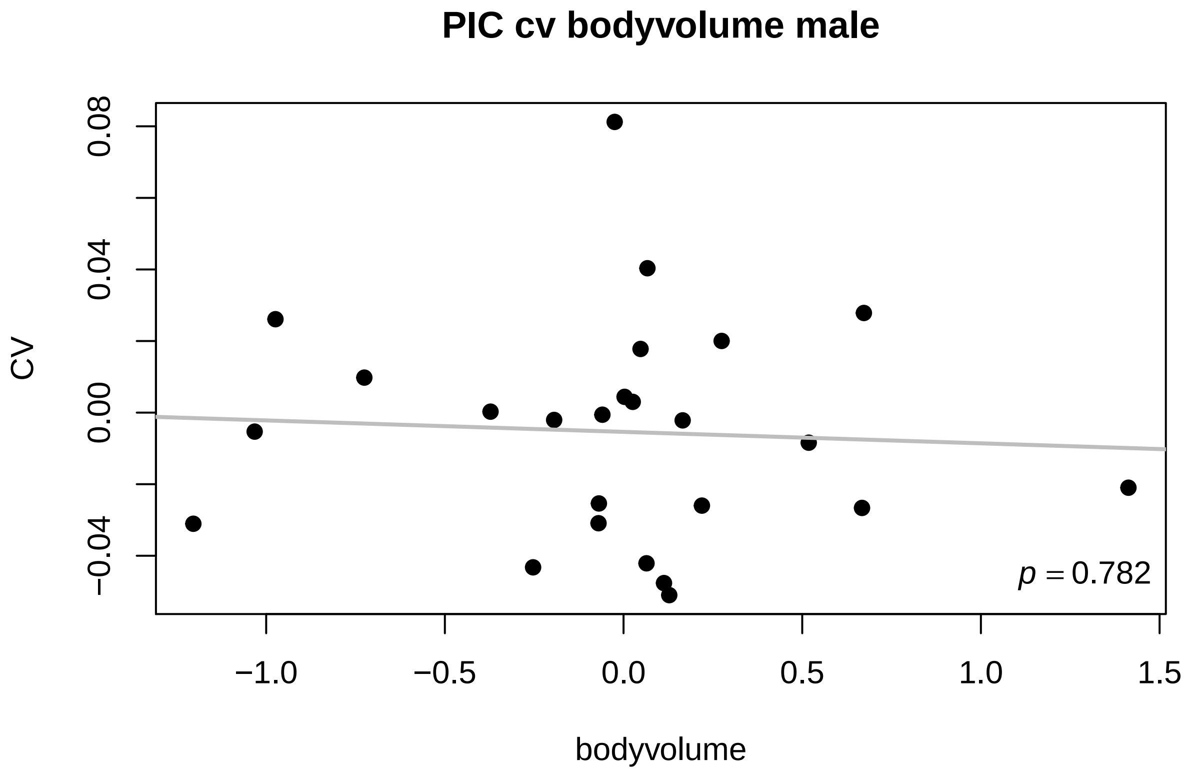


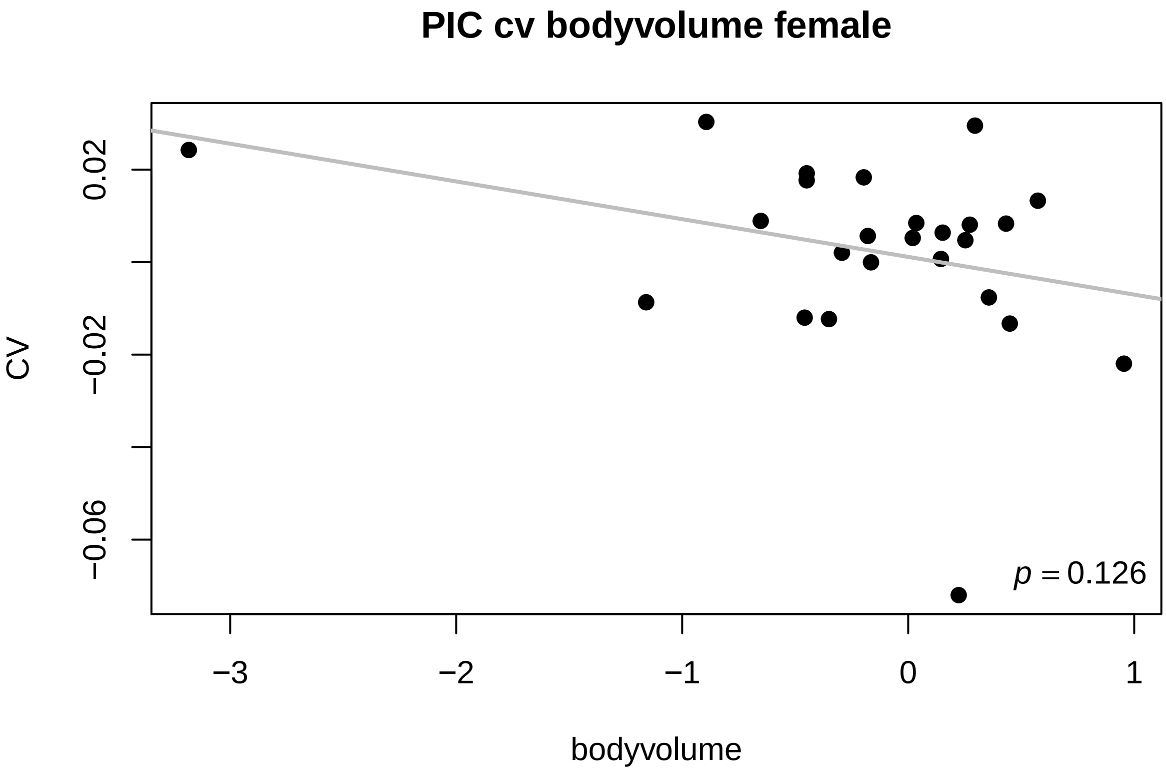
PIC of oocytesize means vs CV

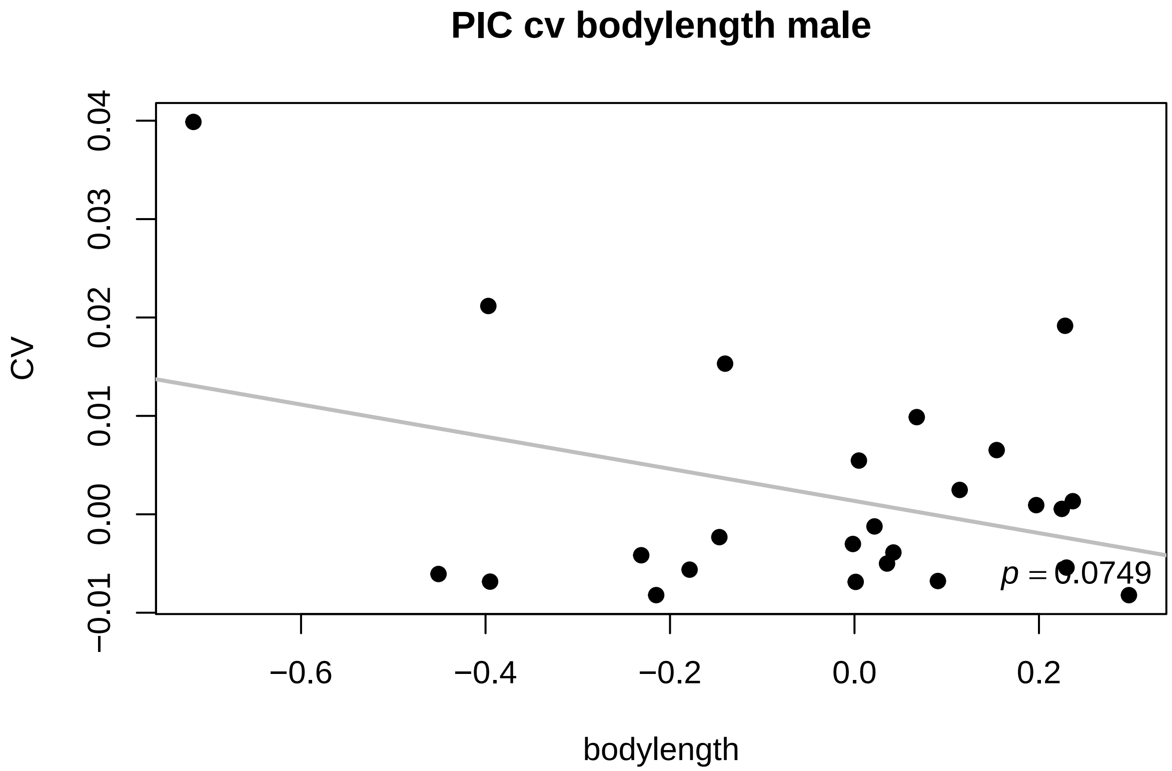


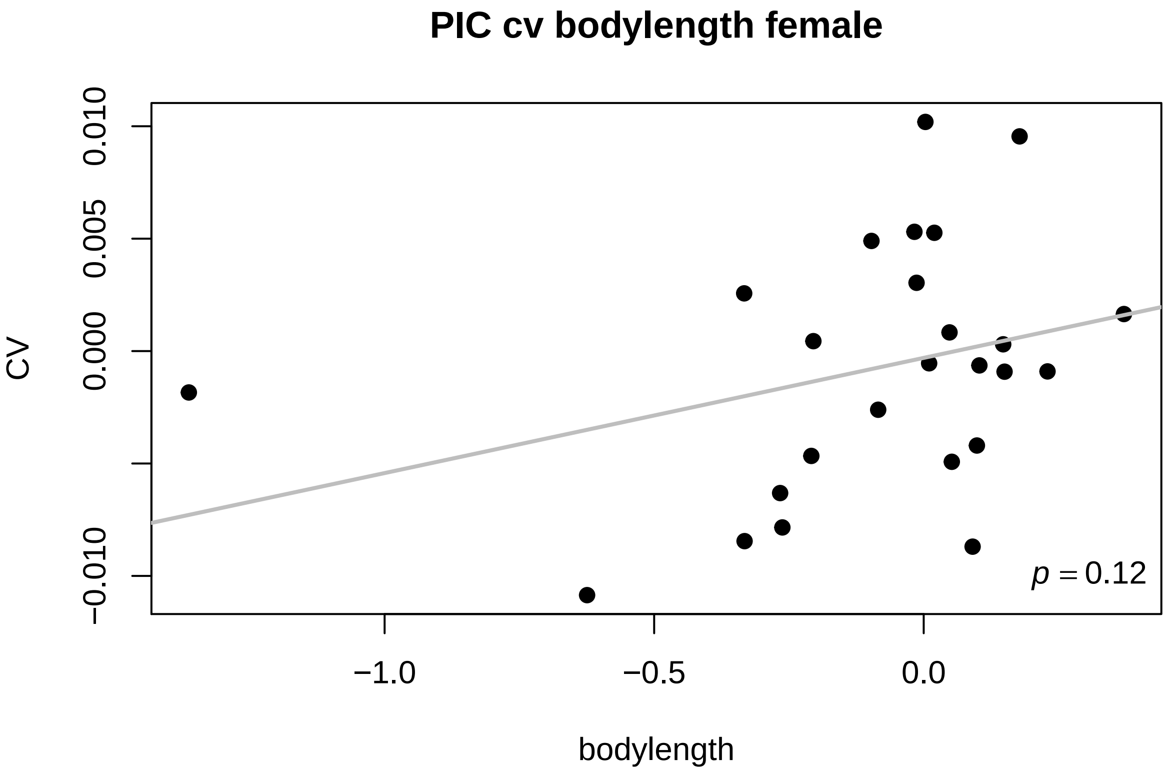

PIC of spermatocytesize means vs CV
CV
0.01
−0.01
−0.03
−0.05
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
spermatocytesize
$p = 0.226$


PIC of oocytesizearea means vs CV
CV
0.10
0.05
0.00
−0.05
−1.0
−0.5
0.0
0.5
1.0
oocytesize
$p = 0.331$

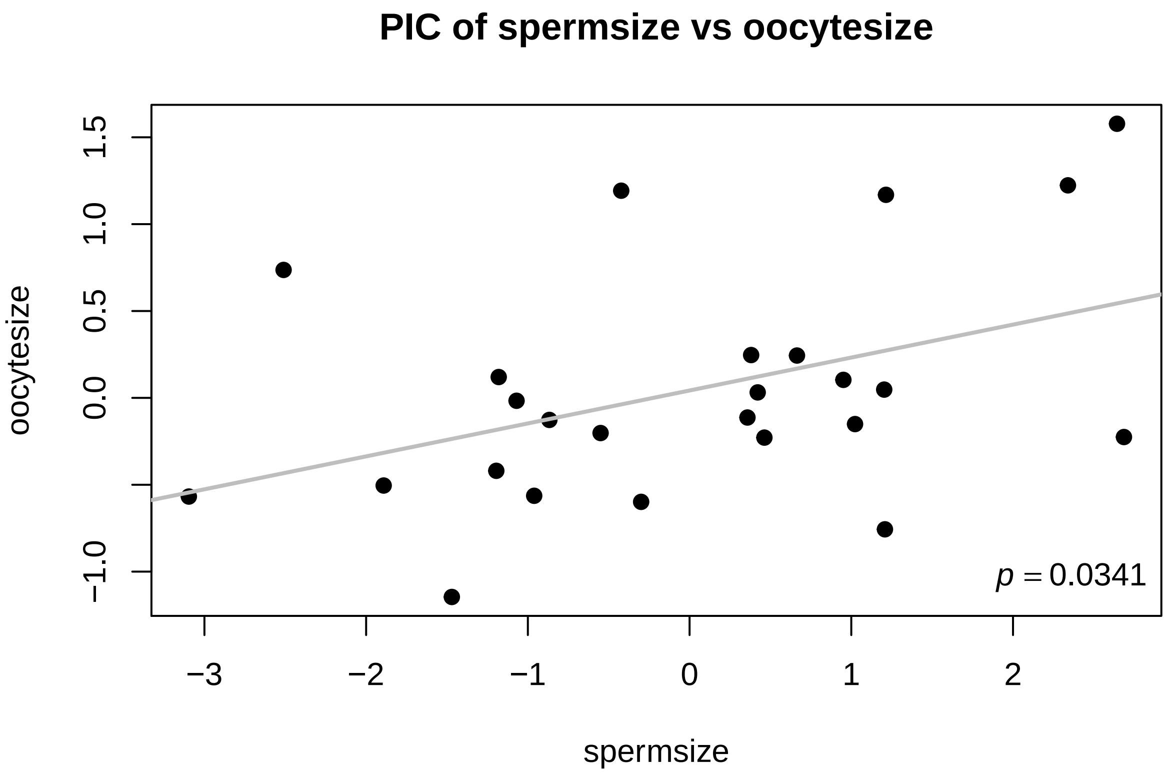

PIC cv bodyvolume male
CV
0.08
0.04
0.00
−0.04
−1.0
−0.5
0.0
0.5
1.0
1.5
bodyvolume
$p = 0.782$

