R Workshop Special Topic: Normalizing qPCR Data in R

Katherine Thompson (katherine.thompson@uky.edu)
Department of Statistics, University of Kentucky
November 4, 2016

0.1 Getting Started

To get started with this module, we will set our working directory, read in our data either using the code below or by importing it into RStudio, and install two packages that have functions built in to normalize qPCR data.

```
########################################
## Set the working directory
########################################t
setwd("C:/Users/ukystat/Dropbox/R_Workshop_2016/")

########################################
## Read in data
########################################
qpcr_plate1=read.table('qpcr_plate1.csv',sep=',',header=TRUE)

########################################
## Load Libraries
########################################
## Install SLqPCR Package from Bioconductor (only run once per computer)
source("https://bioconductor.org/biocLite.R")
biocLite("SLqPCR")

## package 'SLqPCR' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\ukystat\AppData\Local\Temp\Rtmp2B9FoX\downloaded_packages

## Install qpcrNorm Package from Bioconductor (only run once per computer)
source("https://bioconductor.org/biocLite.R")
biocLite("qpcrNorm")

## package 'qpcrNorm' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\ukystat\AppData\Local\Temp\Rtmp2B9FoX\downloaded_packages

## Load packages
library(SLqPCR)
library(qpcrNorm)
```

0.2 Cleaning Data

Next, we need to clean the data so that it is in the correct format for SLqPCR functions.

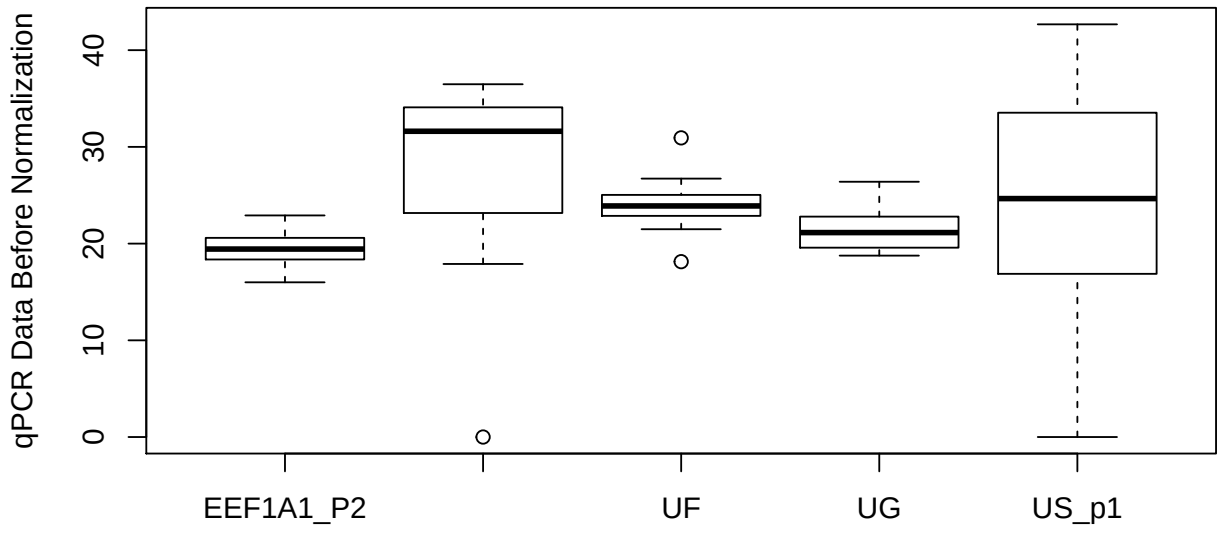
```
## Find number of rows in data
n1=nrow(qpcr_plate1)

## Average over two replicates per subject
```

