

Will climate change cause temperature stress?

Curtis A. Provencher

2016-March-29

Contents

Rationale	1
Experimental protocol	1
loading libraries	1
metadata and loading in dataset	2
paste in your metadata	2
Calculating gene expression	3
Stats!!	4
response:	4
predictor	4
normfinder, findign the best HKG	11

Rationale

Experimental protocol

loading libraries

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:plyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
##
## The following objects are masked from 'package:stats':
```

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

```
library(ggplot2)
library(tidyr)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##   select
```

metadata and loading in dataset

paste in your metadata

```
#warm<-read.csv("2015_warming_prelim.csv") #reading in the data set
warm<-read.csv("../Data//20160211_DF_Aph_Cham_LNSampling.csv")
str(warm) # visualizing the properties of the dataset
```

```
## 'data.frame':   260 obs. of  22 variables:
##  $ N                  : int   1 2 3 4 5 6 7 8 9 10 ...
##  $ Collection.Date     : int  20130702 20130702 20130702 20130702 20130702 20130702 20130702 20130702 20130702 ...
##  $ Site                : Factor w/ 2 levels "DF","HF": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Vial.Name           : Factor w/ 170 levels "", "DF 1.1", "DF 1.2",...: 54 53 52 79 80 78 83 82 ...
##  $ Cham                : int   1 1 1 2 2 2 3 3 3 4 ...
##  $ Sample              : int   1 2 3 1 2 3 1 2 3 1 ...
##  $ Window              : Factor w/ 6 levels "", "A", "B", "C",...: 4 3 3 3 4 2 3 2 4 3 ...
##  $ BaitTemp1           : num   24.2 24.8 25.2 23 23.8 22.4 23 23 22.8 22.4 ...
##  $ BaitTemp2           : num   24.2 24.6 25.2 23.2 23.8 23.6 23 22.8 22.8 22.4 ...
##  $ BaitTemp3           : num   24.4 24.6 25.2 23.2 23.6 23.6 23.6 23 22.8 22.6 ...
##  $ BaitTemp4           : num   24.4 24.4 25 23.4 23.6 23.6 23.2 23.8 22.4 22.6 ...
##  $ RNA.conc.           : Factor w/ 139 levels "", "<2", "10", "10.4",...: 118 48 14 83 136 122 19 1 ...
##  $ Isolation.Date      : int   20150811 20150813 20150731 20150814 20150813 20150813 20150730 20150813 ...
##  $ CT_18s              : num   NA NA 18.5 13.1 NA ...
##  $ CT_40               : num   NA NA 32.1 32.1 NA ...
##  $ CT_70               : num   NA NA 26.1 28.4 NA ...
##  $ CT_83               : num   NA NA 31.3 31.1 NA ...
##  $ RIN_Value           : num   2.1 2.2 3.7 2.8 2.4 7 2.6 2.1 2.4 2.7 ...
##  $ CDNA                 : num   2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 ...
##  $ dilution.factor     : Factor w/ 13 levels "#DIV/0!", "10",...: 2 2 2 2 2 2 2 2 2 2 ...
##  $ vol.cDNA.for.dilution : Factor w/ 13 levels "#DIV/0!", "10.86956522",...: 4 4 4 4 4 4 4 4 4 4 ...
##  $ vol.of.water.for.dilution: Factor w/ 13 levels "#DIV/0!", "37.5",...: 13 13 13 13 13 13 13 13 13 13 ...
```

```
dim(warm) # looking at the dimensions...rows,columns
```

```
## [1] 260 22
```

```
#calculating # of samples per site per chamber
```

```
knitr::kable(ddply(warm,.(Site,as.factor(Cham)),summarize,num=length(N)))
```

Site	as.factor(Cham)	num
DF	1	12
DF	2	12
DF	3	12
DF	4	12
DF	5	12
DF	6	12
DF	7	9
DF	8	10
DF	9	11
DF	10	9
DF	11	12
DF	12	9
DF	13	12
DF	14	9
DF	15	9
HF	1	9
HF	2	9
HF	3	8
HF	4	7
HF	5	9
HF	6	9
HF	7	6
HF	8	8
HF	9	8
HF	10	9
HF	11	7
HF	12	9

Calculating gene expression

GOI= gene of interest

HKG= house keeping gene

$$\frac{2^{(GOI_{mean}-GOI_{sample})}}{2^{(HKG_{mean}-HKG_{sample})}}$$