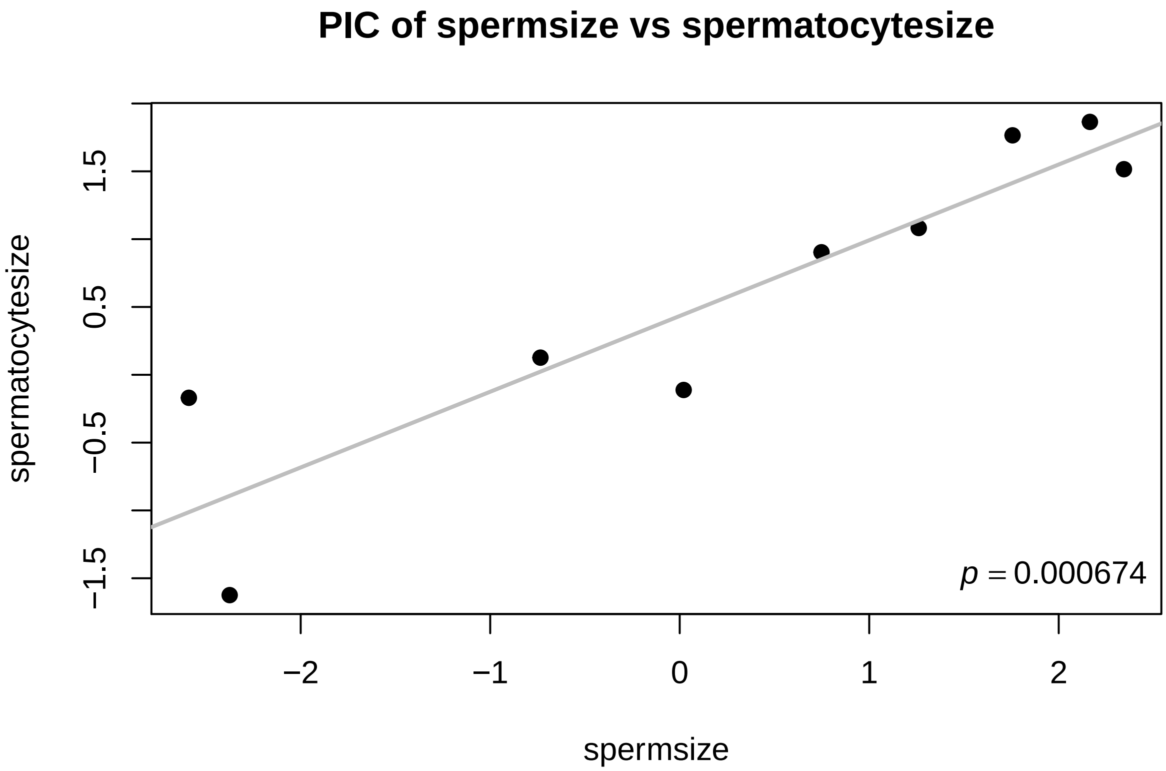

PIC cv bodyvolume female
CV
0.02
−0.02
−0.06
bodyvolume
−3
−2
−1
0
1
$p = 0.126$

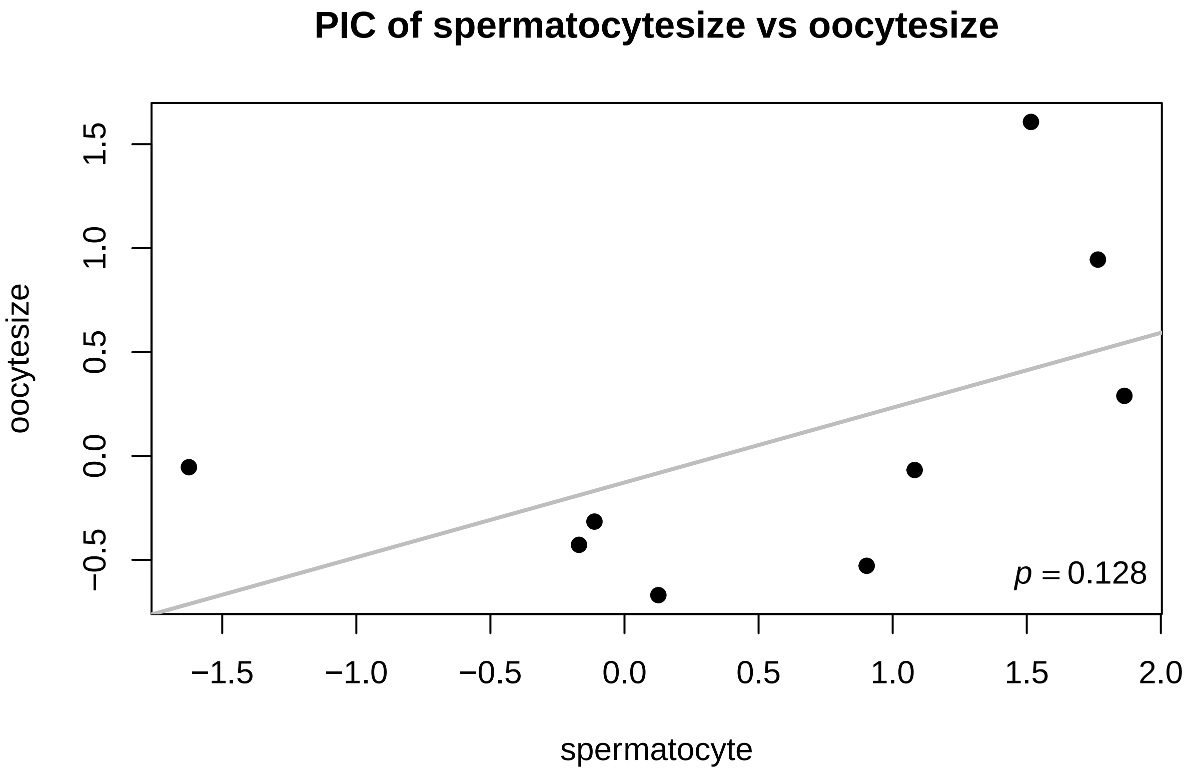

PIC cv bodylength male
CV
0.04
0.03
0.02
0.01
0.00
−0.01
−0.6
−0.4
−0.2
0.0
0.2
bodylength
p = 0.0749

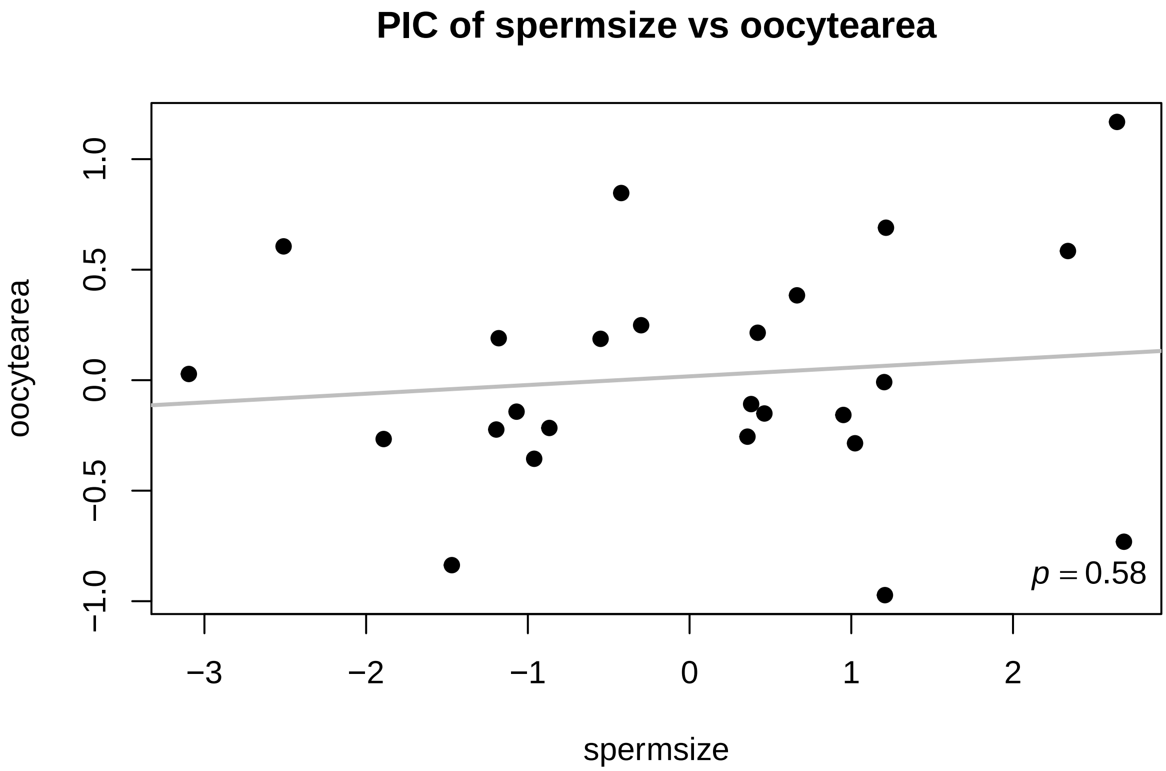

PIC cv bodylength female
CV
0.010
0.005
0.000
−0.010
−1.0
−0.5
0.0
bodylength
$p = 0.12$

