Transcriptome Demo

Lily

2024-05-09

Load required packages (you might have to figure out how to install some of these first. . .)

```
library(ballgown)
library(RColorBrewer)
library(genefilter)
library(dplyr)
library(devtools)
```

Combines the identities and marks files for commit

```
pheno_data<-data.frame(ids = c("plank01", "plank02", "biofilm01", "biofilm02"),
                       stage = c("planktonic", "planktonic", "biofilm", "biofilm"))
```

create Ballgown object and check transcript number-5744

```
samples.c <- paste('ballgown', pheno_data$ids, sep = '/')
bg <- ballgown(samples = samples.c, meas='all', pDdata = pheno_data)
bg
```

```
## ballgown instance with 5744 transcripts and 4 samples
```

<what is this code doing?> Ballgown Filtering of low abundance genes and removes transcripts with a variance across samples less than 1

```
bg_filt = subset(bg,"rowVars(texpr(bg)) >1",genomesubset=TRUE)
bg_filt
```

```
## ballgown instance with 5162 transcripts and 4 samples
```

create a table of transcripts

```
results_transcripts<- stattest(bg_filt, feature = "transcript", covariate = "stage",
getFC = TRUE, meas = "FPKM")
```

```
results_transcripts<-data.frame(geneNames=geneNames(bg_filt),
transcriptNames=transcriptNames(bg_filt), results_transcripts)
```

choose a transcript to examine more closely (this is a demo, you need to choose another)

```
results_transcripts[results_transcripts$transcriptNames == "gene-PA0100", ]
```

```
##     geneNames transcriptNames    feature  id          fc      pval      qval
## 104         .     gene-PA0100 transcript 104 0.007529325 0.1323507 0.9369526
```

what information are you given about this transcript?

#geneName 104, transcriptNames gene-PA0100, feature id, fc 0.00752, pval 0.1323, qval 0.9369

#computes the significance of pairwise differences relative to the mean and variance for the results_transcripts file and it's filtered by pvalues greater than 0.05. The dim function either sets or returns the dimension of the matrix.

```
sigdiff <- results_transcripts %>% filter(pval<0.05)
dim(sigdiff)
```

```
## [1] 213   7
```

organize the table <by what metrics is the table being organized?>

Metrics: geneNames, transcripNames, id, fc, pval, and qval

```
o = order(sigdiff[,"pval"], -abs(sigdiff[,"fc"]), decreasing=FALSE)
output = sigdiff[o,c("geneNames","transcriptNames", "id","fc","pval","qval")]
write.table(output, file="SigDiff.txt", sep="\t", row.names=FALSE, quote=FALSE)
head(output)
```

```
##      geneNames transcriptNames   id           fc         pval      qval
## 2297      lpdV     gene-PA2250 2297 2.044601e-12 0.0002059451 0.9369526
## 2044         .    MSTRG.1712.1 2044 1.336284e-06 0.0007753696 0.9369526
## 5242         .     gene-PA5079 5242 2.634909e+01 0.0008329425 0.9369526
## 1801         .     gene-PA1764 1801 1.622252e-02 0.0010500021 0.9369526
## 3254      ubiG     gene-PA3171 3254 5.053753e+02 0.0014628478 0.9369526
## 4560         .     gene-PA4434 4560 3.374005e+03 0.0014706949 0.9369526
```

load gene names

```
bg_table = texpr(bg_filt, 'all')
bg_gene_names = unique(bg_table[, 9:10])
```

pull out gene expression data and visualize

```
gene_expression = as.data.frame(gexpr(bg_filt))
head(gene_expression)
```

```
##            FPKM.plank01 FPKM.plank02 FPKM.biofilm01 FPKM.biofilm02
## MSTRG.1       405.87982    400.83899      232.31441      181.92555
## MSTRG.10       89.64629     78.57229       35.00898       59.75500
## MSTRG.100     116.43972    106.20566       92.20284       95.31878
## MSTRG.1000     56.71363     84.85225       33.05915       20.13864
## MSTRG.1001     17.20822     21.51570       13.53020       12.65041
## MSTRG.1002   2050.12817   3189.20166     2180.10010     2007.27734
```