```
#calculating the mean bait_temp
```

```
warm$mean.temps<-rowMeans(warm[,8:11],na.rm=T)
```

```
gene.means<-apply(na.omit(warm[,14:17]),2,mean) #taking the means of each gene
gene.means
```

```
## CT_18s CT_40 CT_70 CT_83
## 12.24230 29.47033 24.78352 27.32496
```

```
deltaCT<-as.data.frame(t(apply(warm[,14:17],1,function(x){gene.means-x}))) # subtracting the means of e

dd<-as.data.frame(t(apply(deltaCT,1,function(x){2^x})))# raising everything to 2 power
names(dd)<-c('18s',"hsp40","hsp70","hsp83")
#Calculating fold change!
merg<-as.data.frame(cbind(warm,dd[,2:4]/dd[,1]))
```

Stats!!

response:

predictor

```
#changing collection date into a facto
merg$Collection.Date<-as.factor(as.character(merg$Collection.Date))

#making plots
merg$color<-ifelse(merg$Site=="DF","red","blue")# making color
merg2<-subset(merg,merg$hsp83<100)
#Doing stats
#new<-na.exclude(merg2)
#new1<-subset(merg2,merg2$hsp83 != "NA")
new<-subset(merg2,merg2$hsp70 != "NA")
new2<-subset(merg2,merg2$hsp40 != "NA");new2<-new2[,-37,]

#mod1<-stepAIC(lm(log10(hsp70)~Site*mean.temps+RIN_Value,data=new),direction="backward")
#summary(mod1)

mod1.1<-lm(log10(hsp70)~Site*mean.temps+RIN_Value,data=new)
summary(mod1.1)
#write.csv(new,"test.csv")

#model for hsp40
mod2<-stepAIC(lm(log10(hsp40)~Site*mean.temps+RIN_Value,data=new2),direction="backward");summary(mod2)

mod2.1<-lm(log10(hsp40)~Site*mean.temps+RIN_Value,data=new2)
summary(mod2.1)

#hsp83
mod3<-stepAIC(lm(log10(hsp83)~Site*mean.temps+RIN_Value,data=merg2),direction="backward");summary(mod3)
boxplot(log10(hsp83)~Site,data=merg2)
mod3.1<-lm(log10(hsp83)~Site*mean.temps+RIN_Value,data=merg2)
summary(mod3.1)
```

```

par(mfrow=c(2,2))
plot(mod3)
par(mfrow=c(1,1))
#hsp83 plot
plot(merg2$mean.temps,log10(merg2$hsp83),col=merg2$color,pch=16,ylab="Log10 Hsp83 relative expression",
axis(1,lwd=2);box(lwd=2);axis(2,lwd=2,las=1)
#text(merg2$mean.temps,log10(merg2$hsp83),labels=merg2$Vial.Name)
abline(lm(log10(hsp83)~mean.temps,data=subset(merg2,merg2$Site=="DF")),lty="dotdash",col="red",lwd=3)
abline(lm(log10(hsp83)~mean.temps,data=subset(merg2,merg2$Site=="HF")),lty="dotdash",col="blue",lwd=3)

#hsp70 plot
plot(new$mean.temps,log10(new$hsp70),col=new$color,pch=16,ylab="Log10 Hsp70 relative expression",xlab="1
axis(1,lwd=2);box(lwd=2);axis(2,lwd=2,las=1)
#text(merg2$mean.temps,log10(merg2$hsp83),labels=merg2$Vial.Name)
abline(lm(log10(hsp70)~mean.temps,data=subset(new,new$Site=="DF")),lty="dotdash",col="red",lwd=3)
abline(lm(log10(hsp70)~mean.temps,data=subset(new,new$Site=="HF")),lty="dotdash",col="blue",lwd=3)