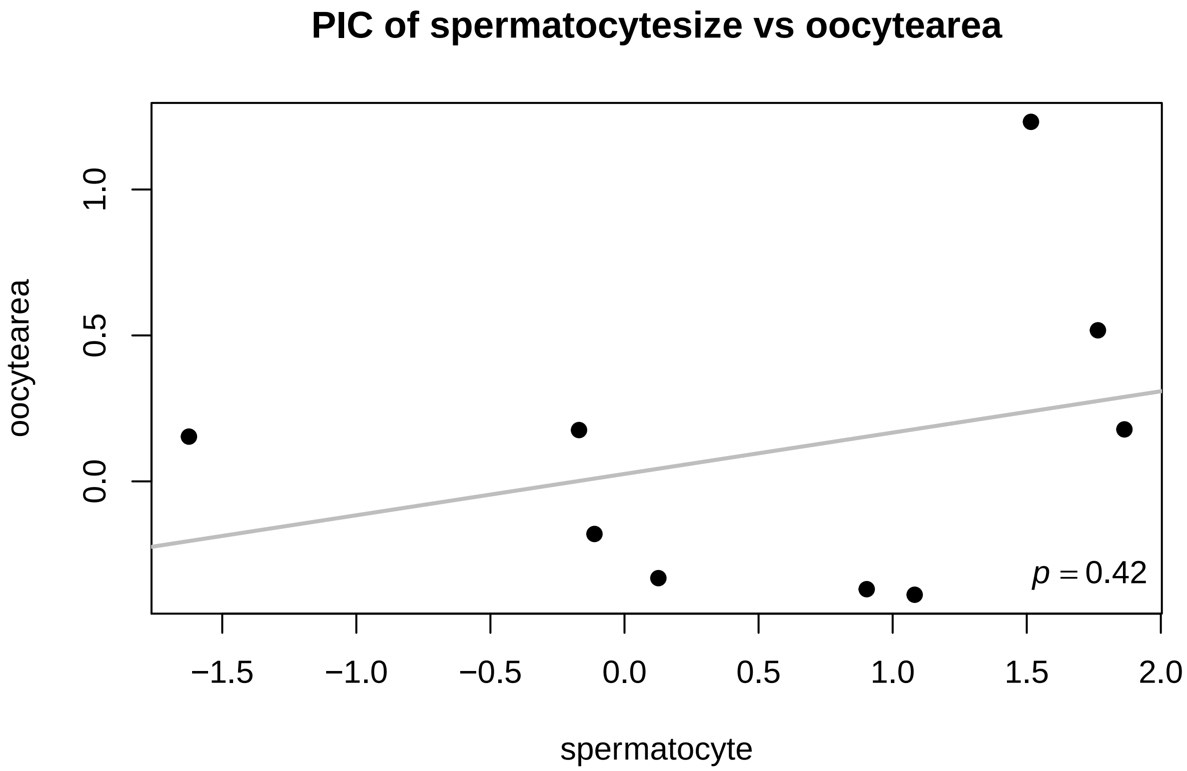
PIC plots - traits vs traits

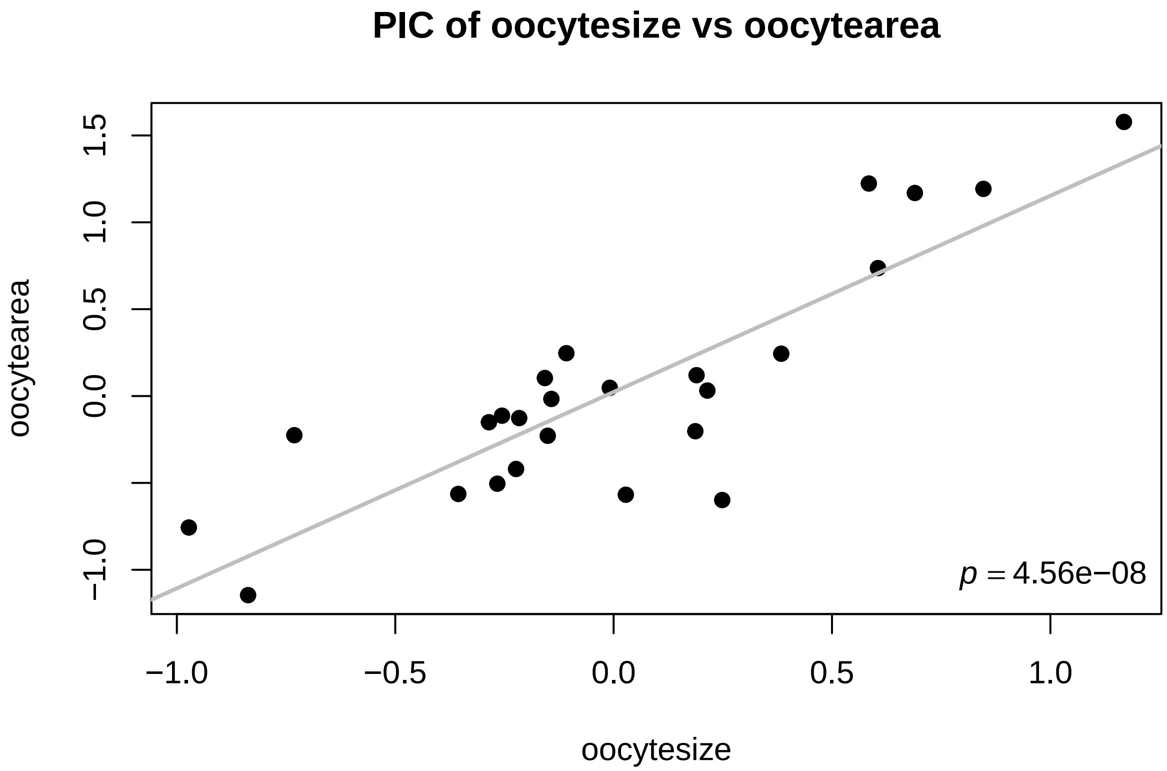


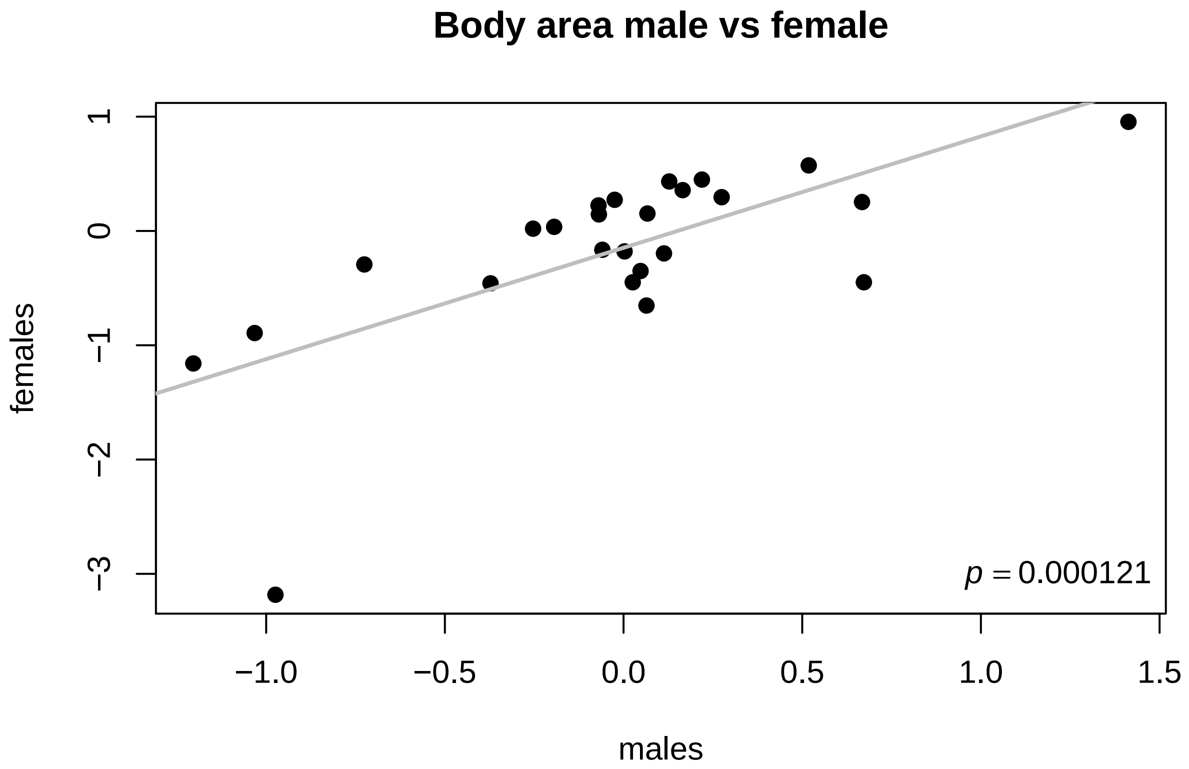


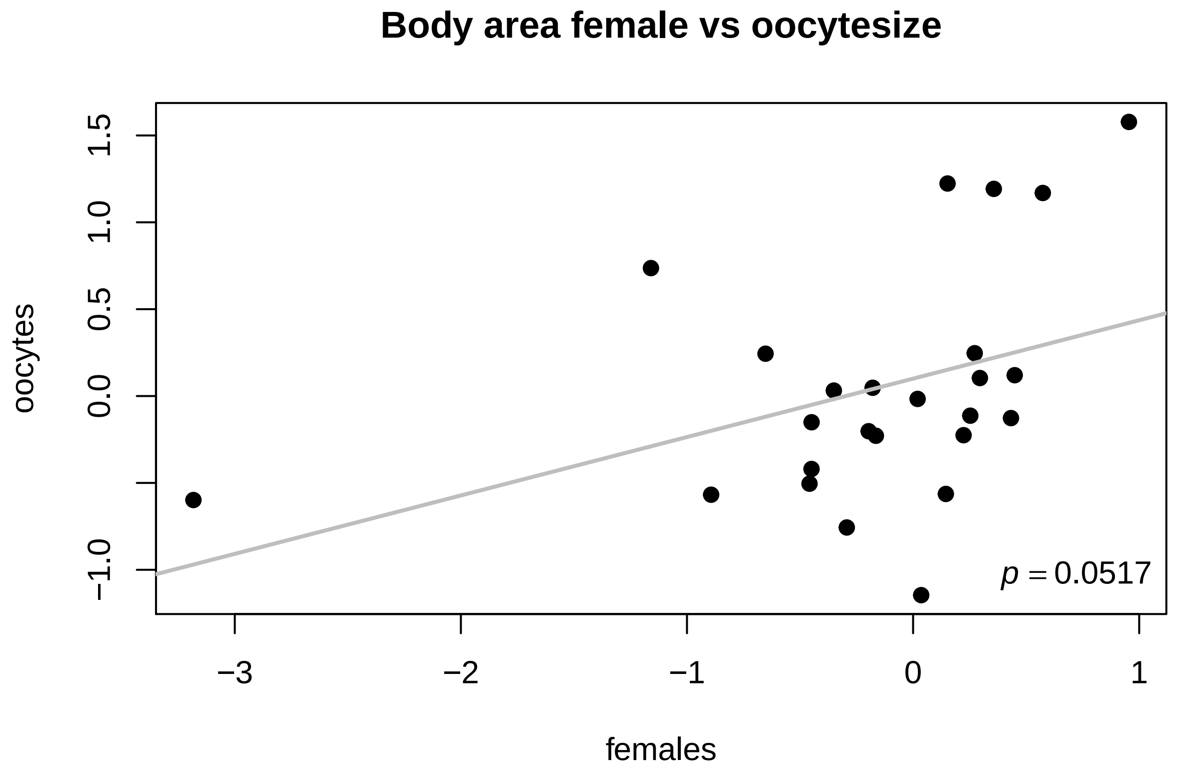

PIC of spermatocytesize vs oocytesize
oocytesize
1.5
1.0
0.5
0.0
−0.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
spermatocyte
$p = 0.128$

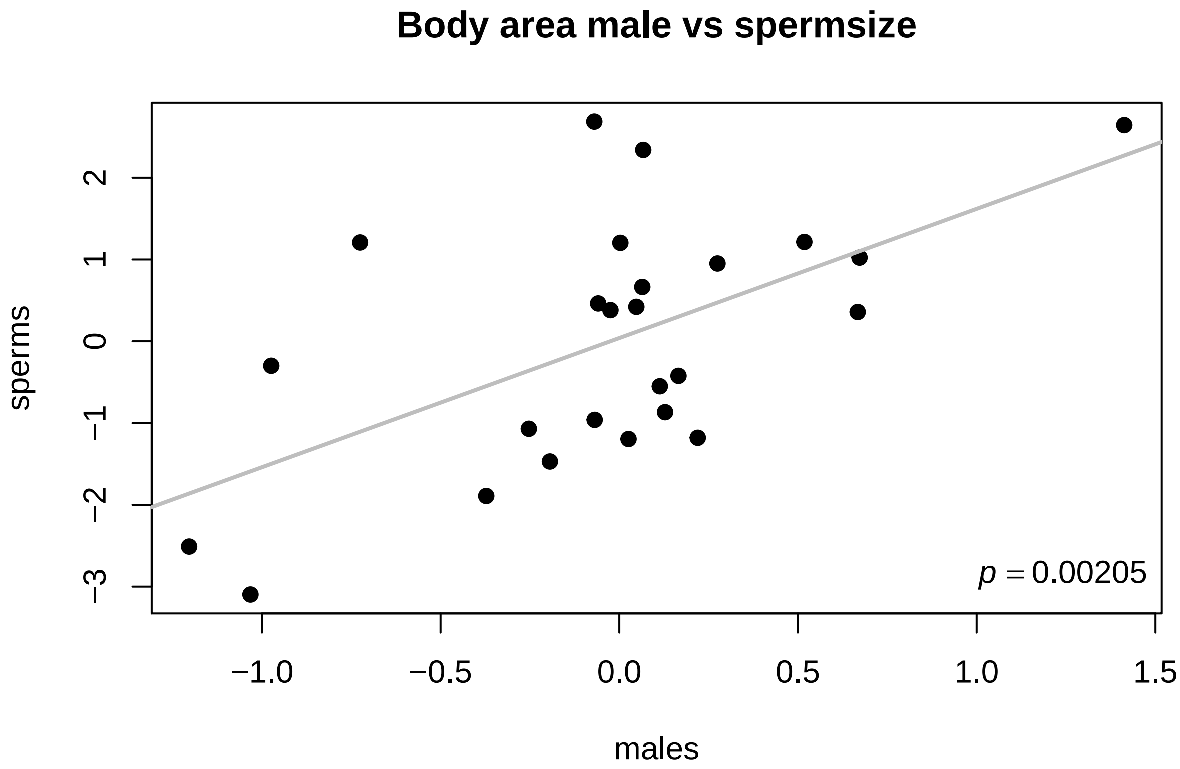

PIC of spermsize vs oocytearea
oocytearea
1.0
0.5
0.0
−0.5
−1.0
−3
−2
−1
0
1
2
spermsize
$p = 0.58$