<what is this code doing? hint:compare the above output of head(gene_expression) to this output>

```
colnames(gene_expression) <- c("plank01", "plank02", "biofilm01", "biofilm02")
head(gene_expression)
```

```
##               plank01    plank02  biofilm01  biofilm02
## MSTRG.1     405.87982  400.83899  232.31441  181.92555
## MSTRG.10     89.64629   78.57229   35.00898   59.75500
## MSTRG.100   116.43972  106.20566   92.20284   95.31878
## MSTRG.1000   56.71363   84.85225   33.05915   20.13864
## MSTRG.1001   17.20822   21.51570   13.53020   12.65041
## MSTRG.1002 2050.12817 3189.20166 2180.10010 2007.27734
```

```
dim(gene_expression)
```

```
## [1] 4592    4
```

load the transcript to gene table and determine the number of transcripts and unique genes

```
transcript_gene_table = indexes(bg)$t2g
head(transcript_gene_table)
```

```
##   t_id    g_id
## 1    1 MSTRG.1
## 2    2 MSTRG.2
## 3    3 MSTRG.3
## 4    4 MSTRG.3
## 5    5 MSTRG.4
## 6    6 MSTRG.5
```

```
length(row.names(transcript_gene_table))
```

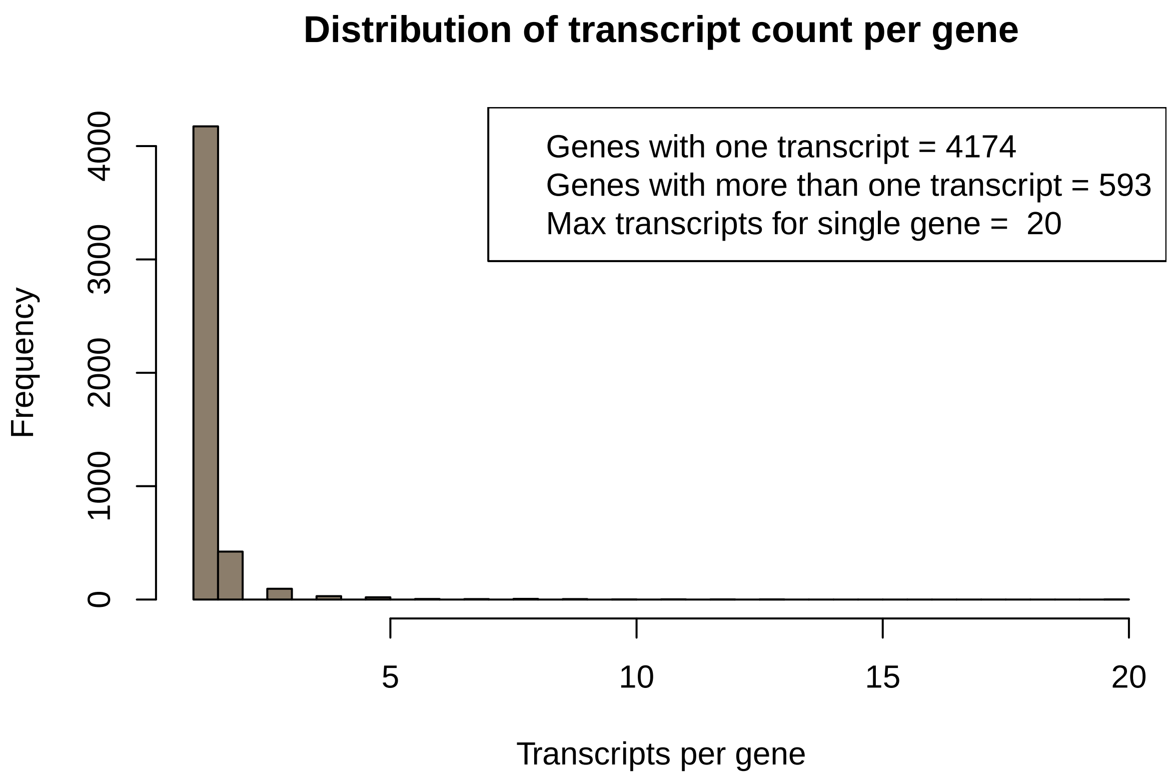
```
## [1] 5744
```

```
length(unique(transcript_gene_table[,"g_id"]))
```

```
## [1] 4767
```