#hsp40
plot(new2$mean.temps,log10(new2$hsp40),col=merg2$color,pch=16,ylab="Log10 Hsp40 relative expression",xl
axis(1,lwd=2);box(lwd=2);axis(2,lwd=2,las=1)
#text(merg2$mean.temps,log10(merg2$hsp83),labels=merg2$Vial.Name)
abline(lm(log10(hsp40)~mean.temps,data=subset(new2,new2$Site=="DF")),lty="dotdash",col="red",lwd=3)
abline(lm(log10(hsp40)~mean.temps,data=subset(new2,new2$Site=="HF")),lty="dotdash",col="blue",lwd=3)

summary(aov(mean.temps~Site,data=subset(merg2,merg2$Year=="2013")))
ggplot(merg2,aes(y=mean.temps,x=Site))+geom_boxplot()

```

```

#using merg 2, convertin xp to 100/ct
names(merg2)

diff.way<-as.data.frame(cbind(merg2[,1:7],apply(merg2[,14:17],2,function(x){1000/x}),merg2[,19]))
names(diff.way)[12]<-"mean.temps"
sd(diff.way$CT_18s)/mean(diff.way$CT_18s)

#summary(lm(CT_83~Site*mean.temps+CT_18s+Year,data=diff.way))
#plot(diff.way$mean.temps,diff.way$CT_83)
#summary(lm(CT_70~Site*mean.temps+CT_18s+Year,data=diff.way))
#summary(lm(CT_40~Site*mean.temps+CT_18s+Year,data=diff.way))

#stepwise
library(MASS)
mod11<-stepAIC(lm(CT_83~Site*mean.temps+CT_18s+Year,data=diff.way),direction="backward")
summary(mod11)
#plot(mod11)

summary(stepAIC(lm(CT_70~Site*mean.temps+CT_18s+Year,data=diff.way),direction="backward"))

summary(stepAIC(lm(CT_40~Site*mean.temps+CT_18s+Year,data=diff.way),direction="backward"))

#18s rRNA

```

```

summary(stepAIC(lm(CT_18s~Site*mean.temps+Year,data=diff.way),direction="backward"))

#only analyzing 2013
diff.way2<-subset(diff.way,diff.way$Year=="2013")

#summary(lm(CT_83~Site*mean.temps+CT_18s,data=diff.way2))
#summary(lm(CT_70~Site*mean.temps+CT_18s,data=diff.way2))
#summary(lm(CT_40~Site*mean.temps+CT_18s,data=diff.way2))

#stepwise
mod22<-stepAIC(lm(CT_83~Site*mean.temps+CT_18s,data=diff.way2),direction="backward");summary(mod22)

mod23<-stepAIC(lm(CT_70~Site*mean.temps+CT_18s,data=diff.way2),direction="backward");summary(mod23)

mod24<-stepAIC(lm(CT_40~Site*mean.temps+CT_18s,data=diff.way2),direction="backward");summary(mod24)

#adjusting sizing
sizzly<- theme_bw()+
  theme(axis.title.y=element_text(size=rel(2)),axis.title.x=element_text(size=rel(3)))+
  theme(axis.text.y=element_text(size=rel(2)),axis.text.x=element_text(size=rel(2)),legend.position=c(.
  theme(legend.title=element_blank(),legend.text=element_text(size=28),panel.grid.major=element_blank())

diff.way$Year<-as.factor(as.character(diff.way$Year))