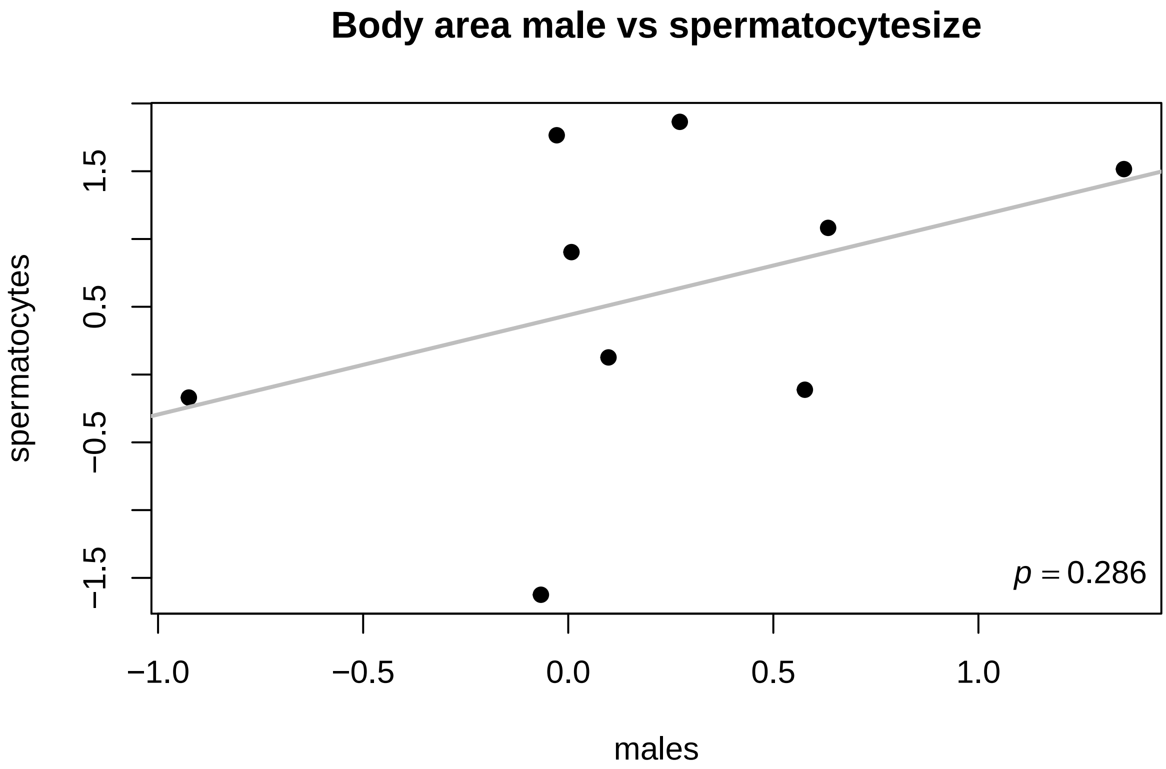

PIC of spermatocytesize vs oocytearea
oocytearea
1.0
0.5
0.0
p = 0.42
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
spermatocyte

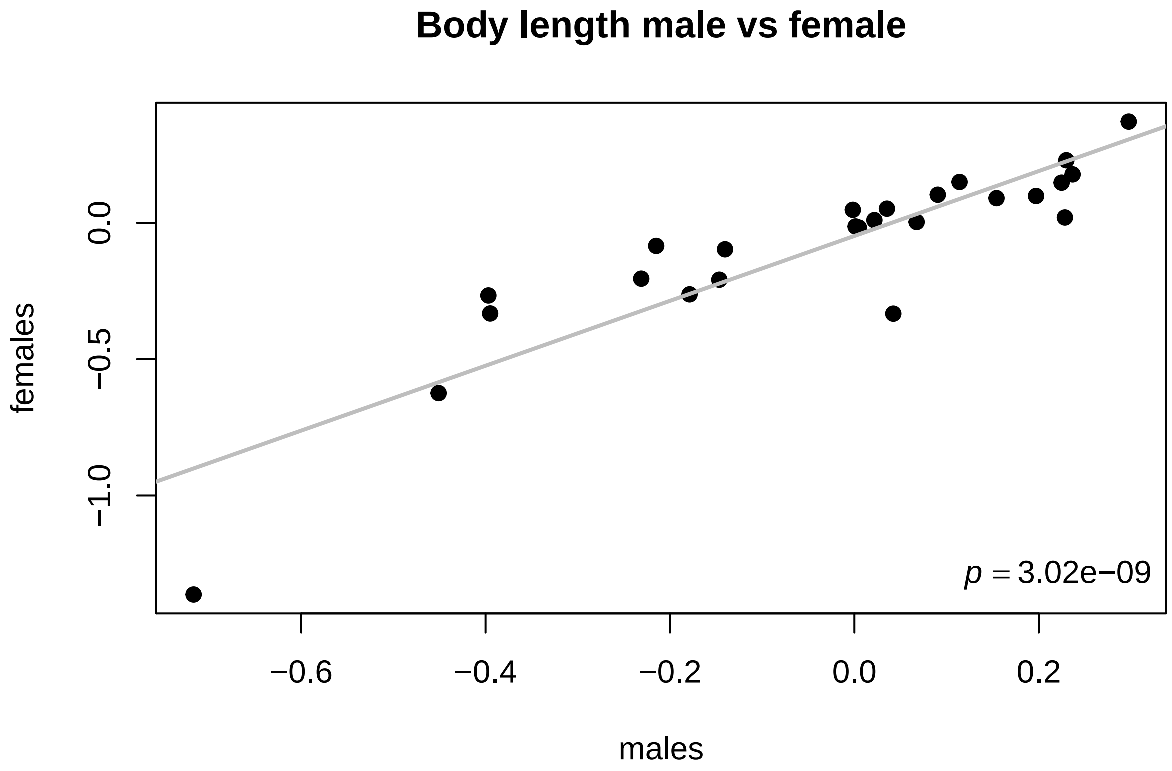

PIC of oocytesize vs oocytearea
oocytearea
1.5
1.0
0.5
0.0
−1.0
oocytesize
−1.0
−0.5
0.0
0.5
1.0
p = 4.56e−08

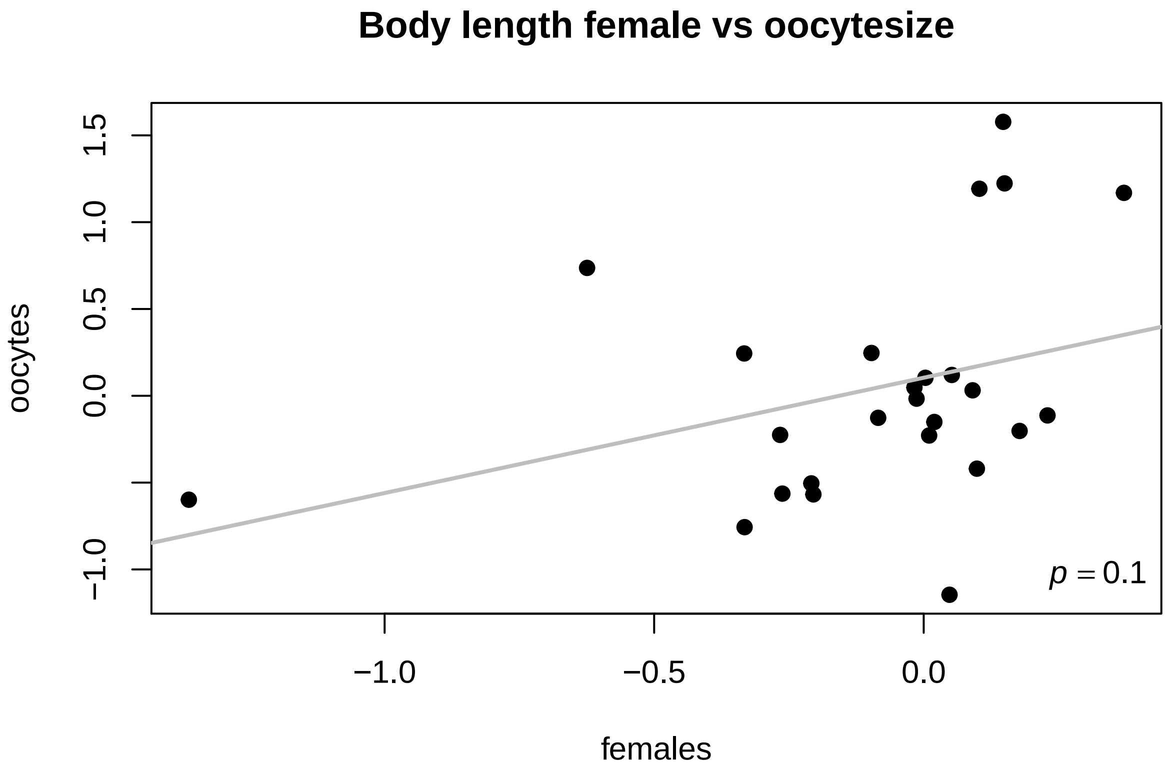

Body area male vs female
females
males
1
0
−1
−2
−3
−1.0
−0.5
0.0
0.5
1.0
1.5
$p = 0.000121$

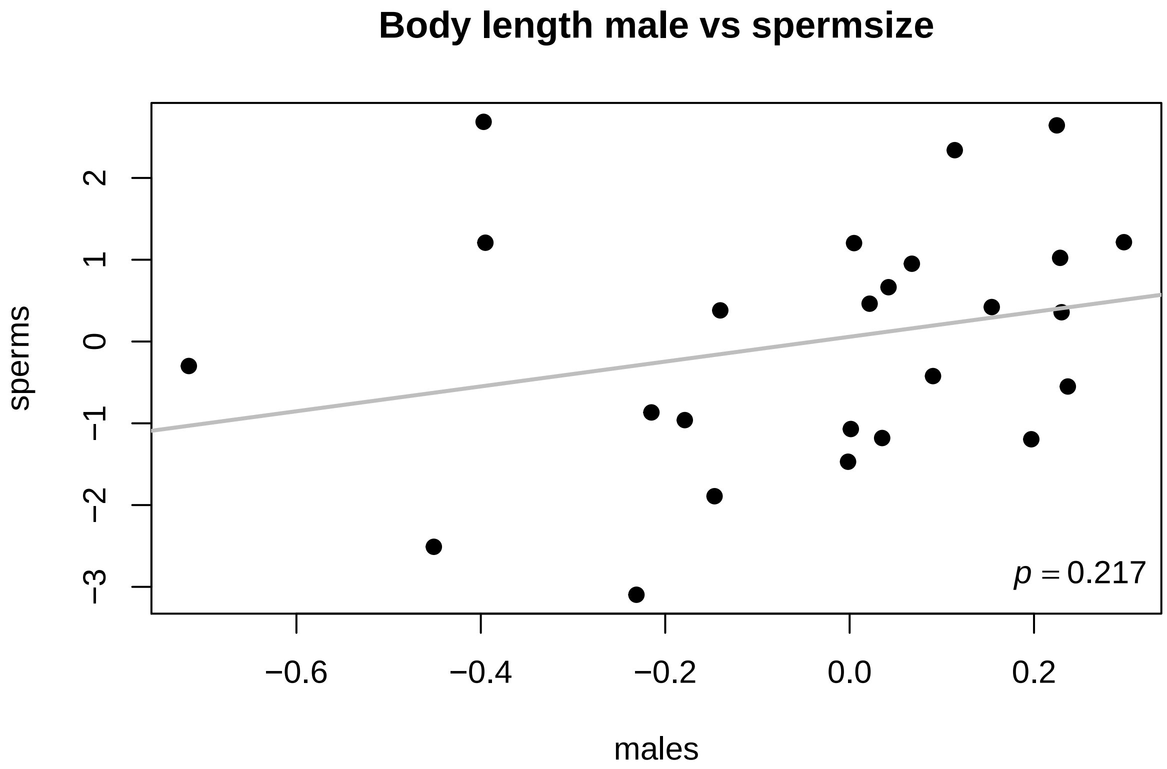

Body area female vs oocytesize
oocytes
1.5
1.0
0.5
0.0
−1.0
females
−3
−2
−1
0
1
$p = 0.0517$

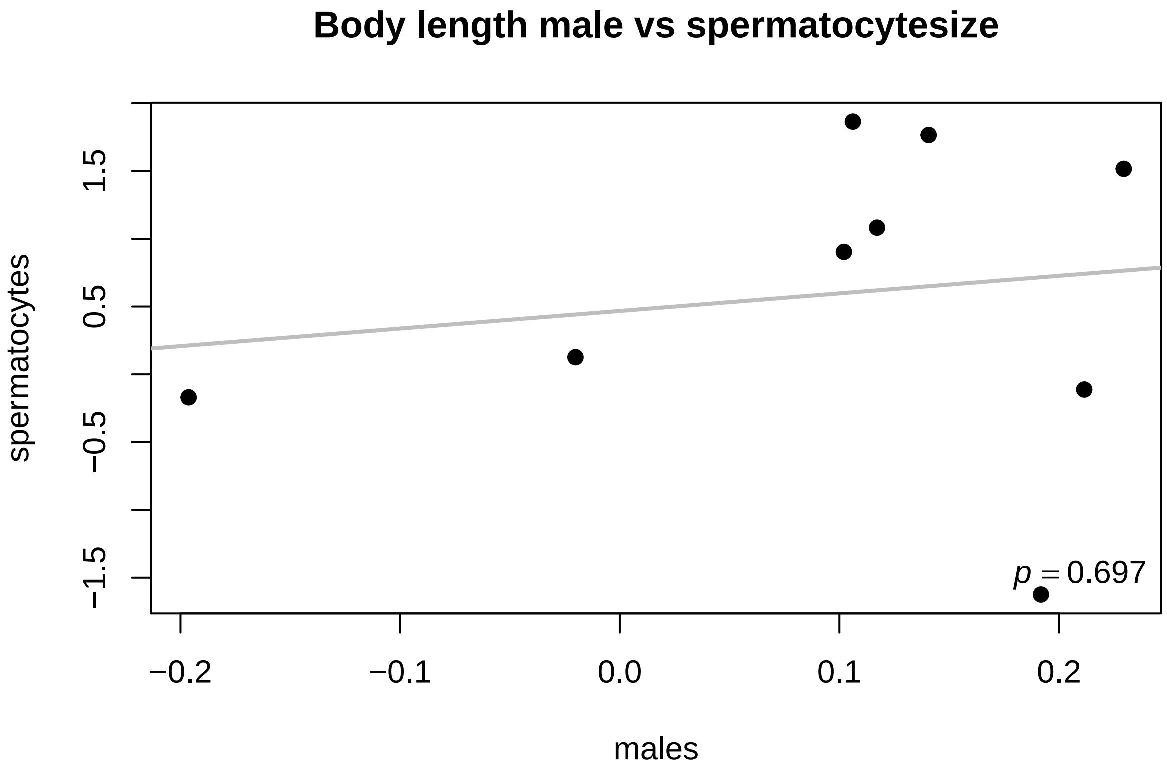

Body area male vs spermsize
sperms
2
1
0
−1
−2
−3
−1.0
−0.5
0.0
0.5
1.0
1.5
males
$p = 0.00205$