plot the number of transcripts per gene

```
counts=table(transcript_gene_table[,"g_id"])
c_one = length(which(counts == 1))
c_more_than_one = length(which(counts > 1))
c_max = max(counts)
hist(counts, breaks=50, col="bisque4", xlab="Transcripts per gene",
main="Distribution of transcript count per gene")
legend_text = c(paste("Genes with one transcript =", c_one),
paste("Genes with more than one transcript =", c_more_than_one),
paste("Max transcripts for single gene = ", c_max))
legend("topright", legend_text, lty=NULL)
```

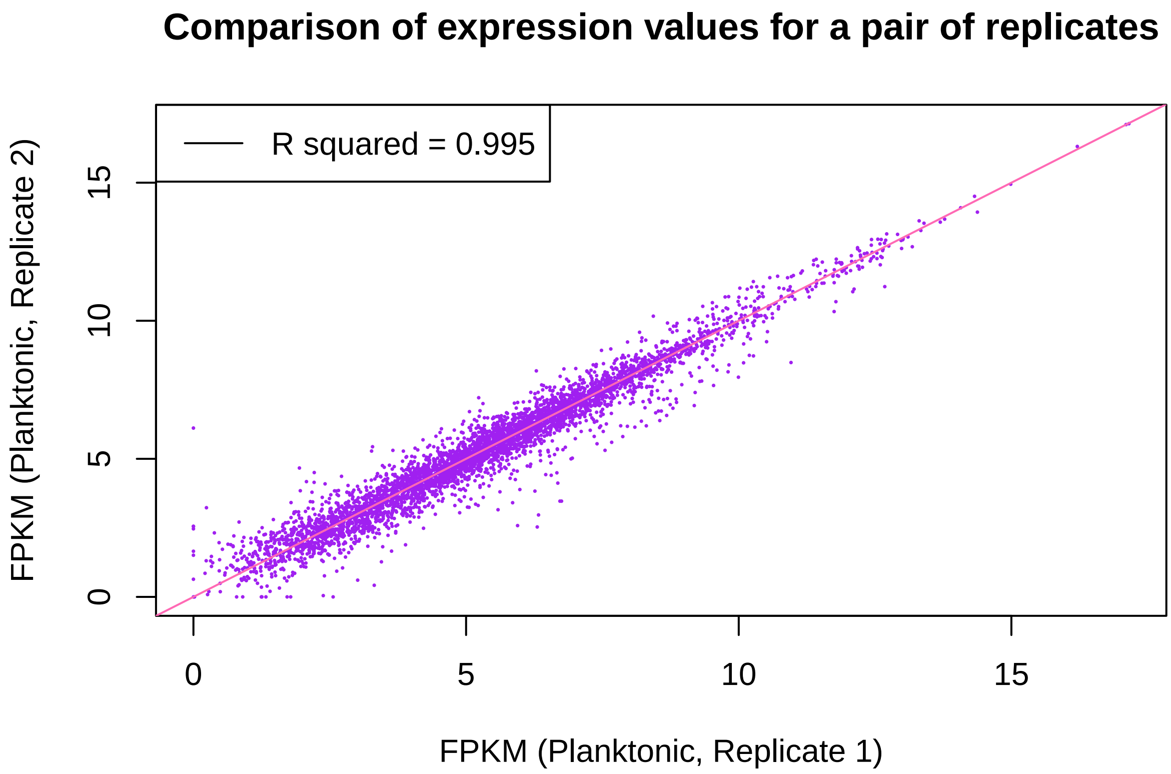


# there are more genes with one transcript about 4174 than the genes with more than one transcript about 593 ##

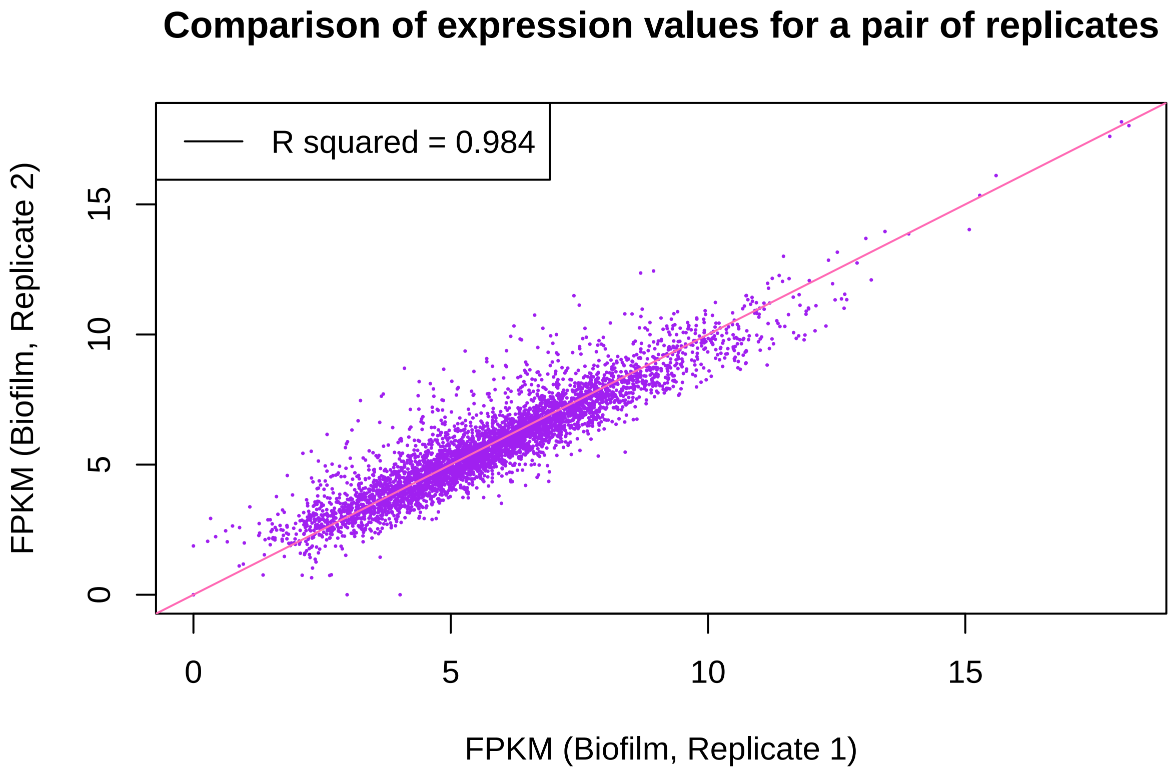
create a plot of how similar the two replicates are for one another. We have two data sets…how can you modify this code in another chunk to create a plot of the other set?

```
x = gene_expression[,"plank01"]
y = gene_expression[,"plank02"]
```

```
min_nonzero=1
plot(x=log2(x+min_nonzero), y=log2(y+min_nonzero), pch=16, col="purple", cex=0.25,
xlab="FPKM (Planktonic, Replicate 1)", ylab="FPKM (Planktonic, Replicate 2)",
main="Comparison of expression values for a pair of replicates")
abline(a=0,b=1, col = "hotpink")
rs=cor(x,y)^2
legend("topleft", paste("R squared = ", round(rs, digits=3), sep=""), lwd=1, col="black")
```



```
x = gene_expression[,"biofilm01"]
y = gene_expression[,"biofilm02"]
min_nonzero=1
plot(x=log2(x+min_nonzero), y=log2(y+min_nonzero), pch=16, col="purple", cex=0.25,
xlab="FPKM (Biofilm, Replicate 1)", ylab="FPKM (Biofilm, Replicate 2)",
main="Comparison of expression values for a pair of replicates")
abline(a=0,b=1, col = "hotpink")
rs=cor(x,y)^2
legend("topleft", paste("R squared = ", round(rs, digits=3), sep=""), lwd=1, col="black")
```