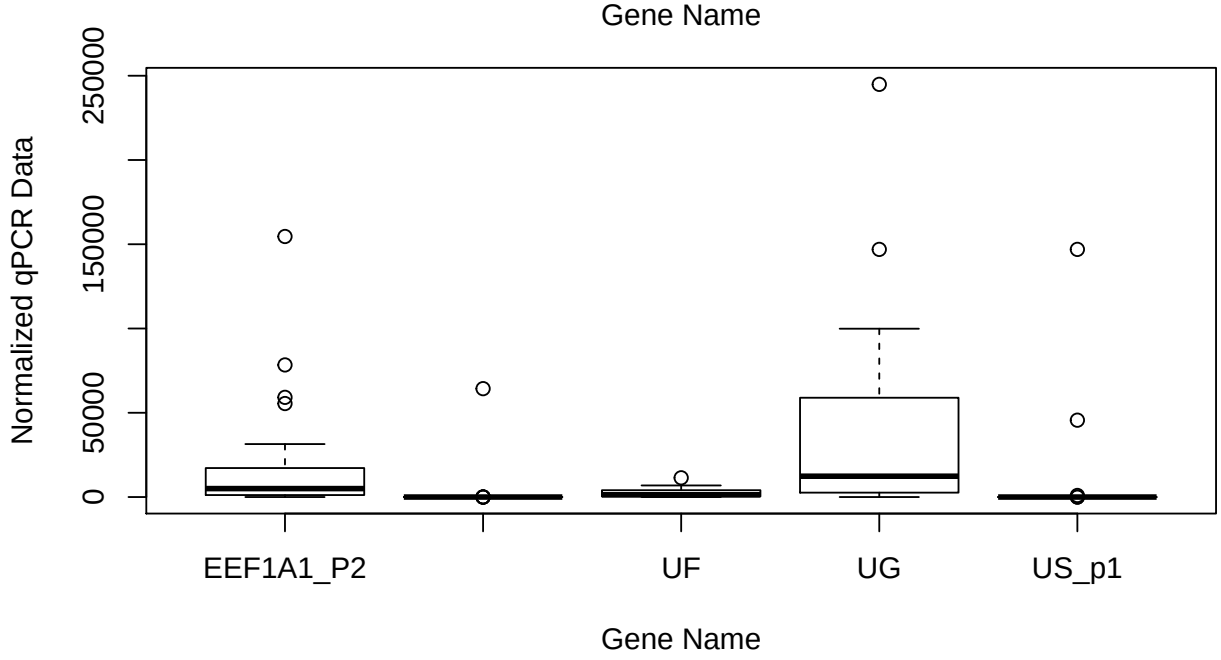
```
qpcr_plate1_avg=aggregate(Ct~Sample*Gene*Stage,
          data=qpcr_plate1,FUN=mean)

## Save information other than qPCR data
sample_information=qpcr_plate1_avg[,c("Sample","Stage")]

## Format data for qpcR functions
##   (genes should be columns)
library(tidyr)

## Warning:  package 'tidyr' was built under R version 3.3.1

qpcr_wide=spread(data=qpcr_plate1_avg,
                 key=Gene,
                 value=Ct)
```

0.3 Normalizing qPCR data with SLqPCR Package

Our first option is to normalize the data using the **SLqPCR** package. Note that we will consider two packages here, but many more exist for normalizing and analyzing qPCR data!

```
## Find relative expression values for realtime quantitative RT-PCR data
##    do not include the columns Sample, Stage, nor Plate here
RelativeData=apply(qpcr_wide[,-(1:3)],2,relQuantPCR)

## Find gene expression stability value M for real-time quantitative RT-PCR data
geneStabM(RelativeData[,c("EEF1A1_P2","GUSB")])

## EEF1A1_P2      GUSB
## 0.9700433 0.9700433

## Normalize real-time quantitative RT-PCR data
expression=normPCR(relData=RelativeData,
                   ## data frame with relative quantities (genes in columns)
                    HKs=c(2,3)) ## Column numbers for housekeeping genes

## Combine normalized data with Sample, Stage, and Plate
qpcr_normalized=data.frame(qpcr_wide[,1:3],expression)

## Make boxplots of raw and normalized data
par(mfrow=c(2,1), ## Make 2 rows and 1 column of plots
    mar=c(4.5,4.5,0.5,0.5))
    ## Make bottom and left margins 4.5, top and right margins 0.5
boxplot(qpcr_wide[,-c(1:3,5:6)],
        ## Omit columns 1-3 (sample information) and 5-6 (housekeeping genes) from plot
        ylab='qPCR Data Before Normalization',
        xlab='Gene Name')
boxplot(qpcr_normalized[,-(1:3)], ## Omit columns 1-3 (sample information) from plot
        ylab='Normalized qPCR Data',
        xlab='Gene Name')
```