Body area male vs spermatocytesize
spermatocytes
males
−1.5
−0.5
0.5
1.5
−1.0
−0.5
0.0
0.5
1.0
p = 0.286


Body length male vs female
females
0.0
−0.5
−1.0
males
−0.6
−0.4
−0.2
0.0
0.2
$p = 3.02\text{e}{-09}$


Body length female vs oocytesize
oocytes
1.5
1.0
0.5
0.0
−1.0
p = 0.1
−1.0
−0.5
0.0
females


Body length male vs spermsize
sperms
2
1
0
−1
−2
−3
$p = 0.217$
−0.6
−0.4
−0.2
0.0
0.2
males



PGLS - testing for correlation with evolutionary models

Output are summary tables from fits. The OU model won't fit the spermatocyte fits due to overparameterisation.

```
##
## Call:
## pgls(formula = meansperm ~ meanooarea, data = compsperm, lambda = "ML")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9643 -0.5988  0.4323  0.6861  1.8263
##
## Branch length transformations:
##
## kappa  [Fix]  : 1.000
## lambda [ ML]  : 0.000
##    lower bound : 0.000, p = 1
##    upper bound : 1.000, p = 0.11287
##    95.0% CI   : (NA, NA)
## delta  [Fix]  : 1.000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.09037    6.57159 -0.9268   0.3812
## meanooarea   1.46171    0.94975  1.5390   0.1624
##
## Residual standard error: 1.057 on 8 degrees of freedom
## Multiple R-squared: 0.2284,  Adjusted R-squared: 0.132
## F-statistic: 2.369 on 1 and 8 DF,  p-value: 0.1624
```

```
##
## Call:
## pgls(formula = meansperm ~ meanoo, data = compsperm, lambda = "ML")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4586 -1.1960  0.6374  1.0989  1.4686
##
## Branch length transformations:
##
## kappa  [Fix]  : 1.000
## lambda [ ML]  : 0.895
##    lower bound : 0.000, p = 1
##    upper bound : 1.000, p = 0.79696
##    95.0% CI   : (NA, NA)
## delta  [Fix]  : 1.000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.31303    6.94067 -1.6300  0.14176
## meanoo        1.50685    0.65983  2.2837  0.05177 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.315 on 8 degrees of freedom
## Multiple R-squared: 0.3946,  Adjusted R-squared: 0.319
## F-statistic: 5.215 on 1 and 8 DF,  p-value: 0.05177
```

```
##
## Call:
## pgls(formula = meansperm ~ meanprimsp, data = compsperm, lambda = "ML")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8244 -0.3185 -0.1789  0.1091  0.5614
##
## Branch length transformations:
##
## kappa  [Fix]  : 1.000
## lambda [ ML]  : 0.000
##    lower bound : 0.000, p = 1
##    upper bound : 1.000, p = 0.0066141
##    95.0% CI   : (NA, 0.708)
## delta  [Fix]  : 1.000
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) -2.63590    0.94589 -2.7867 0.0236793 *
## meanprimsp   1.21417    0.17131  7.0873 0.0001033 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.446 on 8 degrees of freedom
## Multiple R-squared: 0.8626,  Adjusted R-squared: 0.8454
## F-statistic: 50.23 on 1 and 8 DF,  p-value: 0.0001033
```

```
## Generalized least squares fit by REML
##   Model: meansperm ~ meanooarea
##   Data: fullpglsframe
##        AIC      BIC    logLik
##   54.95884 59.67106 -23.47942
##
## Correlation Structure: corMartins
##  Formula: ~1
##  Parameter estimate(s):
##    alpha
## 6.841132
##
## Coefficients:
##                 Value Std.Error    t-value p-value
## (Intercept) -3.044212  3.398811 -0.8956698  0.3793
## meanooarea   1.005070  0.498035  2.0180725  0.0549
##
##  Correlation:
##            (Intr)
## meanooarea -0.999
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -1.7775727 -0.6837636 -0.2886598  0.4237510  2.2862305
##
## Residual standard error: 0.6334179
## Degrees of freedom: 26 total; 24 residual
```

```
## Generalized least squares fit by REML
##   Model: meansperm ~ meanoo
##   Data: fullpglsframe
##        AIC      BIC    logLik
##   50.30285 55.01506 -21.15142
##
## Correlation Structure: corMartins
##  Formula: ~1
##  Parameter estimate(s):
##    alpha
## 5.469326
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept) -7.068653  3.346328 -2.112361  0.0452
## meanoo       1.074887  0.329803  3.259179  0.0033
##
##  Correlation:
```

```
##        (Intr)
## meanoo -0.999
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -1.9084685 -0.7013875 -0.3474741  0.5038124  2.4034645
##
## Residual standard error: 0.5843052
## Degrees of freedom: 26 total; 24 residual
```

Some phylogenetic signal methods - lambda

```
## Loading required package: parallel
```

```
##     Length Class      Mode
## lik   1    bm         function
## bnd   2    data.frame list
## res 400    -none-     numeric
## opt   8    -none-     list
```

```
##     Length Class      Mode
## lik   1    bm         function
## bnd   2    data.frame list
## res 400    -none-     numeric
## opt   8    -none-     list
```

```
##     Length Class      Mode
## lik   1    bm         function
## bnd   2    data.frame list
## res 400    -none-     numeric
## opt   8    -none-     list
```

```
##     Length Class      Mode
## lik   1    bm         function
## bnd   2    data.frame list
## res 400    -none-     numeric
## opt   8    -none-     list
```

Blombergs K

```
##           K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 0.3738911         2.289904              3.920744          0.044
##   PIC.variance.Z
## 1      -1.403023
```

```
##          K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 0.670685        0.4572932              1.224798          0.001
##   PIC.variance.Z
## 1      -1.948734
## [1] "Dropping tips from the tree because they are not present in the data:"
##  [1] "C. sp. 1"        "C. portoensis"   "C. virilis"
##  [4] "C. sp. 8"        "C. angaria"      "C. castelli"
##  [7] "C. sp. 2"        "C. kamaaina"     "C. imperialis"
## [10] "C. afra"         "C. nouraguensis" "C. doughertyi"
## [13] "C. wallacei"     "C. tropicalis"   "C. sp. 5"
## [16] "C. nigoni"
```

```
##          K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 1.218587         1.696502              2.569825          0.035
##   PIC.variance.Z
## 1      -1.818388
```

```
##           K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 0.5877787        0.2622437             0.6166013          0.001
##   PIC.variance.Z
## 1      -1.785553
```

Morans I

```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##     obs = res$obs, alter = alter, names = test.names)
##
## Number of tests:   1
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test         Obs   Std.Obs   Alter Pvalue
## 1    x -0.09396389 -1.384882 greater  0.943
##
## other elements: adj.method call
```

```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##     obs = res$obs, alter = alter, names = test.names)
##
## Number of tests:   1
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test         Obs   Std.Obs   Alter Pvalue
## 1    x -0.01730539 0.6379105 greater  0.219
##
## other elements: adj.method call
```

```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##     obs = res$obs, alter = alter, names = test.names)
```

```
##
## Number of tests:   1
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test        Obs    Std.Obs   Alter Pvalue
## 1    x -0.1482137 -0.8806813 greater  0.805
##
## other elements: adj.method call
```

```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##     obs = res$obs, alter = alter, names = test.names)
##
## Number of tests:   1
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test         Obs   Std.Obs   Alter Pvalue
## 1    x -0.03082416 0.2856757 greater  0.358
##
## other elements: adj.method call
```

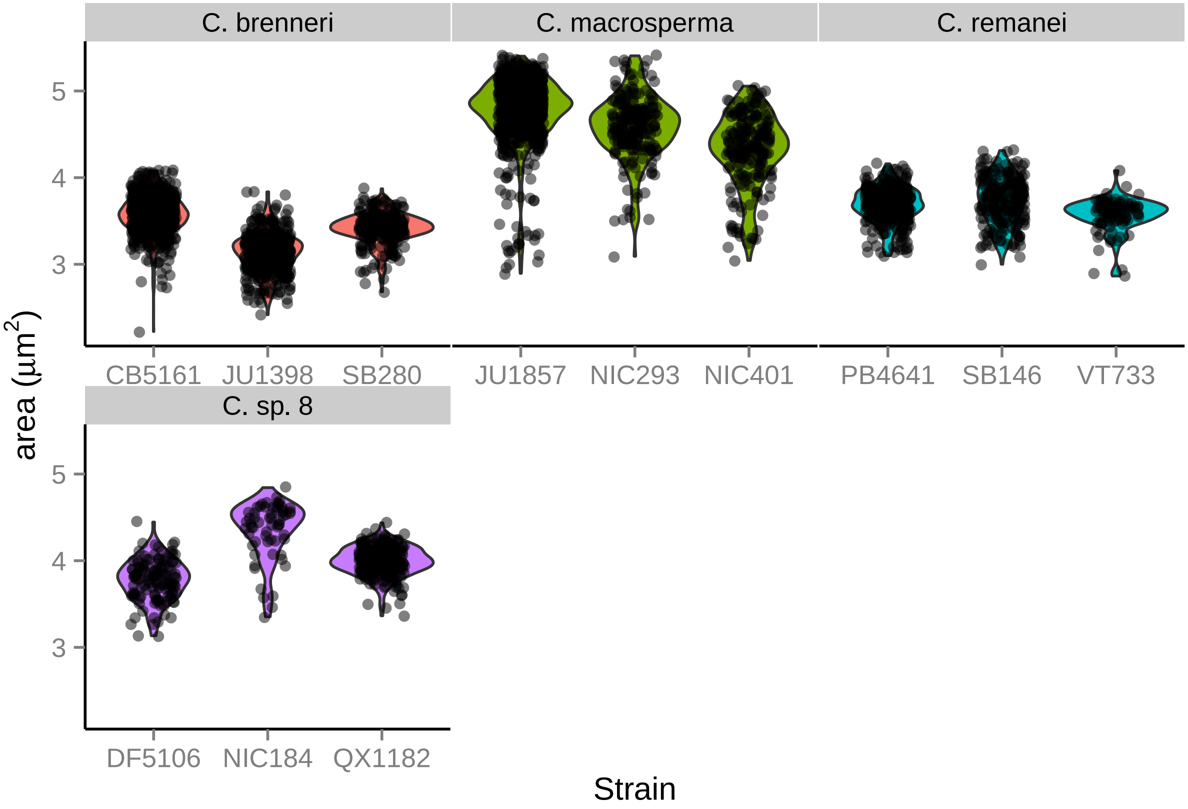
abouheifs cmean

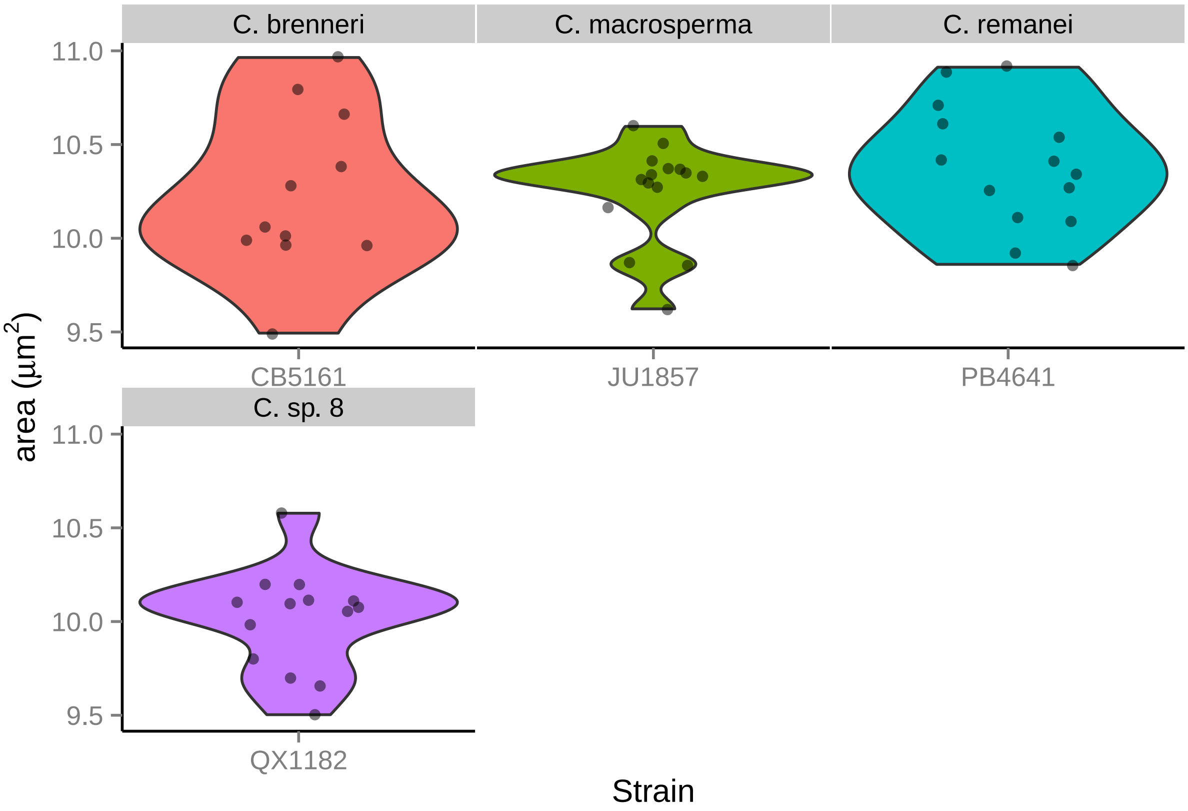
```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##     obs = res$obs, alter = alter, names = test.names)
##
## Number of tests:   1
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test         Obs   Std.Obs   Alter Pvalue
## 1    x -0.09396389 -1.374005 greater   0.94
##
## other elements: adj.method call
```