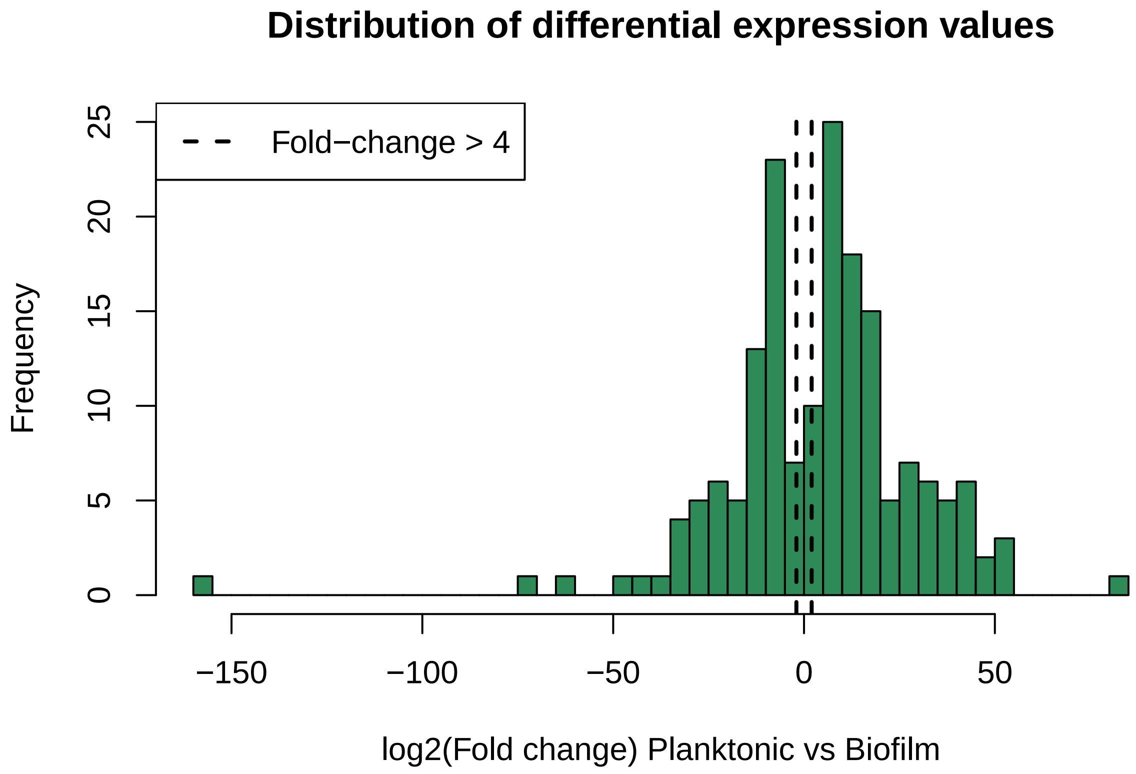


What does it mean if the two data sets are similar?