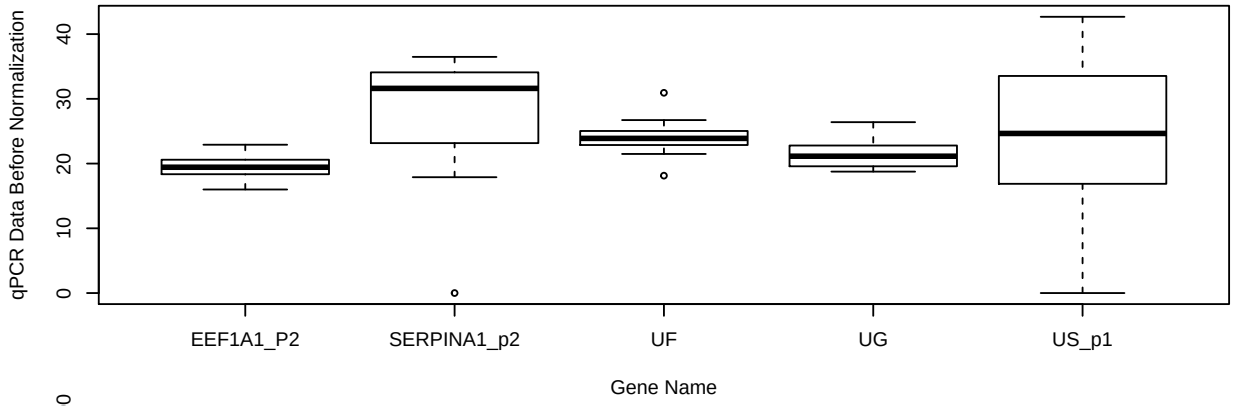


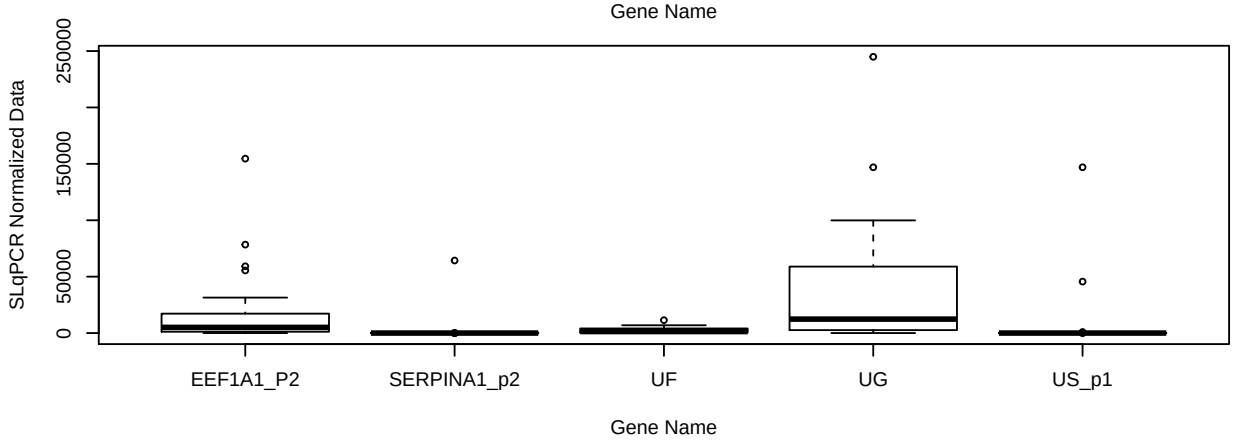

```
## Write results to csv file
write.table(qpcr_normalized,'qPCR_Normalized_with_SLqPCR.csv',
            row.names=FALSE,
            quote=FALSE,
            sep=',')
```