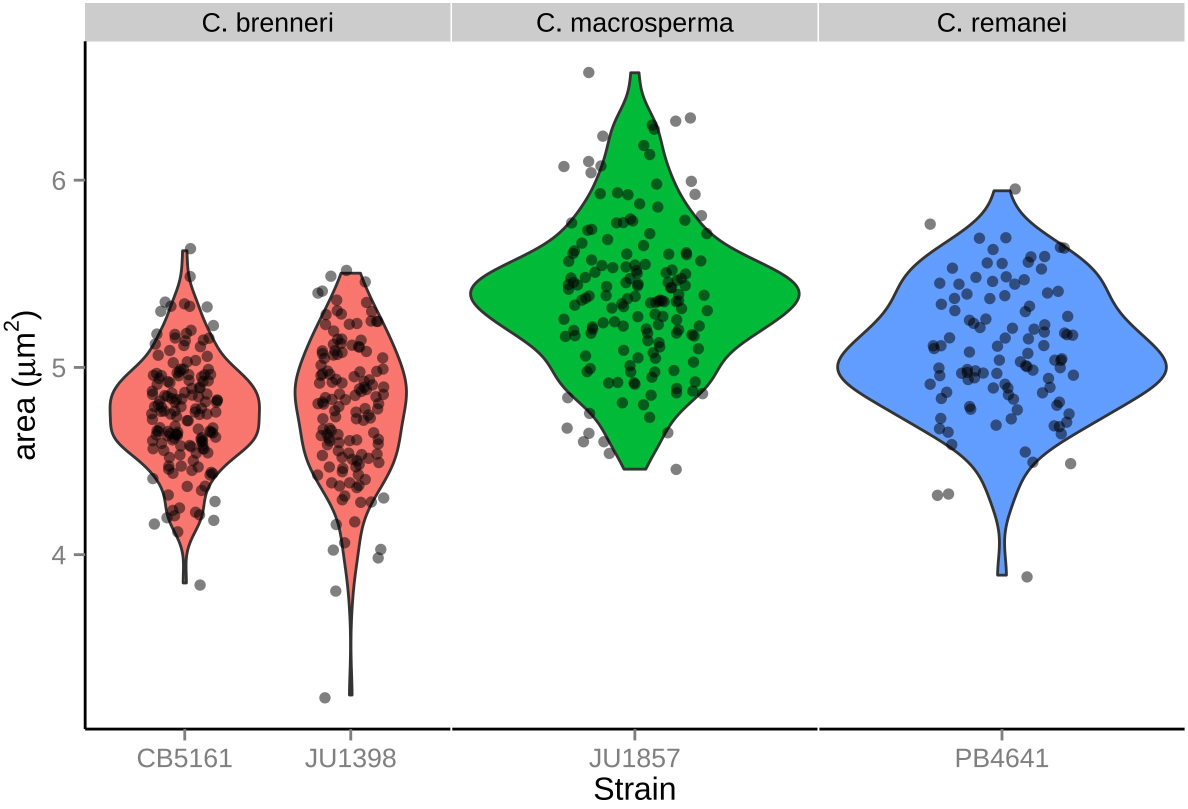
```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##     obs = res$obs, alter = alter, names = test.names)
##
## Number of tests:   1
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test         Obs  Std.Obs   Alter Pvalue
## 1    x -0.01730539 0.632356 greater  0.227
##
## other elements: adj.method call
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##     obs = res$obs, alter = alter, names = test.names)
##
## Number of tests:   1
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test        Obs    Std.Obs   Alter Pvalue
## 1    x -0.1482137 -0.8759883 greater  0.801
##
## other elements: adj.method call
```

```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##     obs = res$obs, alter = alter, names = test.names)
##
## Number of tests:   1
##
## Adjustment method for multiple comparisons:   none
## Permutation number:   999
##   Test         Obs   Std.Obs   Alter Pvalue
## 1    x -0.03082416 0.2717917 greater   0.37
##
## other elements: adj.method call
```

figure 3A with violin plots


C. brenneri
C. macrosperma
C. remanei
C. sp. 8
area (μm²)
5
4
3
CB5161
JU1398
SB280
JU1857
NIC293
NIC401
PB4641
SB146
VT733
DF5106
NIC184
QX1182
Strain


C. brenneri
C. macrosperma
C. remanei
C. sp. 8
area (μm²)
11.0
10.5
10.0
9.5
CB5161
JU1857
PB4641
QX1182
Strain


C. brenneri
C. macrosperma
C. remanei
area (μm^2)
6
5
4
CB5161
JU1398
JU1857
PB4641
Strain

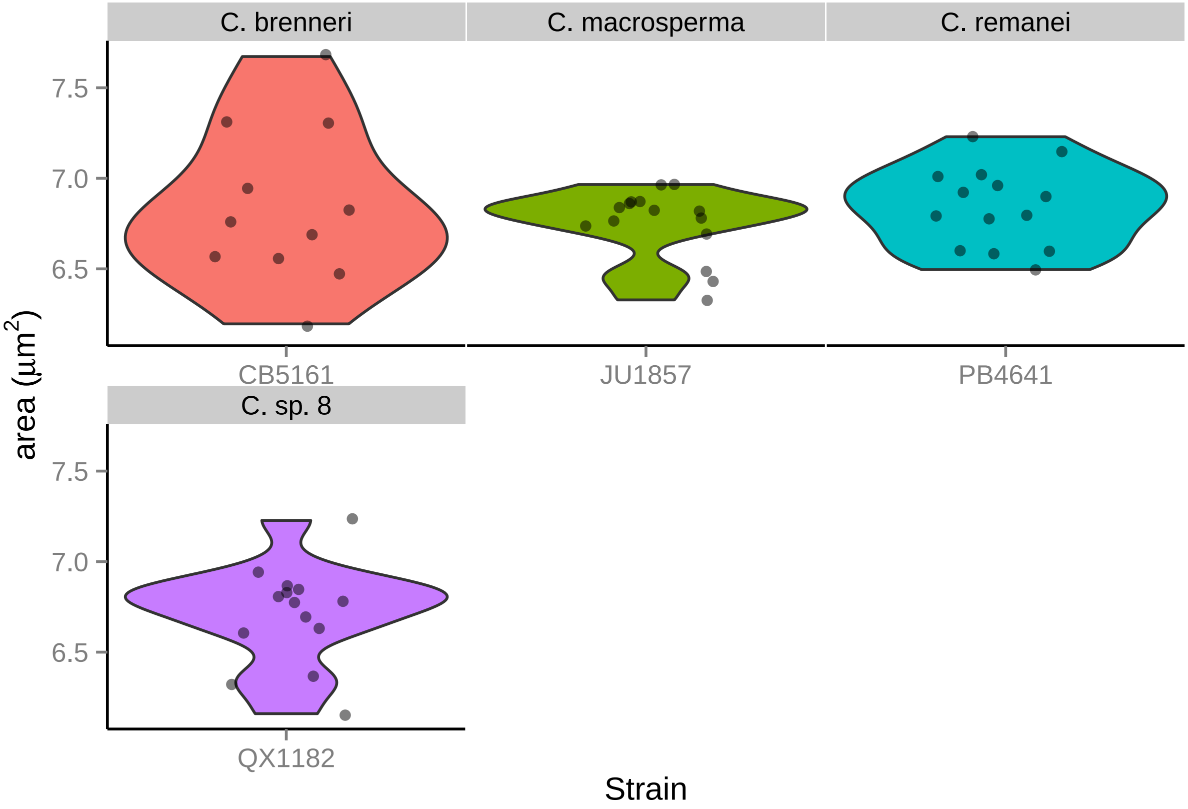
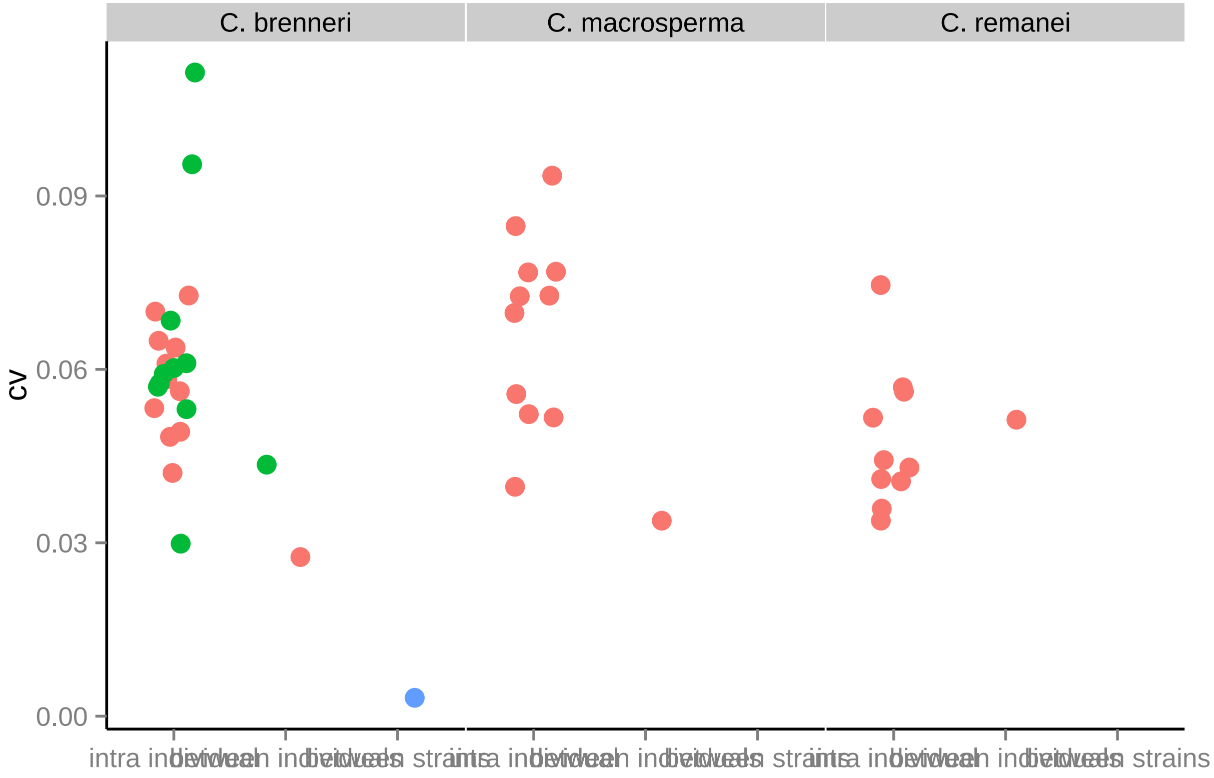
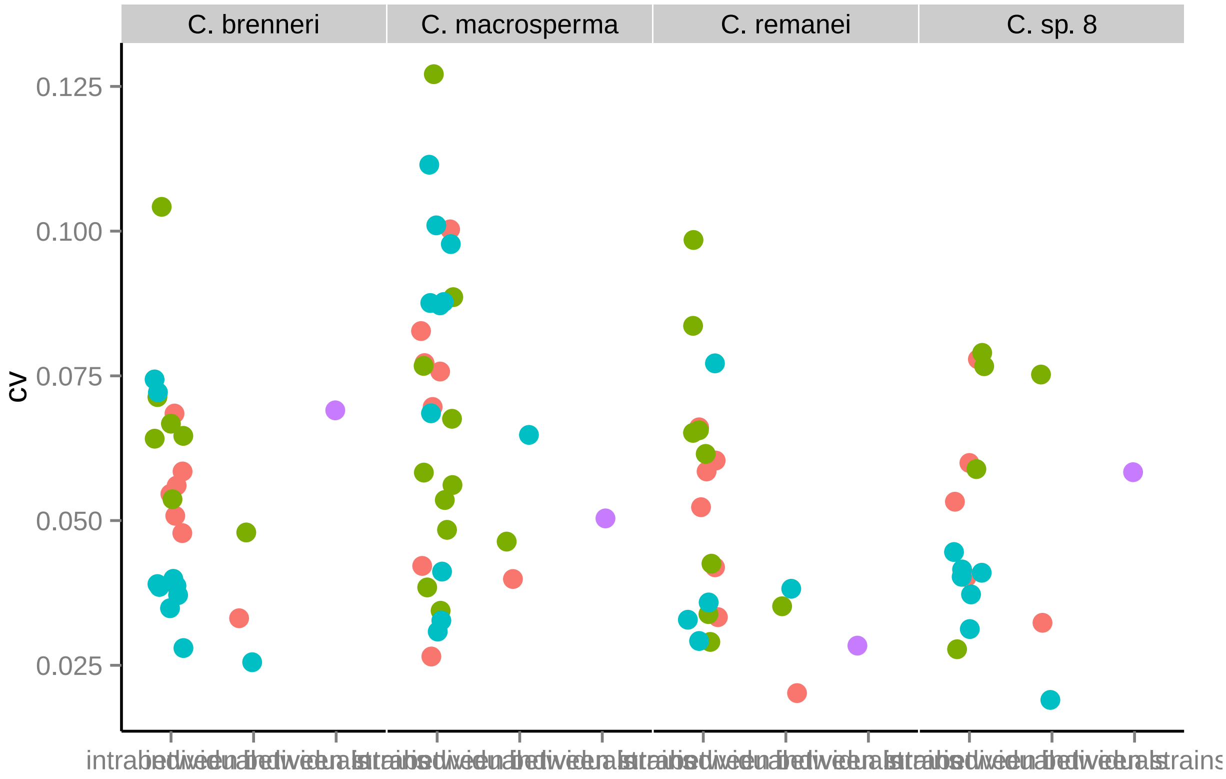
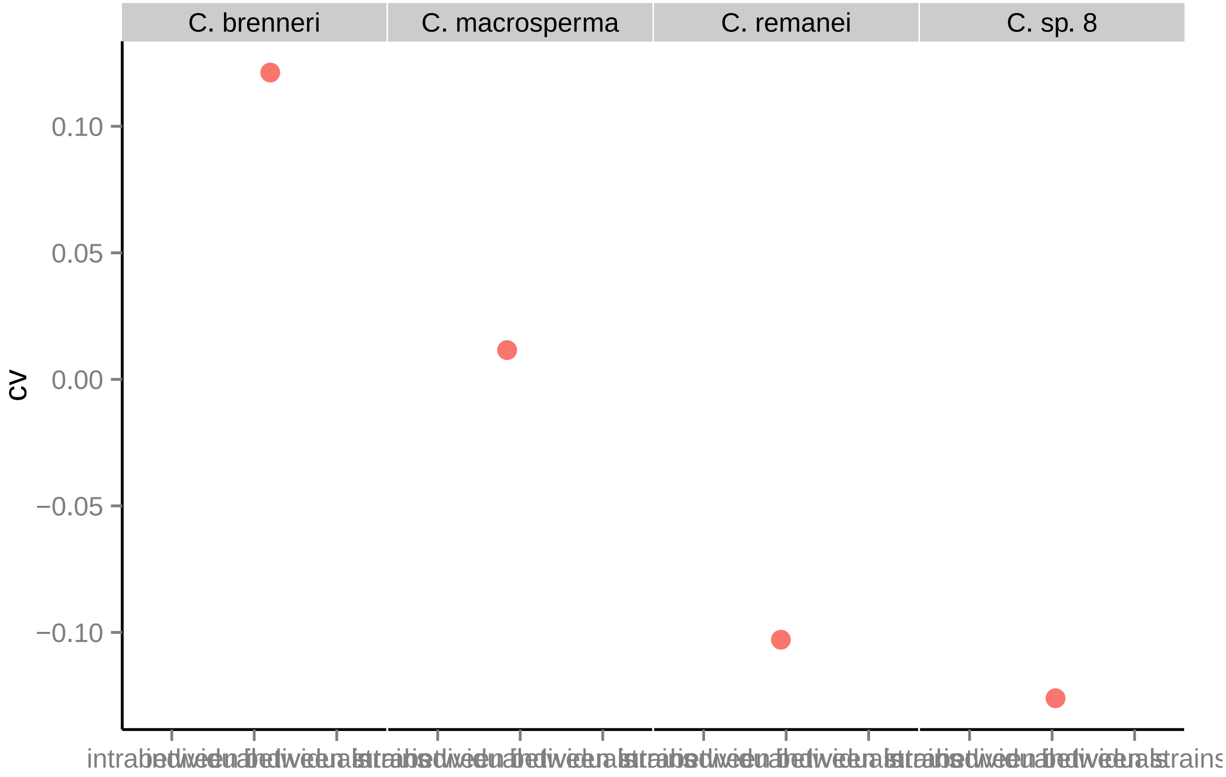