#18s rRNA
diff.way2$col<-ifelse(diff.way2$Site=="DF","red","blue")
plot(diff.way2$mean.temps,diff.way2$CT_18s,col=diff.way2$col,pch=16,ylab="18s rRNA gene expression (100
abline(lm(CT_18s~mean.temps,data=subset(diff.way2,diff.way2$Site=="DF")),lwd=5,lty=2,col="red")
abline(lm(CT_18s~mean.temps,data=subset(diff.way2,diff.way2$Site=="HF")),lwd=5,lty=6,col="blue")
points(diff.way2$mean.temps,diff.way2$CT_83,col=diff.way2$col,pch=17)
#text(diff.way2$mean.temps,diff.way2$CT_18s,labels=diff.way2$Vial.Name)

#hsp83 fig
ggplot(diff.way2,aes(x=mean.temps,y=CT_83,colour=Site,shape=Site))+geom_point(size=3)+scale_x_continuous

ggplot(diff.way,aes(x=mean.temps,y=CT_70,colour=Site))+geom_point(size=3)+scale_x_continuous("Local Temp

#hsp40
ggplot(diff.way2,aes(x=mean.temps,y=CT_40,colour=Site))+geom_point(size=3)+scale_x_continuous("Local Temp

Normfinder=function(filename,Groups=TRUE,ctVal=TRUE,pStabLim=0.25){
#
# If Groups is TRUE the last row contains the group identifier,
# and the last row must be started by a name for that row.
# No spaces are allowed in the gene names, sample names and group identifier.
#
dat0=read.table(filename,header=TRUE,row.names=1,colClasses="character")
#
ntotal=dim(dat0)[2] # number of samples
k0=dim(dat0)[1] # number of rows
#

```

```

if (Groups){
  ngenes=k0-1 # number of genes
  genenames=rownames(dat0)[-k0]
  grId=dat0[k0,]
  dat0=dat0[-k0,]
} else {
  ngenes=k0 # number of genes
  genenames=rownames(dat0)
  grId=rep(1,ntotal)
}
#
dat=matrix(as.numeric(unlist(dat0)),ngenes,ntotal) # matrix instead of list
#
if (!ctVal){dat=log2(dat)} # transform to log2 values
#
samplerNames=colnames(dat0)
grId=factor(unlist(grId)) # group identifier
groupNames=levels(grId) # group names
ngr=length(levels(grId)) # number of groups
# Number of samples in each group:
nsamples=rep(0,ngr)
for (group in 1:ngr){nsamples[group]=sum(grId==groupNames[group])}
#
#
MakeStab=function(da){
  ngenes=dim(da)[1]
  # Sample averages
  sampleAvg=apply(da,2,mean)
  # Gene averages within group
  geneGroupAvg=matrix(0,ngenes,ngr)
  for (group in 1:ngr){
    geneGroupAvg[,group]=apply(da[,grId==groupNames[group]],1,mean)}
  # Group averages
  groupAvg=rep(0,ngr)
  for (group in 1:ngr){groupAvg[group]=mean(da[,grId==groupNames[group]])}

  # Variances
  GGvar=matrix(0,ngenes,ngr)
  for (group in 1:ngr){
    grset=(grId==groupNames[group])
    a=rep(0,ngenes)
    for (gene in 1:ngenes){
      a[gene]=sum((da[gene,grset]-geneGroupAvg[gene,group]-
        sampleAvg[grset]+groupAvg[group])^2)/(nsamples[group]-1)
    }
    GGvar[,group]=(a-sum(a)/(ngenes*ngenes-ngenes))/(1-2/ngenes)
  }
#
# Change possible negative values
geneGroupMinvar=matrix(0,ngenes,ngr)
for (group in 1:ngr){
  grset=(grId==groupNames[group])
  z=da[,grset]

```