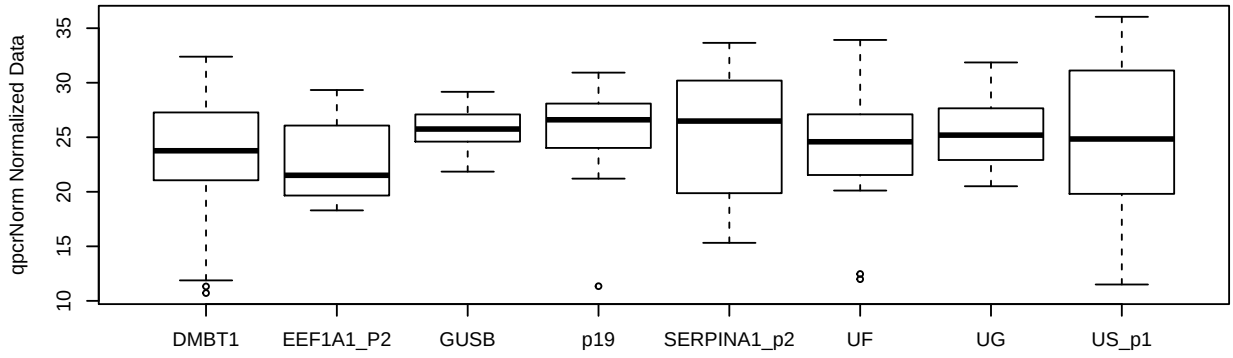
0.4 Normalizing qPCR data with `qpcrNorm` Package

Our first option is to normalize the data using the `qpcrNorm` package. This package requires more data cleaning, and gives different normalization than the SLqPCR package.

```
## Data Cleaning
library(tidyr)
plate1=qpcr_plate1[seq(1,n1-1,by=2),]
plate1$Ct_rep2=qpcr_plate1[seq(2,n1,by=2),"Ct"]
write.table(plate1[,c("Gene",PlateIndex="Sample","Ct","Ct_rep2")],'qpcr_batch.csv',
            sep=',',row.names=FALSE)

write.table(qpcr_plate1_avg[,c("Gene","Sample","Ct")],'qpcr_batch.csv',
            sep=',',row.names=FALSE)

plate1new=readQpcr("qpcr_batch.csv",sep=',',header=TRUE,qc=FALSE)
plate1new@exprs=cbind(plate1new@exprs,plate1$Ct_rep2)

## Normalize data
qnorm_data=normQpcrHouseKeepingGenes(plate1new,hkeep.genes=c("EEF1A1_P2","GUSB"))
normalized_expr=qnorm_data@exprs

## Make boxplots of raw and normalized data
par(mfrow=c(3,1), ## Make 3 rows and 1 column of plots
    mar=c(4.5,4.5,0.5,0.5))
## Make bottom and left margins 4.5, top and right margins 0.5
boxplot(qpcr_wide[,-c(1:3,5:6)],
        ## Omit columns 1-3 (sample information) and 5-6 (housekeeping genes) from plot
        ylab='qPCR Data Before Normalization',
        xlab='Gene Name')
boxplot(qpcr_normalized[,-(1:3)],
        ## Omit columns 1-3 (sample information) from plot
        ylab='SLqPCR Normalized Data',
        xlab='Gene Name')
boxplot(apply(normalized_expr[,1:2],1,mean)~rownames(normalized_expr),
        ylab="qpcrNorm Normalized Data")
```


qPCR Data Before Normalization
0
10
20
30
40
EEF1A1_P2
SERPINA1_p2
UF
UG
US_p1
Gene Name


SLqPCR Normalized Data
0
50000
150000
250000
EEF1A1_P2
SERPINA1_p2
UF
UG
US_p1
Gene Name


qpcrNorm Normalized Data
10
15
20
25
30
35
DMBT1
EEF1A1_P2
GUSB
p19
SERPINA1_p2
UF
UG
US_p1