C. brenneri
C. macrosperma
C. remanei
C. sp. 8
7.5
7.0
6.5
area (μm²)
CB5161
JU1857
PB4641
QX1182
Strain

figure 3B


C. brenneri
C. macrosperma
C. remanei
cv
0.09
0.06
0.03
0.00
intra individual
between individuals
between strains
intra individual
between individuals
between strains
intra individual
between individuals
between strains


C. brenneri
C. macrosperma
C. remanei
C. sp. 8
cv
0.125
0.100
0.075
0.050
0.025
intra individual
between individuals
between strains
intra individual
between individuals
between strains
intra individual
between individuals
between strains
intra individual
between individuals
between strains


C. brenneri
C. macrosperma
C. remanei
C. sp. 8
cv
0.10
0.05
0.00
−0.05
−0.10

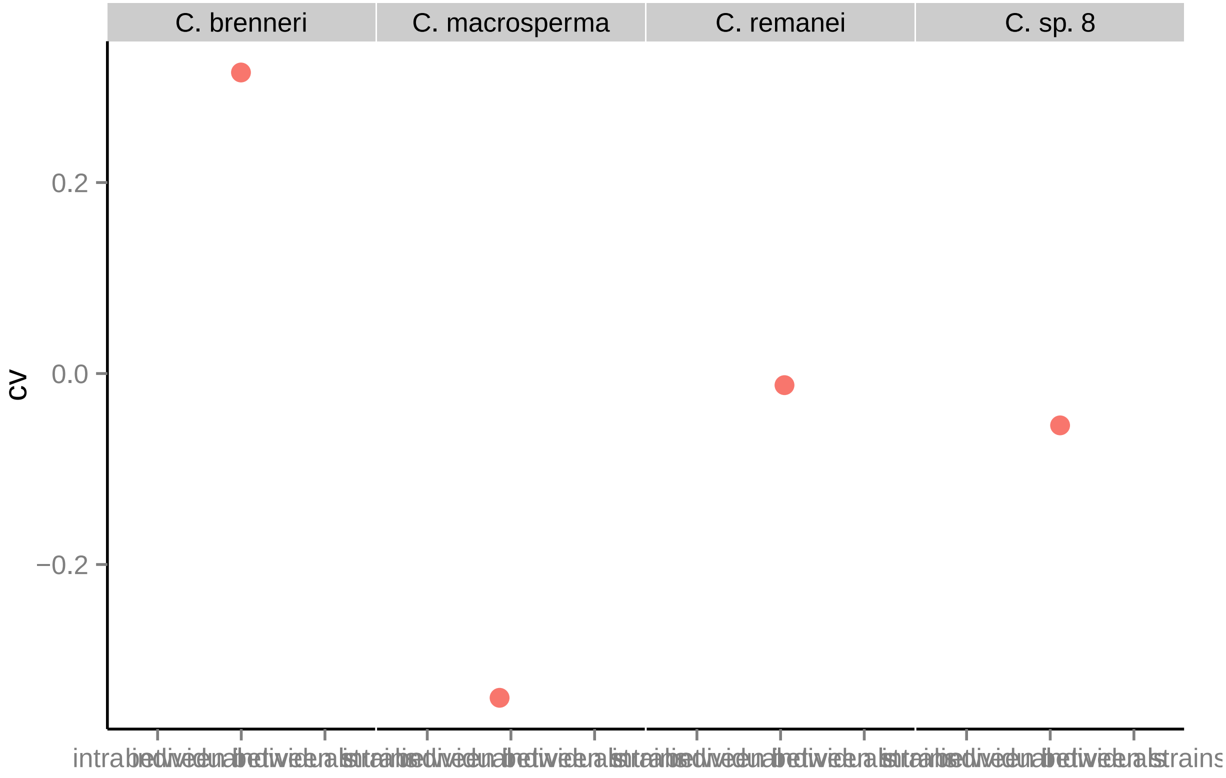


figure mockup a we are starting agin here, as I've modified some of the variables etc above Easier to start with a clean slate Clean and read in data:

mean and sd

```
groupedmeans=ff %>% group_by(Species,sex) %>% summarise(mean=mean(area,na.rm=T),n=n(),sd=sd(area,na.rm=T
limits <- aes(ymax = mean + 1.96*sd/sqrt(n), ymin=mean - 1.96*sd/sqrt(n))
dodge <- position_dodge(width=0.9)
ggplot(data=groupedmeans, aes(x=Species, y=mean, fill=sex)) + geom_bar(stat="identity", position=positio
    scale_fill_manual(values=c("gray","white"))+ylab( expression(paste("area (", mu, m^{2},")")))+xlab(
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
     panel.background = element_blank(),axis.line = element_line(colour = "black"),legend.position = "no
```

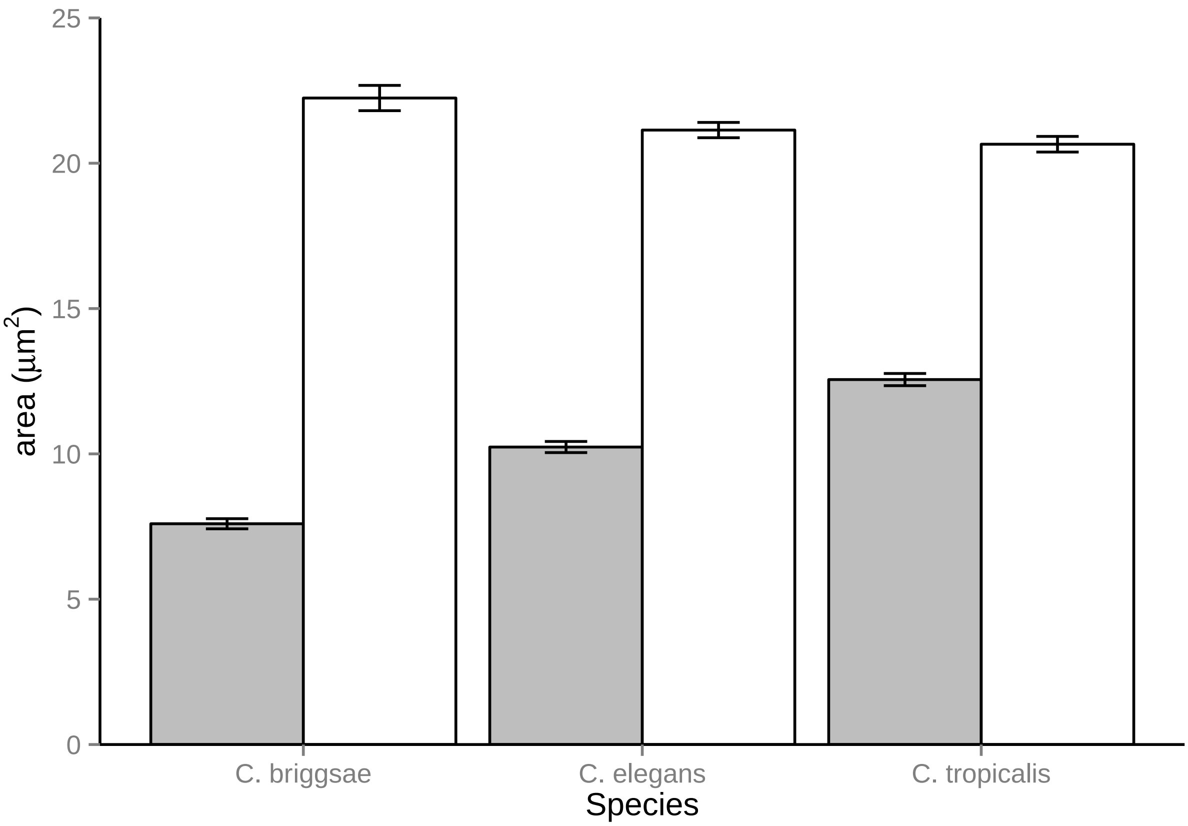


Figure mockup b group to get the data I want it's ugly, but it works

```
groupedstrains=ff %>% group_by(Species,Strain) %>%
    do(data.frame(meanM=mean(.[which(.$sex=="M"),]$area,na.rm=T),meanH=mean(.[which(.$sex=="H"),]$area,
                  sdM=sd(.[which(.$sex=="M"),]$area,na.rm=T),sdH=sd(.[which(.$sex=="H"),]$area,na.rm=T)
groupedstrains$sem=1.96*groupedstrains$sdM/sqrt(groupedstrains$nM)
groupedstrains$seh=1.96*groupedstrains$sdH/sqrt(groupedstrains$nH)
p=ggplot(data=groupedstrains, aes(x=meanM, y=meanH, colour=Species, ymin = meanH - seh,ymax=meanH + seh
p+geom_point(size=5)+coord_cartesian(ylim=c(0,15),xlim=c(0,33))+theme(panel.grid.major = element_blank(
     panel.background = element_blank(),axis.line = element_line(colour = "black"),legend.position = "n
     ylab(expression(paste("Hermaphrodite Sperm area (", mu, m^{2},")")))+xlab(expression(paste("Male S
     geom_smooth(aes(group=Species), method="lm",size=1.5,colour="black")
```