```

for (gene in 1:ngenes){
  varpair=rep(0,ngenes)
  for (gene1 in 1:ngenes){varpair[gene1]=var(z[gene,]-z[gene1,])}
  genegroupMinvar[gene,group]=min(varpair[-gene])/4
}
}
#
# Final variances
GGvar=ifelse(GGvar<0,genegroupMinvar,GGvar)
#
# Old stability measure for each gene is calculated:
#
dif=genegroupavg
difgeneavg=apply(dif,1,mean)
difgroupavg=apply(dif,2,mean)
difavg=mean(dif)
for (gene in 1:ngenes){
  for (group in 1:ngr){
    dif[gene,group]=dif[gene,group]-difgeneavg[gene]-difgroupavg[group]+difavg
  }
}
#
nsampMatrix=matrix(rep(nsamples,ngenes),ngenes,ngr,byrow=T)
vardif=GGvar/nsampMatrix
gamma=sum(dif*dif)/((ngr-1)*(ngenes-1))-sum(vardif)/(ngenes*ngr)
gamma=ifelse(gamma<0,0,gamma)
#
difnew=dif*gamma/(gamma+vardif)
varnew=vardif+gamma*vardif/(gamma+vardif)
Ostab0=abs(difnew)+sqrt(varnew)
Ostab=apply(Ostab0,1,mean)
#
# Measure of group differences:
mud=rep(0,ngenes)
for (gene in 1:ngenes){
  mud[gene]=2*max(abs(dif[gene,]))
}
# Common variance:
genevar=rep(0,ngenes)
for (gene in 1:ngenes){
  genevar[gene]=sum((nsamples-1)*GGvar[gene,])/(sum(nsamples)-ngr)
}
Gsd=sqrt(genevar)
#
# Return results:
#
return(cbind(mud,Gsd,Ostab,rep(gamma,ngenes),GGvar,dif))
} # End of function MakeStab
#
#
MakeComb2=function(g1,g2,res){
gam=res[1,4]
d1=res[g1,(4+ngr+1):(4+ngr+ngr)]; d2=res[g2,(4+ngr+1):(4+ngr+ngr)]

```



```

s1=res[g1,(4+1):(4+ngr)]^2; s2=res[g2,(4+1):(4+ngr)]
rho=abs(gam*d1/(gam+s1/nsamples)+gam*d2/(gam+s2/nsamples))*
  sqrt(ngenes/(ngenes-2))/2
rho=rho+sqrt(s1/nsamples+gam*s1/(nsamples*gam+s1)+
  s2/nsamples+gam*s2/(nsamples*gam+s2))/2
return(sum(rho)/2)
}
#
#
MakeStabOne=function(da){
  ngenes=dim(da)[1]
  # Sample averages
  sampleavg=apply(da,2,mean)
  # Gene averages
  geneavg=apply(da,1,mean)
  totalavg=mean(da)
  #
  # Variances
  genevar0=rep(0,ngenes)
  for (gene in 1:ngenes){
    genevar0[gene]=
      sum((dat[gene,]-geneavg[gene]-sampleavg+totalavg)^2)/
        ((ntotal-1)*(1-2/ngenes))
  }
  genevar=genevar0-sum(genevar0)/(ngenes*ngenes-ngenes)
  #
  # Change possible negative values
  geneMinvar=rep(0,ngenes)
  z=da
  for (gene in 1:ngenes){
    varpair=rep(0,ngenes)
    for (gene1 in 1:ngenes){varpair[gene1]=var(z[gene,]-z[gene1,])}
    geneMinvar[gene]=min(varpair[-gene])/4
  }
  # Final variances
  genevar=ifelse(genevar<0,geneMinvar,genevar)
  #
  return(genevar)
}
# End of function MakeStabOne
#
#####
#
# Main part
#
if (ngr>1){ # More than one group.
#
res=MakeStab(dat)
#
gcand=c(1:ngenes)[res[,3]<pStabLim]
ncand=length(gcand)
if (ncand<4){
  if (ngenes>3){

```