that the expression values are similar

create plot of differential gene expression between the conditions

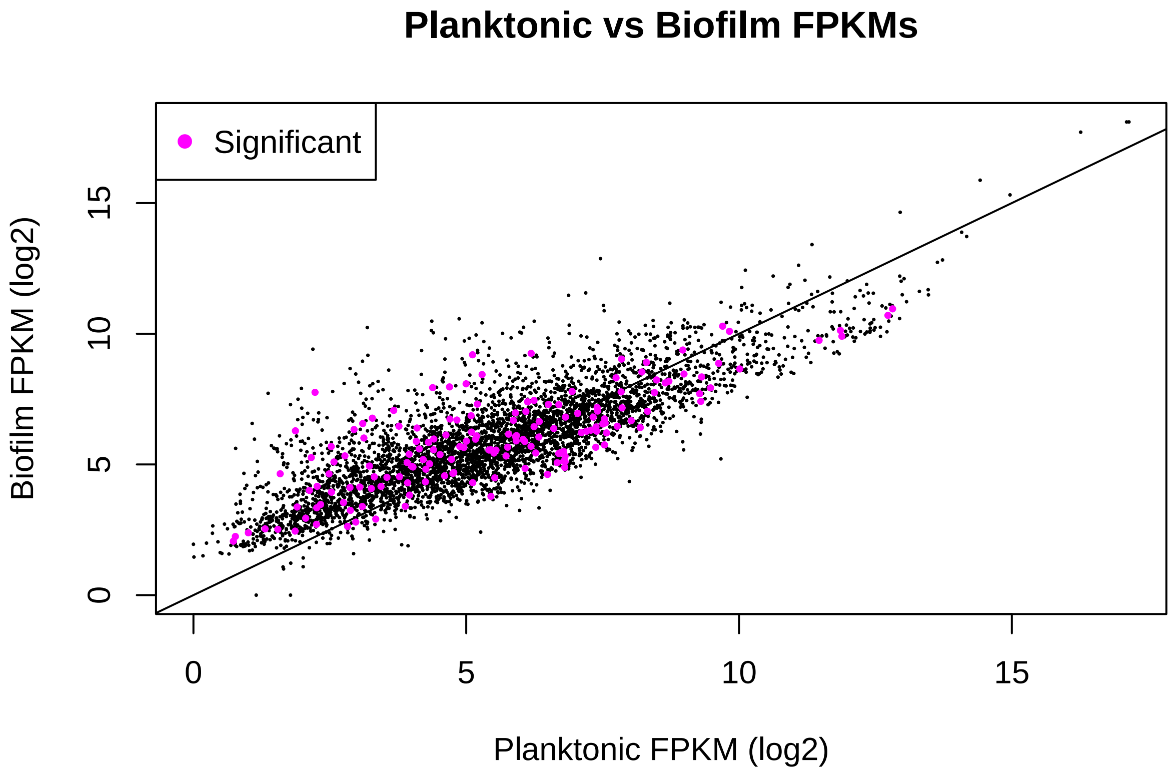
```
results_genes = stattest(bg_filt, feature="gene", covariate="stage", getFC=TRUE, meas="FPKM")
results_genes = merge(results_genes,bg_gene_names,by.x=c("id"),by.y=c("gene_id"))
sig=which(results_genes$pval<0.05)
results_genes[,"de"] = log2(results_genes[,"fc"])
hist(results_genes[sig,"de"], breaks=50, col="seagreen",
xlab="log2(Fold change) Planktonic vs Biofilm",
main="Distribution of differential expression values")
abline(v=-2, col="black", lwd=2, lty=2)
abline(v=2, col="black", lwd=2, lty=2)
legend("topleft", "Fold-change > 4", lwd=2, lty=2)
```



interpret the above figure:

Plot total gene expression highlighting differentially expressed genes

```
gene_expression[,"plank"]=apply(gene_expression[,c(1:2)], 1, mean)
gene_expression[,"biofilm"]=apply(gene_expression[,c(3:4)], 1, mean)
x=log2(gene_expression[,"plank"]+min_nonzero)
y=log2(gene_expression[,"biofilm"]+min_nonzero)
plot(x=x, y=y, pch=16, cex=0.25, xlab="Planktonic FPKM (log2)", ylab="Biofilm FPKM (log2)",
main="Planktonic vs Biofilm FPKMs")
abline(a=0, b=1)
xsig=x[sig]
ysig=y[sig]
points(x=xsig, y=ysig, col="magenta", pch=16, cex=0.5)
legend("topleft", "Significant", col="magenta", pch=16)
```