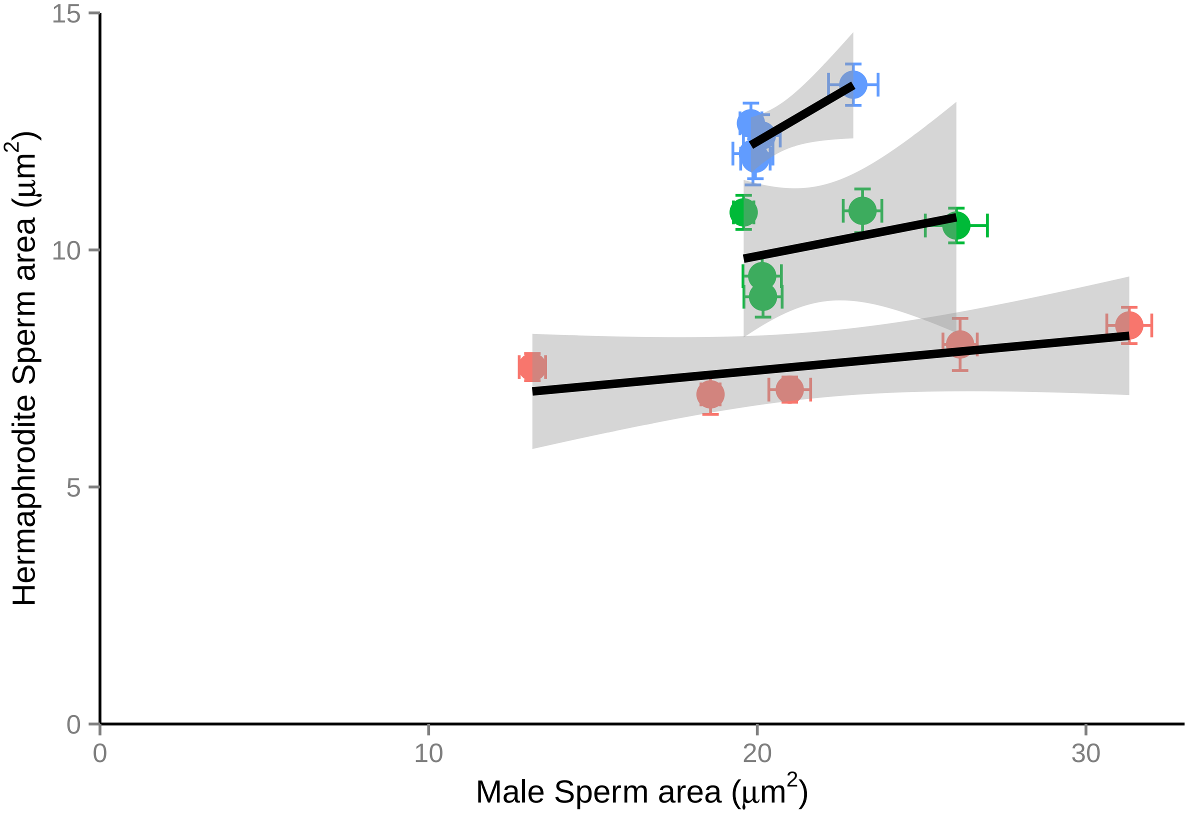


figure1a - bars need to rerun the above

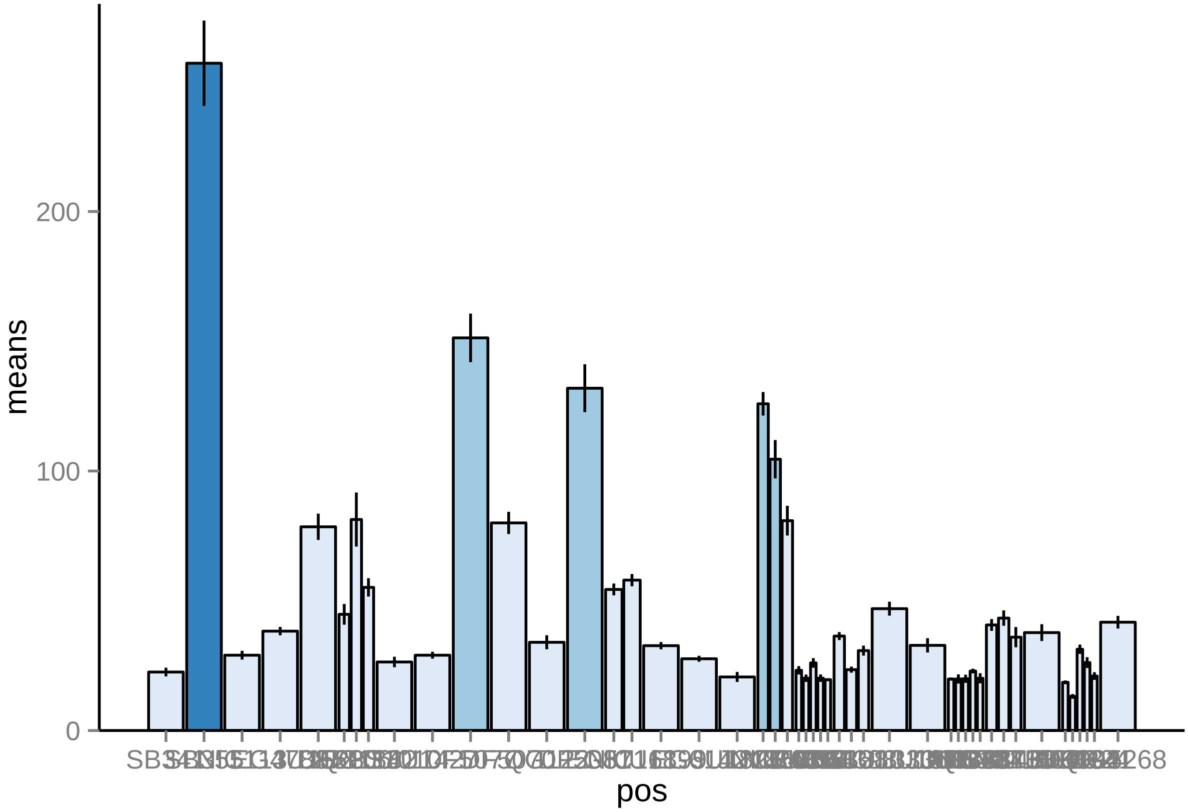
```
ff<-spermsizes<-read.table("C:/Users/jeremy/Desktop/geiger/compileddata.csv",header=T,sep=",")
ff<-renameworms(ff)
spermsize2<-ff %>% group_by(Species,Strain) %>% summarise(means=mean(area,na.rm=T),cv=sd(area,na.rm=T)/
countspec=function(dataframe){
    holding=c()
    for(i in 1:nrow(dataframe)){
        holding=c(holding,sum(dataframe$Species==dataframe[i,]$Species))
    }
    return(holding)
}
spermsize2$Species<-ordered(spermsize2$Species,levels=c(tr$tip.label))
spermsize2<-spermsize2[order(spermsize2$Species),]
spermsize2$Strain<-factor(spermsize2$Strain, levels = spermsize2$Strain)
spermsize2$specnum=countspec(spermsize2)
spermsize2$cols=cut(spermsize2$means,3)
findchange<-function(df){
    holding=c()
    for(i in 1:length(df$Species)){
        if(i==1){
            holding=c(holding,0)
        } else if(df$Species[i]==df$Species[i-1]){
            holding=c(holding,0)
        } else {
            holding=c(holding,1)
        }
    }
    return(holding)
}

spermsize2$change<-findchange(spermsize2)
w<-(1-(spermsize2$specnum-1)*0.05)/spermsize2$specnum
pos <- 0.5 * (cumsum(w) + cumsum(c(0, w[-length(w)])))

gapsizes<-function(df,pos){
    pos1<-pos
    for(i in 1:length(df$specnum)){
        if(df$specnum[i]==1){
            pos1[i:length(pos1)]<-pos1[i:length(pos1)]+0.1
        } else if(df$change[i]==1){
            pos1[i:length(pos1)]<-pos1[i:length(pos1)]+0.1
        } else {
            pos1[i:length(pos1)]<-pos1[i:length(pos1)]+0.05
        }
    }
    return(pos1)
}

pos<-gapsizes(spermsize2,pos)
ggplot(data=spermsize2,aes(x = pos, width = w, y = means, fill=cols,ymin=means-sem,ymax=means+sem)) + sc
    geom_bar(stat = "identity",colour="black") + scale_x_continuous(labels = spermsize2$Strain, breaks =
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
     panel.background = element_blank(),axis.line = element_line(colour = "black"),legend.position = "no
```

```
## Warning: position_stack requires constant width: output may be incorrect
```



large tree first group all the data

```
spermsize1<-ff %>% group_by(Species) %>% summarise(means=mean(area,na.rm=T))
spermsize2<-ff %>% group_by(Species,Strain) %>% summarise(means=mean(area,na.rm=T))
spermsize3<-ff %>% group_by(Species,Strain,individual) %>% summarise(means=mean(area,na.rm=T))
```

function to fix the format of spermsizes, and fit tree

```
fitace <- function(spermsizes,tree){
  holding<-spermsizes$means
  names(holding)<-spermsizes$Species
  return(fastAnc(tree,holding,CI=T))
}
```

function to choose a subset - works on any level, just chooses one from each species

```
subsetter<-function(spermsizes){
  x=lapply(split(spermsizes, spermsizes$Species),function(subdf) subdf[sample(1:nrow(subdf), 1),])
  do.call("rbind",x)
}
```

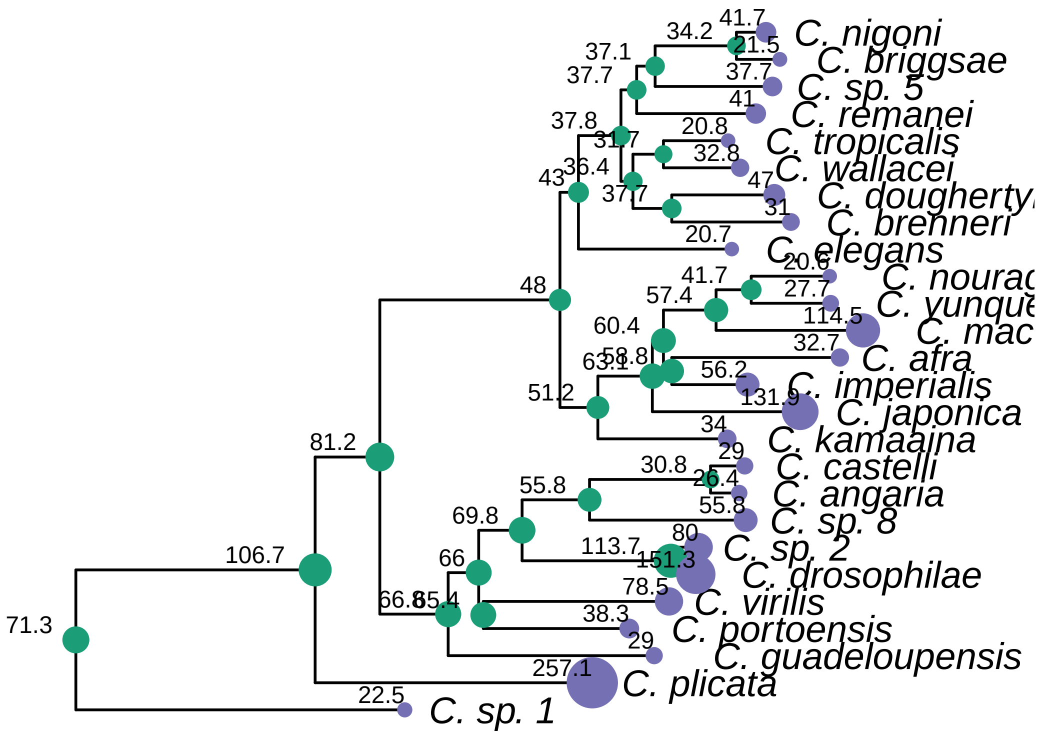
example usage fitace(subsetter(spermsizes2),tr) now need to loop and keep the values

```
looper<-function(spermsizes,reps,tree){
  outputdata=data.frame()
  outputCI=data.frame()
  for(i in 1:reps){
    outputdata=rbind(outputdata,fitace(subsetter(spermsizes),tree)$ace)
    outputCI=rbind(outputCI,fitace(subsetter(spermsizes),tree)$CI95)
  }
  names(outputdata)=seq(from=27,to=51)
  return(c(outputdata,outputCI))
}
```

ok we can call eg x=looper(spermsize3,10,tr) then mean and sd, etc. etc.

```
x<-fitace(spermsize1,tr)
AncSperm=x$ace
#or AncSperm=apply(x,2,mean) if it's bootstrapped
spermsize1$Species <- factor(spermsize1$Species, levels = c("C. sp. 1","C. plicata","C. guadeloupensis"
spermsize=spermsize1[order(spermsize1$Species),]$means
names(spermsize)<-spermsize1[order(spermsize1$Species),]$Species
ggtree(tr)+geom_text(aes(label=label),hjust=-0.19,fontface="italic")+geom_point(subset=.(!isTip),
           color="#1b9e77", size=sqrt(AncSperm/pi))+
           geom_point(subset=.(isTip),color="#7570b3", size=sqrt(spermsize/pi))+
           geom_text(subset=.(isTip),aes(label=round(spermsize,1)), hjust=1, vjust=-0.4, size=3)+
           geom_text(subset=.(!isTip),aes(label=round(AncSperm,1)), hjust=1.5, vjust=-0.4, size=3)+
           scale_x_continuous(expand = c(0.1, 0.1))
```

```
## Warning: Removed 25 rows containing missing values (geom_text).
```


41.7
C. nigoni
34.2
21.5
C. briggsae
37.1
37.7
37.7
C. sp. 5
41
C. remanei
37.8
20.8
C. tropicalis
31.7
32.8
C. wallacei
36.4
43
47
C. dougherty
37.7
31
C. brenneri
20.7
C. elegans
20.6
C. nourag
41.7
27.7
C. yunque
57.4
114.5
C. mac
60.4
32.7
C. afra
63.1
58.8
56.2
C. imperialis
51.2
131.9
C. japonica
34
C. kamaaina
29
C. castelli
30.8
26.4
C. angaria
55.8
55.8
C. sp. 8
69.8
80
C. sp. 2
113.7
151.3
C. drosophilae
66
78.5
C. virilis
66.8
65.4
38.3
C. portoensis
29
C. guadeloupensis
257.1
C. plicata
22.5
C. sp. 1
48
81.2
106.7
71.3