```

li=sort(res[,3])[4]
gcand=c(1:ngenes)[res[,3]<=li]
ncand=length(gcand)
} else {
gcand=c(1:ngenes)
ncand=length(gcand)
}
}
#
vv2=c()
#
for (g1 in 1:(ncand-1)){
  for (g2 in (g1+1):ncand){
    qmeas=MakeComb2(gcand[g1],gcand[g2],res)
    vv2=rbind(vv2,c(gcand[g1],gcand[g2],qmeas))
  }
}
#
ord=order(res[,3])
FinalRes=list(Ordered=
data.frame("GroupDif"=round(res[ord,1],2),"GroupSD"=round(res[ord,2],2),
"Stability"=round(res[ord,3],2),row.names=genenames[ord]),
UnOrdered=
data.frame("GroupDif"=round(res[,1],2),"GroupSD"=round(res[,2],2),
"Stability"=round(res[,3],2),
"IGroupSD"=round(sqrt(res[, (4+1):(4+ngr)]),2),
"IGroupDif"=round(res[, (4+ngr+1):(4+ngr+ngr)],2),
row.names=genenames),
PairOfGenes=
data.frame("Gene1"=genenames[vv2[,1]],"Gene2"=genenames[vv2[,2]],
"Stability"=round(vv2[,3],2)))
#
return(FinalRes)
#
} else {    # End of more than one group: next is for one group only.
#
#
sigma=sqrt(MakeStabOne(dat))
#
siglim=(min(sigma)+0.1)
gcand=c(1:ngenes)[sigma<siglim]
ncand=length(gcand)
#
if ((ncand>=2)&(ngenes>3)){
#
vv2=c()
#
for (g1 in 1:(ncand-1)){
  for (g2 in (g1+1):ncand){
    dat1=rbind(dat[-c(gcand[g1],gcand[g2]),],
    apply(dat[c(gcand[g1],gcand[g2]),],2,mean))
    qmeas=sqrt(MakeStabOne(dat1))
    vv2=rbind(vv2,c(gcand[g1],gcand[g2],qmeas[ngenes-1]))
  }
}
}

```

```

ord=order(sigma)
FinalRes=list(Ordered=
data.frame("GroupSD"=round(sigma[ord],2),row.names=genenames[ord]),
PairOfGenes=
data.frame("Gene1"=genenames[vv2[,1]],"Gene2"=genenames[vv2[,2]],
"GroupSD"=round(vv2[,3],2)))
} else { # No combined genes to consider
ord=order(sigma)
FinalRes=list(Ordered=
data.frame("GroupSD"=round(sigma[ord],2),row.names=genenames[ord]))
} # End ncand<2 or ngenes<=3
#
return(FinalRes)
#
} # End one group only
#
} # End of main function

```

normfinder, findign the best HKG

```

b<-read.csv("../Data/test.csv")
bb<-subset(b,b$CT_18s>1 & b$CT_actin > 1 & b$CT_gapdh > 1)
dim(bb)

bn<-as.data.frame(t(subset(bb,select=c("Site","CT_18s","CT_actin","CT_gapdh"))))
#write.csv(bn,"normfinder_prelim.csv")

aph=Normfinder("20160211_prelim_normfinder.txt")
#single gene stability
aph$Ordered
#pairs of genes
aph$PairOfGenes

sessionInfo()

```

```

## R version 3.2.3 (2015-12-10)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.11.1 (El Capitan)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] MASS_7.3-45  tidyr_0.4.1  ggplot2_2.0.0 dplyr_0.4.3  plyr_1.8.3
##
## loaded via a namespace (and not attached):

```

```
## [1] Rcpp_0.12.3      digest_0.6.8      assertthat_0.1    grid_3.2.3
## [5] R6_2.1.2          gtable_0.1.2      DBI_0.3.1          formatR_1.2
## [9] magrittr_1.5       scales_0.3.0      evaluate_0.7.2     highr_0.5
## [13] stringi_1.0-1      lazyeval_0.1.10   rmarkdown_0.7      tools_3.2.3
## [17] stringr_1.0.0      munsell_0.4.2     yaml_2.1.13        parallel_3.2.3
## [21] colorspace_1.2-6   htmltools_0.2.6   knitr_1.10.5
```