make a table of FPKM values

```
fpkm = texpr(bg_filt,meas="FPKM")
```

choose a gene to determine individual expression (pick a different number than I did)

```
ballgown::transcriptNames(bg_filt)[8]
```

```
##             8
## "gene-PA0008"
```

```
ballgown::geneNames(bg_filt)[8]
```

```
##      8
## "glyS"
```

transform to log2

```
transformed_fpkm <- log2(fpkm[2, ] + 1)
```

make sure values are properly coded as numbers

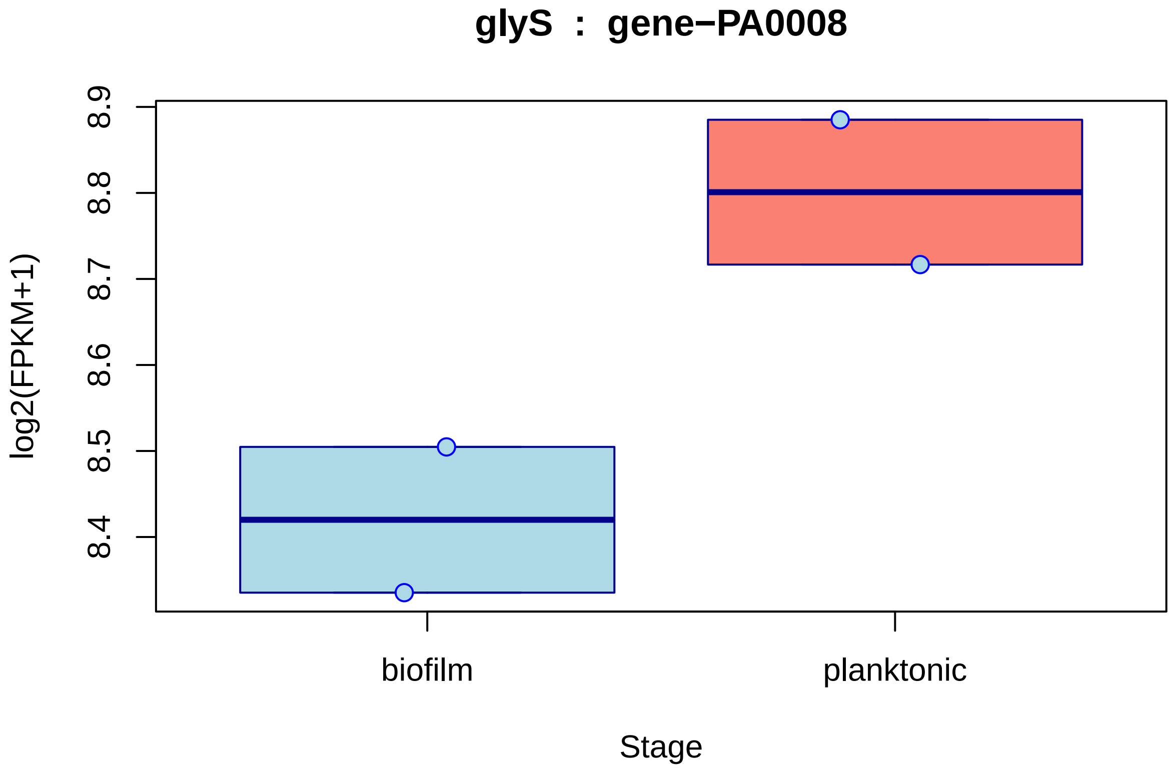
```
numeric_stages <- as.numeric(factor(pheno_data$stage))

jittered_stages <- jitter(numeric_stages)
```

plot expression of individual gene

```
boxplot(transformed_fpkm ~ pheno_data$stage,
        main=paste(ballgown::geneNames(bg_filt)[8], ' : ', ballgown::transcriptNames(bg_filt)[8]),
        xlab="Stage",
        ylab="log2(FPKM+1)",
        col=c("lightblue", "salmon"),
        border="darkblue")

points(transformed_fpkm ~ jittered_stages,
       pch=21, col="blue", bg="lightblue", cex=1.2)
```



interpret the above figure